Systems & Control: Foundations & Applications

Eloy Garcia
Panos J. Antsaklis
Luis A. Montestruque

Model-Based Control of Networked Systems

Birkhäuser

Systems & Control: Foundations & Applications

For further volumes:
http://www.springer.com/series/4895

Eloy Garcia • Panos J. Antsaklis
Luis A. Montestruque

Model-Based Control of Networked Systems

Birkhäuser

Eloy Garcia
Infoscitex Corporation
Dayton, OH, USA

Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN, USA

Luis A. Montestruque
EmNet, LLC
South Bend, IN, USA

ISSN 2324-9749 ISSN 2324-9757 (electronic)
ISBN 978-3-319-07802-1 ISBN 978-3-319-07803-8 (eBook)
DOI 10.1007/978-3-319-07803-8
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014944313

Mathematics Subject Classification (2010): 93C05, 93C15, 93C40, 93C57, 93B52, 93B51, 93D09, 93E12, 93A14, 90B18

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.birkhauser-science.com)

We dedicate this book to our families, who give us support and strength, and provide us with the right priorities and balance.

Preface

Feedback control of dynamical systems has benefited from recent development of sensing and actuation nodes with strong local computation capabilities and from the use of digital communication networks for system interconnections. With all these advancements, the detailed analysis of feedback control systems remains an important part of the complex networked control applications. System uncertainties have always been a central issue in control systems. It is critical to address uncertainty effects on the stability and performance of control systems when it is not possible to obtain continuous feedback measurements. Such is the case in Networked Control Systems (NCS) where the communication channel is of limited bandwidth and it is also shared by different subsystems.

This book presents a specific framework, the Model-Based Networked Control Systems (MB-NCS) framework, for design and analysis of NCS. This approach places special emphasis on model uncertainties. Detailed development of the basic results is provided and a number of important extensions are addressed; multiple examples are given as well. Using the model-based approach we introduce several types of architectures and control strategies that aim at improving performance in NCS. The overall performance of NCS considers the appropriate use of network resources, network bandwidth in particular, in addition to the desired response of the system being controlled. The plant model is used at the controller/actuator side to approximate the plant behavior so that the sensor is able to send data at lower rates, since the model can provide information to generate appropriate control inputs while the system is running in open-loop mode.

It is assumed that readers are familiar with the basics concerning Linear Systems. This book is intended for graduate students, researchers, and practitioners with interest in the study of control over networks, distributed systems, and control with communication constraints. These readers typically have the background or have completed a graduate-level course on Linear Systems.

In this book we take the MB-NCS framework, from its initial configuration and apply it to many complex applications. Different basic problems in control theory are studied in the context of NCS using the model-based approach. The general idea is that, in the absence of continuous feedback, the available knowledge about the

system dynamics provides a useful tool for designing and implementing simple, yet robust controllers that provide good performance in the presence of multiple communication constraints. The problems and scenarios studied in this book provide examples of a large number of potential applications in which the MB-NCS framework brings an innovative advantageous perspective in the analysis and design of control systems.

This book is divided in two parts. Part I focuses on the stability aspects of MB-NCS while considering systems with different types of dynamics: linear, nonlinear, continuous-time, and discrete-time systems. Different network constraints are also considered in Chaps. 2–8 of Part I such as limited bandwidth, network delays, and signal quantization.

Part II, Chaps. 9–14, deals mainly with the performance of MB-NCS. In the following we introduce the main topics covered in each chapter of this book.

Description of Chapter Contents

Chapter 1 introduces Networked Control Systems and several approaches used for analysis of this type of control systems. Detailed literature review of prior and relevant work is addressed in this chapter. Particular approaches are emphasized in Chap. 1: model-based frameworks and event-triggered control.

Part I: Chaps. 2–9

Chapter 2 presents and explains with great detail the MB-NCS framework. The contents of this chapter are fundamental in being able to follow the material in the remaining chapters. Once the main contents of Chap. 2 have been studied there is no strict order for the rest of this book. Besides presenting the MB-NCS framework, Chap. 2 offers the first main results related to this approach which are built on one of the most basic cases we study in the book: state feedback with periodic communication. This chapter considers both continuous-time and discrete-time systems. Recent additional results are provided at the end of the chapter.

In Chap. 3 two important problems are studied: output feedback and network induced delays. Both continuous-time and discrete-time systems are considered. One of the main important lessons of this chapter is that the MB-NCS framework can be applied to many different control problems; the problems in this chapter are two of the main extensions when considering networked systems in general. In the output feedback problem, we introduce a state observer and the overall system now contains three subsystems: plant, model, and observer. A similar approach is used for the network induced delay problem; a third subsystem is introduced that helps in obtaining a current estimate of the system state based on delayed measurements.

The contents of Chap. 4 analyze the stability and the selection of update periods and other network parameters when intermittent feedback is used along with the model-based architecture previously introduced. Two intermittent feedback approaches are described in this chapter. In the first approach the networked system operates in two different modes: closed-loop and open-loop mode. The closed-loop mode of operation requires a strong assumption concerning the ability of transmission of continuous signals over the network. To relax this assumption, we describe a second intermittent feedback approach that operates using two different update rates. The faster update rate now takes the place of the closed-loop mode of operation. Stability results are provided for continuous-time and discrete-time systems and for the state and output feedback cases.

An important aspect present in many networked systems is considered in Chap. 5, namely, time-varying transmission intervals. In many implementations access to the communication channel may be random and strict periodic transmission of information may not be possible. This chapter provides different stability results first when no statistical information about the update periods is known and then when the update periods follow a prescribed statistical distribution.

Chapter 6 also presents an approach for updating the model using time-varying intervals under event-triggered control. The main difference with respect to Chap. 5 is that in event-triggered control the main purpose in using nonperiodic transmission intervals is to adapt the update instants according to the current conditions of the system. Chapter 6 provides details on how to implement event-triggered techniques in the MB-NCS framework. It also describes different strategies that provide different results and varying performance. This approach is also used for systems subject to network induced delays.

The MB-NCS framework is used in Chap. 7 to consider nonlinear systems. Continuous-time systems are considered first and the focus is on finding stability conditions under periodic transmissions. This chapter also considers discrete-time systems using models described by input–output representations; event-triggered control techniques are used in this case.

The work in Chap. 8 addresses another important constraint in NCS, namely, signal quantization. Since measurements need to be transmitted over digital communication networks, they need to be quantized in order to be represented in digital form. This chapter provides a thorough analysis of MB-NCS with quantization of measurements. Different types of quantizers are discussed and the corresponding results provide the design parameters needed in this case which are the quantization parameters and the update intervals. The joint effect of quantization and network delays is also studied and similar results are provided.

Part II: Chaps. 9–14

Two main topics with respect to the optimal performance of MB-NCS are discussed in Chap. 9. The first one is related to the implementation of event-triggered control

techniques for systems that operate optimally assuming continuous feedback. The second topic is concerned with the optimal design (nonperiodic in general) of transmission instants. In the latter case we consider finite-horizon optimal control problems and use Dynamic Programming in order to find optimal decisions concerning transmission of state updates based on the current behavior of the system.

Chapter 10 provides a detailed analysis of the performance of continuous-time MB-NCS. Two different performance metrics are used to quantify the performance of model-based control systems with periodic updates. Design of optimal controllers is also addressed in this chapter, under the assumption of periodic transmission.

Reference input tracking with limited feedback and using the model-based approach is analyzed in Chap. 11. This chapter deals with continuous-time systems using periodic updates and conditions on the update period are given in order to provide a bound on the tracking error. In this chapter we also consider linear discrete-time systems described by input–output representations. Dynamic controllers are considered in this case. One of the advantages of using this type of system representation is that common types of uncertainties such as additive and multiplicative uncertainties can be considered. This implies that knowledge of the system dimension is not assumed and available models with different dimensions than the real system can be used for both controller design and for implementation in the MB-NCS framework.

In Chap. 12 the model of the system is not considered to be fixed. Parameter estimation algorithms are described in the context of networked systems and used to upgrade the model. In other words, by measuring the current response of the system it is possible to find new parameters that better approximate the real values than the initial model parameters. Systems with measurement noise in addition to parameter uncertainties are considered as well. Different types of applications are described where the algorithms for estimation of parameters described in this chapter are of great use. These applications include switched systems, fault identification, and systems with input disturbances.

Chapter 13 studies multirate systems. Several problems are discussed. The first one corresponds to the case where multiple sensors are used to measure the state of an uncertain system. These sensors represent different nodes in an NCS attempting to send their measurements to a centralized controller. The problem considered is to find transmission periods, not necessarily the same for all nodes, which guarantee stability of the networked system. The second problem considers a two-channel networked system where the connection from controller to actuator is implemented using the communication network.

Distributed systems are studied in Chap. 14. Two approaches are considered. First we consider periodic updates for discrete-time systems and discuss both the single-rate and the multirate implementations. Then we consider continuous-time systems using event-based updates. For this problem, we study both centralized and decentralized controller design and implementation.

The Appendix collects different results concerning vector and matrix norms, Jordan canonical forms, similarity transformations, and other linear algebra concepts. It also contains basic control theory definitions and other results about linear systems.

Finally, the appendix provides a summary of definitions and theorems broadly used in Lyapunov analysis.

In the development of the material and in the writing of this book we benefited from comments and suggestions of many colleagues at conferences and during academic visits and presentations. We would also like to acknowledge the help we received from several Ph.D. students at the University of Notre Dame during the past 10 years. We would like to single out and acknowledge the contributions of Tomas Estrada especially in the area of intermittent feedback. We also acknowledge the useful discussions, comments, and suggestions of Michael McCourt and Han (Cherry) Yu.

Dayton, OH, USA E. Garcia
Notre Dame, IN, USA Panos J. Antsaklis
South Bend, IN, USA Luis A. Montestruque

Contents

Chapter 1
Introduction

The last two decades have seen a rapidly increasing use of common-bus network architectures for implementation of control systems. A digital communication network is used to transfer information among the components of a control system: sensors, controllers, actuators, and the systems to be controlled. Such a control system implemented over a network is called a Networked Control System (NCS).

NCSs offer a large number of advantages compared to traditional point-to-point configurations where control systems are interconnected using dedicated wires. In fact, one of the main advantages is that NCSs reduce the volume of wiring; this is of prime importance for instance, in the automobile industry and in the design of efficient aircrafts. NCS can also help to improve efficiency, flexibility, and reliability of the network interconnected system, reducing reconfiguration and maintenance time and costs.

1.1 Networked Control Systems

Figure 1.1 shows multiple systems interconnected with their respective controllers; note that a single controller node may operate on more than one system, that is, NCSs offer a variety of implementations. For instance, a single embedded processor may contain multiple controllers for multiple plants, these controllers may belong to the same node and the network protocol is in charge of matching sensor and actuator nodes with their corresponding controller within a node.

There are many examples where using a network to interconnect control applications is convenient. A typical example is the case of aircraft control. In this case, different sensors and control surfaces among other control components are distributed over the aircraft. Another example is the case of manufacturing in factories, where it is common practice to implement data acquisition systems along the process path. Dozens of sensors are deployed over critical points to make important information about the process available to quality control engineers. Most of the time, these sensors will transmit the collected information to a central computer

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_1,

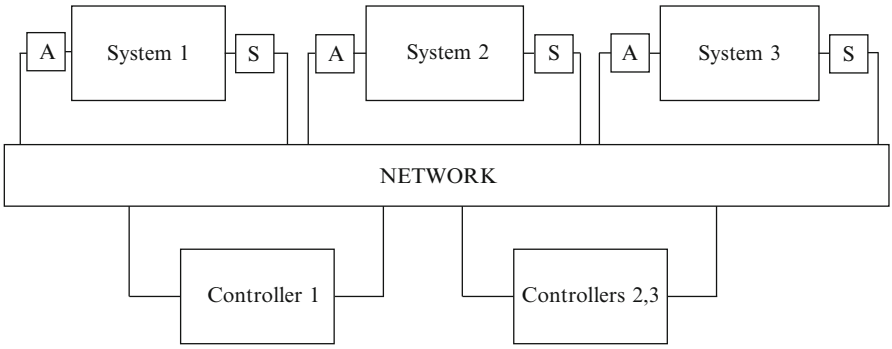

Fig. 1.1 Multiple control systems interconnected using a communication network

using an industrial network. More than often, new control loops need to be added as quality or industrial engineers analyze the data retrieved by the acquisition network. In this case, it seems natural to attach the controllers and actuators to the already existing network and share the data already provided by the deployed sensors. In general, use of a network in a control system is desirable when there is a large number of distributed sensors and actuators.

The complexity of NCS arises from the hybrid nature of the system. That is the continuous plant dynamics interacting with the discrete hardware and software of the network. The hybrid connection to NCS has been explored by Bushnell et al. [33]. Furthermore, tools for analyzing hybrid systems are utilized by Lin et al. [155] to characterize the robust stability of NCS under packet dropouts and network access delays.

One of the main problems to be addressed when considering an NCS is the limited bandwidth of the network. In point-to-point control systems, it is possible to send continuous measurements and control inputs. Bandwidth and dynamic response of a plant are closely related. The faster the dynamics of the plant, the larger is its bandwidth. This usually translates to large frequency content on the controlling signal and a continuous exchange of information between the plant and the controller. In the case of discrete-time plants, the controller acts at spaced instants of time and transmission of continuous signals is not required. However, some discrete-time systems may have a fast internal sampling which results in large bandwidth requirements in terms of the network characteristics.

In NCS, data has to be sampled, quantized, encoded, and sent through the network. An important problem arises when many nodes attempt to broadcast messages in a short interval of time. As the rate of transmission increases and approaches the network bandwidth limit, the presence of packet dropouts becomes more frequent and time delays are longer; therefore, reduction of the transmission of information over the network is of considerable importance for the stability and performance of NCS. This can be addressed by two methods: the first method is to minimize the transfer of information between the sensor and the controller/actuator; the second method is to compress or reduce the size of the data transferred at each transaction.

Popular networks in industry actually deployed include CAN bus, PROFIBUS, Fieldbus Foundation, and Ethernet among others. Each of these protocols and standards has very different characteristics such as network contention resolution or scheduling schemes, and transmission media. Among the shared characteristics are the small transport time and big overhead (network control information included in the packet). This means that data compression by reducing the size of the data transmitted has negligible effects over the overall system performance. So, in general, reducing the number of packets transmitted brings better benefits than data compression when using communication protocols with significant packet overhead. The reduction of the number of packets transmitted through the network can translate into larger minimal transfer times between the components. It is also to be noted that any delay in an information transaction is usually due to network access contention. This translates into what has been already noted by Walsh and Ye [259] (see also [100]); namely that the sensor with a fast sampling rate can send through the network the latest data available resulting in a negligible information transfer delay. But there will still be contention in the network so that, even though the delay is small, the sensor data would not be available at all times to the controller/actuator. This brings us again to the approach of reducing the data transfer rate as much as possible. In this manner more bandwidth will be available to allocate more resources without sacrificing stability and ultimately performance of the overall system.

In this book, we consider the problem of having sensors that are connected to the actuators/controllers by a network; that is, the feedback path is over a network. However, data networks typically have limited bandwidth and transfer information in a discrete-time framework, and this makes the task of designing a control system rather challenging. This book concentrates in the use of the knowledge about the plant to overcome some of the most relevant challenges. This type of NCS is called Model-Based Networked Control System or MB-NCS [186–191].

Extensive work has been done to overcome the shortcomings of NCSs and to understand the implications of the use of a network on the feedback path. Some of the research areas in NCS fall into the following groups: Finite Alphabet Systems, Scheduling Strategies, Discrete Plants with Packet Loss or Delay, and Packetized Control. This work is discussed in the remaining of this section while Sects. 1.2 and 1.3 provide an introduction into the ideas that will be seen later in this book.

1.1.1 Control over Networks Using a Finite Alphabet System

Finite alphabet systems attempt to reduce the bandwidth of the system by sending through the network a symbol from a finite set of symbols or finite alphabet. By doing this the number of bits required to send each symbol through the network can be reduced dramatically. This can be seen as a coding scheme or data compression. Many times the accurate reproduction of the signal sampled by the sensor is not necessary.

In [199, 200] Nair and Evans study the case of an infinite dimensional time-varying discrete plant with unknown initial condition. The plant is being controlled using a network on the feedback path. The only constraint on the network is that a finite set of symbols can be used to send the information from the sensor to the controller/actuator. The sensor then implements a coder that transmits the information to the controller/actuator at each sampling time. The information takes negligible time to get to its destination and the data corruption probability is assumed to be zero. The result is that, under certain technical conditions on the probability density function of the initial condition, the plant is stabilizable asymptotically in the mth output moment and in the infinite horizon if and only if the coder and controller comply with certain characteristics that depend on the alphabet size and some dynamical constants. The special case in which the plant is unstable and LTI the condition is reduced to $R > \log_2|\lambda|$ where R is the transmission rate in bits per second and λ is the unstable open-loop pole with largest magnitude. These results are extended in [201] to include NCS with Markov jump parameters. In the same spirit, Liberzon et al. also consider the stabilization of linear and nonlinear systems with quantized signals in [150–153].

Elia and Mitter [65] propose the design of a quantized controller and state estimator for an LTI discrete system. The result is an optimized controller and state estimator that operate in discrete periodic times with quantized values for the state. It is obvious that the coarser the discretization, the less the bandwidth required for the system to work. This work follows the same line as the one in [199–201]. It is shown that the coarsest or least dense quantizer that quadratically stabilizes the plant is logarithmic and can be computed by solving a special LQR problem. The theory is then extended to continuous LTI plants using constant sampling time intervals. It is shown that the optimal sampling interval time (using the proposed quantizers) is only a function of the sum of the unstable eigenvalues of the system. Ishii and Francis further explore the idea to prevent chattering in the system in [124].

Together with the optimization of the sampling period and quantizer some effort has been made in optimizing the sampling times and control law. This can be viewed as type of scheduling. Several approaches have been proposed. Rehbinder and Sanfrison [215] prove the intuitive idea that the plant with the fastest dynamics should be given more network bandwidth resources. In the same spirit, Xiao et al. [272] study the optimization of the word length, the output scaling, and the controller or estimator gain. Various communication schemes are presented and analyzed.

In [123] Ishii and Francis, extend their Dwell Time Controller [124] for systems with output feedback. In the dwell time controller setup the plant's output is fed to a state observer. The estimated state is then quantized using a logarithmic partition of the state space. The quantized value is sent through the network to the controller/actuator. After decoding the message, the controller will apply a constant input to the plant that corresponds to the received value of the quantized state. The logarithmic partition is coarser as the state's distance to the origin is bigger, and it is finer when the state is closer to the origin. This seems to be reasonable since fine

steering of the state is more useful when the state is close to the origin. The logarithmic partitions are made overlapping so that the system can tolerate some noise generated by the sensors. Also a dwell time is specified to reduce fast chattering produced by the controller when switching control inputs. To do so, the time interval between switching control inputs is enforced to be bigger or equal than the dwell time. This can be done by timing the messages sent to the controller. In our approach we will use the natural choice of having a state observer at the sensor side of the network.

In [209] Petersen and Savkin present an algorithm for the stabilization of a multi-input/multi-output discrete-time linear system via a limited capacity channel. The approach taken is a deterministic multirate state-space approach that leads to a nonlinear dynamic feedback controller. The network channel is assumed to be noiseless and with time delays associated with transmissions. An important feature of the approach is that it is a multirate approach in which symbols are transmitted across the channel at a slower rate that the control inputs are applied to the discrete-time plant. It is also shown how the results can be extended to the general output feedback case by using a form of deadbeat observer. Both coder and decoder are dynamic systems that are "synchronized" by an evolving state that is known at both sides. The state space is partitioned dynamically as the system approaches its steady state, in this way asymptotic behavior is proved achievable. The actuator (or sensor) can use the state synchronization proposed here to predict the behavior of the sensor (or actuator) in the intervals where there is no communication.

The work in [202] offers a framework that applies to nonlinear systems subject to quantization. In addition, network scheduling is considered using periodic updates in terms of the maximal allowable transfer interval, MATI (see below). Copies of the plant dynamics are used at both ends of the communication channel. Model uncertainties are not considered, the only difference between the plant and the copies being that the copies operate using the quantized variables instead of the real ones. The transmission of quantized measurements at times dictated by event-based strategies has been considered by several authors. The references [17, 182] discuss this approach when considering the most convenient times to sample a signal. The authors of [140] also consider event-triggered transmission for stabilization of systems that communicate using a limited-bandwidth network and using a similar model-based approach as the one discussed in this book. Event-triggered control techniques will be used in many chapters of this book and details and literature review concerning these techniques will be provided in Sect. 1.3 of this chapter.

1.1.2 Control over Networks Using Scheduling Strategies

Another important problem associated with control over networks is deciding about scheduling strategies. One of the main purposes for using a network is that many control and data acquisition systems can share its resources. Since many nodes will

be transmitting information over the network, it is natural to determine when is the best time for these nodes to transmit in order to avoid congestion, collisions, and achieve the control goal.

In [260], Walsh et al. presents a protocol that uses dynamic scheduling and a zero-order-hold at the controller input. The notion of maximal allowable transfer interval, MATI, is introduced to place an upper bound on the time between transfers of information from the sensor to the controller. In this case the controller is designed without taking the network into account, a desirable feature. However, serious behavior degradation can result if the MATI is too large and the network slow. Also a dynamic scheduling scheme is introduced: Try-Once-Discard or TOD protocol. In TOD each sensor has a transmission priority that is proportional to the error between the last data sent and the actual measured value. The sensor with biggest error is given maximum priority to transmit. Additionally, if a sensor is denied access to the network by contention, it will discard the packet and construct a new one with fresh data before trying again to transmit. These results are extended to nonlinear plants in [257]. Tolerance of these systems under different types of noise is studied on [20].

The effects of different scheduling schemes for the TOD protocol are studied in [258]. It is implied that the plant performance is improved if an appropriate scheduling scheme is used. Scheduling is of utmost importance when there are a number of sensors, actuators and controllers competing for network resources. It determines the nature of the delays, transmission rates, etc. A deterministic scheduling scheme is presented by Hristu-Varsakelis in [117]. Deterministic communication sequences are easier to analyze and sometimes can have a superior performance than non-deterministic scheduling schemes but can also be difficult to enforce.

In [21] Beldiman et al. extend the results in [260] to include a state predictor, for LTI systems, to estimate the state in between updates. Two types of state predictors are defined. The first one is the so-called open-loop predictor, which is basically a plant model that is updated with an invertible transformation of the state vector available at the plant output. The model assumes complete knowledge of the plant. In other words, there is no uncertainty in the model dynamics and they match perfectly with the plant dynamics. Once the model's state has been updated, it can provide the controller with an estimate of the plant state vector. The second predictor, called closed-loop predictor, has the same structure as a standard linear state observer. It receives the output of the plant and tries to recreate the plant output in between transmissions. This predicted output is then fed to the controller. This closed observer needs the network to be very fast in order for the observer to converge. Sufficient conditions are given for the stability of this NCS setup.

The optimization of switching times and state estimation through a network is covered by A.S. Matveev and Savkin in [173]. In [173] a linear discrete-time partially observed system perturbed by a white noise is studied. The observations are transmitted to the controller via communication channels with irregular transmission times. Various measurements signals may incur independent delays or arrive at the estimator out of order. The estimator can dynamically control which

sensors it will connect to. The minimum variance state estimate and the optimal sensor switching strategy are obtained. Basically a Kalman-like state estimator that is able to connect to its inputs a limited array of the plant output sensors and deal with a variable delay to optimally estimate the plant state.

Branicky et al. study in [30] the design of a scheduling policy based on the Rate Monotonic Scheduling. Here a number of independent plants are allowed to close their feedback loops through a shared network. Each system is given a constant transmission time where contention is solved via the Rate Monotonic Scheduling. The problem is posed as an optimization problem in which optimal transmission times are obtained based on traditional cost functions and schedulability constraints. Extensive simulations of the obtained results are shown in [28].

A model-based approach was used in [66] and in [242, 243] in order to schedule updates for the problem of distributed systems that broadcast information using a digital communication network. In this problem, there are a finite number of subsystems and each one of them is coupled to a subset of those systems. Each subsystem broadcasts its state at some time instants in order for the coupled systems to use that information for self-control. In these references the model-based approach is used to schedule the updates using a preassigned single update rate in a sequential manner, i.e., a single rate approach in which the agents send updates at different time instants.

1.1.3 Control over Networks Using Discrete Plants with Packet Losses or Delays

Due to the discrete nature of the network, the analysis of discrete plants with packet loss or delays is greatly simplified. Some natural questions arising in this area deal with stability or performance of the NCS under stochastically described packet delays, bounded packet delays, Markov chain-driven packet loss models, etc.

Bauer et al. analyze the problem on a network with random delays in [19]. The paper proposes the use of a Smith predictor in a discrete framework to eliminate the delay induced by the network. The Smith predictor is placed in front of the controller and uses knowledge about the plant to propagate forward the delayed information from the sensor and make it accessible to the controller. We will use the intuitive idea that knowledge about the plant dynamics can help to relax the network quality of service requirements without sacrificing the performance of the NCS.

In [174] Matveev and Savkin present an NCS with an estimator/central controller, and several semi-independent subsystems. The central controller receives information from the different subsystems about the uncontrollable dynamics. It compresses and processes the data and sends it to the different subsystems. The data arrives to the local controllers at each subsystem. The local controller selects the right information from the central controller message. This message contains

propagated versions of the estimated state and a time stamp so that the local controller can choose the right propagated version. This is done since the central controller does not know the value of the transmission delay. The local controller estimates the control-induced part of the controllable states of the subsystem and computes the state of the controllable state by adding the term corresponding to the uncontrollable dynamics that was received from the central controller. The problem is solved in a quadratic optimization framework.

In [156] Ling and Lemmon study the performance of a discrete system in which the sensor sends the output of the plant to the controller/actuator at each sampling time. The packets sent from the sensor can be dropped with a certain probability; it is also assumed that there is no delay associated with the packet transmission. The packet dropout process is assumed to be identically and independently distributed. The controller/actuator is assumed to hold the last received sensor data until the next packet arrival. Conditions for mean square stability are first stated. Ergodicity and wide sense stationarity conditions are then derived. Finally, the power spectral density of the output is computed as a measure of the control system's performance. These results are then extended to the case of Markov chain-driven dropouts in [157, 158] and used to calculate optimal dropout policies.

Azimi-Sadjadi in [13] studies the stability of NCS in the presence of packet losses. In this setup the controller receives information from the sensors through the network, calculates the control signal, and sends through the network the control signal to the actuators. Multiple numbers of actuators and sensors are allowed. The packets are sent simultaneously from all the sensors and to all the actuators. The packets are delivered with identical and independent probability distribution. An "uncertainty threshold" is calculated based on the system dynamics and the statistical properties of the network under which the system is no longer stable.

The work in [172] presents a configuration that stabilizes an NCS with large constant delays using passivity and the scattering transformation. The results presented in [46, 113] derive general models of NCSs that consider time-varying sampling intervals and delays.

The excellent survey papers [112, 227] describe typical problems and approaches concerning packet losses and time delays in NCS.

1.1.4 Control over Networks Using Packetized Control

A different approach to reducing transmission rates over a network is based on transmitting large amounts of information in single packets. Georgiev and Tilbury [100] use more efficiently the packet structure, that is, reduction on communication is obtained by sending packets of information using all data bits available (excluding overhead) in the structure of the packet. For the sequence of sensor data received, the controller needs to find a control sequence instead of a single control value. Packet size is a new feature in NCS compared to point-to-point architectures and it is a variable that depends on the protocol or type of network being

implemented. For example, the minimum effective load in an Ethernet packet is 46 bytes and, if 2 bytes are used to represent a sensed quantity, which is good enough for most applications since those 16 bits can encode $2^{16} = 65536$ different levels of sensed signals, then we can send at least 23 sensed signals in such a packet. Other networks that specialize in control applications require a considerably smaller minimum size packet. For instance, the CAN protocol is optimized for small messages. With an overhead of 47 bits (minimum packet size), and a maximum data load of 8 bytes encourages designers to use all bits available to send different input elements of a computed input sequence. Quevedo et al. [212] follow a similar approach; they focus on large data packets like the ones used in Ethernet-based protocols and, employing a predictive controller, they are able to encapsulate a sequence of input values in a single packet. The work in [293] presents a strategy that reduces data transmission in both the sensor to controller and the controller to actuator channels of an NCS. The sensing strategy is based in a "sensor transmission rule" (STR) which broadcasts a sensed measurement if there is significant change in the output of the system. The control action consists of a sequence of control inputs that are sent in one packet; the control inputs are based on a time-varying controller found by solving a series of Linear Matrix Inequalities (LMI).

1.2 Model-Based Frameworks

A particular approach for the design of controllers in NCS that has been gaining attention in the NCS research community is called Model-Based Networked Control Systems (MB-NCS) and it was introduced in [186–191]. In this framework a nominal, usually inexact, model of the physical system or plant is used to generate a control input for the actuator allowing the system to run in open loop for a finite interval of time without need for feedback in this period. The state of the model is then updated when a measurement arrives from the sensor. When we update the model state, we reset any possible mismatch or difference between the model and the plant states that is produced by allowing the plant to operate in open-loop mode for a time interval. It has been shown [188] that using this framework it is possible to stabilize a system by sending sporadic measurements to the controller, considerably reducing the number of information packets broadcasted through the network. In this way we are able to reduce time delays and the number of packet dropouts in the network.

Figure 1.2 shows a basic MB-NCS configuration, where the network exists only on the sensor-controller side while the controller is connected directly to the actuator and the plant, where $u = K\hat{x}$, and the matrices $\hat{A}, \hat{B}$ represent the available model of the system matrices A, B.

In [188] necessary and sufficient conditions for stability were provided when the updates from the system are periodic (h is constant) and with negligible delay for

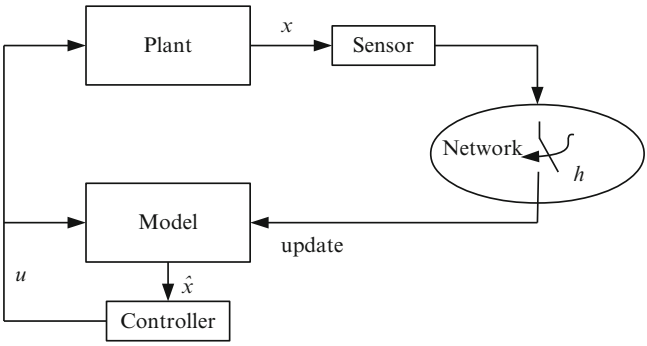

Fig. 1.2 The Model-Based Networked Control System (MB-NCS) architecture

linear continuous and discrete time-invariant systems and for two cases: state feedback and output feedback.

In the case of output feedback, a Luenberger observer is implemented in the sensor node. The sensor contains a copy of the model and controller parameters to generate the input u at the sensor node and along with continuous measurements of the plant outputs; it is capable of generating an estimate of the plant state and sending it periodically to the controller node where it is used to update the state of the model. Similar results were also obtained when the plant and its model are described using discrete-time systems [188]. The same authors also considered the situation when the state measurement arrives at the controller node with a considerable time delay. A propagation unit is implemented in this case to find an estimate of the current plant state based on the delayed measurement. An important extension of this work considered time-varying updates [189]. In that case, two stochastic scenarios were studied; in the first, the assumption is that transmission times are identically independent distributed, in the second, transmission times are driven by a finite Markov chain. In both cases, conditions were derived for almost sure (probability 1) and mean square stability.

In all the work described above, it was assumed that the measurements sent from the sensor to the controller contained data representing infinite precision. This is not the case when using a digital communication network as a medium to broadcast information since measurements have to be sensed, quantized, and encoded and the exact value of the sensed variables cannot be recovered at the receiving node but only an approximation of it. The precision of this approximation depends mainly on the number of bits used for quantization and the encoding algorithm that is being used. The practical assumption that was made before is based on the fact that many industrial networks use a large number of bits to represent data, neglecting the quantization error, i.e., the difference between the real variable and the quantized variable is of a very small order. Different quantization methods were proposed and its implementation in MB-NCS was studied in [191].

Nonlinear systems have also been considered using the MB-NCS configuration. Different authors have provided stability conditions and stabilizing update rates for

nonlinear MB-NCS with and without uncertainties and for nonlinear MB-NCS with time delays [160, 185, 210].

Estrada and Antsaklis introduced the notion of intermittent MB-NCS [67–71]. In this case the measurement updates are not only given for a time instant, but they last for a period of time making the system to operate in closed-loop mode for a finite interval $\tau < h$. The authors show that the maximum allowable value for h in a particular problem can be increased as a function of τ and they also developed similar approaches to the original MB-NCS with instantaneous feedback for the cases of output feedback, network induced delays, and when the interval updates are time varying with different probabilistic distributions. Some preliminary performance-related results when the system is driven by a reference input signal are presented in [67] and they will be discussed in more detail in Chap. 11.

The concept of intermittent control in MB-NCS has also been modeled in the framework of switched systems [295]. This type of systems provides a natural and simple way of representing the behavior of intermittent MB-NCS. In this case the switched system contains two modes corresponding to the open-loop and closed-loop responses. Stability analysis follows from slow switching theory by restricting the time that the system remains in open loop and by choosing an average dwell time sufficiently large.

Different authors have dealt with similar problems in different types of applications. Motivated by activities that involve human operators, the authors of [80, 126] point out that in general a human operator scans information intermittently and operates the controlled system continuously; the intermittent characteristic in this case refers to the same situation presented in [188], that is, a single measurement is used to update the internal model and generate the control input. For a skillful operator, the information is scanned less frequently. Between update intervals the control input is generated the same way as in MB-NCS, that is, an imperfect model of the system is used to generate an estimate of the state and periodic measurements are used to update the state of this model. In the output feedback case, a stochastic estimator is implemented with the assumption that the statistical properties of the measurement noise are known. In both cases the authors provide conditions for stability based on the length of the sampling interval.

Chaillet and Bicchi [41] also use a model that produces the input for the plant (possibly nonlinear) and considers a network in both sides of the plant. The actuator is assumed to have an embedded computer that decodes and resynchronizes the large packets sent from the controller that contain the predicted control input obtained by the model. It also considers delays and partial updates of the state and access to the network is based on error size. The authors consider two situations: first, the actuator has computation capabilities to generate an input sequence based on the estimate of the controller and second, the actuator can only receive, decode, and resynchronize an input sequence. It is clear that the actuator capabilities in the first case make it possible to implement model and controller computations; therefore a controller node is not necessary and the result can be seen as the MB-NCS configuration of Fig. 2.1. The second case also requires some computational capabilities in the actuator but supposing these are limited and

do not allow the implementation of the model and controller, then the performance depends, especially for continuous-time systems, on the size of the packet. The result is a bounded output in contrast to the asymptotic stability property of the first case and the allowable delays and transfer intervals depend directly on the size of the packet.

In [111] Hespanha et al. uses differential pulse code modulation techniques together with a model of the plant dynamics to reduce the amount of bandwidth needed to stabilize the plant. Both sensor and controller/actuator have a model of the plant (assumed to be exact). Both models are run simultaneously, this is assuming that the sensor node can reproduce the control signal applied by the controller/actuator to the plant. The sensor node then calculates the difference between the actual state of the plant and the plant model's state. This difference is then quantized and sent over to controller/actuator node to update its state. It is assumed that the same procedure will be done locally with the sensor node's plant model so to have both sensor and controller/actuator nodes synchronized. The controller/actuator node uses its model to generate the control signal to be applied to the real plant. It is assumed, at first, that the sensor can measure directly all states. In the cases where this is not possible, an observer is proposed. This observer can be placed at the sensor side node. It is not clear whether this will guarantee the stability of the plant, since the observer will need some time to converge to the actual plant state. Plant noise and disturbances can prevent the observer from obtaining the correct plant state estimate. The effects of these practical issues need to be looked at more closely.

Zhivoglyadov and Middleton in [294] study an NCS where a plant model is used in the controller. The discrete plant transmits the plant state to the controller every sampling time. If the communication is interrupted or if a packet is dropped then the model is used to predict the actual plant state. State estimation is also addressed when the sensor transmits a number of the past outputs in a packet. The approach in this book is similar but it also addresses the stability of the system in the presence of plant-model mismatch obtaining necessary and sufficient conditions by explicitly computing the time response of the system.

Yook et al. [278] also approach the problem of reducing the bandwidth utilization by making use of a plant model; here the update of the model is event driven as opposed to time driven. The model is updated when any of the states differ from the computed value for more than a certain threshold. Some stability and performance conditions are derived as functions of the plant, threshold, and magnitude of the plant-model mismatch.

There exist also intrinsic relations between the MB-NCS approach and the well-known area of Model Predictive Control (MPC) [81, 192]. In the MPC approach, an explicit model of the system is used to predict the future output behavior; a tracking error is defined using this prediction and the desired reference and the control action are computed online. The purpose of the control action is to drive the state of the system to a reference position in an optimal fashion while satisfying the existing constraints. The authors of [98, 99, 219] introduced intermittent control in the context of MPC to take account of the open-loop inter-sample behavior of an

underlying continuous-time system. Recent work on NCS has been reported where model predictive controllers have been successfully applied to deal with the usual input and output constraints, but also consider the bandwidth limitations of the network. The additional constraint aims to reduce the traffic in the communication network by using the explicit model to generate the appropriate input in the absence of continuous feedback. The formulation proposed by Bernardini and Bemporad [24] reduces communication rate in the network using MPC techniques; a similar idea is also presented by Zhang et al. [288]. This problem was motivated by the implementation of wireless sensors that are supposed to consume much more energy in broadcasting the information than other tasks such computing and data reception. The solution can be applied to a more general class of networked systems for which a reduction of communication rate is desired. Varutti et al. [255] apply MPC based on events to stabilize a nonlinear continuous-time system; they extended their approach to compensate for time delays and packet losses when the system and the model predictive controller are implemented using a communication network. A common feature in the last few approaches [24, 255, 288] is that a mismatch between the available model and the real dynamics of the plant is not considered; their work is developed assuming the model and the plant dynamics are the same. In traditional MPC this may not be an important problem since we can continuously access the real state, but in NCS due to delays, packet dropouts, and bandwidth limitations we need to consider in one way or another the always present uncertainty in the model that, in the above cases, may produce significant error in the predictions.

Another control technique known as Internal Model Control (IMC) [193] uses a plant model to subtract the effect of the manipulated variables from the plant output. That is, assuming the plant is stable, the model is used to obtain a measure of the disturbances affecting the system and the inaccuracies of the model. This signal is then used to feed the IMC controller. For unstable plants this approach leads to serious problems, limiting their use in many applications.

Lately, the MB-NCS framework introduced in [188] has been used for stabilization of coupled subsystems using periodic updates [66] and also using event-triggered control techniques [241].

A similar model-based approach has been developed by Lunze and Lehman [165]. In their approach, the model is assumed to match the dynamics of the system exactly; however, the system is subject to unknown input disturbances. The main idea of the approach in [165] is the same as in the ideas of this book, that is, to use the nominal model to generate estimates of the current state of the system. Since the system is subject to unknown disturbances and the model is executed with zero input disturbance, then a difference between the states is expected and the sensor updates transmitted over a digital communication network are used to reset this difference between the states of the plant and of the model. The same authors have extended this approach to consider the output feedback, quantization, and network delay cases.

1.3 Event-Triggered Control

Event-triggered control represents a new method for sampling of signals that has gained substantial popularity in the control systems community. In event-triggered broadcasting [10, 11, 246, 247, 261, 262, 264, 265] a subsystem sends its local state to the network only when it is necessary, that is, when a measure of the local subsystem state error is above a specified threshold. Event-triggered control schemes offer a new point of view, with respect to conventional time-driven strategies, on how information could be sampled for control purposes.

One of the works that laid the foundations for this type of sampling is [11]; it provided an interesting comparison between conventional time-driven (Riemann) sampling and the new event-driven (Lebesgue) sampling, emphasizing the practical advantages of the latter. It also provided preliminary analysis of simple systems controlled using this technique. Tabuada [247] showed, more formally, the stabilizing properties of the event-triggered control strategy. He presented a triggering condition based on the norms of the state and the state error $e = x(t_i) - x(t)$, that is, the last measured state minus the current state of the system. This means that the measurement received in the controller node is held constant until a new measurement arrives; when this happens, the error is reset to zero and starts growing until it triggers a new execution or measurement update. In the case that delays are not negligible, the control task should be executed before the regular (no delay) execution condition takes place in order to account for those time delays, but the control task should not be executed too soon and provoke accumulation points. This work also provided lower bounds on inter-execution time intervals avoiding such situation. Accumulation points, also called Zeno behavior [290] due to the hybrid nature of a networked system having plants and controllers with continuous-time dynamics and discrete-time updates, represent a fundamental problem in event-triggered control. Zeno behavior in event-triggered control specifically refers to possible case in which the sensor attempts to transmit an infinite number of updates to the controller in a finite period of time. For any event-triggered control approach that involves continuous-time systems, it is essential to guarantee that such behavior will never occur. Event-triggered strategies for discrete-time systems have also been studied; in this situation the triggering conditions are evaluated only once per system transition then the minimum inter-execution time interval is given by the underlying discrete-time period of the system.

The stability conditions provided in [247] are only sufficient and, in many cases, conservative execution intervals are obtained. Wang and Lemmon [264] presented a different method to design stabilizing controllers based on the event-triggered control strategy by noting that the closed-loop system Lyapunov function V needs not to be monotonically decreasing for all time but an appropriate subsequence of V needs to be. In this case the triggering condition is based on the Lyapunov function and not in error norms. Longer inter-execution intervals are obtained using this event-driven controller.

The papers [58, 59] discussed extensions involving event-triggered control to systems with output feedback and implementing decentralized controllers. The authors use an impulsive system approach to model the system, controller, and event-based communication scheme. Stability and performance were analyzed using LMIs.

Considering distributed systems, Yook et al. [278] proposed a framework in which all nodes in a distributed system have identical estimators of states of all remote nodes and an estimator for their own state. The estimated values of the remote outputs are used for control and each node compares its actual state value to its estimate. If the error is greater than an established threshold, the node needs to broadcast the true value of its state to the rest of the nodes.

In [102, 104, 261] the authors offered communication protocols based on events that stabilize the coupled systems in the presence of network delays and packet losses. Other references discussing decentralized implementations of event-triggered control [177, 178, 262, 265].

Event-triggered control and passivity techniques have been used for control of systems with output feedback and subject to time-varying network induced delays [280]. Combinations of periodic sampling and event-triggered control have also been studied. In [105–107] the conditions that trigger events of continuous-time systems do not need to be verified continuously but only at discrete and periodic time instants.

Recent work has applied event-triggered control strategies in cooperative control. The references [52–54, 95, 96, 228, 281] considered the event-triggered consensus problem. Consensus problems have been widely studied assuming continuous communication between agents and its neighbors [127, 194, 216] and using sampled-data implementations [35, 36]. Although the sampled-data implementation does not require continuous communication between agents, it does require synchronization in order for all agents to broadcast their measurements at the same sampling instants. The event-triggered control techniques described in [95, 96, 228, 281] offer a higher level of decentralization since the agents are permitted to select, independently from one another, their own broadcasting instants. The work in [95] considered agents described by continuous-time single integrator dynamics and extended to the case of measurement quantization. The paper [228] considered both, single and double integrator dynamics. The case of event-triggered synchronization of multi-agent systems with general linear dynamics was addressed in [96].

In a similar topic [281] provided conditions for output synchronization of multiple systems described by more general passive dynamics. In an extension, quantization of measurements was considered as well and similar conditions for output synchronization were derived.

A closely related approach to event-triggered control is self-triggered control [1, 2, 9, 175, 176, 205, 263]. This technique also implements non-periodic measurement of the state and computation of the control law. The main difference with respect to event-triggered control is that a measure of the state is not being compared constantly against a predefined threshold. Instead, the current state measurement is used to determine its next deadline, i.e., the next time that the

sensor is required to send a measurement to the controller. A trade-off between increased complex computations and hardware that is necessary for continuously sensing and comparing is made in the sensor and controller nodes by applying this new control strategy. That is, the sensor does not need to continuously measure the state, compute the error, and compare this error against the threshold. Now, it only has to measure the state at some specific times. In the other side of the network, the controller now has to perform more intense computations every time that a measurement arrives in order to obtain the next time that a measurement is needed. Accurate models of the system and disturbances are needed in order to obtain good performance concerning the computation of the inter-sampling intervals.

Part I
Stability

Chapter 2
Model-Based Control Systems: Stability

In this book we study networked control systems that make explicit use of existing knowledge of the plant dynamics, encapsulated in the mathematical model of the plant, to enhance the performance of the system. This class of networked systems is called Model-Based Networked Control Systems (MB-NCS). The performance of a networked control system depends on the performance of the communication network in addition to traditional control systems performance measures. The bandwidth of the communication network used by the control system is of major concern, since other control and data acquisition systems will typically be sharing the same digital communication network.

It turns out that stability margins, controller robustness, and other stability and performance measures may be significantly improved when knowledge of the plant dynamics is explicitly used to predict plant behavior. Note that the plant model is always used to design controllers in standard control design. The difference here is that the plant model is used explicitly in the controller implementation to great advantage. This is possible today because existing inexpensive computation power allows the simulation of the model of the plant in real time.

In this chapter we lay the foundations for the type of networked architecture that we call MB-NCS. We also provide a thorough analysis of the behavior of the system. The focus of the chapter is in obtaining conditions that result in a stable networked system. Stabilization depends on the chosen control gain, the accuracy of the model, and the update interval. We derive necessary and sufficient stabilizing conditions for both continuous and discrete-time linear time-invariant systems. In the following chapters we significantly extend these results to consider different scenarios, for instance, we include the more general case when only a linear combination of the states (system output) is available for measurement and we also consider the effect of network induced delays when state feedback is possible.

The contents of this chapter are as follows: In Sect. 2.1 the Model-Based Control architecture is introduced. In Sect. 2.2 the continuous-time case with state feedback is considered and in Sect. 2.3 the discrete-time case is addressed. Alternative stability conditions are offered in Sect. 2.4. Notes and references are in Sect. 2.5.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_2,

2.1 Fundamentals: Model-Based Control Architecture

We consider here the control of a linear time-invariant dynamical system where the state sensors are connected to controllers/actuators via a network; more complex models are considered in later chapters. The main goal is to reduce the network usage using knowledge of the plant dynamics. Specifically, the controller uses an explicit model of the plant that approximates the plant dynamics and makes possible the stabilization of the plant even under slow network conditions. Although in principle, we can use the same framework to study the problem of packet dropouts in NCS, the aim here is to purposely avoid frequent broadcasting of unnecessary sensor measurements so to reduce traffic in the network as much as possible, which in turn reduces the presence of the problems associated with high network load such as packet collisions and network induced delays.

We will concentrate on characterizing the time interval between successive transmissions of data from the sensor to the controller/actuator (update intervals of the state of the model); Transmission of data from sensor to controller and from controller to actuator is considered later in the book. *Our goal in this chapter will be to identify the maximum update intervals that can be used to transmit measurement updates between the sensor and the actuator while keeping the system stabl*e. This will reduce the bandwidth required from the network and will free it for other tasks such as other control loops using the network and/or non-control information exchange. In order to increase the update time we will use the knowledge we have of the plant dynamics. The plant model is used at the controller/actuator side to approximate the plant behavior. The sensor is able to reduce the rate at which it transmits data, since the model can provide information to generate appropriate control inputs while the systems is running open loop. *Note that in standard digital control a zero-order-hold keeps the input constant in between samples. Here the input between samples is calculated based on the plant model and it is reasonable to assume that it will be better suited than the constant input.*

The main idea is to update the state of the model using the actual state of the plant provided by the sensor. The rest of the time the control action is based on a plant model that is incorporated in the controller/actuator and is running open loop for a period of h seconds. The control architecture is shown in Fig. 2.1.

In our control architecture having knowledge of the plant at the actuator side enables us to run the plant in open loop, while the update of the model state provides the closed-loop information needed to overcome model uncertainties and plant disturbances. In the remaining of this chapter we assume that all states can be measured and these measurements can be transmitted through the network to update the model of the system in the controller node. The more general case when only a linear combination of the states can be sensed is studied in the following chapter together with the network induced delays case. In this chapter we provide necessary and sufficient conditions for stability that result in a maximum update time, which depends mainly on the model inaccuracies but also on the designed control gain.

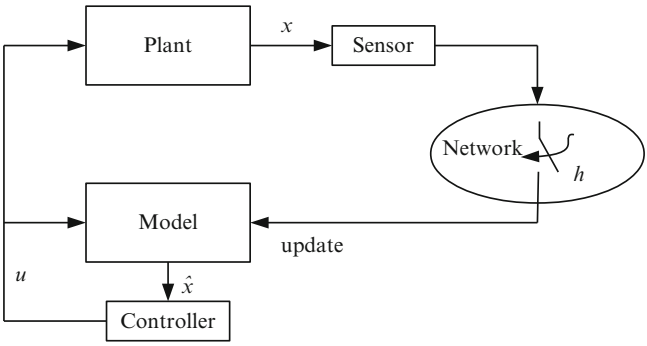

Fig. 2.1 The model-based networked control system (MB-NCS) architecture

2.2 Continuous-Time LTI Systems: State Feedback

In this section, we introduce the foundations of our Model-Based approach. We consider multi-input, multi-output linear time-invariant continuous-time systems and their state variable representations, and we assume a constant linear state feedback control law. Necessary and sufficient conditions are derived for the stability of the compensated system in Theorem 2.3, the main result of the section. Illustrative examples are also included. Discrete-time systems are considered in the next section and output feedback in the next chapter.

If all the states are available for measurement, then the sensors can send this information through the network to update the model's vector state. We will assume that the compensated model is stable, which is typical in control systems, and that the transportation delay is negligible, which is completely justifiable in most of the popular network standards like CAN bus or Ethernet. We will assume that the frequency at which the network updates the state in the controller is constant. The goal is to find the largest constant update period at which the network must update the model state in the controller for stability, that is, we are seeking an upper bound for h the update time. Usual assumptions in the literature include requiring a stable plant or in the case of a discrete controller, a smaller update time than the sampling time. Here we do not make any of these assumptions. The original plant may be open-loop unstable.

Consider the control system of Fig. 2.1 where the plant, the plant model and the controller are described by:

$$\begin{aligned} &\text{Plant :} && \dot{x} = Ax + Bu, \\ &\text{Plant model :} && \dot{\hat{x}} = \hat{A}\,\hat{x} + \hat{B}\,u, \\ &\text{Controller :} && u = K\hat{x}\,. \end{aligned} \tag{2.1}$$

Since the sensor has the full state vector available, its function will be to send the state information through the network every h seconds. The state error is defined as:

$$\text{State Error :} \quad e = x - \hat{x}\,, \tag{2.2}$$

and represents the difference between the plant state and the model state. The modeling error matrices: $\widetilde{A} = A - \hat{A}$ and $\widetilde{B} = B - \hat{B}$ represent the difference between the plant and the model. The periodic update time instants are denoted by t_k, where

$$t_k - t_{k-1} = h \quad \text{for} \quad k = 1, 2, \ldots \tag{2.3}$$

(here, h is a constant). The choice of h, being a constant, is simple to implement and also results in a simple analysis procedure as shown below. Update intervals that are not constant will be considered as well; in Chap. 5, we address time-varying update intervals and in Chap. 6 we study event (error)-based update intervals.

Since the model state is updated every t_k s,

$$e(t_k) = 0 \quad \text{for} \quad k = 1, 2, \ldots, . \tag{2.4}$$

This resetting of the state error at every update time instant is a key characteristic of our control system. Now for $t \in [t_k, t_{k+1})$, we have that: $u = K\hat{x}$ so the overall system is described by

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{2.5}$$

with initial conditions $\hat{x}(t_k) = x(t_k)$.

Introducing the error $e(t) = x(t) - \hat{x}(t)$, it is easy to see that the dynamics of the overall system for $t \in [t_k, t_{k+1})$ can be described by

$$\begin{aligned} &\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \\ &\begin{bmatrix} x(t_k) \\ e(t_k) \end{bmatrix} = \begin{bmatrix} x(t_k^-) \\ 0 \end{bmatrix}, \\ &\quad \forall t \in [t_k, t_{k+1}), \quad \text{with} \quad t_{k+1} - t_k = h. \end{aligned} \tag{2.6}$$

Define the augmented state vector $z(t)$ and the open-loop augmented state matrix Λ:

$$z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \tag{2.7}$$

$$\Lambda = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \tag{2.8}$$

so that (2.6) can be rewritten as

$$\dot{z} = \Lambda z \quad \text{for} \quad t \in [t_k, t_{k+1}). \tag{2.9}$$

We will now express $z(t)$ in terms of the initial condition $x(t_0)$. Then we will show under what conditions the system will be stable.

Proposition 2.1 *The system described by (2.6) with initial conditions* $z(t_0) = \begin{bmatrix} x(t_0) \\ 0 \end{bmatrix} = z_0$ *has the following response:*

$$\begin{aligned} z(t) &= e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \\ t &\in [t_k, t_{k+1}), \quad \text{with} \quad t_{k+1} - t_k = h. \end{aligned} \tag{2.10}$$

Proof On the interval $t \in [t_k, t_{k+1})$, the system response is:

$$z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = e^{\Lambda(t-t_k)} \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix} = e^{\Lambda(t-t_k)} z(t_k). \tag{2.11}$$

Now, note that at times t_k, $z(t_k) = \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix}$, that is, the error is reset to 0, that is, $e(t_k) = 0$ for $k = 1,2,\ldots,$. We can represent this by

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z(t_k^-). \tag{2.12}$$

Using (2.11) to calculate $z(t_k^-)$ we obtain

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1}). \tag{2.13}$$

In view of (2.11) we have that if at time $t = t_0$, $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ is the initial condition, then:

$$\begin{aligned} z(t) &= e^{\Lambda(t-t_k)} z(t_k) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-1}) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} z(t_{k-2}) \\ &= e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k z_0 \end{aligned} \tag{2.14}$$

Now we know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h}$ is of the form $\begin{bmatrix} M & N \\ 0 & 0 \end{bmatrix}$ and so $\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k$ has the form $\begin{bmatrix} M^K & P \\ 0 & 0 \end{bmatrix}$. Additionally we note the special form of the initial condition $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ so that:

$$\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} = \begin{bmatrix} M^k x_0 & 0 \\ 0 & 0 \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix}. \quad (2.15)$$

In view of (2.15) it is clear that we can represent the system response as in (2.10):

$$z(t) = e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0$$
$$t \in [t_k, t_{k+1}), \quad \text{with} \quad t_{k+1} - t_k = h.$$

♦

A necessary and sufficient condition for stability of the networked system will now be presented. For this, the following definition for global exponential stability [5] is needed.

Definition 2.2 The equilibrium $z = 0$ of a system described by $\dot{z} = f(t, z)$ with initial condition $z(t_0) = z_0$ is exponentially stable at large (or globally) if there exists $\alpha > 0$ and for any $\beta > 0$, there exists $k(\beta) > 0$ such that the solution

$$\|\phi(t, t_0, z_0)\| \leq k(\beta)\|z_0\| e^{-\alpha(t-t_0)}, \qquad \forall t \geq t_0 \quad (2.16)$$

whenever $\|z_0\| < \beta$.

With this definition of stability we state the following theorem characterizing the necessary and sufficient conditions for the system described by (2.6) to have global exponential stability around the solution $z = 0$. The norm used here is the 2-norm but any other consistent norm can also be used.

Theorem 2.3 *The system described by (2.6) is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are strictly inside the unit circle.*

Proof Sufficiency. Taking the norm of the solution described in (2.10), (in Proposition 2.1):

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\|$$
$$\leq \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\|. \tag{2.17}$$

Now let us analyze the first term on the right-hand side of (2.17):

$$\left\| e^{\Lambda(t-t_k)} \right\| \leq 1 + (t-t_k)\overline{\sigma}(\Lambda) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda)^2 \ldots = e^{\overline{\sigma}(\Lambda)(t-t_k)} \leq e^{\overline{\sigma}(\Lambda)h}$$
$$= K_1 \tag{2.18}$$

where $\overline{\sigma}(\Lambda)$ is the largest singular value of Λ. In general this term can always be bounded since the time difference $t - t_k$ is always smaller than h. In other words even when Λ has eigenvalues with positive real part, $\left\| e^{\Lambda(t-t_k)} \right\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the term $\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 e^{-\alpha_1 k} \tag{2.19}$$

with $K_2, \alpha_1 > 0$. Since k is a function of time, we can bound the right term of (2.19) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} t} = K_3 e^{-\alpha t} \tag{2.20}$$

with $K_3,\ \alpha > 0$. So from (2.17) using (2.18) and (2.20) we can conclude:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \leq K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\|. \tag{2.21}$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the state feedback MB-NCS is stable and that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, the sequence of periodic samples of the response should converge to 0 with time. We will take the sample at times t_{k+1}^-, that is, just before the update. We will concentrate on a

specific term: the state of the plant $x(t_{k+1}^-)$, which is the first element of $z(t_{k+1}^-)$. We will call $x(t_{k+1}^-)$, $\xi(k)$.

Now assume $e^{\Lambda\tau}$ has the following form:

$$e^{\Lambda\tau} = \begin{bmatrix} W(\tau) & X(\tau) \\ Y(\tau) & Z(\tau) \end{bmatrix}. \tag{2.22}$$

In view of (2.11) we can express the response $z(t)$ as:

$$\begin{aligned} & e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \\ &= \begin{bmatrix} W(t-t_k) & X(t-t_k) \\ Y(t-t_k) & Z(t-t_k) \end{bmatrix} \begin{bmatrix} (W(h))^k & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(t-t_k)(W(h))^k & 0 \\ Y(t-t_k)(W(h))^k & 0 \end{bmatrix} z_0. \end{aligned} \tag{2.23}$$

Now the values of the response at times t_{k+1}^-, that is just before the update, are

$$z\left(t_{k+1}^-\right) = \begin{bmatrix} W(h)(W(h))^k & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0. \tag{2.24}$$

We also know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This implies that the first element of $z(t_{k+1}^-)$, which we call $\xi(k)$, will in general grow with k. In other words we cannot ensure $\xi(k)$ will converge to 0 for general initial condition x_0. That is

$$\left\| x\left(t_{k+1}^-\right) \right\| = \|\xi(k)\| = \left\| (W(h))^{k+1} x_0 \right\| \to \infty \quad \text{as} \quad k \to \infty \tag{2.25}$$

which implies that the system cannot be stable, and thus we have a contradiction. ♦

Example 2.1 Consider the following unstable system (plant) dynamics:

$$A = \begin{bmatrix} 0.14 & 1.25 \\ 0 & 0.08 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1.07 \end{bmatrix}. \tag{2.26}$$

The exact parameters of the plant are unknown. In general, a model containing nominal parameters of the real parameters is available. Assume that (2.26) is modeled using the following nominal parameters

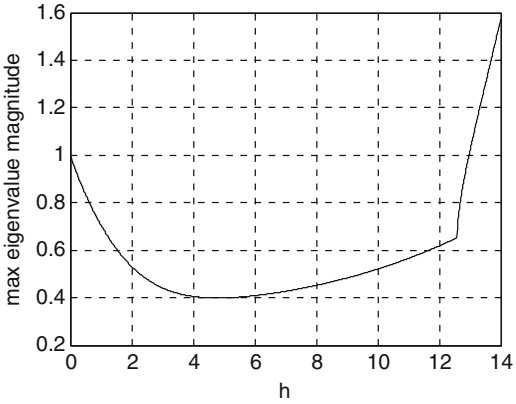

Fig. 2.2 Maximum eigenvalue magnitude of the test matrix M vs. the update time h

$$\hat{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.27}$$

We will use the state feedback controller with control gain given by $K = [-0.2 \ -0.9]$. We now search for the largest h such that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has its eigenvalues inside the unit circle. Figure 2.2 shows the eigenvalue with maximum magnitude of this matrix. From this search we can use values of $h < 12.96$ s in order to design a stable networked system using the model-based approach. Figure 2.3 shows the response for different values of h and the same initial conditions $x_0 = [1 \ -1]^T$, while the model initial conditions are set equal to 0. Note that the time scale is different for the responses shown in the second column in order to show that the response is stable for $h = 12.9$ s and unstable for $h = 13$ s, as expected.

In order to draw a comparison with other approaches commonly used in networked control, we now use a model of the system equivalent to a zero-order-hold (ZOH) model. This is also the same type of implementation used in traditional sampled-data control systems. Here, the ZOH holds the value of the most recent measurement constant until a new measurement arrives to update the controller. We can use the results of this section to find the largest possible value of the update period under this scenario since the ZOH is just a special case of our approach that can be easily modeled using the following:

$$\hat{A} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.28}$$

From Fig. 2.4 we can see that the admissible values of the update period that preserve stability are $h < 2.13$. Figure 2.5 shows the response for two choices $h = 1.5$ (stable) and $h = 2.15$ (unstable). The initial conditions are the same in both cases. We can see in this example that there is a significant difference in the range of values for the update periods when using the ZOH model (2.28) and using

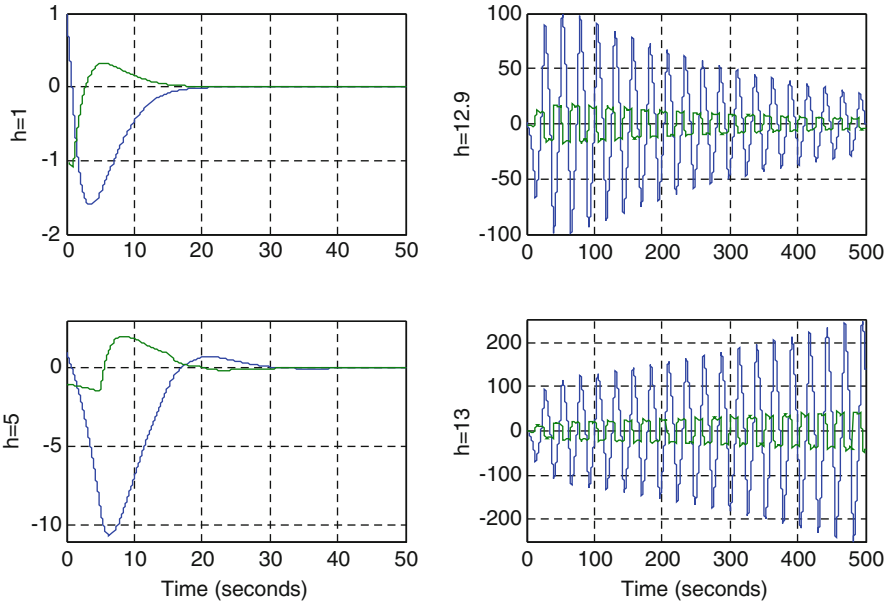

Fig. 2.3 System response for different values of h. For $h = 1$, $h = 5$, and $h = 12.9$ s the system is stable. For $h = 13$ s the system is unstable

the nominal model (2.27). Even though the nominal model does not represent the exact dynamics of the system, it provides a more accurate estimation of the real state between updates that allows for an implementation of the networked system with longer update periods. This is a clear benefit of the model-based approach in reducing the necessary bandwidth for stability.

Example 2.2 Applicability of State Feedback MB-NCS Results. Regarding applicability to real world cases, we see that, as stated in Theorem 2.3, the stability properties of a networked control system can be determined by studying the eigenvalues of the matrix:

$$M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \text{ where } \Lambda = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}. \tag{2.29}$$

The matrix Λ can also be expressed in terms of the model and the errors only:

$$\Lambda = \begin{bmatrix} \hat{A} + \widetilde{A} + \left(\hat{B} + \widetilde{B}\right)K & -\left(\hat{B} + \widetilde{B}\right)K \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}. \tag{2.30}$$

In real applications the state-space model is usually obtained by studying the structure and behavior of the plant. The uncertainties can frequently be expressed as

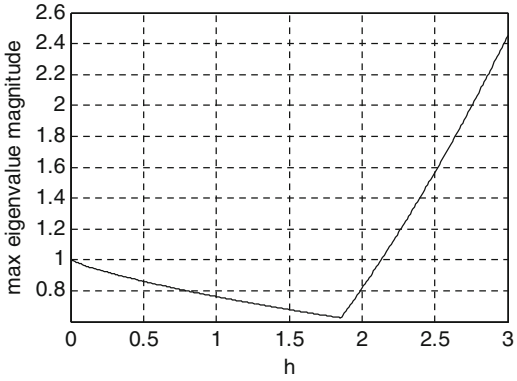

Fig. 2.4 Maximum eigenvalue magnitude of the test matrix M vs. the update time h for the ZOH model

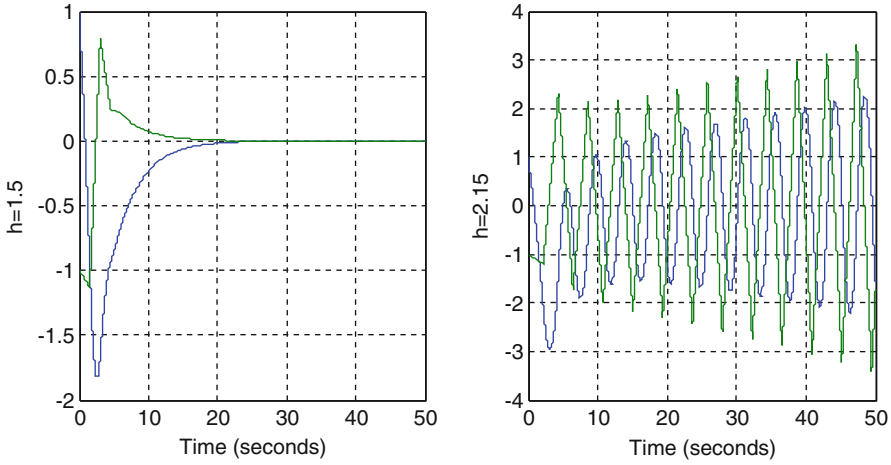

Fig. 2.5 System response for different values of h using a ZOH model. For $h = 1.5$ s the system is stable. For $h = 2.15$ s the system is unstable

tolerances over the different measured parameter values of the plant. This can be mapped into structured or parametric uncertainties on the state-space matrices. The following is an example that provides insight on how the theorem can be applied if two entries on the A matrix of the model can vary within a certain interval.

$$\begin{aligned}
&\text{model}: \quad \hat{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \\
&\text{plant}: \quad A = \begin{bmatrix} 0 & 1+\widetilde{a}_{12} \\ 0+\widetilde{a}_{21} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \\
&\qquad \text{with } \widetilde{a}_{12} = [-0.5, 0.5], \widetilde{a}_{21} = [-0.5, 0.5] \\
&\text{controller}: K = [-1, -2].
\end{aligned} \tag{2.31}$$

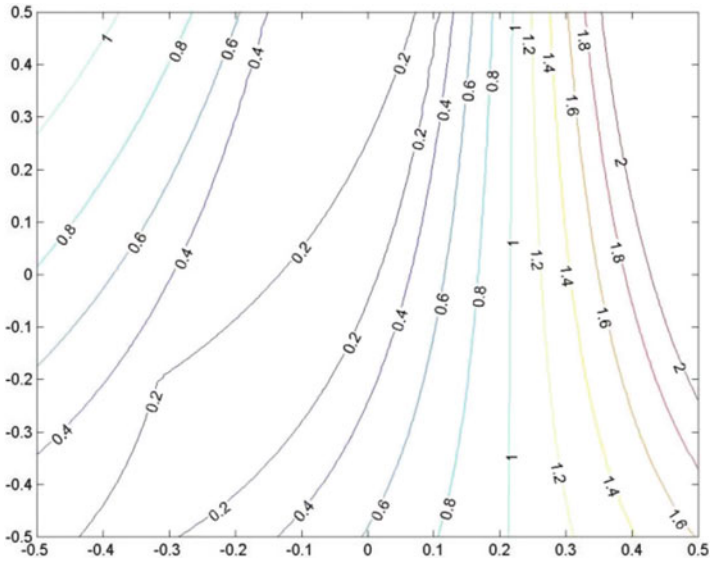

Fig. 2.6 Contour plot maximum eigenvalue. Magnitude vs. model error

The system will now be tested for an update time of $h = 2.5$ time units. Figure 2.6 represents a contour plot where the contour of height equal to 1 separates the stable and unstable regions. In Fig. 2.7 we have plotted the surface representing the maximum eigenvalue magnitude for the test matrix M as a function of the (1,2) and (2,1) entries and for a given selection of values of those entries.

It is easy to isolate the stable and unstable regions in the uncertainty parameter plane. The stable region is between the lines labeled as 1 in Fig. 2.6. The same procedure can be used in real applications to verify the stability of a networked control system with a certain model and update time. The procedure can be repeated for different values of uncertainties on as many elements of A as needed and for different update intervals as well.

Example 2.3 In this example we consider the instrument servo (dc motor driving an inertial load) dynamics from Example 6A in [77]

$$\begin{bmatrix} \dot{e} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix} \begin{bmatrix} e \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \end{bmatrix} u \tag{2.32}$$

where $e = \theta - \theta_r$ represents the error between the current angular position θ and the desired position θ_r, where the desired position is assumed to be constant, ω is the angular velocity, and u is the applied voltage. The parameters α and β represent constants that depend on the physical parameters of the motor and load.

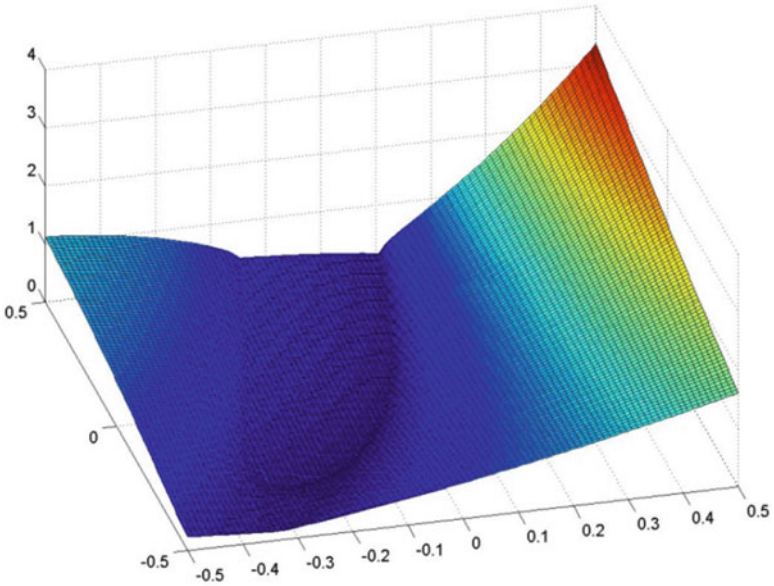

Fig. 2.7 Maximum eigenvalue magnitude vs. model error

The nominal parameters are $\hat{\alpha} = 1$, $\hat{\beta} = 3$. The real parameters are given by $\alpha = 0.75$, $\beta = 2.58$. The model parameters are used for controller design: $K = [-0.1667 \ -0.1667]$. According to the eigenvalue search described in this section, the networked system is stable for any value $h > 0$; however the transient response will significantly differ for different choices of the update period. Figure 2.8 shows the response for two different update periods where we can see that the networked system converges more rapidly for $h = 1$ s than for $h = 10$ s.

Remark Models in which uncertainty is represented in terms of norms as a ball around the model can also be derived, but the conditions for stability would only be sufficient. The results in this section then offer a less conservative approach that can be readily applied to real applications as well. The conditions in Theorem 2.3 assume that noiseless measurements are transmitted at the update instants t_k. In Chap. 12 we consider noisy measurements within the MB-NCS framework.

Remark Optimization problem. The design of optimal controllers for MB-NCS represents an interesting research problem. In addition to optimizing the response of the control systems and penalizing the excessive use of control effort, we should also weight the use of network resources. The overall problem requires not only the design of the optimal control law but also the design of the optimal scheduling law.

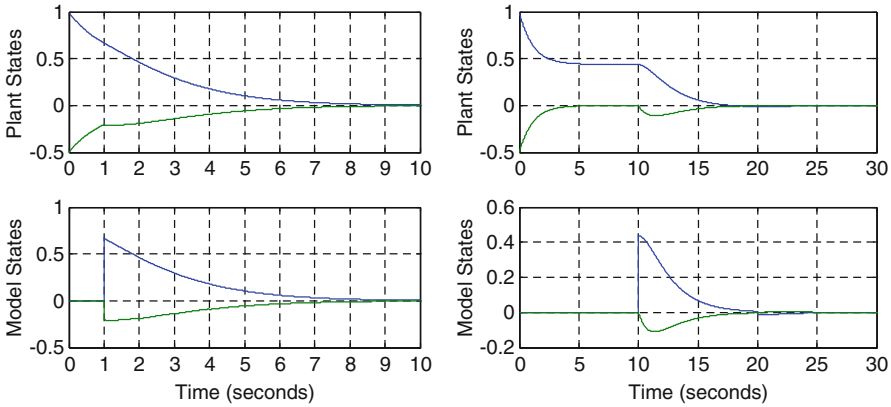

Fig. 2.8 Plant and model states. *Left*: $h = 1$ s. *Right*: $h = 10$ s

The scheduling law can, in general, be a time-varying transmission policy. Chapter 9 addresses this type of problem by using the following cost function:

$$\min_{u,\beta} J = \frac{1}{2}x^T(T)Gx(T) + \int_0^T \frac{1}{2}\left(x^T(t)Qx(t) + u^T(t)Ru(t)\right)dt + \text{cost of transmission.} \tag{2.33}$$

Note that this problem considers the uncertain nature of the model parameters with respect to the real parameters of the system which significantly increases the complexity.

Remark It is of interest to study the eigenvalues of the networked control system matrix $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ and express them, if possible, in terms of h and the error in the plant model $\widetilde{A}$ and $\widetilde{B}$. To do so, we first apply a transformation to the matrix Λ to obtain a diagonal matrix that will facilitate the computation of the exponential part.

We choose the transformation $P = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$ with inverse $P^{-1} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$. Applying this transformation on Λ we obtain:

$$\overline{\Lambda} = P\Lambda P^{-1} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix}$$

Using this transformation we obtain:

$$M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} P^{-1} e^{\overline{\Lambda} h} P \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\overline{\Lambda} h} \begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}$$

The matrix exponential $e^{\overline{\Lambda}h}$ may be found directly or by considering a Laplace transform-based approach. For the latter approach, we will change the variable h to t.

$$
\begin{aligned}
L\{M\} &= L\left\{ \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\overline{\Lambda}t} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right\} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} L\left\{ e^{\overline{\Lambda}t} \right\} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} (sI-A)^{-1} & (sI-A)^{-1}BK\left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1} \\ 0 & \left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix} \\
&= \begin{bmatrix} (sI-A)^{-1} + (sI-A)^{-1}BK\left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1} & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}
$$

Note that only the upper left block contains the critical eigenvalues. Using the inverse Laplace transform:

$$
\begin{aligned}
& L^{-1}\left\{(sI-A)^{-1} + (sI-A)^{-1}BK\left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1}\right\} \\
&= L^{-1}\left\{(sI-A)^{-1}\left(I+BK\left(sI-\hat{A}-\hat{B}K\right)^{-1}\right)\right\} \\
&= L^{-1}\left\{(sI-A)^{-1}\left(sI-\hat{A}-\hat{B}K+BK\right)\left(sI-\hat{A}-\hat{B}K\right)^{-1}\right\} \\
&= L^{-1}\left\{(sI-A)^{-1}\left(sI-A+\widetilde{A}+\widetilde{B}K\right)\left(sI-\hat{A}-\hat{B}K\right)^{-1}\right\} \\
&= L^{-1}\left\{\left(I+(sI-A)^{-1}\left(\widetilde{A}+\widetilde{B}K\right)\right)\left(sI-\hat{A}-\hat{B}K\right)^{-1}\right\} \\
&= L^{-1}\left\{\left(sI-\hat{A}-\hat{B}K\right)^{-1} + (sI-A)^{-1}\left(\widetilde{A}+\widetilde{B}K\right)\left(sI-\hat{A}-\hat{B}K\right)^{-1}\right\} \\
&= e^{\left(\hat{A}+\hat{B}K\right)t} + e^{At}\int_0^t e^{-A\tau}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau
\end{aligned}
$$

That is the eigenvalues in question are exactly the eigenvalues of:

$$
N = e^{\left(\hat{A}+\hat{B}K\right)h} + e^{Ah}\int_0^h e^{-A\tau}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau
$$

Then the eigenvalues of $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ are inside the unit circle if and only if the eigenvalues of N are inside the unit circle. One can gain a better insight of the system by observing the structure of N. To start with, we observe that the eigenvalues of the compensated model appear in the first term of N. In that sense we can see the term $\Delta = e^{Ah}\int_0^h e^{-A\tau}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau$ as a perturbation over the desired eigenvalues. The important variables in the perturbation term Δ are the uncertainties and the update intervals. In general, for unstable open-loop systems, the perturbation Δ increases as either the update intervals or the model uncertainties increase. On the other hand, even if the eigenvalues of the original plant were unstable the perturbation Δ can be made small enough by having $\widetilde{A}+\widetilde{B}K$ small and thus minimizing their impact over the eigenvalues of the compensated plant.

Remark An important discussion concerning the control of continuous-time systems using the MB-NCS framework is the following. When the update interval h is getting small and approaches 0, then according to the analysis just described in this section, the model/controller tries to update its internal state very fast or in other words it tries to switch infinitely many times in finite intervals of time without letting the model to execute and estimate the plant state. If we compute the eigenvalues of the matrix given in Theorem 2.3 (or the eigenvalues of N in the previous remark) for $h = 0$ we obtain eigenvalues equal to one which represent this behavior of attempting to update the same value over and over again. However, in a physical continuous-time system we cannot update the controller infinitely fast in real time (compare to a typical sampled-data system in the same situation) and undesired system response may occur. The case when $h = 0$ may also be considered in a different way. The case when $h = 0$ corresponds to the controller receiving continuous feedback measurements from the sensor; and instead of updating the model many times we can use directly the plant measurements to control the system. We may also consider the networked system as operating in two modes: In closed-loop mode, that is, when the controller receives continuous feedback from the sensor ($h = 0$), the model is unnecessary and the feedback information is immediately used by the controller to compute the control input. In open-loop mode, the model is now used to estimate the state of the plant using the last received feedback value of the real state as the initial condition. We will study this case in more detail in Chap. 4.

2.3 Discrete-Time LTI Systems: State Feedback

In this section, we present results for the discrete-time case. They are analogous to the continuous-time results of the previous section. We consider multi-input, multi-output linear time-invariant discrete-time systems and their state variable representations, and we assume constant linear feedback control law. Necessary and sufficient conditions are derived for the stability of the compensated systems in Theorem 2.6, the main result of the section. Illustrative examples are included. Output feedback is considered in the next chapter.

So far we have studied continuous-time plants. We will extend our results to discrete-time plants of the form:

$$x(n+1) = Ax(n) + Bu(n) \tag{2.34}$$

with model

$$\hat{x}\,(n+1) = \hat{A}\,\hat{x}\,(n) + \hat{B}\,u(n) \tag{2.35}$$

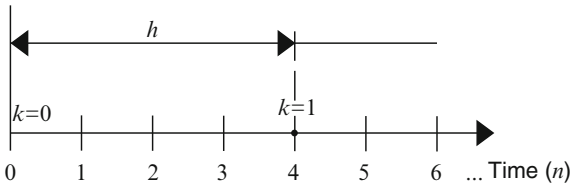

Fig. 2.9 Representation of the state updates for the discrete-time case

where $n = 0,1,2,\ldots$ The control input and the state error are defined, respectively, by:

$$\begin{aligned} u(n) &= K\hat{x}\left(n\right) \\ e(n) &= x\left(n\right) - \hat{x}\left(n\right). \end{aligned} \tag{2.36}$$

There are some assumptions we need to make before we carry our results over to the discrete-time domain. First, in order to have consistency in updates from the sensor side to the actuator side, we must ensure that both the sensor and the actuator/controller are synchronized in the sense that both will carry out their respective tasks at the same sampling time. Moreover state updates will occur only at some of these sampling times. This implies that the update interval h will be an integer number, representing the number of time units between updates of the actuator's model. For example, if $h = 4$, see Fig. 2.9, then the sensor will send measurement updates every 4 time units as indexed by n.

A slightly different way to explain these ideas is by considering two sampling periods. One of them occurs at time instants indexed by n and corresponds to the sampling period of the original plant (2.34). The other one occurs at time instants indexed by hn (for positive integer h) and corresponds to the selected update period at which the sensor sends information to the controller in order to update the state of the model. When the update intervals are constant the networked system can be seen as a linear periodic system. When $h = 1$ the controller is updated at each time index n of the system's clock, i.e., the discrete-time system will operate in closed-loop mode. In Sect. 2.4 we pursue this idea further, but for the moment we will focus on the first explanation provided on the previous paragraph.

The approach that we are going to follow to determine the stability of the networked system is analogous to the one used for continuous plants. The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described by:

$$\begin{bmatrix} x(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}, \tag{2.37}$$
$$\text{for } n \in \left[n_k, n_{k+1}\right), \quad n_{k+1} - n_k = h, \quad \text{and} \quad e(n_{k+1}) = 0.$$

We also use the definitions (2.7) and (2.8) to express (2.37) in compact form as follows:

$$z(n+1) = \Lambda_D z(n) \text{ for } n \in [n_k, n_{k+1}), \quad \text{and} \quad z(n_{k+1}) = \begin{bmatrix} x(n_{k+1}) \\ 0 \end{bmatrix}. \tag{2.38}$$

where

$$z(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}$$

$$\Lambda_D = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}$$

Proposition 2.4 *The system described by* (2.38) *with initial condition* $z(n_0) = \begin{bmatrix} x(n_0) \\ 0 \end{bmatrix} = z_0$, *has the response:*

$$z(n) = {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0, \tag{2.39}$$

$$\text{for } n \in [n_k, n_{k+1}), \quad n_k - n_{k+1} = h.$$

Proof On the interval $n \in [n_k, n_{k+1})$, the system response is

$$z(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} = {\Lambda_D}^{n-n_k} \begin{bmatrix} x(n_k) \\ 0 \end{bmatrix} = {\Lambda_D}^{n-n_k} z(n_k). \tag{2.40}$$

Now, note that at times n_k, $z(n_k) = \begin{bmatrix} x(n_k) \\ 0 \end{bmatrix}$, that is, the error $e(n)$ is reset to 0. We can represent this by

$$z(n_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \breve{z}(n_k). \tag{2.41}$$

Here $\breve{z}(n_k)$ is the value assumed by $z(n)$ when $n = n_k$ using (2.40) for the interval $n \in [n_{k-1}, n_k)$. Using this value of $\breve{z}(n_k)$ we obtain:

$$z(n_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h z(n_{k-1}). \tag{2.42}$$

In view of (2.40) we have that if at time $n = n_0$, $z(n_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$\begin{aligned} z(n) &= {\Lambda_D}^{n-n_k} z(n_k) \\ &= {\Lambda_D}^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} z(n_{k-1}) \\ &= {\Lambda_D}^{n-n_k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} z(n_{k-2}) \\ &\cdots \\ &= {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \right)^k z_0. \end{aligned} \tag{2.43}$$

Now we know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_D^h$ is of the form $\begin{bmatrix} M & N \\ 0 & 0 \end{bmatrix}$ and so $\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \right)^k$ has the form $\begin{bmatrix} M^K & P \\ 0 & 0 \end{bmatrix}$. Additionally we note the special form of the initial condition $z(n_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ so that

$$\begin{aligned} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix} &= \begin{bmatrix} M^k x_0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} M^k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \\ &= \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \begin{bmatrix} x_0 \\ 0 \end{bmatrix}. \end{aligned} \tag{2.44}$$

In view of (2.44) it is clear that the system response as in (2.39) may be written as:

$$z(n) = {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0,$$
$$\text{for } n \in [n_k, n_{k+1}), \quad n_k - n_{k+1} = h.$$

♦

Note that the main difference between this theorem and the continuous version in Proposition 2.1 is in the state transition matrix used for the dynamics of the system in between updates (${\Lambda_D}^{n-n_k}$ instead of $e^{A(t-t_k)}$). We now introduce an exponential global stability definition for the case of discrete plants [131].

Definition 2.5 The equilibrium $z = 0$ of a discrete time system described by $z(n+1) = f(n, z)$ with initial condition $z(n_0) = z_0$ is globally exponentially stable if there exists $\alpha > 0$ and $0 < \gamma < 1$ such that the solution

$$\|z(n)\| \leq \alpha \|z_0\| \gamma^n, \qquad \forall n \geq 0.$$

With this definition of stability we can now state the necessary and sufficient conditions for the exponential global stability of the system described by (2.38). Theorem 2.6 below for the discrete case is the corresponding result to Theorem 2.3 of the previous section for the continuous case. Again the norm used here is the 2-norm but any other consistent norm can also be used.

Theorem 2.6 *The system described by* (2.38) *is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $M_D = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are inside the unit circle.*

Proof Sufficiency. Taking the norm of the solution described as in Proposition 2.4:

$$\begin{aligned} \|z(n)\| &= \left\| {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \\ &\leq \|{\Lambda_D}^{n-n_k}\| \cdot \left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\|. \end{aligned} \tag{2.45}$$

Now, the first term on the right-hand side of (2.45) satisfies:

$$\|{\Lambda_D}^{n-n_k}\| \leq (\overline{\sigma}(\Lambda_D))^{n-n_k} \leq (\overline{\sigma}(\Lambda_D))^h = K_1 \tag{2.46}$$

where $\overline{\sigma}(\Lambda_D)$ is the largest singular value of Λ_D. In general, this term can always be bounded since the time difference $n - n_k$ is always smaller than h. In other words even when Λ_D has eigenvalues with magnitude greater than one, $\|{\Lambda_D}^{n-n_k}\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the term $\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 \gamma_1^k \tag{2.47}$$

with $K_2 > 0,\ 0 < \gamma_1 < 1$.

Also note that k is a function of time so we can express the right term of (2.47) in terms of n:

$$K_2 \gamma_1^k = K_2 \gamma_1^{n/h} = K_2 \left(\gamma_1^{1/h} \right)^n = K_2 \gamma^n \tag{2.48}$$

where $0 < \gamma < 1$ since $h \geq 1$.

So from (2.45) using (2.46) and (2.48) we can conclude:

$$\|z(n)\| = \left\| {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^{k} z_0 \right\| \leq K_1 \cdot K_2 \gamma^n \cdot \|z_0\|. \tag{2.49}$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the system is stable and that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to 0 with time. We will take the sample at times n_{k+1}, in other words, just at the update. Even further we will concentrate on a specific term: the state of the plant $x(n_{k+1})$, which is the first element of $z(n_{k+1})$. We will call $x(n_{k+1})$, $\xi(k+1)$.

Now assume ${\Lambda_D}^{j}$ has the following form:

$${\Lambda_D}^{j} = \begin{bmatrix} W(j) & X(j) \\ Y(j) & Z(j) \end{bmatrix}. \tag{2.50}$$

Then we can express the solution $z(n_{k+1})$ as:

$$\begin{aligned} & {\Lambda_D}^{n-n_k} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^{k} z_0 \\ &= \begin{bmatrix} W(n-n_k) & X(n-n_k) \\ Y(n-n_k) & Z(n-n_k) \end{bmatrix} \begin{bmatrix} (W(h))^{k} & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(n-n_k)(W(h))^{k} & 0 \\ Y(n-n_k)(W(h))^{k} & 0 \end{bmatrix} z_0 \\ & \forall n \in \left[n_k, n_{k+1} \right). \end{aligned} \tag{2.51}$$

Now let us check the values of the solution at times n_{k+1}, that is the update time. We know that because of the update at this time the error is 0, and therefore:

$$z(n_{k+1}) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W(h)(W(h))^{k} & 0 \\ Y(h)(W(h))^{k} & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ 0 & 0 \end{bmatrix} z_0. \tag{2.52}$$

We also know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^{h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$.

This means that the first element of $z(n_{k+1})$, which we call $\xi(k+1)$, will in general grow with k. In other words we can't ensure $\xi(k+1)$ will converge to 0 for general initial condition x_0. That is,

$$\|x(n_{k+1})\| = \|\xi(k+1)\| = \left\|(W(h))^{k+1}x_0\right\| \to \infty \qquad \text{as} \qquad k \to \infty. \tag{2.53}$$

This clearly means that the system cannot be stable, and thus we have a contradiction. ♦

Example 2.4 Consider an example of the full state feedback setup using a discrete-time plant with parameters:

$$A = \begin{bmatrix} 0.89 & 1.23 \\ 0.08 & 0.98 \end{bmatrix}, \quad B = \begin{bmatrix} -0.04 \\ 1.19 \end{bmatrix} \tag{2.54}$$

The exact parameters of the plant are unknown. In general, a model containing nominal parameters of the real parameters is available. Assume that (2.54) is modeled using the following nominal parameters

$$\hat{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.55}$$

We will use the state feedback controller with control gain given by $K = [-0.12 \quad -0.7]$. We now search for the largest h such that $M_D = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} {\Lambda_D}^h \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has its eigenvalues inside the unit circle. Figure 2.10 shows the eigenvalue with maximum magnitude of this matrix. Recall that in the discrete-time case we only search for integer values of the update period equal or greater than 1. In this example we are able to use $h \leq 11$ in order to design a stable networked system using the model-based approach. Figure 2.11 shows the response for $h = 8$ and $h = 12$ and using the same initial conditions $x_0 = [-1 \ \ 0.2]^T$ and the model initial conditions are equal to 0. Linear interpolation is used when plotting the response of the systems. For each choice of h this figure shows the response of the plant and the model as well.

Example 2.5 Consider a discretized version of the instrument servo shown in Example 2.3. The nominal parameters are $\hat{\alpha} = 1$, $\hat{\beta} = 3$. The real parameters are given by $\alpha = -0.75$, $\beta = 2.58$. The discretization period is $T = 0.01$ s. The resulting discrete-time model parameters are used to design the controller: $K = [-30.1502 \ \ -5.8807]$. The same discrete-time model parameters are also used to implement the model in our discrete-time MB-NCS setup. Similar to the continuous-time implementation, stability is obtained for any integer value $h \geq 1$. The difference between implementing longer values of h is given by the transient response as it can be seen in Fig. 2.12 where the plant states converge in significantly longer time for $h = 50$ than for $h = 10$.

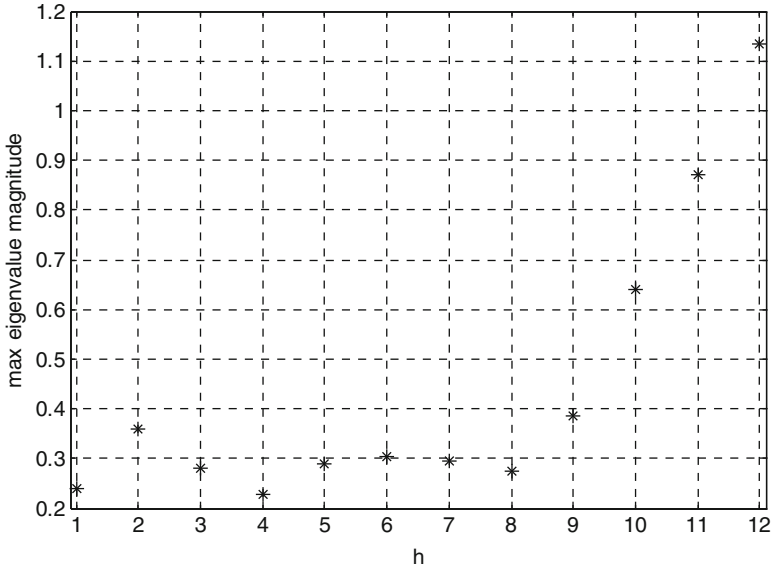

Fig. 2.10 Maximum eigenvalue magnitude of the test matrix M vs. the update time h

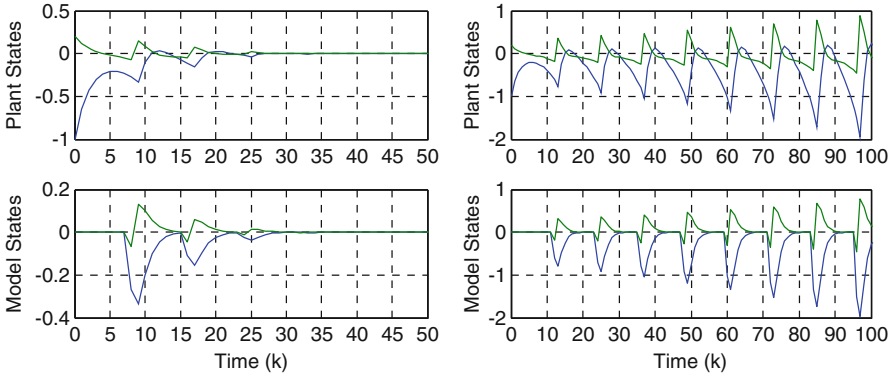

Fig. 2.11 Response of plant and model states. *Left*: $h = 8$. *Right*: $h = 12$

Remark Optimization problem. A similar optimization problem to the continuous-time case can also be considered in the discrete-time case. The problem is to find the optimal control input in the presence of model uncertainties and in the absence of feedback measurements for extended periods of time. Reducing network communication is also important when regulating the states of the system and minimizing the control effort.

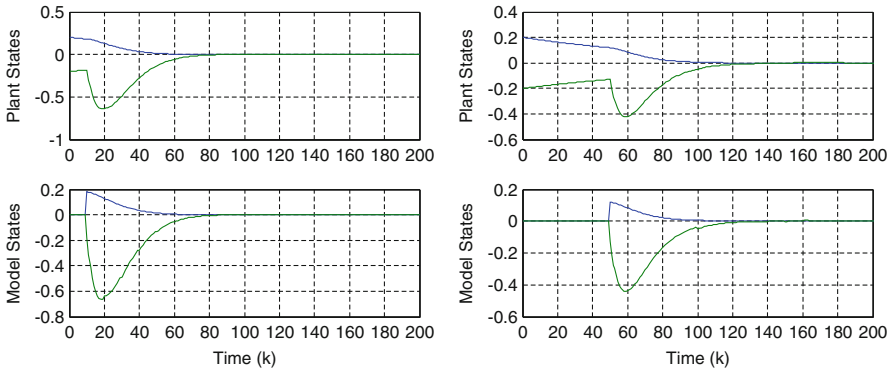

Fig. 2.12 Response of plant and model states. *Left*: $h = 10$. *Right*: $h = 50$

Mathematically, we try to solve the following problem:

$$\min_{u,\beta} x_T^T Q_T x_T + \sum_{n=0}^{T-1} x_n^T Q x_n + u_n^T R u_n + S\beta_n \tag{2.56}$$

where Q and Q_T are real, symmetric, and positive semi-definite matrices; R is a real, symmetric, and positive definite matrix; S is a positive weighting factor that penalizes network communication; x_T is the terminal state; and β_n is constrained to take on only two different values:

$$\beta_n = \begin{cases} 1 & measurement\ \ x_n\ \ is\, sent \\ 0 & measurement\ \ x_n\ \ is\, not\, sent \end{cases} \tag{2.57}$$

The optimal sequence of u_n and β_n, $n = 0,1,2,\ldots,T\text{-}1$, are of interest. This cost criterion can be applied to a discrete-time system or to a discretized version of continuous-time systems. The objective is to find the optimal control input and the optimal scheduling decisions (2.57). Solving this problem is the topic of Chap. 9.

2.4 Alternative Conditions for Stability of MB-NCS

In this section the stability of discrete-time MB-NCS is studied using a lifting approach. The stability of continuous-time systems is also considered using norms and a Lyapunov-based approach.

2.4.1 A Lifting Approach for Stability of Discrete-Time MB-NCS

In this section the traditional configuration for discrete-time MB-NCS will be studied using lifting techniques. In the lifting process, the input and output spaces are extended appropriately in order to obtain a Linear Time-Invariant (LTI) system description for sampled-data, multi-rate, or linear time-varying periodic systems. Since the lifted system is an LTI system, the available tools and results for LTI systems are applicable to the lifted system as well.

Applying the lifting approach to the traditional setup in MB-NCS, the stability problem is reformulated and necessary and sufficient conditions for discrete-time systems are derived which are the same conditions given in Sect. 2.3 derived using another approach. The main advantage of lifting here is that this strategy can be extended to the case when a communication network exists on both sides of the control loop, from sensor to controller and from controller to actuator. This topic will be covered in later chapters.

Lifting discrete-time signals and systems. This section provides a brief discussion on lifting discrete-time signals and systems based on [42]. Suppose there exist two periods h and h_s in a discrete-time setup and they are related by $h_s = h/r$, where r is some positive integer. For a discrete-time signal $v(k)$ referred to the sub-period h/r, that is, $v(0)$ occurs at time $t = 0$, $v(1)$ at $t = h/r$, $v(2)$ at $t = 2\ h/r$, and so on, the lifted signal $\underline{v}$ is defined as follows:

If $v = \{v(0), v(1), v(2), \ldots\}$, then

$$\underline{v} = \left\{ \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(r-1) \end{bmatrix}, \begin{bmatrix} v(r) \\ v(r+1) \\ \vdots \\ v(2r-1) \end{bmatrix}, \ldots \right\}. \tag{2.58}$$

The dimension of the lifted signal $\underline{v}(n)$ is r times the dimension of the original signal $v(n)$ and is regarded to the base period, i.e., $\underline{v}(n)$ occurs at time $t = nh$.

The lifting operator L is defined to be the map $v \rightarrow \underline{v}$. The inverse operator L^{-1} exists and is defined as follows:

If

$$\psi = \left\{ \begin{bmatrix} \psi_1(0) \\ \psi_2(0) \\ \vdots \\ \psi_n(0) \end{bmatrix}, \begin{bmatrix} \psi_1(1) \\ \psi_2(1) \\ \vdots \\ \psi_n(1) \end{bmatrix}, \ldots \right\} \tag{2.59}$$

and $v = L^{-1}\psi$, then $v = \{\psi_1(0), \psi_2(0), \ldots, \psi_n(0), \psi_1(1), \psi_2(1), \ldots, \psi_n(1), \ldots\}$.

A relevant feature of lifting is that it preserves inner products and norms. Let us see this for the case of l_2-norms. The norm of a signal $v \in l_2(\mathbb{Z}_+, \mathbb{R}^m)$ is given by:

$$\|v\|_2^2 = v(0)^T v(0) + v(1)^T v(1) + ... \tag{2.60}$$

and the norm of its lifted version $\underline{v} \in l_2(\mathbb{Z}_+, \mathbb{R}^{rm})$ is:

$$\begin{aligned} \left\|\underline{v}\right\|_2^2 &= \underline{v}(0)^T \underline{v}(0) + \underline{v}(1)^T \underline{v}(1) + ... \\ &= \begin{bmatrix} v(0) \\ : \\ v(r-1) \end{bmatrix}^T \begin{bmatrix} v(0) \\ : \\ v(r-1) \end{bmatrix} + \begin{bmatrix} v(r) \\ : \\ v(2r-1) \end{bmatrix}^T \begin{bmatrix} v(r) \\ : \\ v(2r-1) \end{bmatrix} + ... \\ &= v(0)^T v(0) + v(1)^T v(1) + ... \\ &= \|v\|_2^2. \end{aligned} \tag{2.61}$$

Now, let us consider a discrete-time finite dimensional LTI system G_d with underlying period h/r. Lifting the input and output signals so that the lifted signals correspond to the base period h results in the lifted system: $\underline{G}_d = LG_dL^{-1}$.

Assuming the state-space representation of the original system G_d is known and given by:

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + Du(n). \end{aligned} \tag{2.62}$$

Then the state-space representation for the lifted system $\underline{G}_d$ is given by:

$$\begin{aligned} x\big((n+1)h\big) &= A^r x(nh) + \begin{bmatrix} A^{r-1}B & A^{r-2}B & \dots & B \end{bmatrix} \underline{u}(nh) \\ \underline{y}(nh) &= \begin{bmatrix} C \\ CA \\ : \\ CA^{r-1} \end{bmatrix} x(nh) + \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ : & & & \\ CA^{r-2}B & CA^{r-3}B & \dots & D \end{bmatrix} \underline{u}(nh). \end{aligned} \tag{2.63}$$

Lifting discrete-time MB-NCS. As it was described in Sect. 2.1, MB-NCS makes use of an explicit model of the plant which is added to the controller node to compute the control input based on the state of the model rather than on the plant state.

The dynamics of a discrete-time plant and model are given respectively by:

$$x(n+1) = Ax(n) + Bu(n) \tag{2.64}$$

$$\hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n) \tag{2.65}$$

where x is the state of the plant, $\hat{x}$ is the state of the model, and the matrices $\hat{A}, \hat{B}$ represent the available model of the system matrices A,B.

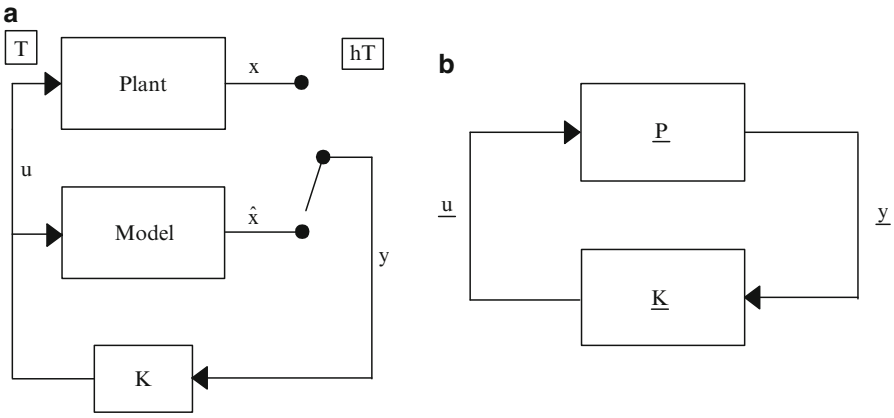

Fig. 2.13 Equivalent systems to MB-NCS. (**a**) Linear time-varying periodic system. (**b**) Lifted system

It will be advantageous to define the input u as:

$$u(n) = \begin{cases} Kx(n) & n = ih \\ K\hat{x}(n) & n = ih + j \end{cases} \tag{2.66}$$

for $i = 0, 1, 2....$ and $0 < j < h$ (j is also an integer), that is, the input is a function of the state of the plant at the time instant when we update the model, and a function of the state of the model otherwise.

A state feedback discrete-time MB-NCS can be seen as a linear time-varying periodic system as shown in (Fig. 2.13a) by considering an output y that is equal to $\hat{x}$ when the loop is open and equal to x when we have an update (closed loop). The system after applying lifting is represented in part b) of the same figure and is regarded as an LTI system with higher dimension input and output.

Note that the original period of the system is denoted by T and the period of the network by hT. Then, using the definition at the beginning of this section, we have that for this case $r = h$ since $hT/h = T$. The input $\underline{u}$ for the lifted system $\underline{P}$ and its output $\underline{y}$ are given by the following equations:

$$\underline{u}(nh) = \begin{bmatrix} u(nh) \\ u(nh+1) \\ \vdots \\ u\big((nh+h-1)\big) \end{bmatrix} = \begin{bmatrix} Kx(nh) \\ K\hat{x}(nh+1) \\ \vdots \\ K\hat{x}\big((nh+h-1)\big) \end{bmatrix} \tag{2.67}$$

$$\underline{y}(nh) = \begin{bmatrix} I \\ \hat{A} \\ \hat{A}^2 \\ \vdots \\ \hat{A}^{h-1} \end{bmatrix} x(nh) + \begin{bmatrix} 0 & 0 & \ldots & 0 \\ \hat{B} & 0 & \ldots & 0 \\ \hat{A}\hat{B} & \hat{B} & \ldots & 0 \\ \vdots & & & \vdots \\ \hat{A}^{h-2}\hat{B} & \hat{A}^{h-3}\hat{B} & \ldots & 0 \end{bmatrix} \underline{u}(nh). \tag{2.68}$$

The dimension of the state is preserved and the state equation expressed in terms of the lifted input is given by:

$$x((n+1)h) = A^h x(nh) + \begin{bmatrix} A^{h-1}B & A^{h-2}B & \ldots & B \end{bmatrix} \underline{u}(nh). \tag{2.69}$$

The new controller is of the form: $\underline{K} = \begin{bmatrix} K & 0 & 0 & \ldots \\ 0 & K & 0 & \\ 0 & 0 & K & \\ \vdots & & & \end{bmatrix}$ where each zero block has the same dimensions as K.

Definition 2.7 [5] Asymptotic stability of discrete-time LTI system. The equilibrium $x = 0$ of a system described by $x(n+1) = Ax(n)$ is asymptotically stable if and only if all eigenvalues of A are within the unit circle of the complex plane (i.e., if $\lambda_1,\ldots,\lambda_n$ denote the eigenvalues of A, then $|\lambda_i| < 1,\ i = 1,\ldots,n$). In this case we say that the matrix A is Schur stable or simply, the matrix A is stable.

Theorem 2.8 *The lifted system is asymptotically stable if only if the eigenvalues of*

$$N = A^h + \sum_{j=0}^{h-1} A^{h-1-j} BK\left(\hat{A} + \hat{B}K\right)^j \tag{2.70}$$

lie strictly inside the unit circle.

Proof To prove this theorem, we note that (2.69) is the same as the state equation that characterizes the autonomous LTI system:

$$x((n+1)h) = \left(A^h + \sum_{j=0}^{h-1} A^{h-1-j} BK\left(\hat{A} + \hat{B}K\right)^j\right) x(nh). \tag{2.71}$$

Equation (2.71) can be obtained by directly substituting (2.67) in (2.69), and then substituting the value of each individual output by its equivalent in terms of the state *x(kh)*, i.e.,

$$\begin{aligned} &\hat{x}(nh+1) = \hat{A}x(nh) + \hat{B}u(nh) = \left(\hat{A} + \hat{B}K\right)x(nh) \\ &\hat{x}(nh+2) = \left(\hat{A} + \hat{B}K\right)^2 x(nh) \\ &\vdots \end{aligned} \tag{2.72}$$

The resulting equation can be simply expressed as (2.71) which characterizes a discrete-time LTI system of the form given in Definition 2.7, then the networked system is asymptotically stable if only if the eigenvalues of (2.70) lie inside the unit circle. ♦

2.4.2 Relation to Previous Results

The results presented in the previous section using a lifting approach are directly related to the stability conditions in Sect. 2.3. Since both results present necessary and sufficient conditions for stability of the discrete-time MB-NCS, we should be able to demonstrate that the conditions are the same. This is shown below.

From Theorem 2.6 a necessary and sufficient condition for stability of an MB-NCS with instantaneous feedback is given if the eigenvalues of

$$M_D(h) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \Lambda_D^h \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \tag{2.73}$$

lie inside the unit circle, where

$$\Lambda_D = \begin{pmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{pmatrix}.$$

In order to establish the relation between the two theorems consider the transformation:

$$\overline{\Lambda} = P\Lambda_D P^{-1} = \begin{pmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{pmatrix} \tag{2.74}$$

where

$$P = \begin{pmatrix} I & 0 \\ I & -I \end{pmatrix}, \quad P^{-1} = \begin{pmatrix} I & 0 \\ I & -I \end{pmatrix} \tag{2.75}$$

and find the Z-transform of $M_D(n)$, $n = 0,1,2,\ldots$

$$Z\{M_D(n)\} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} Z\{\Lambda_D^n\} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}.$$

The Z-transform of Λ_D^n can be obtained according to:

$$Z\{\Lambda_D^n\} = (zI - \Lambda_D)^{-1} z = \left(zI - P^{-1}\overline{\Lambda}P\right)^{-1} z = P^{-1}\left(zI - \overline{\Lambda}\right)^{-1} Pz. \tag{2.76}$$

Substituting the last equation in the Z-transform of $M_D(n)$ we obtain:

$$Z\{M_D(k)\} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} (zI - \overline{\Lambda})^{-1} \begin{pmatrix} I & 0 \\ I & 0 \end{pmatrix} z$$
$$= \begin{pmatrix} (zI - A)^{-1} + (zI - A)^{-1} BK\left(zI - \left(\hat{A} + \hat{B}K\right)\right)^{-1} & 0 \\ 0 & 0 \end{pmatrix} z.$$

We now proceed to obtain the inverse Z-transform of the upper left sub-matrix which contains the eigenvalues of interest:

$$Z^{-1}\left\{(zI - A)^{-1} z + (zI - A)^{-1} BK\left(zI - \left(\hat{A} + \hat{B}K\right)\right)^{-1} z\right\} = A^n + \sum_{j=0}^{n-1} A^{n-1-j} BK\left(\hat{A} + \hat{B}K\right)^j. \tag{2.77}$$

Let $n = h$ in the above equation to obtain:

$$A^h + \sum_{j=0}^{h-1} A^{h-1-j} BK\left(\hat{A} + \hat{B}K\right)^j \tag{2.78}$$

which is exactly the same result obtained in Theorem 2.8, i.e., the eigenvalues of the first term of $M_D(h)$ correspond to the eigenvalues of (2.70). The other eigenvalues of $M_D(h)$ are not necessary in the analysis since they are always equal to 0 (they always lie inside the unit circle).

2.4.3 Additional Approaches for Stability of MB-NCS

We end this section by describing two additional approaches that provide conditions for stability of continuous-time MB-NCS that do not require exact knowledge of the plant parameters. These results can be used directly to compute the admissible range for h based only on certain bounds on the norms of the uncertainties. It is also important to clarify that the next conditions are sufficient only and, in general, very conservative, that is, the admissible update intervals h computed here are typically shorter than those obtained in Sect. 2.2.

Approach based on the norm of the state. In the first approach we set the requirement that the norm of the state of the plant should decrease at every sampling instant, that is, $\|x(t_{k+1})\| < \|x(t_k)\|$, for all $k = 0,1,\ldots$ and $h = t_{k+1} - t_k$. The plant and the model dynamics are given by (2.1) with $u = K\hat{x}$. The state error was defined in (2.2) and the error uncertainty matrices are expressed as follows.

$$\widetilde{A} = A - \hat{A}\,, \quad \widetilde{B} = B - \hat{B}\,. \tag{2.79}$$

By finding a control gain K that asymptotically stabilizes the model, i.e., $\hat{A} + \hat{B}K$ is Hurwitz, we can use the following bound:

$$\left\| e^{\left(\hat{A}+\hat{B}K\right)t} \right\| \leq \alpha e^{-\beta t}, \qquad \alpha, \beta > 0. \tag{2.80}$$

Theorem 2.9 *Assume that the pair* $\left(\hat{A}, \hat{B}\right)$ *is stabilizable. The model-based networked system described by* (2.1) *is asymptotically stable if:*

$$1 - \alpha\left(e^{-\beta h} + \frac{\widetilde{K}}{\beta + \beta_2}\left(e^{\beta_2 h} - e^{-\beta h}\right)\right) > 0 \tag{2.81}$$

where $\|A\| \leq \left\|\hat{A}\right\| + \left\|\widetilde{A}\right\| \leq \beta_2$, $\left\|\widetilde{A} + \widetilde{B}K\right\| \leq \widetilde{K}$, $h = t_{k+1} - t_k$.

Proof From (2.2) we can obtain the next expression for the state error:

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ae + \left(\widetilde{A} + \widetilde{B}K\right)\hat{x}\,. \tag{2.82}$$

The response of the state error at any given time as a function of the last received measurement is:

$$\begin{aligned} e(t) &= e^{A(t-t_k)}e\left(t_k\right) - \int_{t_k}^{t} e^{A(t-\tau)}\left(\widetilde{A} + \widetilde{B}K\right)\hat{x}\,(\tau - t_k)d\tau \\ &= -\int_{t_k}^{t} e^{A(t-\tau)}\left(\widetilde{A} + \widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)(\tau - t_k)}d\tau \cdot x(t_k). \end{aligned} \tag{2.83}$$

then, a bound for the norm of the state error is:

$$\|e(t)\| \leq \frac{\alpha\widetilde{K}\|x(t_k)\|}{\beta + \beta_2}\left(e^{\beta_2(t-t_k)} - e^{-\beta(t-t_k)}\right). \tag{2.84}$$

It is also clear from (2.2) that the norm of the state satisfies:

$$\|x(t)\| < \|\hat{x}\,(t)\| + \|e(t)\|. \tag{2.85}$$

Substituting (2.84) in (2.85) we have, using again the bound (2.80):

$$\|x(t)\| \leq \alpha e^{-\beta(t-t_k)}\|x(t_k)\| + \frac{\alpha\widetilde{K}\|x(t_k)\|}{\beta + \beta_2}\left(e^{\beta_2(t-t_k)} - e^{-\beta(t-t_k)}\right). \tag{2.86}$$

In order to satisfy the initial condition on the norm of the state we require that:

$$\|x(t_k)\| - \|x(t_k)\|\left(\alpha e^{-\beta h} + \frac{\alpha\widetilde{K}}{\beta+\beta_2}\left(e^{\beta_2 h} - e^{-\beta h}\right)\right) > 0 \tag{2.87}$$

which is equivalent to (2.81). ♦

Approach based on Lyapunov functions. In the second approach we set the requirement that the discrete-time Lyapunov function along the trajectories of the system decreases at every sampling instant, i.e., $V(x(t_{k+1})) - V(x(t_k)) < 0$, for $k = 0,1,\ldots$

Theorem 2.10 *Assume that the pair* $\left(\hat{A},\hat{B}\right)$ *is stabilizable. The model-based networked system described by* (2.1) *is asymptotically stable if*:

$$2\overline{\sigma}(F)G_h + G_h^2 < \frac{q}{\overline{\sigma}(P)} \tag{2.88}$$

where $F = e^{\left(\hat{A}+\hat{B}K\right)h}$, $G_h = \frac{\widetilde{\alpha K}}{\beta+\beta_2}\left(e^{\beta_2 h} - e^{-\beta h}\right) \geq \overline{\sigma}(\Delta(h))$, $\Delta(h) = \int_0^h e^{A(h-\tau)}$ $\left(\widetilde{A} + \widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau$, $\|A\| \leq \left\|\hat{A}\right\| + \left\|\widetilde{A}\right\| \leq \beta_2, \alpha$ and β are given by (2.80).

Proof Using (2.2) and (2.83), the state of the plant can be expressed by:

$$\begin{aligned} x(t_{k+1}) &= \left(e^{\left(\hat{A}+\hat{B}K\right)h} + \int_0^h e^{A(h-\tau)}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau\right)\cdot x(t_k) \\ &= (F + \Delta(h))x(t_k). \end{aligned} \tag{2.89}$$

Next we define a quadratic Lyapunov function $V = x^T P x$, that is evaluated at the update instants as follows:

$$V(x(t_{k+1})) - V(x(t_k)) = x(t_k)^T\Big[(F+\Delta(h))^T P(F+\Delta(h)) - P\Big]x(t_k). \tag{2.90}$$

Note that a stabilizing controller for the model can be found. F is a stable discrete-time matrix that satisfies:

$$F^T PF - P = -Q \tag{2.91}$$

for symmetric, positive definite matrices P and Q. Then a sufficient condition for stability of the system is given in terms of the perturbation $\Delta(h)$, as follows

$$\begin{aligned} &(F+\Delta(h))^T P\big(F+\Delta(h)\big) - P \\ &\quad = -Q + \Delta(h)^T PF + F^T P\Delta\big(h\big) + \Delta\big(h\big)^T P\Delta\big(h\big) \\ &\quad \leq -\underline{\sigma}(Q) + \overline{\sigma}\big[\Delta\big(h\big)^T PF + F^T P\Delta\big(h\big) + \Delta\big(h\big)^T P\Delta\big(h\big)\big] \\ &\quad \leq -q + 2\overline{\sigma}(PF)G_h + \overline{\sigma}\big(P\big)G_h^2 < 0 \end{aligned} \tag{2.92}$$

where $q = \underline{\sigma}(Q)$ represents the smallest singular value of Q and $\overline{\sigma}(P)$ represents the largest singular value of P. If (2.88) holds, then (2.92) is satisfied. ♦

Recall that the application of the results in Sect. 2.2 to real world problems involves a repeated computation of eigenvalues for different values of parameters uncertainties as explained in Example 2.2. The results in this subsection can be used in a direct way based on a priori information about uncertainty bounds. The disadvantage is that these conditions are sufficient only and may be conservative in general, resulting in smaller update intervals h.

2.5 Notes and References

In this chapter we introduced the basic MB-NCS architecture that will be used throughout this book. This framework uses the available knowledge of the real plant dynamics encapsulated in the plant model to perform an open-loop estimation of the real plant state that is used to compute the control input for the intervals of time that the controller does not receive any measured feedback information. The main results of this chapter were presented in Theorem 2.3 and Theorem 2.6. Theorem 2.3 provides necessary and sufficient conditions for exponential stability of the networked system as a function of the update intervals which are assumed to be periodic. Theorem 2.6 provides the corresponding conditions for discrete-time systems. In both cases, it is assumed that the sensor is able to measure the entire state of the system. Alternative stability conditions were given in Sect. 2.4. In subsequent chapters, different extensions, alternative sampling methodologies, and architectures will be explored. Additionally, different common problems in control theory will be discussed from the model-based networked perspective.

The MB-NCS architecture shown in Sect. 2.1 was first proposed by Montestruque and Antsaklis [186, 187] and the results shown in Sects. 2.2 and 2.3 are primarily based on work published in those two references and also in [188]. The lifting approach for MB-NCS discussed in Sect. 2.4 was first published by Garcia and Antsaklis, [82]. The lifting methods used in that section are based on common lifting techniques presented in [42]. See also [16, 26, 130] for additional discussion of lifting techniques.

The alternative stability conditions for continuous-time systems of Sect. 2.4 represent original work presented in this book and is an adaptation of the work by Montestruque [185] on nonlinear systems.

The MB-NCS framework has been followed by different authors to consider specific problems in NCS. Orihuela et al. [207] have derived conditions for stability of MB-NCS with parametric uncertainties based on the theory of interval matrices. The authors of [197] compare different control decisions for a nonlinear NCS subject to sensor losses. The options include zero control, last available control, and open-loop control; the last one corresponds to using the nominal nonlinear model of the plant. It is proved in that paper that under certain conditions the use of

the model exceeds in performance the other two choices to decide the control input in the absence of feedback information. The MB-NCS structure is used by Yu et al. [283] to study singularly perturbed systems where the sensor is connected to the controller/actuator node using a communication network. Singularly perturbed systems refer to the two-time scale systems that appear due to the presence of small parasitic parameters multiplying the time derivatives of some states of the system. It is difficult, in general, to control this type of systems since the controller has to react simultaneously to both the slow and fast modes of the system. Mu et al. [196] provided a different analysis of MB-NCS with output feedback and constant network delays where the model is updated using directly the received output $y(t_k) = Cx(t_k)$.

Recent work has been produced independently and it shares many characteristics of the MB-NCS framework, indicating the fundamental advantages of using the model dynamics when operating and controlling dynamical systems in the absence of continuous feedback information. Motivated by human operations, the authors of [80] and [126] point out that in general a human operator scans information intermittently and operates continuously the controlled system; the intermittent characteristic in this case refers to the same situation presented in this chapter, that is, a single measurement is used to update the internal model and generate the control input. For a skillful operator, the information is scanned less frequently. Between update intervals the control input is generated the same way as it was shown in the MB-NCS framework, that is, an imperfect model of the system is used to generate an estimate of the state and periodic measurements are used to update the state of this model. In the output feedback case a stochastic estimator is implemented with the assumption that the statistical properties of the measurement noise are known. In both cases the authors provide conditions for stability based on the length of the sampling interval.

In the networked framework shown in [140–142, 165], the model is assumed to match the dynamics of the system exactly; however, the system is subject to unknown input disturbances. The main idea of the approach in those papers is the same as the one described in this chapter, that is, to use the nominal model to generate estimates of the current state of the system. Since the system is subject to unknown disturbances and the model is executed with zero input disturbance, then a difference between the states is expected and the sensor updates transmitted over a digital communication network are used to reset this difference between the states of the plant and of the model at the update time instants.

There exist similarities and significant differences between the MB-NCS approach and the developed and mature Model Predictive Control (MPC) approach. MPC also uses model of the system for control; specifically, the model is used to predict the future output behavior; a tracking error is defined using this prediction and the desired reference and the control action are computed online. The purpose of the control action is to drive the state of the system to a reference position in an optimal fashion while satisfying the existing constraints. MPC relies in frequent feedback measurements in order to update the predicted control sequences. Meanwhile, in MB-NCS the main constraint is imposed in the form of reducing

measurement update rates. Recent work related to NCS [24, 255, 288] has been reported where model predictive controllers have been successfully applied to deal with the usual input and output constraints, but also consider the bandwidth limitations of the network. The additional constraint aims to reduce the traffic in the communication network by using the explicit model to generate the appropriate input in the absence of continuous feedback.

Chapter 3
Model-Based Control Systems: Output Feedback and Delays

In the previous chapter, Chap. 2, we considered the case when the full state vector is available for measurement. We now extend our approach to include continuous and discrete time systems, where the state is not directly measurable. Here, state observers are used to obtain an estimate of the state vector. It is assumed that the state observer is collocated with the sensor.

The use of state observers is not the only approach that is used in this book when dealing with the output feedback case. The output feedback problem is studied again in Chap. 11 where the output measurements are used directly to update the model without need of state observers.

Another problem discussed in this chapter is the control under network induced delays. In this problem, the MB-NCS framework provides the double advantage of using the model parameters for estimating the state of the system during the open-loop intervals, as it was shown previously, and also for estimating the current state of system at the update time instants based on delayed measurements.

The contents of the present Chapter are as follows: Output feedback using state observers is studied in Sect. 3.1 for continuous time systems. Section 3.2 presents similar results for output feedback discrete time systems. Network delays are addressed in Sects. 3.3 and 3.4. Notes and references are presented in Sect. 3.5.

3.1 Output Feedback Using State Observers for Continuous Time Systems

We use the plant model parameters $\left(\hat{A}, \hat{B}, \hat{C}, \hat{D}\right)$ in (3.1) to design the state observer. See Fig. 3.1. The observer has as inputs the output and input of the plant. In the implementation, in order to acquire the input of the plant, which is at the other side of the communication link, the observer should also include a version of the model and controller, and also knowledge of the update time h. In this way, the output of the controller, that is the input to the plant, can be simultaneously and continuously

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_3,

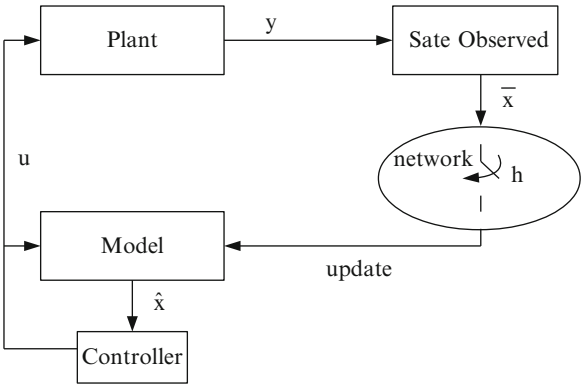

Fig. 3.1 Proposed configuration of an output feedback networked control system

generated at both ends of the feedback path with the only requirement being that the observer makes sure that the model has been updated. This last requirement ensures that both the controller and the observer are synchronized. Handshaking protocols provided by most networks can be used.

The observer has the form of a standard asymptotic state estimator, a Luenberger observer, with gain L. It makes use of the plant model parameters.

In summary, the system dynamic equations are:

$$\begin{aligned}
&\text{Plant :} && \dot{x} = Ax + Bu,\ y = Cx + Du \\
&\text{Model :} && \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u,\ \hat{y} = \hat{C}\hat{x} + \hat{D}u \\
&\text{Controller :} && u = K\hat{x} \\
&\text{Observer :} && \dot{\bar{x}} = \left(\hat{A} - L\hat{C}\right)\bar{x} + \begin{bmatrix}\hat{B} - L\hat{D} & L\end{bmatrix}\begin{bmatrix}u \\ y\end{bmatrix}, \\
&&& \text{for } t \in [t_k, t_{k+1})
\end{aligned} \tag{3.1}$$

We now proceed in a similar way as in the previous case of full feedback. Namely, there will be an update interval h, after which the observer updates the controller's model state $\hat{x}$ with its estimate $\bar{x}$. We will also define an error e that will be the difference between the controller's model state and the observer's estimate: $\bar{e} = \bar{x} - \hat{x}$.

It is clear that at times t_k, where $t_k - t_{k-1} = h$, the error $\bar{e}$ will be equal to 0, since the update $\hat{x}(t_k) = \bar{x}(t_k)$ takes place.

$$\bar{e}(t) = \left\{ \begin{array}{ll} \bar{x}(t) - \hat{x}(t) & t \in (t_k, t_{k+1}) \\ 0 & t = t_k \end{array} \right\} \tag{3.2}$$

Also we will define the modeling error matrices in the same way as before (Chap. 2): $\widetilde{A} = A - \hat{A}$, $\widetilde{B} = B - \hat{B}$, $\widetilde{C} = C - \hat{C}$, $\widetilde{D} = D - \hat{D}$.

Now for $t \in [t_k, t_{k+1})$, $u = K\hat{x}$, so we have:

$$\begin{aligned}
\dot{x} &= Ax + BK\hat{x} \\
\dot{\hat{x}} &= \left(\hat{A} + \hat{B}K\right)\hat{x} \\
\dot{\bar{x}} &= \left(\hat{A} - L\hat{C}\right)\bar{x} + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} K\hat{x} \\ Cx + DK\hat{x} \end{bmatrix} \\
&= \begin{bmatrix} LC & \hat{B}K + L\tilde{D}K & \hat{A} - L\hat{C} \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \\ \bar{x} \end{bmatrix}
\end{aligned} \tag{3.3}$$

with initial conditions $\hat{x}(t_k) = \bar{x}(t_k)$. Using the same approach as before, we express the system dynamics in terms of the states that will not change at the update times. Then the dynamics of the overall system for $t \in [t_k, t_{k+1})$ can be described by

$$\begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \\ \dot{\bar{e}} \end{bmatrix} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\tilde{D}K & -\hat{B}K - L\tilde{D}K \\ LC & L\tilde{D}K - L\hat{C} & \hat{A} - L\tilde{D}K \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \\ \bar{e} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x(t_k) \\ \bar{x}(t_k) \\ \bar{e}(t_k) \end{bmatrix} = \begin{bmatrix} x(t_k^-) \\ \bar{x}(t_k^-) \\ 0 \end{bmatrix}.$$

$$t \in [t_k, t_{k+1}), \qquad \text{with} \quad t_{k+1} - t_k = h \tag{3.4}$$

Define $z = \begin{bmatrix} x \\ \bar{x} \\ \bar{e} \end{bmatrix}$, and $\Lambda_o = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\tilde{D}K & -\hat{B}K - L\tilde{D}K \\ LC & L\tilde{D}K - L\hat{C} & \hat{A} - L\tilde{D}K \end{bmatrix}$

so that (3.4) can be represented by:

$$\dot{z} = \Lambda_o z \text{ for } t \in [t_k, t_{k+1}) \text{ and } z(t_k) = \begin{bmatrix} x(t_k^-) \\ \bar{x}(t_k^-) \\ 0 \end{bmatrix} \tag{3.5}$$

Proposition 3.1 *The system with dynamics described by* (3.5) *with initial conditions* $z(t_0) = \begin{bmatrix} x(t_0) \\ \bar{x}(t_0) \\ 0 \end{bmatrix} = z_0, \quad t_0 = 0,$ *has the following response*:

$$\begin{aligned}
z(t) &= e^{\Lambda_o(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0, \\
&\text{for } t \in [t_k, \quad t_{k+1}), \quad \text{with} \quad t_{k+1} - t_k = h
\end{aligned} \tag{3.6}$$

Proof On the interval $t \in [t_k, t_{k+1})$, the system response is

$$z(t) = \begin{bmatrix} x(t) \\ \overline{x}(t) \\ \overline{e}(t) \end{bmatrix} = e^{\Lambda_o(t-t_k)} \begin{bmatrix} x(t_k) \\ \overline{x}(t_k) \\ 0 \end{bmatrix} = e^{\Lambda_o(t-t_k)} z(t_k). \tag{3.7}$$

Now, note that at times t_k, $z(t_k) = \begin{bmatrix} x(t_k) \\ \overline{x}(t_k) \\ 0 \end{bmatrix}$, that is, the error $\overline{e}(t)$ is reset to 0. We can represent this by

$$z(t_k) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} z\left(t_k^-\right). \tag{3.8}$$

Using (3.7) to calculate $z(t_k^-)$ we obtain

$$z(t_k) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} z(t_{k-1}). \tag{3.9}$$

In view of (3.7) we have that if at time $t = t_0$, $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ \overline{x}_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$\begin{aligned} z(t) &= e^{\Lambda_o(t-t_k)} z\left(t_k\right) \\ &= e^{\Lambda_o(t-t_k)} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} z(t_{k-1}) \\ &= e^{\Lambda_o(t-t_k)} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} z\left(t_{k-2}\right) \\ &\cdots \\ &= e^{\Lambda_o(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \right)^k z_0. \end{aligned} \tag{3.10}$$

Now we know that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h}$ is of the form $\begin{bmatrix} M_1 & M_2 & N_1 \\ M_3 & M_4 & N_2 \\ 0 & 0 & 0 \end{bmatrix}$ and so

$\left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h}\right)^k$ has the form $\begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k & \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix}$. Additionally we note the special form of the initial condition $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ \overline{x}_0 \\ 0 \end{bmatrix}$ so that

$$\begin{aligned} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h}\right)^k \begin{bmatrix} x_0 \\ \overline{x}_0 \\ 0 \end{bmatrix} &= \begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k \begin{bmatrix} x_0 \\ \overline{x}_0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} \\ &= \begin{bmatrix} \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ \overline{x}_0 \\ 0 \end{bmatrix} \\ &= \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k \begin{bmatrix} x_0 \\ \overline{x}_0 \\ 0 \end{bmatrix}. \end{aligned} \tag{3.11}$$

In view (3.11) it is clear that we can represent the system response as in (3.6), that is:

$$z(t) = e^{\Lambda_o (t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)^k z_0$$
$$t \in [t_k, \quad t_{k+1}), \quad \textit{with} \quad t_{k+1} - t_k = h.$$

♦

We will present now the necessary and sufficient conditions for this system to be exponentially stable at large (or globally). For the definition of stability please refer to Definition 2.2 in Sect. 2.2.

Theorem 3.2 *The system described by* (3.5) *is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ \overline{x} \\ \overline{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $M_o = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ *are strictly inside the unit circle.*

Proof Sufficiency. Taking the norm of the solution described as in Proposition 3.1:

$$\|z(t)\| = \left\| e^{\Lambda_o(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0 \right\|$$

$$\leq \left\| e^{\Lambda_o(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\|. \tag{3.12}$$

Now let us analyze the first term on the right hand side of (3.12):

$$\left\| e^{\Lambda_o(t-t_k)} \right\| \leq 1 + (t-t_k)\overline{\sigma}(\Lambda_o) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda_o)^2 \ldots = e^{\overline{\sigma}(\Lambda_o)(t-t_k)} \leq e^{\overline{\sigma}(\Lambda_o)h} = K_1 \tag{3.13}$$

where $\overline{\sigma}(\Lambda_o)$ is the largest singular value of Λ_o. In general this term can always be bounded since the time difference $t - t_k$ is always smaller than h. In other words, even when Λ_o has eigenvalues with positive real part, $\left\| e^{\Lambda_o(t-t_k)} \right\|$ can only grow a certain finite amount. This growth is completely independent of k.

We now study the term $\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 e^{-\alpha_1 k} \tag{3.14}$$

with K_2, $\alpha_1 > 0$.

Since k is a function of time we can bound the right term of (3.14) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h}t} = K_3 e^{-\alpha t} \tag{3.15}$$

with K_3, $\alpha > 0$.

So from (3.12) using (3.13) and (3.15) we can conclude:

$$\|z(t)\| = \left\| e^{\Lambda_o(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \leq K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\|.$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the system is stable and that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be bounded and converging to 0 with time. We will take the sample at times t_{k+1}^-, just before the update. Even further we will concentrate on the combined state of the plant $x(t_{k+1}^-)$ and observer $\overline{x}\left(t_{k+1}^-\right)$, which are the first two elements of $z(t_{k+1}^-)$. We will call $\begin{bmatrix} x\left(t_{k+1}^-\right) \\ \overline{x}\left(t_{k+1}^-\right) \end{bmatrix}$, $\xi(k)$. Now assume $e^{\Lambda_o \tau}$ has the following form:

$$e^{\Lambda_o \tau} = \begin{bmatrix} W_1(\tau) & W_2(\tau) & X_1(\tau) \\ W_3(\tau) & W_4(\tau) & X_2(\tau) \\ Y_1(\tau) & Y_2(\tau) & Z(\tau) \end{bmatrix} \tag{3.16}$$

For simplicity, let us define:

$$W(\tau) = \begin{bmatrix} W_1(\tau) & W_2(\tau) \\ W_3(\tau) & W_4(\tau) \end{bmatrix}, \quad X(\tau) = \begin{bmatrix} X_1(\tau) \\ X_2(\tau) \end{bmatrix}, \quad Y(\tau) = \begin{bmatrix} Y_1(\tau) & Y_2(\tau) \end{bmatrix} \tag{3.17}$$

Then we can express the solution $z(t)$ as:

$$\begin{aligned} & e^{\Lambda_o (t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0 \\ &= \begin{bmatrix} W(t-t_k) & X(t-t_k) \\ Y(t-t_k) & Z(t-t_k) \end{bmatrix} \begin{bmatrix} (W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(t-t_k)(W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(t-t_k)(W(h))^k & 0 \end{bmatrix} z_0 \end{aligned} \tag{3.18}$$

Now let us check the values of the solution at times t_{k+1}^-, or just before the update:

$$z\left(t_{k+1}^-\right) = \begin{bmatrix} W(h)(W(h))^k & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0. \tag{3.19}$$

We also know that $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This means that the first two elements of $z(t_{k+1}^-)$, which we call $\xi(k)$, will in general grow with k. In other words, we cannot ensure $\xi(k)$ will converge to 0 for general initial conditions x_0, $\overline{x}_0$, that is, in this case

$$\left\| \begin{bmatrix} x(t_{k+1}^-) \\ \overline{x}(t_{k+1}^-) \end{bmatrix} \right\| = \|\xi(k)\| = \left\| (W(h))^{k+1} \begin{bmatrix} x_0 \\ \overline{x}_0 \end{bmatrix} \right\| \to \infty \qquad \text{as} \qquad k \to \infty. \quad (3.20)$$

This clearly means the system cannot be stable, and thus we have a contradiction. ♦

Remark The state observer can also be implemented at the controller node. In this case the sensor sends measurements of the output $y(t_k)$ every h time units. At the controller node, this measurement is used by the state observer to obtain an estimate of the state of the plant and use that estimate to update the state of the model. Since the observer does not receive continuous measurements of the plant output, it does not provide accurate estimations of the state $x(t)$ when h is large. In this case, conditions for stability require a very short update interval which means that communication rate is not significantly reduced compared to the case when the observer is collocated with the sensor.

3.1.1 Separation Principle

The conditions for stability depend on several factors including the choice of update interval, the model uncertainties, the control gain, and (for the output feedback case) the observer gain as well. The results in Proposition 3.1 and Theorem 3.2 are more complex than the usual observer based output feedback closed loop design that follows the Separation Principle. The complexity is due, of course, to the communication limitations, but is also due to the consideration of model uncertainties. The Separation Principle [76] widely used in the design of closed-loop control systems does not apply to the case when there exists model mismatch.

Consider again the plant and observer dynamics, where the observer uses the nominal parameters, which are different from the real plant parameters:

Plant :	$\dot{x} = Ax + Bu \quad y = Cx + Du$
Observer :	$\dot{\overline{x}} = \left(\hat{A} - L\hat{C}\right)\overline{x} + \left(\hat{B} - L\hat{D}\right)u + Ly$
Error matrices :	$\widetilde{A} = A - \hat{A} \quad \widetilde{B} = B - \hat{B} \quad \widetilde{C} = C - \hat{C} \quad \widetilde{D} = D - \hat{D}$
Estimation error	$\varepsilon = x - \overline{x}$

Assume that the plant and model parameters are the same, that is, the error matrices are all equal to 0. Then the separation principle follows by finding the dynamic equation of the estimation error

$$\dot{\varepsilon} = \dot{x} - \dot{\bar{x}} = Ax + Bu - \left(\hat{A} - L\hat{C}\right)\bar{x} - \left(\hat{B} - L\hat{D}\right)u - Ly = (A - LC)\varepsilon. \quad (3.21)$$

The control input is based on the observer state $u = K\bar{x}$. Then the augmented vector containing the state of the plant and the estimation error is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix}. \quad (3.22)$$

Since this matrix is triangular, its eigenvalues are the eigenvalues of $A + BK$ and A-LC. Then the stability of the observer and the closed loop plant are independent. Clearly in this case each gain (observer and controller gains) can be designed separately, thus simplifying the design.

In a more general situation, as is the case in MB-NCS, if we analyze the estimation error dynamics by explicitly taking into account the difference between the plant and model parameters, then it is simple to show that the error dynamics are described by the next equation:

$$\dot{\varepsilon} = \dot{x} - \dot{\bar{x}} = \left(\hat{A} - L\hat{C}\right)\varepsilon + \left(\widetilde{B} - L\widetilde{D}\right)u + \left(\widetilde{A} - L\widetilde{C}\right)x. \quad (3.23)$$

This equation shows that in order to obtain asymptotic estimates of the system states using imperfect parameters of that system, we need a zero reference input (u does not have external component) and a state x that tends asymptotically to zero as time tends to infinity.

The augmented state that contains the state of the closed-loop plant and the estimation error is now given by:

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} - L\widetilde{C} + \widetilde{B}K - L\widetilde{D}K & \hat{A} - L\hat{C} - \widetilde{B}K + L\widetilde{D}K \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix}. \quad (3.24)$$

We can see that the stability of the closed loop plant and the observer are not obtained just by separate designs of the gains K and L, but the eigenvalues of the whole matrix in (3.24) have to be considered. Note that (3.22) is a special case of (3.24), when the model parameters exactly match those of the plant and the uncertainty matrices are all equal to 0.

Example 3.1 Consider the instrument servo example described in Example 2.3 in Sect. 2.2. We now consider the case where only the angular position error can be measured, that is, the output matrices are given by $C = [1 \quad 0]$ and $D = 0$. The state equations have the same form as (2.32) in Chap. 2. Here, we use the nominal

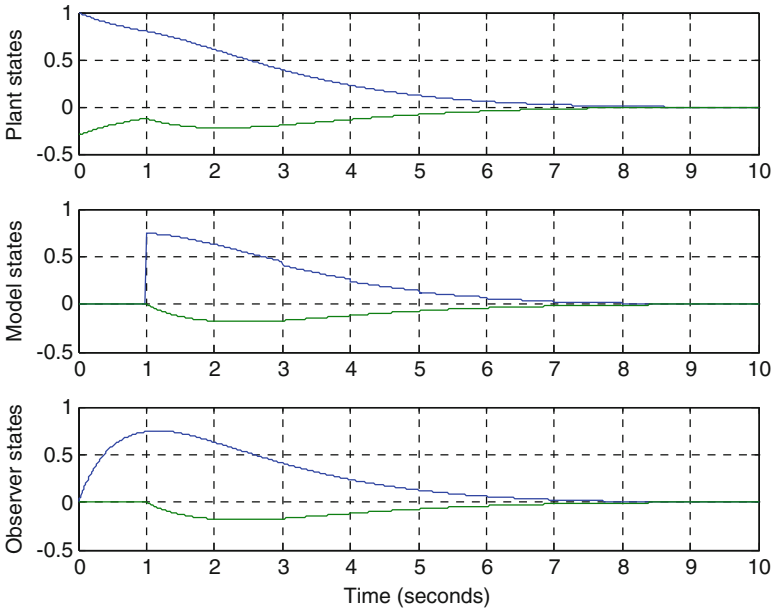

Fig. 3.2 States of the networked instrument servo system for $h = 1$ s

parameters $\hat{\alpha} = 1, \hat{\beta} = 3$. The real parameters are $\alpha = 0.89$, $\beta = 2.79$. We will use the state feedback controller $K = [-0.1667 \ \ -0.1667]$ and the state estimator gain $L = [2 \ 0]^T$.

According to the eigenvalue search using the results of Theorem 3.2, the networked system is stable for any value $h > 0$; however, the transient response will be significantly different for different choices of the update period. Figure 3.2 shows the response for $h = 1$ s. This figure shows the states of the plant, the model, and the observer. Figure 3.3 shows the same states for the case $h = 10$ s. In both cases only the first state of the plant can be measured and used by the state observer but in the figures we plot both of them in order to compare the response for different update intervals.

Example 3.2 In this example we use the inverted pendulum on a moving cart dynamics (linearized dynamics) described in Example 2E in [77].

The linearized dynamics can be expressed using the state vector $x = \left[y \ \theta \ \dot{y} \ \dot{\theta}\right]^T$, where y represents the displacement of the cart with respect to some reference point and θ represents the angle that the pendulum rod makes with respect to the vertical. The matrices corresponding to the state space representation (3.1) are given by

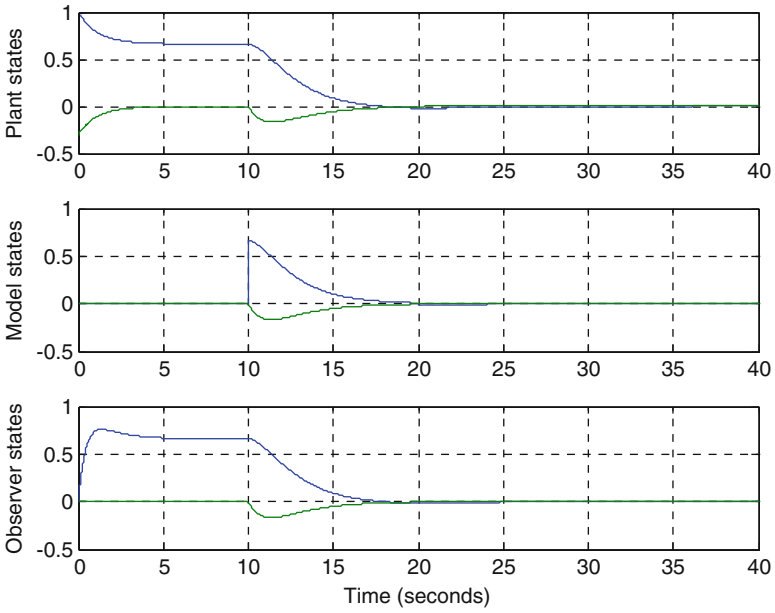

Fig. 3.3 States of the networked instrument servo system for $h = 10$ s

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -mg/M & 0 & 0 \\ 0 & (M+m)/Ml & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1/M \\ -1/Ml \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.25}$$

where the nominal parameters of the model are given by: $\hat{m} = 0.1$, $\hat{M} = 1$, $\hat{l} = 1$. The real parameters represent values close to the nominal parameters, but not exactly the same due to uncertainties in the measurements and specification of these parameters. The physical parameter values are given by: $m = 0.1023$, $M = 0.9965$, $l = 1.0061$. We have that $\hat{g} = g = 9.8$. The input u represents the external force applied to the cart. The open-loop plant and model dynamics are unstable. In this example we use the following gains

$$K = [0.3673 \quad 23.0473 \quad 1.1020 \quad 6.8020], \quad L = \begin{bmatrix} 12.5 & 0 \\ 0 & 10.5 \\ 39 & -0.98 \\ 0 & 32.28 \end{bmatrix}.$$

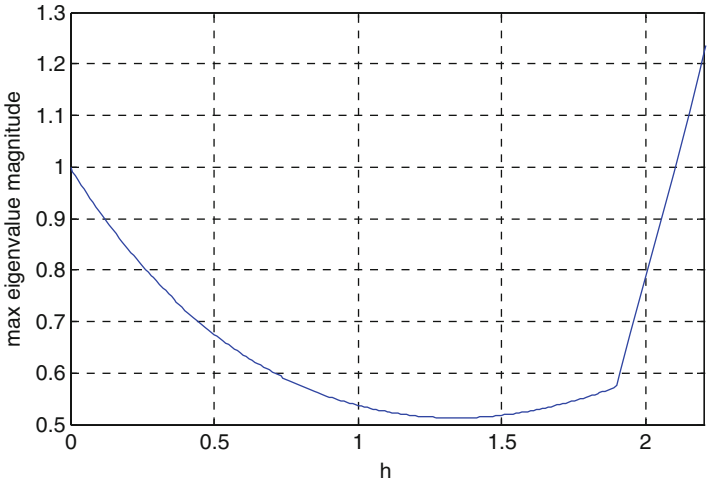

Fig. 3.4 Eigenvalue with maximum magnitude for inverted pendulum on moving cart example

Figure 3.4 shows the eigenvalue with maximum magnitude of $M_o = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ $e^{\Lambda_o h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ for different values of h. Figures 3.5 and 3.6 show the response of the system for $h = 0.5$ s. Figure 3.5 shows the states of the plant, the model, and the observer corresponding to the positions y and θ. Figure 3.6 shows the velocities $\dot{y}$, $\dot{\theta}$ corresponding to the same subsystems.

Example 3.3 Consider the distillation column example shown in Example 2G in [77]. The process describes an extractive column used for separation of isopropanol from a mixture of water. A controlled amount of heating steam, denoted as Δu_1, is introduced near the bottom of the column and the amount of vapor side stream, ΔS, can also be controlled. The objective is to stabilize two key vertical positions subject to changes in the feed composition, Δx_{FA1}, and the flow rate, ΔF_A, disturbances. The positions z_1, z_2 represent interphase changes between the substances in the column. A sharp temperature gradient is associated with each one of the positions. These positions can be determined by measuring corresponding temperature changes $\Delta T_1 = c_{13}\Delta z_1$ and $\Delta T_2 = c_{24}\Delta z_2$ where Δz_1, Δz_2 represent changes in the corresponding positions.

The state vector is given by $x = [\Delta Q_l \; \Delta V_l \; \Delta z_1 \; \Delta z_2]^T$, where ΔQ_l represents heat flow to reboiler and ΔV_l represents the vapor flow rate. $u = [\Delta u_1 \;\; \Delta S]^T$, $x_0 = [\Delta x_{FA1} \; \Delta F_A]^T$, and $y = [\Delta T_1 \; \Delta T_2]^T$ represent the control input, the input disturbance, and the measureable output, respectively. The system dynamics are given by:

$$\dot{x} = Ax + Bu + Ex_0, \quad y = Cx$$

with

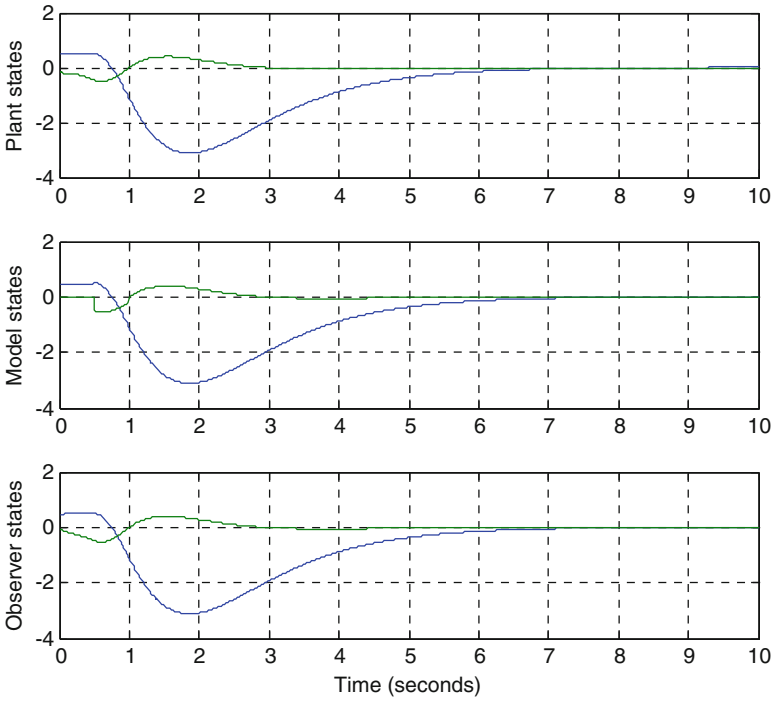

Fig. 3.5 Inverted pendulum on a moving cart. Positions. $h = 0.5$ s

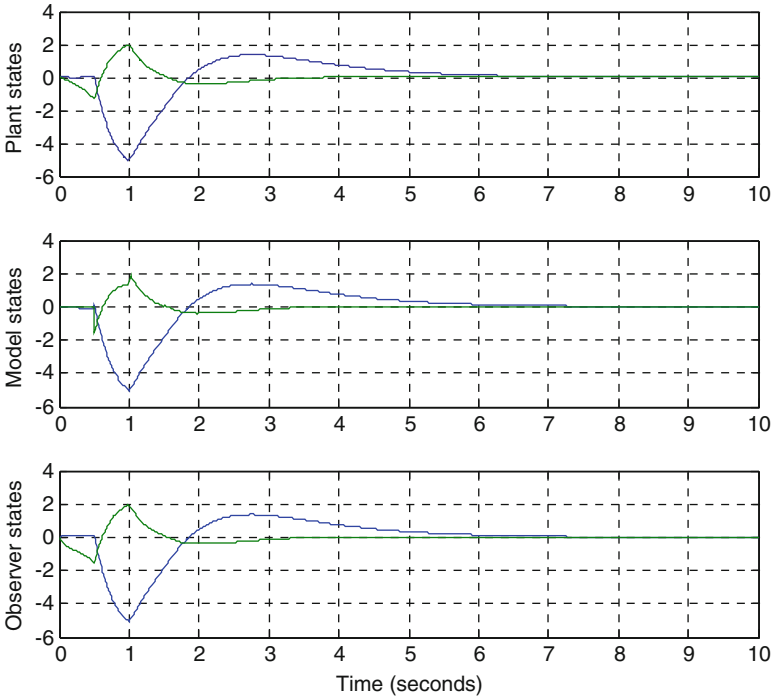

Fig. 3.6 Inverted pendulum on a moving cart. Velocities. $h = 0.5$ s

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & a_{32} & 0 & 0 \\ 0 & a_{42} & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} b_{11} & 0 \\ 0 & 0 \\ 0 & b_{32} \\ 0 & b_{42} \end{bmatrix}, \; E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ f_{31} & f_{32} \\ 0 & f_{42} \end{bmatrix}, \; C = \begin{bmatrix} 0 & 0 & c_{13} & 0 \\ 0 & 0 & 0 & c_{24} \end{bmatrix} \tag{3.26}$$

where time is measured in hours and temperature in degrees Celsius. The nominal model parameters are as follows:

$$\begin{array}{lll} a_{11} = -30.3 & b_{11} = 6.15 \times 10^5 & f_{31} = 62.2 \\ a_{21} = 1.2 \times 10^{-4} & b_{32} = 3.04 & f_{32} = 5.76 \\ a_{22} = -6.02 & b_{42} = 0.052 & f_{42} = 5.12 \\ a_{32} = -3.77 & c_{13} = -7.3 & \\ a_{42} = -2.8 & c_{24} = -25 & \end{array}$$

Note that when implementing the model in the controller node we have $\hat{x}_0 = 0$ since the disturbance cannot be measured. The real parameters represent values within ± 2 % of the nominal parameters. The system is open loop stable, but it does not reject the disturbances. The following controller and observer gains are chosen:

$$K = \begin{bmatrix} 2.5704 \times 10^5 & 1.4590 \times 10^8 & -3.2105 \times 10^8 & 0 \\ 0 & 53.8462 & 0 & -9.6154 \end{bmatrix}$$

$$L = \begin{bmatrix} 2.6526 \times 10^5 & 1.2404 \times 10^5 \\ -28.0331 & 27.1634 \\ -2.0983 & -3.1061 \\ 1.9248 & -3.5914 \end{bmatrix}.$$

Figure 3.7 shows the output of the system subject to a pulse disturbance that occurs at time 0.5 h and lasts for 0.05 h. The figure shows three different instances of the problem. First, it shows the response of the open-loop system where clearly the system is not able to reject the disturbance effect. Second, we implement a model-based networked system with $h = 1$ h and the system can be stabilized and disturbances are rejected. Finally, with $h = 5$ h we can obtain similar results except that the transient response has deteriorated.

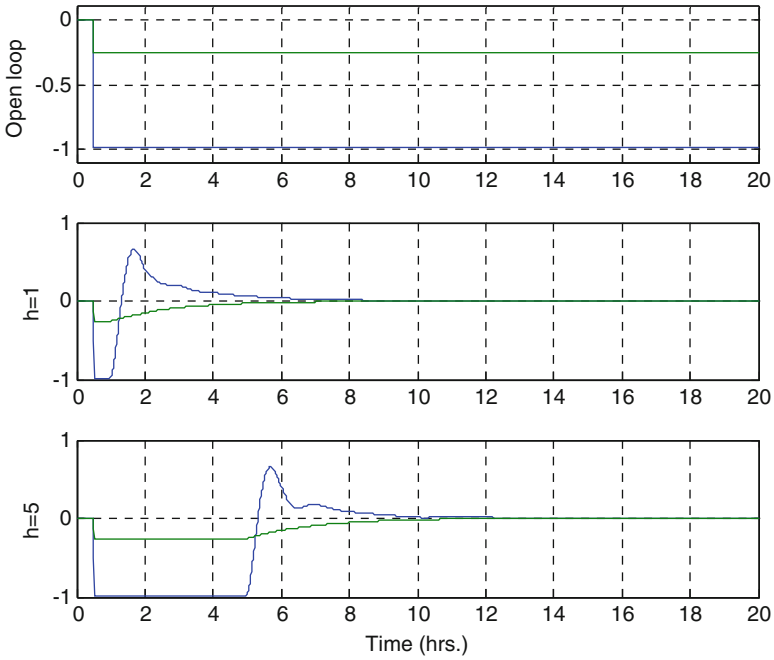

Fig. 3.7 Response (output) of distillation column to a disturbance occurring at time 0.5 h and for three cases: open-loop, model-based networked system with h = 1 h, and model-based networked system with h = 5 h

3.2 Output Feedback Using State Observers for Discrete Time Systems

We will now present the corresponding results for discrete time systems with state observers. The translation from continuous-time to discrete-time domain involves the same practical aspects described in Sect. 2.3. The proofs for Proposition 3.3 and Theorem 3.4 below follow directly from the results provided in the current and previous sections and are, therefore, omitted.

Let us consider the following equations:

$$\begin{aligned}
&\text{Plant}: x(n+1) \quad = Ax(n) + Bu(n), y(n) = Cx(n) + Du(n) \\
&\text{Model}: \hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n), y(n) = \hat{C}\hat{x}(n) + \hat{D}u(n) \\
&\text{Controller}: \quad u(n) = K\hat{x}(n) \\
&\text{Observer}: \bar{x}(n+1) = \left(\hat{A} - L\hat{C}\right)\bar{x}(n) + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} u(n) \\ y(n) \end{bmatrix} \\
&\text{for } n \in [n_k, n_{k+1}), \quad \text{with} \quad n_{k+1} - n_k = h
\end{aligned} \tag{3.27}$$

As with the continuous plant case, we use the same model-based framework to include a standard state observer at the output of the plant. This observer will provide estimates of the state of the plant at every time step n since it has access to the measurements at every sampling period $n = 0,1,2,\ldots$; but it will send the state estimate to the controller every h sampling periods, where $h \geq 1$ is an integer. The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described by:

$$\begin{bmatrix} x(n+1) \\ \overline{x}(n+1) \\ \overline{e}(n+1) \end{bmatrix} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\widetilde{D}K & -\hat{B}K - L\widetilde{D}K \\ LC & L\widetilde{D}K - L\hat{C} & \hat{A} - L\widetilde{D}K \end{bmatrix} \begin{bmatrix} x(n) \\ \overline{x}(n) \\ \overline{e}(n) \end{bmatrix},$$

for $n \in [n_k, n_{k+1})$, with $n_{k+1} - n_k = h$, and $\overline{e}(n_{k+1}) = 0$. (3.28)

Define $z = [x \quad \overline{x} \quad \overline{e}]^T$, and $\Lambda_F = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\widetilde{D}K & -\hat{B}K - L\widetilde{D}K \\ LC & L\widetilde{D}K - L\hat{C} & \hat{A} - L\widetilde{D}K \end{bmatrix}$

so that (3.28) can be written as:

$$z(n+1) = \Lambda_F z(n) \quad \text{for} \quad n \in [n_k, n_{k+1}) \text{ and } z(n_k) = \begin{bmatrix} x(n_k^-) \\ \overline{x}(n_k^-) \\ 0 \end{bmatrix} \tag{3.29}$$

Proposition 3.3 *The system with dynamics described by* (3.29) *with initial conditions* $z(n_0) = \begin{bmatrix} x(n_0) \\ \overline{x}(n_0) \\ 0 \end{bmatrix} = z_0$, *has the following response*:

$$z(n) = \Lambda_F{}^{n-n_k} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda_F{}^{h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)^k z_0, \tag{3.30}$$

for $n \in [n_k, \quad n_{k+1}), with\ n_{k+1} - n_k = h$

♦

The following theorem presents necessary and sufficient conditions for system (3.29) to be exponentially stable at large (or globally). For the definition of stability refer to Definition 2.5 in Sect. 2.3.

Theorem 3.4 *The system described by (3.29) is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ \overline{x} \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $M_F = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} {\Lambda_F}^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ *are strictly inside the unit circle.* ♦

Proofs of Proposition 3.3 and Theorem 3.4 are similar to the continuous time counterparts and have been omitted.

3.3 Network Induced Delays: Small Delay Case

Previously we assumed that the network delays were negligible. This is usually true for plants with slow dynamics relative to the network bandwidth. When this is not the case the network delay cannot be neglected. In this section, we introduce an approach where appropriate state estimates for the controller are generated based on delayed information and using the plant model. Theorem 3.6, which is the main result, presents necessary and sufficient conditions for stability. The results are illustrated by an example.

There are three important network delay sources: Processing time, Media access contention, Propagation and Transmission time. The first one, processing time, occurs at both ends of the communication channel. On the transmitter side, the processing time is the time elapsed from the time the transmitting process makes the request to the operating system to transmit a message, to the time the message is actually ready to be sent. On the receiver side this is the time interval that occurs from the time the last bit of the message is received by the receiver, to the time the message is delivered by the operating system to the receiver process. The media access contention time is the time the transmitter has to wait until the communication channel is not busy. This is usually the case when several transmitters have to share the same media.

The propagation and transmission time is the time the message takes to be streamed out by the sender on the network media and to travel through the network to reach the receiver. In local area networks the time the message takes to travel or propagate through the media is small in comparison to wide area networks or internetworks like the Internet. The time the message takes to be placed on the network depends on the size of the message and the baud rate.

If the control network is a local area network, as is common practice in industry, the propagation and transmission time can be established forehand with good accuracy. Similar observations can be made regarding the processing time. If real time operating systems are used, the processing time can be accurately calculated. Finally, media access contention delay can be fixed with the use of a communication protocol with scheduling. Fast data communication networks like Token Ring,

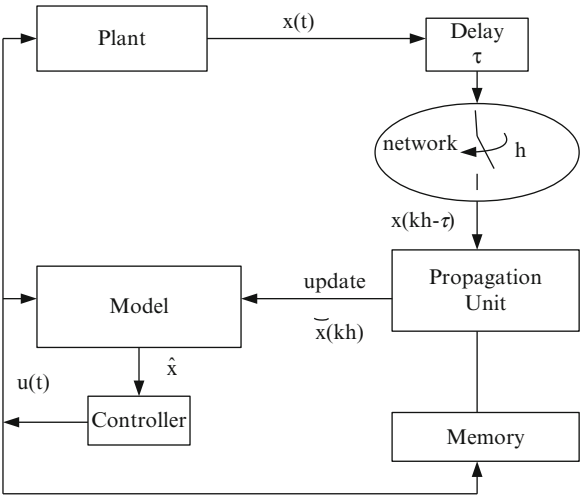

Fig. 3.8 Proposed configuration of a state feedback networked control system in the presence of network delays

Token Bus, and ArcNet fall into this category. Industry oriented control networks like Foundation Fieldbus also implement a scheduler through its LAS or Link Active Scheduler. Even the inherently non-deterministic Ethernet has addressed the problem of not having a specified contention time with the so-called Switched Ethernet.

In view of the above, the delay of a message from the sender to the receiver on a local area network can be at least bounded if standard scheduling techniques are used on the operating system of the nodes and for network contention resolution.

3.3.1 Continuous Time Systems with Delays

In the following, we present the MB-NCS setup that also includes the presence of transmission delays. We will assume that the update time h is larger than the delay time τ. We call this particular case: "small delay." In Sect. 3.4 we consider the more general case in which the delay τ is greater than the update intervals h and we called this scenario "large delay" case.

As before we will assume that the update time h is constant. We will also assume at this time that the delay τ is constant. Time-varying delays will be discussed in Chap. 6. We will present here the case of full state feedback systems.

So, at times kh-τ the sensor transmits the state data to the controller/actuator. This data will arrive τ seconds later. So, at times kh the controller/actuator will receive the state vector value $x(kh\text{-}\tau)$. The main idea is to use the plant model in the controller/actuator to calculate the present value of the state. Then, the obtained state approximation can be used to update the controller's model as in previous setups. The system is depicted in Fig. 3.8.

The Propagation Unit uses the plant model and the past values of the control input $u(t)$ to calculate an estimate of actual state $\breve{x}(kh)$ from the received data $x(kh-\tau)$. This estimate is then used to update the model that with the controller will generate the control signal for the plant.

The system is described by the following equations:

$$\begin{aligned}
&\text{Plant :} && \dot{x} = Ax + Bu \\
&\text{Model :} && \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \\
&\text{Controller :} && u = K\hat{x}, t \in \left[t_k, t_{k+1}\right) \\
&\text{Propagation Unit :} && \dot{\breve{x}} = \hat{A}\breve{x} + \hat{B}u, t \in [t_{k+1} - \tau, t_{k+1}] \\
&\text{Update law :} && \left\{ \begin{array}{l} \breve{x} \leftarrow x,\ t = t_{k+1} - \tau \\ \hat{x} \leftarrow \breve{x},\ t = t_{k+1} \end{array} \right\}.
\end{aligned} \tag{3.31}$$

To ease the analysis, we initialize the propagation unit at time $t_{k+1}-\tau$ with the state vector that the sensor obtains. We then run the plant, model, and propagation unit together until t_{k+1}. At this time, the model is updated with the propagation unit state vector, as described in the update law of (3.31). This is equivalent to having the propagation unit receive the state vector $x(t_{k+1}-\tau)$ at t_{k+1} and propagate it instantaneously to t_{k+1}.

We define the errors $\hat{e} = \breve{x} - \hat{x}$ and $\breve{e} = x - \breve{x}$. We also make the following definitions:

$$\Lambda_T = \begin{bmatrix} A + BK & -BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K & -\widetilde{B}K \\ 0 & 0 & \hat{A} \end{bmatrix},\ z = \begin{bmatrix} x \\ \breve{e} \\ \hat{e} \end{bmatrix} \tag{3.32}$$

where $\widetilde{A} = A - \hat{A}, \widetilde{B} = B - \hat{B}$ represent the modeling error matrices. The dynamics of the overall system (3.31) for $t \in [t_k, t_{k+1})$ can be described by:

$$\begin{aligned}
&\dot{z}(t) = \Lambda_T z(t), \\
&\text{with state reset equations : } z(t_k) = \begin{bmatrix} x(t_k^-) \\ \breve{e}(t_k^-) \\ 0 \end{bmatrix}, \\
&\text{and } z(t_{k+1} - \tau) = \begin{bmatrix} x\left((t_{k+1} - \tau)^-\right) \\ 0 \\ \breve{e}\left((t_{k+1} - \tau)^-\right) + \hat{e}\left((t_{k+1} - \tau)^-\right) \end{bmatrix}
\end{aligned} \tag{3.33}$$

where $t_{k+1} - t_k = h,\ 0 < \tau < h$.

Proposition 3.5 *The system with dynamics described by (3.33) with initial conditions* $z(t_0) = \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix} = z_0, t_0 = 0,$ *has the following response:*

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix} = e^{\Lambda_T (t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right)^k \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$$

for $t \in \left[t_k, t_{k+1} - \tau\right)$

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix} = e^{\Lambda_T (t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right)^k \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$$

for $t \in \left[t_{k+1} - \tau, t_{k+1}\right)$. (3.34)

with $t_{k+1} - t_k = h,\ \tau < h$.

Proof Assume the system starts at time t_0 with initial conditions $z(t_0) = \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}$. On the interval $t \in [t_0, t_1 - \tau)$, the system response is:

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \end{bmatrix} = e^{\Lambda_T (t-t_0)} \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}. \tag{3.35}$$

At $t = (t_1 - \tau)^-$:

$$z((t_1 - \tau)^-) = \begin{bmatrix} x\left((t_1 - \tau)^-\right) \\ \breve{e}\left((t_1 - \tau)^-\right) \\ \hat{e}\left((t_1 - \tau)^-\right) \end{bmatrix} = e^{\Lambda_T (h-\tau)} \begin{bmatrix} x_0 \\ \breve{e}_0 \\ \hat{e}_0 \end{bmatrix}. \tag{3.36}$$

According to the update law, at $t = t_1 - \tau$, $\breve{e} \leftarrow 0$ and $\hat{e} \leftarrow x - \hat{x} = \breve{e} + \hat{e}$, so:

$$z(t_1-\tau)=\begin{bmatrix} x(t_1-\tau)\\ \breve{e}\,(t_1-\tau)\\ \hat{e}\,(t_1-\tau)\end{bmatrix}=\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}\begin{bmatrix} x_0\\ \breve{e}_0\\ \hat{e}_0\end{bmatrix}. \tag{3.37}$$

Continuing with the interval $t\in[t_1-\tau,t_1)$

$$z(t)=\begin{bmatrix} x(t)\\ \breve{e}\,(t)\\ \hat{e}\,(t)\end{bmatrix}=e^{\Lambda_T(t-t_1+\tau)}\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}\begin{bmatrix} x_0\\ \breve{e}_0\\ \hat{e}_0\end{bmatrix} \tag{3.38}$$

At $t=t_1^{-}$:

$$z(t_1^{-})=\begin{bmatrix} x(t_1^{-})\\ \breve{e}\,(t_1^{-})\\ \hat{e}\,(t_1^{-})\end{bmatrix}=e^{\Lambda_T\tau}\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}\begin{bmatrix} x_0\\ \breve{e}_0\\ \hat{e}_0\end{bmatrix}. \tag{3.39}$$

Now, according to the update law, at $t=t_1$, $\hat{e}\leftarrow 0$, so:

$$z(t_1)=\begin{bmatrix} x(t_1)\\ \breve{e}\,(t_1)\\ \hat{e}\,(t_1)\end{bmatrix}=\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}e^{\Lambda_T\tau}\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}\begin{bmatrix} x_0\\ \breve{e}_0\\ \hat{e}_0\end{bmatrix} \tag{3.40}$$

It is easy to see that the response is given by (3.34):

$$z(t)=\begin{bmatrix} x(t)\\ \breve{e}\,(t)\\ \hat{e}\,(t)\end{bmatrix}=e^{\Lambda_T(t-t_k)}\left(\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}e^{\Lambda_T\tau}\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}\right)^k\begin{bmatrix} x_0\\ \breve{e}_0\\ \hat{e}_0\end{bmatrix}$$

for $t\in\left[t_k,t_{k+1}-\tau\right)$

$$z(t)=\begin{bmatrix} x(t)\\ \breve{e}\,(t)\\ \hat{e}\,(t)\end{bmatrix}=e^{\Lambda_T(t-t_{k+1}+\tau)}\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}$$
$$\left(\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}e^{\Lambda_T\tau}\begin{bmatrix} I & 0 & 0\\ 0 & 0 & 0\\ 0 & I & I\end{bmatrix}e^{\Lambda_T(h-\tau)}\right)^k\begin{bmatrix} x_0\\ \breve{e}_0\\ \hat{e}_0\end{bmatrix}$$

for $t\in\left[t_{k+1}-\tau,t_{k+1}\right)$.

♦

We will present now the necessary and sufficient conditions for this system to be exponentially stable at large (or globally).

Theorem 3.6 *The system described by (3.33) is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ \breve{e} \\ \hat{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $M_T = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)}$ *are inside the unit circle.*

Proof Sufficiency. Taking the norm of the solutions described in (3.34):

$$\|z(t)\| = \left\| e^{\Lambda_T (t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right)^k z_0 \right\|$$

$$\leq \left\| e^{\Lambda_T (t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right)^k \right\| \cdot \|z_0\|$$

for $t \in \left[t_k, t_{k+1} - \tau \right)$

$$\|z(t)\| = \left\| e^{\Lambda_T (t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right)^k z_0 \right\|$$

$$\leq \left\| e^{\Lambda_T (t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T (h-\tau)} \right)^k \right\| \cdot \|z_0\|$$

for $t \in \left[t_{k+1} - \tau, t_{k+1} \right)$. (3.41)

Now lets analyze the first terms on the right hand side of the inequalities in (3.41), we obtain:

$$\left\| e^{\Lambda_T (t-t_k)} \right\| \leq 1 + (t - t_k) \overline{\sigma}(\Lambda_T) + \frac{(t-t_k)^2}{2!} \overline{\sigma}(\Lambda_T)^2 \ldots = e^{\overline{\sigma}(\Lambda_T)(t-t_k)} \leq e^{\overline{\sigma}(\Lambda_T) h} = K_a \tag{3.42}$$

and

$$\left\| e^{\Lambda_T(t-t_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)} \right\| \leq \left\| e^{\Lambda_T(t-t_{k+1}+\tau)} \right\| \cdot \left\| e^{\Lambda_T(h-\tau)} \right\| = \left\| e^{\Lambda_T(t-t_{k+1}+\tau)} \right\| \cdot C$$

$$\leq \left(1 + (t-t_k+\tau)\overline{\sigma}(\Lambda_T) + \frac{(t-t_k+\tau)^2}{2!}\overline{\sigma}(\Lambda_T)^2 \ldots \right) \cdot C$$

$$= e^{\overline{\sigma}(\Lambda_T)(t-t_{k+1}+\tau)} \cdot C \leq e^{\overline{\sigma}(\Lambda_T)\tau} \cdot C = K_b \tag{3.43}$$

where $\overline{\sigma}(\Lambda_T)$ is the largest singular value of Λ_T. We define a new constant $K_I = \max(K_a, K_b)$.

We now study the term $\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)} \right)^k \right\|$. It is clear that this term will be bounded if and only if the eigenvalues of $M = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)} \right)^k \right\| \leq K_2 e^{-\alpha_1 k} \tag{3.44}$$

with $K_2, \alpha_1 > 0$.

Since k is a function of time we can bound the right term of (3.44) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} t} = K_3 e^{-\alpha t} \tag{3.45}$$

with K_3, $\alpha > 0$.

So from (3.41) using (3.42), (3.43), and (3.45) we can conclude:

$$\|z(t)\| \leq K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\|. \tag{3.46}$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the system is stable and that $M = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)}$ has at least one eigenvalue outside the unit circle. Since the system is stable, a periodic sample of the response should be bounded. In other words, the sequence of

a periodic sample of the response should converge to zero with time. We will take the sample at times t_{k+1}^-. We can express the solution $z(t_{k+1}^-)$ as:

$$\begin{aligned} z\left(t_{k+1}^-\right) &= \xi(k) \\ &= e^{\Lambda_T(\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_T \tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)} \right)^k z_0 \\ &= e^{\Lambda_T(\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} e^{\Lambda_T(h-\tau)} M^k z_0. \end{aligned} \tag{3.47}$$

We also know that M has at least one eigenvalue outside the unit circle. This means that $z(t_{k+1}^-)$ will in general grow with k. In other words, we cannot ensure $z(t_{k+1}^-)$, and $x(k)$, will converge to zero for general initial condition z_0.

$$\left\| z\left(t_{k+1}^-\right) \right\| = \|\xi(k)\| \to \infty \qquad \text{as} \qquad k \to \infty \tag{3.48}$$

this clearly means the system cannot be stable, and thus we have a contradiction. ♦

It is interesting to note that the results in Proposition 3.5 and Theorem 3.6 can be seen as a generalization of Proposition 2.1 and Theorem 2.3. The networked system (2.6) can be seen as a special case of (3.32) when $\tau = 0$. For this particular value of τ the error $\hat{e} = \breve{x} - \hat{x}$ is always equal to 0 and the error $\breve{e}$ can be expressed as $\breve{e} = x - \hat{x}$; consequently (3.32) reduces to (2.6).

Example 3.4 We present a numerical example using the linearized dynamics of the inverted pendulum on a moving cart shown in Example 3.2. Assume in this case that the whole state can be measured and transmitted to the controller node.

In this example we consider a constant delay equal to 0.3 s. By using the results in Theorem 3.6 we find that the networked system is stable for values of $0.3 < h < 1.98$. Figure 3.9 shows the response of the plant and the model for $h = 1$ s.

3.3.2 *Discrete Time Systems with Delays*

In this section we consider multi-input, multi-output linear time-invariant discrete-time systems and models that are also subject to network delays. In this case the updates will occur at some of the discrete time instants indexed by n that correspond

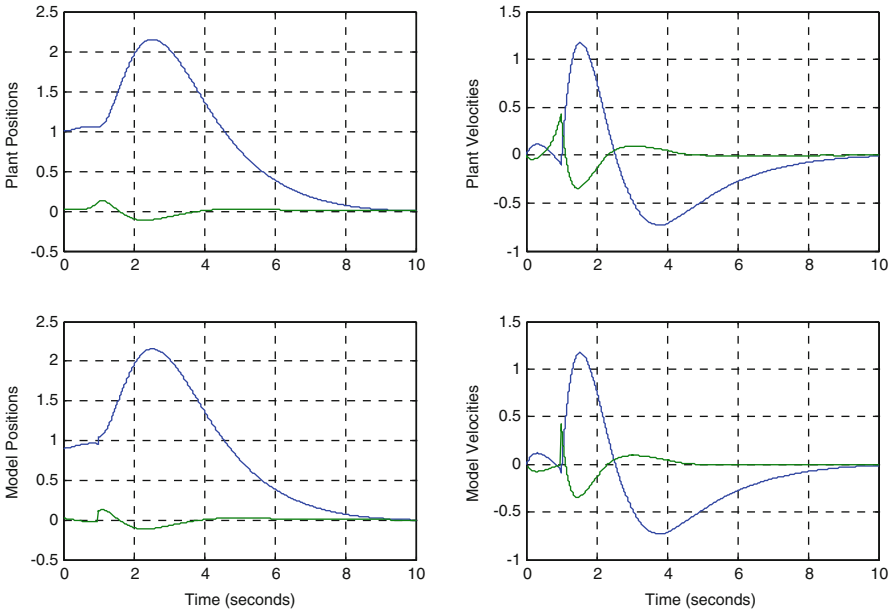

Fig. 3.9 Response of system in Example 3.4 for $h = 1$ s, $\tau = 0.3$ s

to the operation and sampling of the plant. This implies that the update interval h will be an integer number, representing the number of plant discrete instants between two consecutive measurements that the sensor broadcasts.

Note that in this section the delay can be greater than the sampling time of the system indexed by n but smaller than the update interval h at which the sensor decides to send information. For discrete-time systems we assume that the update time h is constant and an integer number. We also assume the delay τ is constant and an integer number; it represents the number of system sampling instants that the information is delayed. We will present here the case of full state feedback systems.

For this case we have that at times $n_k - \tau$ the sensor transmits the state data to the controller/actuator. This packet will arrive τ plant samplings later. So, at times kh the controller/actuator receives the state vector value $x(n_k - \tau)$. The main idea is to use the plant model in the controller/actuator to estimate the present value of the state. After this, the estimated state is used to update the controller's model.

The Propagation Unit uses the plant model and the past values of the control input $u(n)$ to calculate an estimate $\breve{x}(n_k)$ of the current state $x(n_k)$ from the received data $x(n_k - \tau)$. This estimate is then used to update the model that with the controller will generate the control signal for the plant.

The system is described by the following equations:

$$\begin{aligned}
&\text{Plant :} && x(n+1) &&= Ax(n) + Bu(n) \\
&\text{Model :} && \hat{x}(n+1) &&= \hat{A}\hat{x}(n) + \hat{B}u(n), \quad n \in [n_k, n_{k+1}) \\
&\text{Controller :} && u(n) &&= K\hat{x}(n) \\
&\text{Propagation} \\
&\text{Unit :} && \breve{x}(n+1) &&= \hat{A}\breve{x}(n) + \hat{B}u(n), \quad n \in [n_k - \tau, n_k] \\
&\text{Update law :} && \left\{ \begin{array}{l} x \to \breve{x}, \; n = n_k - \tau \\ \breve{x} \to \hat{x}, \quad n = n_k \end{array} \right\}.
\end{aligned} \tag{3.49}$$

To ease the analysis, we initialize the propagation unit at time n_k-τ with the state vector that the sensor obtains. We then run the plant, model, and propagation unit together until n_k. At this time, the model is updated with the propagation unit state vector, as described in the update law of (3.49). This is equivalent to having the propagation unit receive the state vector $x(n_k-\tau)$ at n_k and to instantaneously compute an estimate $\breve{x}(n_k)$ of the current state $x(n_k)$.

We define the errors $\hat{e}(n) = \breve{x}(n) - \hat{x}(n)$ and $\breve{e}(n) = x(n) - \breve{x}(n)$. It can be shown that the dynamics of the state and the errors can be represented by:

$$\begin{aligned}
x(n+1) &= (A + BK)x(n) - BK\breve{e}(n) - BK\hat{e}(n) \\
\breve{e}(n+1) &= (\widetilde{A} + \widetilde{B}K)x(n) + (\hat{A} - \widetilde{B}K)\breve{e}(n) - \widetilde{B}K\hat{e}(n) \\
\hat{e}(n+1) &= \hat{A}\hat{e}(n).
\end{aligned} \tag{3.50}$$

Define the augmented state vector $z = \left[x^T \;\; \breve{e}^T \;\; \hat{e}^T \right]^T$ then the augmented system can be represented in compact form as:

$$z(n+1) = \Lambda z(n), \tag{3.51}$$

where $\Lambda = \begin{bmatrix} A + BK & -BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K & -\widetilde{B}K \\ 0 & 0 & \hat{A} \end{bmatrix}$.

According to the update laws in (3.49) we have the augmented state reset equations:

$$\begin{aligned}
z(n_k - \tau) &= \begin{bmatrix} x\left((n_k - \tau)^-\right) \\ 0 \\ \breve{e}\left((n_k - \tau)^-\right) + \hat{e}\left((n_k - \tau)^-\right) \end{bmatrix}, \\
z(n_k) &= \begin{bmatrix} x(n_k^-) \\ \breve{e}(n_k^-) \\ 0 \end{bmatrix}
\end{aligned} \tag{3.52}$$

where $n_{k+1} - n_k = h,\ 0 < \tau < h$. The dynamics of the overall system for $n \in [n_k, n_{k+1})$ can be described as shown next.

Proposition 3.7 *The system with dynamics described by (3.49) with initial conditions* $z(n_0) = \left[x_0^T \ \breve{e}_0^T \ \hat{e}_0^T\right]^T = z_0$, $n_0 = 0$, *has the following response*:

$$z(n) = \begin{bmatrix} x(n) \\ \breve{e}(n) \\ \hat{e}(n) \end{bmatrix} = \Lambda^{(n-n_k)}\Sigma^k z_0 \tag{3.53}$$

$$\text{for } n \in \left[n_k, n_{k+1} - \tau\right)$$

$$z(n) = \begin{bmatrix} x(n) \\ \breve{e}(n) \\ \hat{e}(n) \end{bmatrix} = \Lambda^{(n-n_{k+1}+\tau)} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)}\Sigma^k z_0$$

$$\text{for } n \in \left[n_{k+1} - \tau, n_{k+1}\right).$$

where $\Sigma = \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} \right)$.

Theorem 3.8 *The networked system described by (3.49) with constant updates* h *and constant delays* $\tau < h$ *is globally exponentially stable around the solution* $z_0 = 0$ *if and only if the eigenvalues of*

$$\Sigma = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I & I \end{bmatrix} \Lambda^{(h-\tau)} \tag{3.54}$$

are within the unit circle in the complex plane.

Remark The analysis presented above for discrete-time systems offers an important advantage compared to its continuous-time counterpart discussed in the previous section. In order to compute the propagated state based on the delayed measurements we need to store the previous values of the control inputs that were used over the interval $[n_k - \tau, n_k)$. For discrete-time systems the control input history over $[n_k - \tau, n_k)$ can be represented by a finite number of values but for continuous-time systems it is not possible to store in digital memory an infinite number of values that characterize the input $u(t)$. In this case we need to sample sufficiently fast in order to obtain a good approximation of the continuous control input $u(t)$.

3.4 Network Induced Delays: Large Delay Case

In this section we extend the approach discussed in previous sections to consider the case when $\tau > h$, that is, the delay τ is larger than the update interval h. We consider both continuous and discrete-time systems. The aim is to obtain conditions for stability in the presence of delays, when the delays are larger than the periodic update intervals. The solution to this problem is obtained by considering an increased number of propagation state variables which, in turn, requires the state vector z to be augmented to include additional error variables.

3.4.1 Continuous Time Systems

For continuous time systems h and τ can be real numbers, they are not restricted to be integers. For simplicity we consider first the case when $h < \tau < 2h$. The general case $ah < \tau < (a+1)h$ for any positive integer a can be solved using the same approach but adding more error variables.

We consider two propagation state variables $\breve{x}_1$ and $\breve{x}_2$. We update the state $\breve{x}_2$ at time $t_k - \tau$ using the real state $x(t_k - \tau)$. At time t_k we perform two updates in sequential manner. First, we update the state of the model $\hat{x}(t_k)$ using the value $\breve{x}_1(t_k)$ and then we update $\breve{x}_1(t_k)$ using the value $\breve{x}_2(t_k)$. The overall setup for continuous-time systems can be described as follows:

$$
\begin{aligned}
&\text{Plant :} && \dot{x}(t) = Ax(t) + Bu(t) \\
&\text{Model :} && \dot{\hat{x}}(t) = \hat{A}\,\hat{x}(t) + \hat{B}\,u(t), \quad t \in [t_k, t_{k+1}) \\
&\text{Controller :} && u(t) = K\hat{x}(t) \\
&\text{Propagation} && \dot{\breve{x}}_i(t) = \hat{A}\,\breve{x}_i(t) + \hat{B}\,u(t), \quad t \in [t_k - \tau, t_k], \quad \text{for } i = 1, 2 \\
&\text{Unit :} && \\
&\text{Update law :} && \left\{ \begin{array}{c} x \to \breve{x}_2,\ t = t_k - \tau \\ \breve{x}_1 \to \hat{x}, \quad \text{then} \quad \breve{x}_2 \to \breve{x}_1, \quad t = t_k \end{array} \right\}.
\end{aligned}
\tag{3.55}
$$

Using this representation we ensure that the model of the system that generates the state $\hat{x}(t)$ is updated at time t_k with the propagated variable that is computed based on the measurement that was sent by the sensor at time $t_k - \tau$, see Fig. 3.10. In this figure we can see that at time t_k the plant model in the controller is updated with information generated at time $t_k - \tau$. Even though a new measurement has been sent by the senor node at time $t_{k+1} - \tau$, this new information has not arrived yet to the controller node due to large delay $\tau > h$.

Remark Note that in the actual implementation we only need one propagation unit (and only one propagation state variable) that receives the delayed measurement

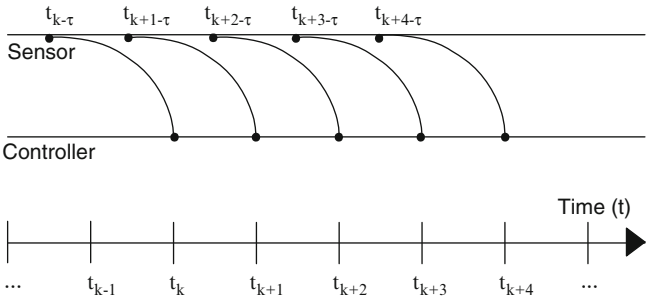

Fig. 3.10 Representation of update intervals in the presence of delays $h < \tau < 2h$

and propagates it instantaneously to time t_k using the previous control input $u(t)$ stored over the interval $[t_k - \tau, t_k]$.

Define the errors $\breve{e}(t) = x(t) - \breve{x}_2(t)$, $\hat{e}(t) = \breve{x}_1(t) - \hat{x}(t)$ and $\widehat{e}(t) = \breve{x}_2(t) - \breve{x}_1(t)$. It can be shown that the dynamics of the state and the errors can be written as:

$$\begin{aligned}
\dot{x}(t) &= (A + BK)x(t) - BK\breve{e}(t) - BK\hat{e}(t) - BK\,\widehat{e}(t) \\
\dot{\breve{e}}(t) &= (\widetilde{A} + \widetilde{B}K)x(t) + (\hat{A} - \widetilde{B}K)\breve{e}(t) - \widetilde{B}K\hat{e}(t) - \widetilde{B}K\,\widehat{e}(t) \\
\dot{\hat{e}}(t) &= \hat{A}\,\hat{e}(t) \\
\dot{\widehat{e}}(t) &= \hat{A}\,\widehat{e}(t).
\end{aligned} \tag{3.56}$$

Define the augmented state vector $z = \left[x^T \;\; \breve{e}^T \;\; \hat{e}^T \;\; \widehat{e}^T\right]^T$ then the augmented system can be described by:

$$\dot{z}(t) = \Lambda z(t), \tag{3.57}$$

where $\Lambda = \begin{bmatrix} A + BK & -BK & -BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K & -\widetilde{B}K & -\widetilde{B}K \\ 0 & 0 & \hat{A} & 0 \\ 0 & 0 & 0 & \hat{A} \end{bmatrix}$. From the update laws (3.55) we obtain the augmented state reset equations:

$$z(t_k - \tau) = \begin{bmatrix} x\left((t_k - \tau)^-\right) \\ 0 \\ \hat{e}\left((t_k - \tau)^-\right) \\ \breve{e}\left((t_k - \tau)^-\right) + \widehat{e}\left((t_k - \tau)^-\right) \end{bmatrix}, \quad z(t_k) = \begin{bmatrix} x\left(t_k^-\right) \\ \breve{e}\left(t_k^-\right) \\ \widehat{e}\left(t_k^-\right) \\ 0 \end{bmatrix} \tag{3.58}$$

where $t_{k+1} - t_k = h, \quad h < \tau < 2h.$

Proposition 3.9 *The system with dynamics described by* (3.55) *with initial conditions* $z(t_0) = \begin{bmatrix} x_0^T & \breve{e}_0^T & \hat{e}_0^T & \widehat{e}_0^T \end{bmatrix}^T = z_0$, $t_0 = 0$, *has the following response*:

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \\ \widehat{e}(t) \end{bmatrix} = e^{\Lambda(t-t_k)}\Sigma^k z_0 \tag{3.59}$$

$$\text{for } t \in \left[t_k, t_{k+1} - \tau'\right)$$

$$z(t) = \begin{bmatrix} x(t) \\ \breve{e}(t) \\ \hat{e}(t) \\ \widehat{e}(t) \end{bmatrix} = e^{\Lambda(t-t_{k+1}+\tau')} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')}\Sigma^k z_0$$

$$\text{for } t \in \left[t_{k+1} - \tau', t_{k+1}\right)$$

$$\text{where } \Sigma = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau'} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')}, \text{ and } \tau' = \tau - h.$$

Theorem 3.10 *The networked system described by* (3.55) *with constant updates* h *and constant delays* $h < \tau < 2h$ *is globally exponentially stable around the solution* $z_0 = 0$ *if and only if the eigenvalues of*

$$\Sigma = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau'} \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')} \tag{3.60}$$

are within the unit circle of the complex plane.

Remark The most important advantage of the results presented in this section compared to the special case when the delays are restricted to be smaller than the update interval can be better appreciated in the next scenario: for a given plant, model, controller, and network delay τ there may not be an update interval $h > \tau$ that stabilizes the system (as shown in Example 3.3 below), but using the results in this section, we can find some smaller interval $h < \tau$ that results in a stable system.

We can extend the previous results in order to establish necessary and sufficient conditions for stability for the general case when $ah < \tau < (a+1)h$ by adding a additional propagation (where a is a positive integer) state variables with respect to the small delay case described in Sect. 3.3. For instance, in the previous section we had that $h < \tau < 2\,h$ so $a = 1$; there we added one additional propagation variable with respect to the small delay case and we had a total of two propagation variables.

Theorem 3.11 *The networked system described by (3.55) with constant updates h, constant delays $ah < \tau < (a+1)h$, and with augmented state $z = \left[x^T \;\breve{e}^T \;\hat{e}^T \;\breve{e}_1^T \ldots \;\breve{e}_a^T\right]^T$ is globally exponentially stable around the solution $z_0 = 0$ if and only if the eigenvalues of*

$$\Sigma = \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & I \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix} e^{\Lambda\tau'} \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & I & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & I \end{bmatrix} e^{\Lambda(h-\tau')} \tag{3.61}$$

are inside the unit circle, where $\tau' = \tau - ah$, $\breve{e}(t) = x(t) - \breve{x}_{a+1}(t)$, $\hat{e}(t) = \breve{x}_1(t) - \hat{x}(t)$, $\breve{e}_i(t) = \breve{x}_{i+1}(t) - \breve{x}_i(t)$, for $i = 1,2\ldots a$, and the $(a+3) \times (a+3)$ matrix Λ is given by:

$$\Lambda = \begin{bmatrix} A + BK & -BK & -BK & -BK & \ldots & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K & -\widetilde{B}K & -\widetilde{B}K & \ldots & -\widetilde{B}K \\ 0 & 0 & \hat{A} & 0 & \ldots & 0 \\ 0 & 0 & 0 & \hat{A} & \ldots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \hat{A} \end{bmatrix}. \tag{3.62}$$

Example 3.5 Consider the following unstable continuous-time plant, model, and controller gain:

$$A = \begin{bmatrix} -0.349 & 0.65 \\ -0.316 & 0.787 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

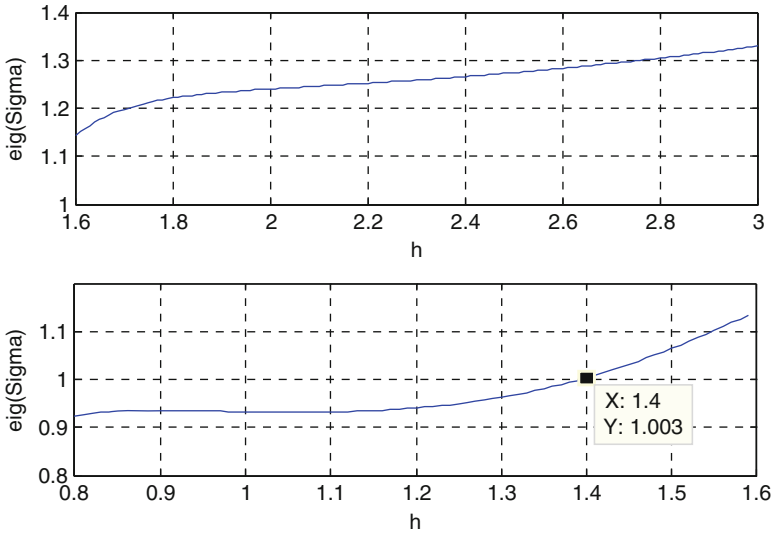

Fig. 3.11 $\tau = 1.6$ s. Eigenvalue of Σ with largest magnitude for $\tau < h$ (*top*) and for $h < \tau < 2h$ (*bottom*)

$$\hat{A} = \begin{bmatrix} -0.5 & 1 \\ 0 & 0.6 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} -0.008 \\ 1 \end{bmatrix} \tag{3.63}$$
$$K = [\,-5.4621 \quad -11.1658\,].$$

Suppose that the networked induced delay is constant and equal to 1.6 s. For this delay there is no update interval $h > \tau$ that provides stability as shown in the top part of Fig. 3.11. If we decrease the update interval, there exist values of h as shown in the bottom part of Fig. 3.11, for which the system is stable. An example of a stable response of the system for $\tau = 1.6$ time units is shown in Fig. 3.13a using $h = 1.2$ time units.

Suppose now that the delay is larger, $\tau = 1.85$ s. Figure 3.12 shows that stabilizing values of h exist for the range for $2h < \tau < 3h$ (c), but not for $h < \tau < 2h$ (b) or for $\tau < h$ (a). The response of the system for $\tau = 1.85$ s and using $h = 0.73$ s is shown in Fig. 3.13b.

3.4.2 *Discrete-Time Systems*

The results of the previous section can be easily extended to consider discrete time system with large delays. Keep in mind that the update intervals h and the delays τ are constrained to be integers. The following theorem provides necessary and

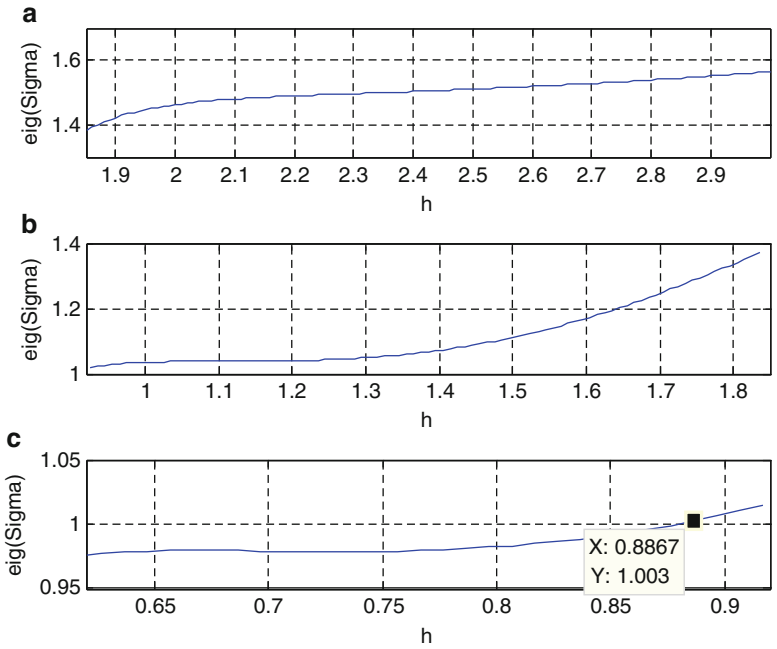

Fig. 3.12 $\tau = 1.85$ s. Eigenvalue of Σ with largest magnitude for $\tau < h$ (**a**), for $h < \tau < 2h$ (**b**), and for $2h < \tau < 3h$ (**c**)

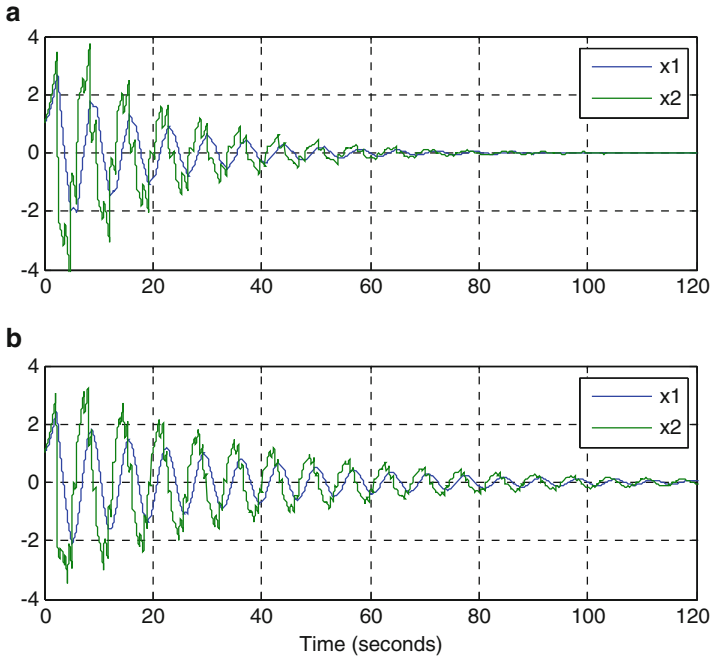

Fig. 3.13 Response of networked system in example 1. (**a**) For $\tau = 1.6$ s and h = 1.2 s. (**b**) For $\tau = 1.85$ s and $h = 0.73$ s

sufficient conditions for stability of discrete time networked control systems with delays. The proof follows directly from the results in Sects. 3.3.2 and 3.4.1.

Theorem 3.12 *The networked system described by* (3.49) *with constant updates* h, *constant delays* $ah < \tau < (a+1)h$, *and with augmented state* $z = \left[x^T \;\; \breve{e}^T \;\; \hat{e}^T \;\; \breve{e}_1^T \ldots \;\; \breve{e}_a^T\right]^T$ *is globally exponentially stable around the solution* $z_0 = 0$ *if and only if the eigenvalues of*

$$\Sigma = \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & I \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix} \Lambda^{\tau'} \begin{bmatrix} I & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & I & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & I & \ldots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I & 0 \\ 0 & I & 0 & 0 & \ldots & 0 & I \end{bmatrix} \Lambda^{(h-\tau')} \tag{3.64}$$

are within the unit circle of the complex plane, where $\tau' = \tau - ah$, $\breve{e}(n) = x(n) - \breve{x}_{a+1}(n)$, $\hat{e}(n) = \breve{x}_1(n) - \hat{x}(n)$, $\breve{e}_i(n) = \breve{x}_{i+1}(n) - \breve{x}_i(n)$, *for* $i = 1,2\ldots a$, *and the* $(a+3) \times (a+3)$ *matrix* Λ *is given by* (3.62).

3.5 Notes and References

This chapter extended the initial results on MB-NCS to control systems where it is only possible to measure the output of the plant but not the entire state.

The approach described in Sect. 3.1 uses a traditional Luenberger observer [76] and studies the associated convergence and stability results when the observed states are used to update the model of the system at the controller node. A similar analysis was presented in Sect. 3.2 for discrete time systems. In both cases we design the controller and observer gains and then analyze the stability of the system given these gains and the uncertainties of the system. It was shown that more complex expression than the simple, separate design of the controller and observer gains needs to be followed due to the system uncertainties and the absence of feedback for intervals of time to be determined.

The MB-NCS framework was also extended to consider a particular problem in NCS, namely, network induced delays. One of the main advantages using the model-based approach in comparison to typical methods in NCS is that the delayed measurements are propagated in order to obtain a more accurate estimation of the current states.

The work in Sects. 3.1 and 3.2 was first published in [187] and [188]. The work in Sect. 3.3 appeared first in [188]. The extensions to consider large delays using multiple propagation units are based on [86].

For the case in which the observer is implemented in the sensor node, the computational load of an observer in the sensor is justified by the fact that, typically sensors which can be connected to a network have an embedded processor inside. Ishii and Francis [124] give a similar justification. In their approach an observer is placed at the output of the plant to reconstruct the state vector. The result is then quantized and sent over the network to the controller.

Chapter 11 revisits the output feedback problem when considering tracking of reference inputs. Different system representation and controller designs are used and state observers are not required.

Many different approaches exist for output feedback stabilization of networked systems. For instance, [292] presented a method based on linear matrix inequalities. The paper [225] discussed the detectability and output feedback stabilizability problems for NCS of nonlinear systems using algebraic Riccati equations. In [229] the output feedback system is modeled as a Markov chain and a two-mode controller is designed.

Similarly, interest in NCS with time delays has increased in the recent years. Assumptions about network induced delays vary from constant delays to time-varying deterministic delays and also time-varying stochastic delays that may follow one of many different probability distributions. The authors of [291] explored different problems related to NCS including time delays and offer solutions using hybrid systems techniques. Other techniques that consider network induced delays in NCS are given in [132] and [230]. In more recent work such as [276] the authors use linear matrix inequalities for controller design in the presence of stochastic delays from the sensor to the controller and from the controller to the system that follow a Bernoulli distribution. The work in [46] provides general models of NCSs that consider time-varying sampling intervals and delays. In [45] the authors use the Jordan canonical form of the continuous-time plant in order to derive stability conditions considering uncertain, time varying delays that can be smaller or larger than the sampling interval of the network.

Chapter 4
Model-Based Control Systems with Intermittent Feedback

The general idea using the intermittent feedback approach is to operate a control system in open-loop mode for as long as possible in order to reduce the use of resources (communication resources when applied to NCS) and to change the mode of operation to closed-loop mode in order to recover some desired performance by applying a continuous feedback control action. The present chapter develops this idea in the context of MB-NCS. The overall networked architecture and approach are similar to those presented before in Chaps. 2 and 3. The main difference is that the update strategy in previous chapters uses instantaneous feedback, that is, a single set of measurements is sent from the sensor node to the controller node once every h time units. In contrast, the intermittent approach establishes a closed-loop operation mode in which multiple measurements are transmitted starting at every h time units.

Intermittent feedback is a very natural way of controlling the response of a system. A simple example can be found in human learning. When a person is performing certain operation or action repeatedly, s/he first pays special attention to learn it and eventually the same person can perform the same action with "closed eyes" which in this case it means open loop or lack of visual feedback. If the operation or process that the person is controlling changes or if at some moment it is easily perceived that the outcome is significantly different from the desired one then it is necessary to obtain visual feedback to discern how his actions (control) affect the results and make all corrections and adjustments as needed (to compute a different control input).

The implications and characteristics of the use of intermittent feedback in automatic control as explained here have not been fully studied. Specifically, the potential of intermittent feedback in NCS is of great interest. The central idea in this chapter is to take advantage of the preexisting work in model-based networked control systems and of the concept of intermittent feedback to develop new results and ideas that can lead us to a more complete and comprehensive understanding of analysis and design of networked control systems. In addition to what we have said in the previous paragraph about intermittent feedback, the difference between closed-loop and open-loop dynamics suggests a hybrid nature for the system (switching between two modes), and since networked control systems display characteristics of hybrid systems as well, the application of the intermittent

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_4,

feedback concept to networked control systems seems especially intuitive. Also, since, under an intermittent feedback scenario we have entire intervals of time when the system is running closed loop, it makes it far more natural to relate dynamics in networked control systems to results in classical, continuous-time control theory.

In this chapter we extend the key results of Chaps. 2 and 3 in order to consider the intermittent behavior that may be encountered in several types of networks. Proofs for most theorems are similar to the corresponding proofs in the previous chapters and, therefore, have been omitted. The present chapter is organized as follows: In Sect. 4.1 we present the Model-Based Control with intermittent feedback architecture and provide stability results for continuous-time systems. In Sect. 4.2 we introduce a different approach for intermittent feedback that relaxes the use of continuous feedback during the closed-loop times. Discrete-time systems are studied in Sect. 4.3. In Sect. 4.4 an alternative lifting approach is used to study stability of discrete-time systems. Notes and references are in Sect. 4.5.

4.1 Continuous-Time Systems with State Feedback and Closed-Loop Mode

In this section the architecture of MB-NCS with intermittent feedback is first discussed. Then conditions for stability are presented based on the two modes of operation: open loop and closed loop. It is assumed in this section that when the networked system operates in closed-loop the controller receives continuous feedback from the sensor. This is a restrictive assumption since the network has limited bandwidth. This assumption will be relaxed later in Sect. 4.2. Nevertheless, the ideas presented in this section provide a clear idea about the behavior of this type of system.

4.1.1 Model-Based Control with Intermittent Feedback Architecture

We consider a similar architecture to the one we used for MB-NCS in Chap. 2. A nominal model of the plant that approximates the dynamics of the real plant is implemented in the controller node in order to generate an estimate $\hat{x}$ of the plant state x during the open-loop intervals. The main difference with respect to the traditional MB-NCS setup, as it can be seen in Fig. 4.1, is that when the networked system operates in closed-loop mode the control input is a function of the real plant state $x(t)$ and $\hat{x}(t) = x(t)$ during the whole closed-loop interval.

In dealing with intermittent feedback, we have two key time parameters: how frequently we want to close the loop, which we shall denote by h; and how long we wish the loop to remain closed, which we shall denote by τ, see Fig. 4.2. Naturally, in the more general cases both h and τ can be time-varying. For the purposes on this chapter, however, we will deal only with the case where both h and τ are fixed.

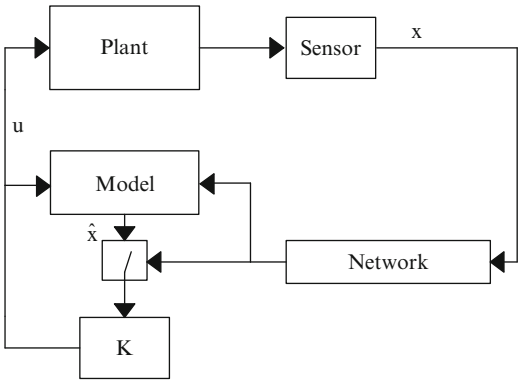

Fig. 4.1 MB-NCS with intermittent feedback (basic architecture)

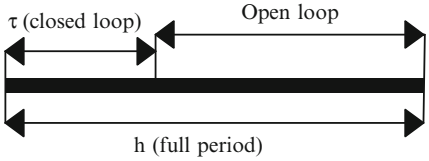

Fig. 4.2 Intermittent feedback. Open and closed-loop periods

We consider then a system such that the loop is closed periodically, every h seconds, and where each time the loop closes, it remains so for a time of τ seconds. The loop is closed at times t_k for $k = 1, 2, \ldots$. Thus, there are two very clear modes of operation: closed loop and open loop. *The system will be operating in closed-loop mode for the intervals* $[t_k, t_k + \tau)$ *and in open loop for the intervals* $[t_k + \tau, t_{k+1})$. Intuitively, we should be able to achieve much better results the longer the loop is closed. Since the level of degradation of the information increases the longer the system is running open loop, it is expected that intermittent feedback should yield better results than those for traditional MB-NCS with the same period h. This is indeed the case. The remaining sections summarize the corresponding results described in Chaps. 2 and 3 but now using the intermittent feedback approach.

4.1.2 Stability of Continuous-Time MB-NCS with Intermittent Feedback

In this section we consider uncertain linear time invariant systems that are interconnected as shown in Fig. 4.1. The uncertain plant and the nominal plant model are represented by:

$$\begin{aligned} \text{Plant} \quad & \dot{x} = Ax + Bu \\ \text{Model} \quad & \dot{\hat{x}} = \hat{A}\,\hat{x} + \hat{B}\,u \end{aligned} \tag{4.1}$$

and the controller is given by

$$u = \begin{cases} Kx & for \;\; t \in [t_k, t_k + \tau) \\ K\hat{x} & for \;\; t \in [t_k + \tau, t_{k+1}). \end{cases} \tag{4.2}$$

The state error is defined as

$$e(t) = x(t) - \hat{x}\,(t) \tag{4.3}$$

and represents the difference between the plant state and the model state. The modeling error matrices $\widetilde{A} = A - \hat{A}$ and $\widetilde{B} = B - \hat{B}$ represent the difference between the plant and the model. We also define the augmented state vector $z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix}$.

Proposition 4.1 *The system described by (4.1) with input (4.2) and initial conditions* $z(t_0) = \begin{bmatrix} x(t_0) \\ 0 \end{bmatrix} = z_0$, *has the following response:*

$$z(t) = \begin{cases} e^{\Lambda_c (t - t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0, \quad t \in [t_k, t_k + \tau) \\ e^{\Lambda_o (t - (t_k + \tau))} e^{\Lambda_c (\tau)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0, \quad t \in [t_k + \tau, t_{k+1}) \end{cases} \tag{4.4}$$

where $\Sigma = e^{\Lambda_o (h - \tau)} e^{\Lambda_c (\tau)}$, $\Lambda_c = \begin{bmatrix} A + BK & -BK \\ 0 & 0 \end{bmatrix}$, $\Lambda_o = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}$, *and* $t_{k+1} - t_k = h$.

Proof We will now proceed to derive the response of the MB-NCS with intermittent feedback to prove this proposition in a direct way. To this effect, let us separately investigate the state response when the system is operating under closed and open-loop conditions, respectively.

During the open-loop case, that is, when $t \in [t_k + \tau, t_{k+1})$, according to (4.2) we have that $u = K\hat{x}$ so

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{4.5}$$

with initial conditions $\hat{x}\,(t_k + \tau) = x(t_k + \tau)$.

Rewriting in terms of x and e, that is, of the vector z:

$$\begin{aligned}
\dot{z}(t) &= \begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \\
z(t_k+\tau) &= \begin{bmatrix} x(t_k+\tau) \\ e(t_k+\tau) \end{bmatrix} = \begin{bmatrix} x\left(t_k^- + \tau\right) \\ 0 \end{bmatrix}, \\
&\forall t \in \left[t_k+\tau, t_{k+1}\right).
\end{aligned} \tag{4.6}$$

Thus, we have

$$\dot{z} = \Lambda_o z, \text{ where } \Lambda_o = \begin{bmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{bmatrix} \text{ for } t \in \left[t_k+\tau, t_{k+1}\right). \tag{4.7}$$

The closed-loop case is a simplified version of the case above, as the difference resides in the fact that the state error is always zero. Thus, for $t \in [t_k, t_k+\tau)$ we have

$$\dot{z} = \Lambda_c z, \text{ where } \Lambda_c = \begin{bmatrix} A+BK & -BK \\ 0 & 0 \end{bmatrix} \text{ for } t \in \left[t_k, t_k+\tau\right). \tag{4.8}$$

This should be clear in that the error is always zero, while the state progresses in the same way as before. From the analysis above, given an initial condition $z(t=0)=z_0$, and assuming without loss of generality that we start the cycle in closed loop then after a certain time $t \in [0,\tau)$ the solution of the trajectory of the vector is given by

$$z(t) = e^{\Lambda_c(t)} z_0, \quad t \in \left[0, \tau\right). \tag{4.9}$$

In particular, at time τ, $z(\tau) = e^{\Lambda_c(\tau)} z_0$.

Once the loop is opened, the open-loop behavior takes over, so that

$$z(t) = e^{\Lambda_0(t-\tau)} z(\tau) = e^{\Lambda_o(t-\tau)} e^{\Lambda_c(\tau)} z_0, \quad t \in \left[\tau, t_1\right). \tag{4.10}$$

In particular, when the time comes to close the loop again, that is, after time h; then

$$z\left(t_1^-\right) = e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} z_0. \tag{4.11}$$

Now, note that at times $t_k, z(t_k) = \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix}$, that is, the error $e(t)$ is reset to zero. We can represent this by pre-multiplying by $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ before we analyze the closed-loop

trajectory for the next cycle; because we wish to always start with an error that is set to zero, we should actually multiply by $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ at the beginning of the cycle as well.

Then, after k cycles, going through this analysis yields a solution.

$$z(t_k) = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \quad (4.12)$$

where $\Sigma = e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)}$.

The final step is to consider the last (partial) cycle that the system goes through. If the system is in closed loop, that is, $t \in [t_k, t_k + \tau)$ then the solution can be achieved merely by pre-multiplying $z(t_k)$ by $e^{\Lambda_c(t-t_k)}$. In the case when the system is in open loop, that is, $t \in [t_k + \tau, t_{k+1})$ then clearly we must pre-multiply by $e^{\Lambda_o(t-(t_k+\tau))} e^{\Lambda_c(\tau)}$. ♦

A necessary and sufficient condition for stability of the networked system will now be presented. For the definition of stability please refer to Definition 2.2 in Sect. 2.2.

Theorem 4.2 *The system described by (4.1) with input (4.2) is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are strictly inside the unit circle, where* $\Sigma = e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)}$.

Proof Sufficiency. Taking the norm of the solution described as in Proposition 4.1:

$$\begin{aligned} \|z(t)\| &= \left\| e^{\Lambda_c(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \\ &\leq \left\| e^{\Lambda_c(t-t_k)} \right\| \cdot \left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \cdot \|z_0\| \end{aligned} \quad (4.13)$$

Notice that here we only consider the case $t \in [t_k, t_k + \tau)$, however the process is exactly the same for the intervals where $t \in [t_k + \tau, t_{k+1})$. Now let's analyze the first factor of the right hand side of (4.13):

$$\left\| e^{\Lambda_c(t-t_k)} \right\| \leq 1 + (t - t_k)\overline{\sigma}(\Lambda_c) + \frac{(t-t_k)^2}{2!}\overline{\sigma}(\Lambda_c)^2 \ldots = e^{\overline{\sigma}(\Lambda_c)(t-t_k)} \leq e^{\overline{\sigma}(\Lambda_c)\tau} = K_1 \quad (4.14)$$

where $\overline{\sigma}(\Lambda_c)$ is the largest singular value of Λ_c. In general this term can always be bounded since the time difference $t - t_k$ is always smaller than τ. In other words even when Λ_c has eigenvalues with positive real part, $\left\| e^{\Lambda_c(t-t_k)} \right\|$ can only grow a certain amount. This growth is completely independent of k.

We now study the factor $\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\|$. It is clear that it will be bounded if and only if the eigenvalues of $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ lie inside the unit circle:

$$\left\| \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k \right\| \leq K_2 e^{-\alpha_1 k} \tag{4.15}$$

with $K_2, \alpha_1 > 0$.

Since k is a function of time we can bound the right hand side of (4.15) in terms of t:

$$K_2 e^{-\alpha_1 k} < K_2 e^{-\alpha_1 \frac{t-1}{h}} = K_2 e^{\frac{\alpha_1}{h}} e^{-\frac{\alpha_1}{h} t} = K_3 e^{-\alpha t} \tag{4.16}$$

with $K_3, \alpha > 0$.

So from (4.13) using (4.14) and (4.16) we can conclude that:

$$\|z(t)\| = \left\| e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \right\| \leq K_1 \cdot K_3 e^{-\alpha t} \cdot \|z_0\|. \tag{4.17}$$

Necessity. We will now prove the necessity part of the theorem by contradiction. Assume the state feedback MB-NCS is stable and that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle. Let us define $\Sigma(h) = e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)}$. Since the system is stable, a periodic sample of the response should be stable as well. In other words the sequence product of a periodic sample of the response should converge to zero with time. We will take the sample at times t_{k+1}^-, that is, just before the update. We will concentrate on a specific term: the state of the plant $x(t_{k+1}^-)$, which is the first element of $z(t_{k+1}^-)$. We will call $x(t_{k+1}^-)$, $\xi(k)$.

Now assume $\Sigma(\eta)$ has the following form:

$$\Sigma(\eta) = \begin{bmatrix} W(\eta) & X(\eta) \\ Y(\eta) & Z(\eta) \end{bmatrix}$$

Then we can express the solution $z(t)$ as:

$$\begin{aligned} & e^{\Lambda_c(t-t_k)}\left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\Sigma(h)\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)^k z_0 \\ &= \begin{bmatrix} W(t-t_k) & X(t-t_k) \\ Y(t-t_k) & Z(t-t_k) \end{bmatrix}\begin{bmatrix} (W(h))^k & 0 \\ 0 & 0 \end{bmatrix} z_0 \\ &= \begin{bmatrix} W(t-t_k)(W(h))^k & 0 \\ Y(t-t_k)(W(h))^k & 0 \end{bmatrix} z_0 \end{aligned} \tag{4.18}$$

Now the values of the solution at times t_{k+1}^-, that is just before the update, are

$$z\left(t_{k+1}^-\right) = \begin{bmatrix} W(h)(W(h))^k & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0 = \begin{bmatrix} (W(h))^{k+1} & 0 \\ Y(h)(W(h))^k & 0 \end{bmatrix} z_0$$

We also know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} e^{\Lambda_c(\tau)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ has at least one eigenvalue outside the unit circle, which means that those unstable eigenvalues must be in $W(h)$. This means that the first element of $z(t_{k+1}^-)$, which we call $\xi(k)$, will in general grow with k. In other words we can't ensure $\xi(k)$ will converge to zero for general initial condition x_0. In general,

$$\left\|x\left(t_{k+1}^-\right)\right\| = \|\xi(k)\| = \left\|(W(h))^{k+1} x_0\right\| \to \infty \quad \text{as} \quad k \to \infty \tag{4.19}$$

which means that the system cannot be stable, and thus we have a contradiction.♦

4.2 A Different Approach: Fast-Slow Update Rates

In practice when considering continuous-time systems we are not really capable of implementing a closed-loop mode of operation, since it requires continuous feedback from the sensor to the controller using a limited-bandwidth digital network as a communication medium.

A more realistic way of implementing intermittent feedback for continuous-time systems is to consider fast and slow update rates. The difference of the approach in this section compared to one in the previous section is that when the closed-loop mode turns on the controller does not receive continuous measurements from the sensor; instead the sensor sends measurements at discrete points of time which are more frequent than when the open-loop mode was used.

Open loop. No measurements are received.

Closed loop. Updates are received at higher rate but continuous feedback is not assumed.

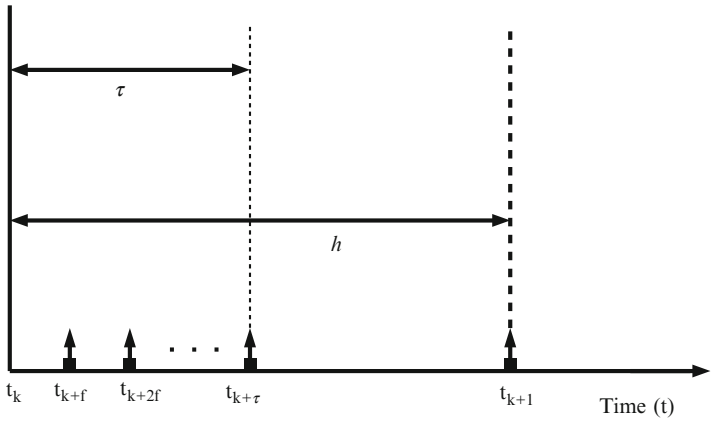

Fig. 4.3 Representation of parameters involved in the model-based approach with intermittent feedback

4.2.1 State Feedback

There are several parameters that are used for the analysis of this type of intermittent feedback MB-NCS (Fig. 4.3).

h: represents the duration of the entire cycle.
τ: represents the interval of time when the controller is receiving updates at higher frequency; this corresponds to the closed-loop duration of the cycle.
f: is the time interval between successive fast updates.
$\tau_f = \frac{\tau}{f}$: represents the number of fast updates during the closed-loop duration of the cycle and it is assumed to be an integer.

Consider the plant, control input, and state error defined in (4.1)–(4.3). The response of the MB-NCS with fast and slow update rates is given in the following proposition.

Proposition 4.3 *The system described by (4.1) and input (4.2) with initial conditions* $z(t_0) = \begin{bmatrix} x(t_0) \\ 0 \end{bmatrix} = z_0$, *has the following response:*

$$z(t_k) = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^{t_f} \right)^k z_0 \tag{4.20}$$

where $\Lambda = \begin{bmatrix} A + BK & -BK \\ \widetilde{A} + \widetilde{B}K & \hat{A} - \widetilde{B}K \end{bmatrix}$, $z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix}$, and $t_{k+1} - t_k = h$.

Proof The times t_k $k=0,1,\ldots$ represent the beginning of a new cycle, that is, they represent the first time instant that updates are received after an open-loop period. Let $z(t_k)$ represents the augmented state after the first update takes place. It follows that:

$$z(t) = e^{\Lambda(t-t_k)} z(t_k), \quad t \in \left[t_k, t_k + f\right). \tag{4.21}$$

In particular, at time $t_k + f$ we have $z(t_k + f) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} z(t_k)$ because of the second update. The response of the augmented system during the following period can be expressed in a similar way.

$$z(t) = e^{\Lambda(t-(t_k+f))} z(t_k + f), \quad t \in \left[t_k + f, t_k + 2f\right). \tag{4.22}$$

At time $t_k + 2f$ we obtain $z(t_k + 2f) = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^2 z(t_k)$. The response of the system at every update is of similar form including the last update of the cycle which takes place at time $t_k + \tau$ and it is given by (in terms of $z(t_k)$):

$$z(t_k + \tau) = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^{\tau_f} z(t_k). \tag{4.23}$$

The second part of the cycle corresponds to the open-loop mode when no update is transmitted for the time interval $t \in [t_k + \tau, t_{k+1})$; the response of the augmented system can be represented by:

$$z(t_k + \tau) = e^{\Lambda(t-(t_k+\tau))} z(t_k + \tau), \quad t \in \left[t_k + \tau, t_{k+1}\right). \tag{4.24}$$

At time t_{k+1} we begin a new cycle and the first update of that cycle takes place at that same instant. The response of the system, including the first update, in terms of $z(t_k)$ can be written as follows:

$$z(t_{k+1}) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^{\tau_f} z(t_k). \tag{4.25}$$

Due to the periodicity of the open loop and closed loop as defined in this section and the periodic fast updates during the closed-loop periods, the response of the augmented system to initial conditions $z(t_0) = \begin{bmatrix} x(t_0) \\ 0 \end{bmatrix} = z_0$ is given by:

$$z(t_k) = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^{t_f} \right)^{k} z_0.$$

The last expression provides the state of the system at time instants t_k, $k = 0,1,\ldots$ The response at any time $t \in (t_k, t_{k+1})$ can be obtained by pre-multiplying (4.20) by the corresponding partial response. ♦

A necessary and sufficient condition for stability of the networked system will now be presented. For the definition of stability please refer to Definition 2.2 in Sect. 2.2.

Theorem 4.4 *The system described by (4.1) with input (4.2) and with fast-slow intermittent update rates is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^{t_f}$ *are strictly inside the unit circle.*

The proof is similar to the one for Theorem 4.2 and it is omitted.

Example 4.1 Consider the following open-loop unstable continuous-time system:

$$A = \begin{bmatrix} -0.76 & 2.23 \\ 1.87 & -2.56 \end{bmatrix}, \quad B = \begin{bmatrix} 1.14 \\ 0 \end{bmatrix}.$$

Let the nominal model dynamics be given by:

$$\hat{A} = \begin{bmatrix} 1 & 2 \\ 2 & -3 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Let the control gain, selected based on the available model parameters, be:

$$K = [\, -5.4621 \quad -11.1658 \,].$$

We choose $\tau = 0.4$ s and $f = 0.1$ s. We want to find the range of values for h that result in stability. Using the results in Theorem 4.4 we can find that stability is obtained for choices of h (greater than 0.4 s) to about 1.83 s. Figure 4.4 shows the response of the system and model for $h = 1$ s.

Example 4.2 Theorem 4.4 offers necessary and sufficient conditions for stability in terms of different parameters including the parameters of the system which are unknown. In this example we use the same results to estimate the admissible uncertainties for given model parameters, i.e., those uncertainties that result in a stable model-based control system.

This example considers a first order system for simplicity but the same type of search can be followed for systems of higher order by searching over an increased number of scalar parameters corresponding to the elements of the real system matrices. Let the model parameters be given by: $\hat{A} = 1, \hat{B} = 1,$ and $K = -2.$

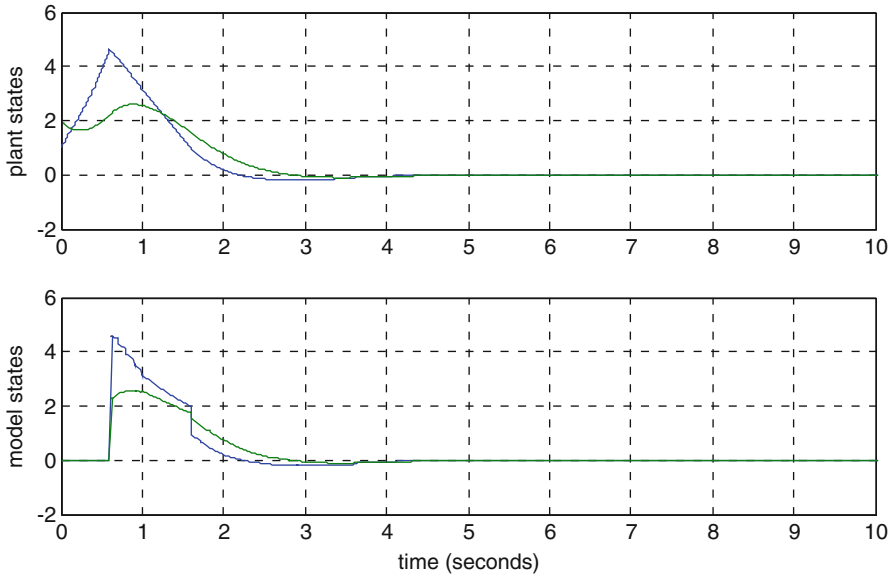

Fig. 4.4 Response of system and model for $h = 1$ s

Let us fix the network parameters: $h = 2$ s, $\tau = 0.5$ s, $f = 0.1$ s. Figure 4.5 shows the eigenvalue of the matrix $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda(h-\tau)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{f\Lambda} \right)^{t_f}$ with maximum magnitude for different values of A and B, the real parameters. The figure shows which combinations of plant parameters result in a stable control system for the given selection of model and network parameters. For instance, we can see that values close to the model parameters result in a stable model-based system, as expected.

4.2.2 Output Feedback

In the previous section we considered plants where the full vector of the state was available at the output. When the state is not directly measurable, one can implement a state observer as in Sect. 3.1. When the networked system operates in closed loop the control input is now based on the state of the observer, i.e., $u = K\overline{x}$. In the open-loop mode we use the state of the model as before, that is, $u = K\hat{x}$. In this section we extend our results to this situation. As in the architecture used in Sect. 3.1 for instantaneous model-based feedback, we assume that the state observer is collocated with the sensor. We use the plant model to design the state observer and the configuration is similar as the one shown in Fig. 4.1.

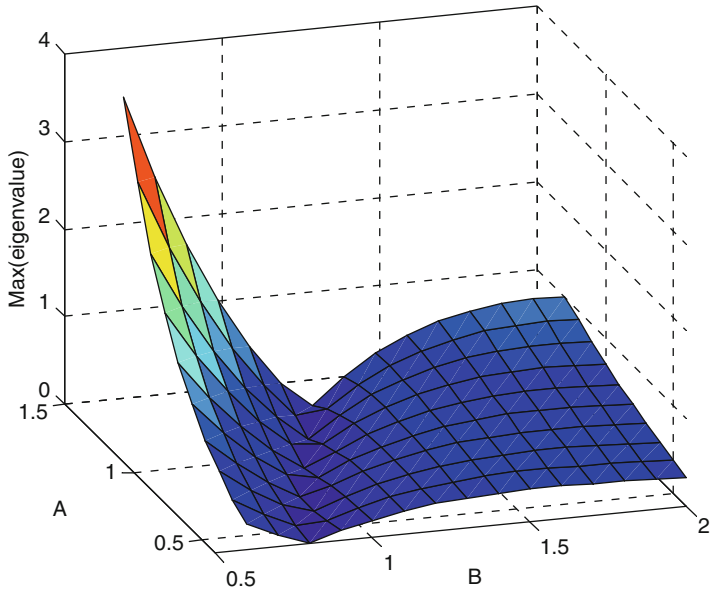

Fig. 4.5 Maximum eigenvalue of matrix in Theorem 4.4 for different values of A and B

We consider systems of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du. \end{aligned} \tag{4.26}$$

Let the nominal model corresponding to system (4.26) be given by:

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\,\hat{x} + \hat{B}\,u \\ \hat{y} &= \hat{C}\,\hat{x} + \hat{D}\,u. \end{aligned} \tag{4.27}$$

The approach we follow in this section is to implement a state observer at the sensor node. The observer is given by:

$$\dot{\overline{x}} = \left(\hat{A} - L\hat{C}\right)\overline{x} + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix}. \tag{4.28}$$

The parameters used by the observer are the available model parameters and it is assumed that the observer has continuous access to the output of the system since it is implemented at the sensor node. In order for the observer to obtain access to the system's input, a copy of the model is also implemented at the sensor node and the state of that model is updated at the same updated instants as the model in the controller node.

The state of the model is now updated using the observer state. Define the observer-model state error

$$\overline{e} = \overline{x} - \hat{x} \tag{4.29}$$

and the error matrices $\widetilde{A} = A - \hat{A}$, $\widetilde{B} = B - \hat{B}$, $\widetilde{C} = C - \hat{C}$, $\widetilde{D} = D - \hat{D}$.

Proposition 4.5 *The system described by (4.26)–(4.28) with control input* $u = K\hat{x}$ *and initial conditions* $z(t_0) = \begin{bmatrix} x(t_0) \\ \overline{x}(t_0) \\ \overline{e}(t_0) \end{bmatrix} = z_0$, $t_0 = 0$, *has the following response:*

$$z(t_k) = \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{f\Lambda_o} \right)^{t_f} \right)^{k} z_0 \tag{4.30}$$

where $\Lambda_o = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\widetilde{D}K & -\hat{B}K - L\widetilde{D}K \\ LC & L\widetilde{D}K - L\hat{C} & \hat{A} - L\widetilde{D}K \end{bmatrix}$ $z(t) = \begin{bmatrix} x(t) \\ \overline{x}(t) \\ \overline{e}(t) \end{bmatrix}$ *and* $h = t_{k+1} - t_k$. ♦

The proof follows the steps of Proposition 4.3 and considering the augmented system state.

Theorem 4.6 *The system described by (4.26)–(4.28) with control input* $u = K\hat{x}$ *and with fast-slow intermittent update rates is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ \overline{x} \\ \overline{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of*

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda_o(h-\tau)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{f\Lambda_o} \right)^{t_f} \tag{4.31}$$

are within the unit circle of the complex plane. ♦

The proof is similar to the one for Theorem 4.2 and it is omitted.

Example 4.2 Consider the inverted pendulum on a moving cart example described in Example 3.2 in Sect. 3.1. The nominal parameters, real parameters, controller gain, and observer gain are the same as in Example 3.2. Let us consider the case where $\tau = 0.3$ s and $f = 0.1$ s.

Figure 4.6 shows the eigenvalue with maximum magnitude of matrix (4.31) corresponding to this example. The networked system remains stable for values of h between 0.3 s and about 2.45 s. Figure 4.7 shows the states corresponding to the positions of the plant, the model, and the observer for the choice of $h = 1$ s and Fig. 4.8 shows the corresponding velocities of the same subsystems.

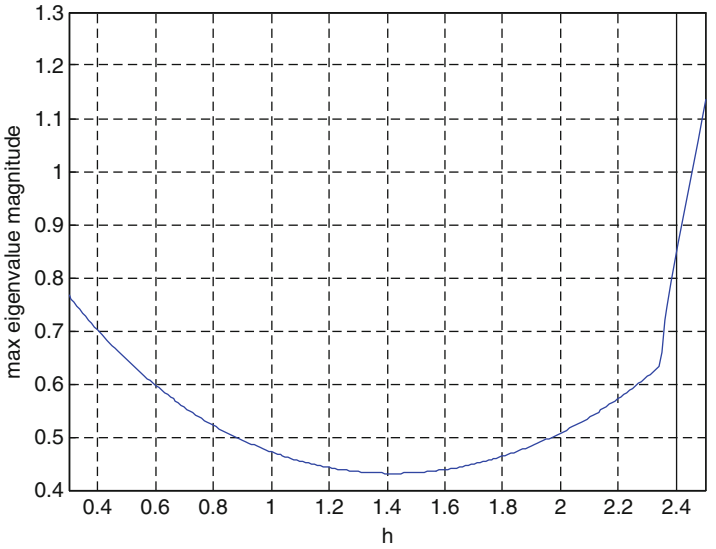

Fig. 4.6 Eigenvalue with maximum magnitude for Example 4.1

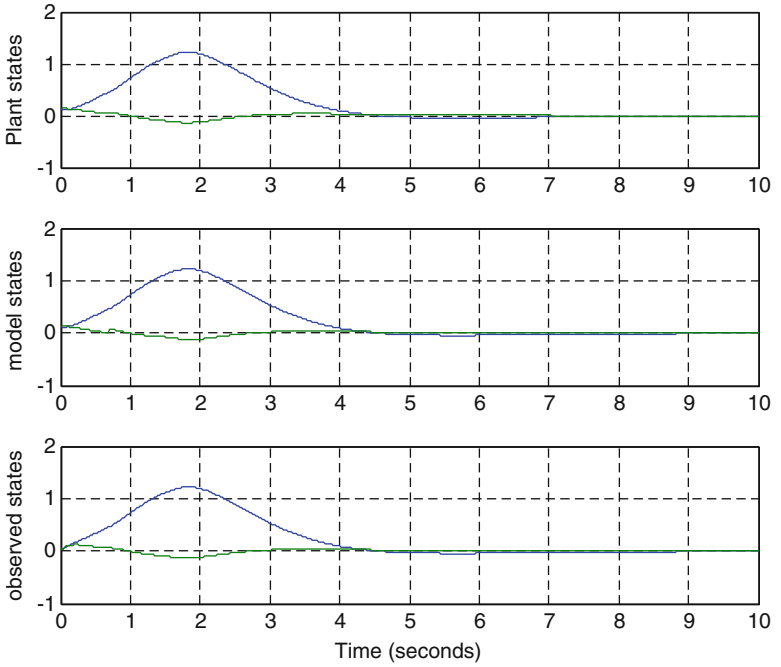

Fig. 4.7 Positions of inverted pendulum on a moving cart with intermittent feedback and $h = 1$ s

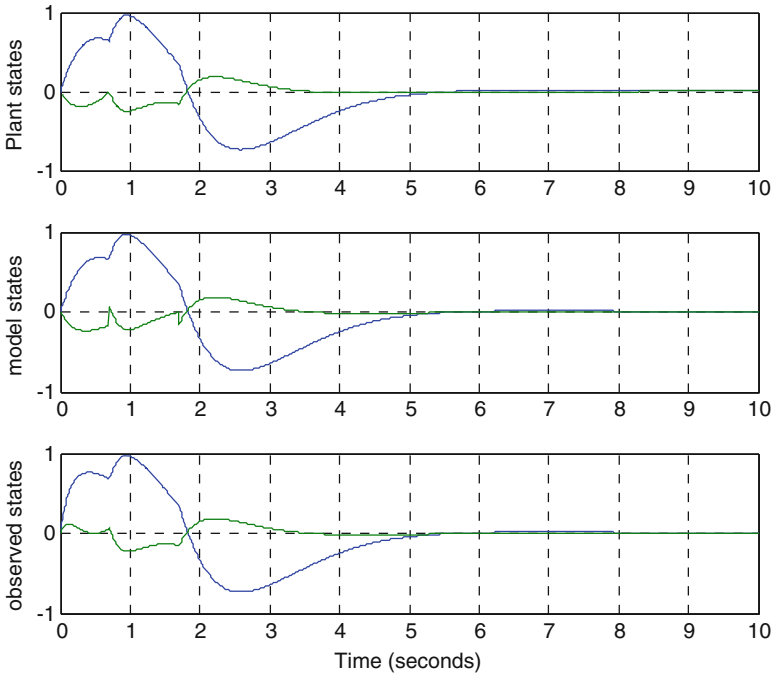

Fig. 4.8 Velocities of inverted pendulum on a moving cart with intermittent feedback and $h = 1$ s

4.3 Discrete-Time Systems with Intermittent Feedback

The architecture for discrete-time MB-NCS with intermittent feedback is essentially the same as that for continuous time, see Fig. 4.1. We make here similar assumptions to the ones in Sect. 2.3 for the instantaneous feedback case, where both the sensor and actuator sides are synchronized and updates occur at the same instants of time. The two key parameters in the implementation of MB-NCS with intermittent feedback h and τ, are now restricted to be integers, as they represent the number of ticks of the clock in the corresponding interval.

The treatment in this section is similar to Sect. 4.1 when we considered two modes of operation: open- and closed-loop modes. Although it is possible to consider fast and slow rates as in Sect. 4.2 the case of discrete-time systems allows, for reasonable sampling rates of the underlying plant, for the implementation of a closed-loop mode since for discrete-time systems only a finite number of measurements needs to be transmitted for the duration of the closed-loop cycle.

4.3.1 State Feedback

We consider a discrete-time system where the loop is closed periodically, every h ticks of the clock, and where each time the loop is closed, it remains closed for a time of τ ticks of the clock. The loop is closed at times n_k for $k = 1, 2, \ldots$. The system will be operating in closed-loop mode for the intervals $[n_k, n_k + \tau)$ and in open loop for the intervals $[n_k + \tau, n_{k+1})$ with $n_{k+1} - n_k = h$. When the loop is closed, the control decision is based directly on the information of the state of the plant, but we will keep track of the error nonetheless.

Let the discrete-time plant and its nominal model be given by:

$$\begin{aligned} \text{Plant} &: x(n+1) = Ax(n) + Bu(n) \\ \text{Model} &: \hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n). \end{aligned} \tag{4.32}$$

The control input can be represented by the following:

$$u(n) = \begin{cases} K\hat{x}(n) & \text{for } n \in [n_k + \tau, n_{k+1}) \\ Kx(n) & \text{for } n \in [n_k, n_k + \tau) \end{cases} \tag{4.33}$$

and the state error is given by:

$$e(n) = x(n) - \hat{x}(n). \tag{4.34}$$

We also define the augmented vector $z(n) = \begin{bmatrix} x(n) \\ e(n) \end{bmatrix}$.

Proposition 4.7 *The system described by (4.32) with input (4.33) and initial conditions* $z(n_0) = \begin{bmatrix} x(n_0) \\ 0 \end{bmatrix} = z_0$, *has the following response:*

$$z(n) = \begin{cases} \Lambda_{Dc}^{(n-n_k)} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0, & n \in [n_k, n_k + \tau) \\ \Lambda_{Do}^{(n-(n_k+\tau))} \Lambda_{Dc}^{\tau} \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0, & n \in [n_k + \tau, n_{k+1}) \end{cases} \tag{4.35}$$

where $\Sigma = \Lambda_{Do}^{(h-\tau)} \Lambda_{Dc}^{\tau}$, $\Lambda_{Do} = \begin{bmatrix} A + BK & -BK \\ \tilde{A} + \tilde{B}K & \hat{A} - \tilde{B}K \end{bmatrix}$, $\Lambda_{Dc} = \begin{bmatrix} A + BK & -BK \\ 0 & 0 \end{bmatrix}$ *and* $n_{k+1} - n_k = h$.

Proof We will derive the response of the augmented system. The approach is similar to the one used in the previous section for the continuous-time case. To this effect, let us separately investigate the partial response of the system under each mode, closed loop and open loop.

During the open-loop case, that is, for $n \in [n_k+\tau, n_{k+1})$, we have that $u(n) = K\hat{x}(n)$ so we have that

$$\begin{aligned} z(n+1) &= \begin{bmatrix} x(n+1) \\ e(n+1) \end{bmatrix} = \begin{bmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{bmatrix} \begin{bmatrix} x(n) \\ e(n) \end{bmatrix} \\ z(n_k+\tau) &= \begin{bmatrix} x(n_k+\tau) \\ e(n_k+\tau) \end{bmatrix} = \begin{bmatrix} x(n_k+\tau^-) \\ 0 \end{bmatrix} \end{aligned} \tag{4.36}$$

for $n \in [n_k+\tau, n_{k+1})$. Thus we have that

$$z(n+1) = \Lambda_{Do} z(n) \tag{4.37}$$

for $n \in [n_k+\tau, n_{k+1})$, where $\Lambda_{Do} = \begin{bmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{bmatrix}$.

The closed-loop case is a simplified version of the case above, as the difference resides in the fact that the error is always zero. Thus, for $n \in [n_k, n_k+\tau)$ we have

$$z(n+1) = \Lambda_{Dc} z(n), \tag{4.38}$$

for $n \in [n_k, n_k+\tau)$, where $\Lambda_{Dc} = \begin{bmatrix} A+BK & -BK \\ 0 & 0 \end{bmatrix}$.

Given an initial condition $z(0) = z_0$ and assuming without loss of generality that we start the cycle in closed-loop mode, then after a certain time $n \in [0, \tau)$ the solution of the trajectory of the vector is given by

$$z(n) = \Lambda_{Dc}^n z_0 \tag{4.39}$$

for $n \in [0, \tau)$. In particular, at time τ, $z(\tau) = \Lambda_{Dc}^\tau z_0$.

Once the loop is opened, the open-loop mode takes over, so that

$$z(n) = \Lambda_{Do}^{(n-\tau)} z(\tau) = \Lambda_{Do}^{(n-\tau)} \Lambda_{Dc}^\tau z_0, \tag{4.40}$$

for $n \in [\tau, n_1)$. In particular, when the time comes to close the loop again, that is, after time h; then $z(n_1) = \Lambda_{Do}^{(h-\tau)} \Lambda_{Dc}^\tau z_0$.

Notice, however, that at this instant the error is reset to zero. We can represent this by pre-multiplying by $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ before we analyze the closed-loop trajectory for the next cycle. Because we wish to always start with an error that is set to zero, we should actually multiply by $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ at the beginning of the cycle as well.

So, after k cycles, the response

$$z(n_k) = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Lambda_{Do}^{(h-\tau)} \Lambda_{Dc}^{\tau} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right)^k z_0 \quad (4.41)$$

is obtained, where $\Sigma = \Lambda_{Do}^{(h-\tau)} \Lambda_{Dc}^{\tau}$.

The final step is to consider the last (partial) cycle that the system goes through. If the system is in closed loop, that is, $n \in [n_k, n_k + \tau)$ then the solution can be achieved merely by pre-multiplying $z(n_k)$ by $\Lambda_{Dc}^{(n-n_k)}$. In the case of the system being in open loop, that is, $n \in [n_k + \tau, n_{k+1})$ then clearly we must pre-multiply by $\Lambda_{Do}^{(n-(n_k+\tau))} \Lambda_{Dc}^{\tau}$. ♦

A necessary and sufficient condition for stability of the networked system will now be presented. For the definition of stability please refer to Definition 2.5 in Sect. 2.3.

Theorem 4.8 *The system described by (4.32) with input (4.33) is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *if and only if the eigenvalues of* $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$ *are strictly inside the unit circle, where* $\Sigma = \Lambda_{Do}^{(h-\tau)} \Lambda_{Dc}^{\tau}$.

The proof for Theorem 4.8 follows directly from the results provided in the previous section on intermittent feedback for continuous-time systems and the results in Sect. 2.3, therefore, the proof has been omitted.

4.3.2 Discrete-Time Systems with Output Feedback

A similar analysis to the one in the previous section can be performed to obtain the response and stability conditions for discrete-time LTI systems using state observer and intermittent feedback.

Let us consider the following equations:

$$\begin{aligned}
&\text{Plant}: \quad x(n+1) = Ax(n) + Bu(n), y(n) = Cx(n) + Du(n) \\
&\text{Model}: \quad \hat{x}(n+1) = \hat{A}\hat{x}(n) + \hat{B}u(n), \hat{y}(n) = \hat{C}\hat{x}(n) + \hat{D}u(n) \\
&\text{Controller}: \quad u(n) = \begin{cases} K\hat{x}(n) & \text{for } n \in [n_k + \tau, n_{k+1}) \\ K\bar{x}(n) & \text{for } n \in [n_k, n_k + \tau) \end{cases} \\
&\text{Observer}: \quad \bar{x}(n+1) = (\hat{A} - L\hat{C})\bar{x}(n) + \begin{bmatrix} \hat{B} - L\hat{D} & L \end{bmatrix} \begin{bmatrix} u(n) \\ y(n) \end{bmatrix}.
\end{aligned} \quad (4.42)$$

We also have that $n_{k+1} - n_k = h$, $\tau < h$ and both h and τ are integers.

Define the augmented state vector as follows: $z = \begin{bmatrix} x \\ \overline{x} \\ e \end{bmatrix}$. The following proposition specifies the state response of the system.

Proposition 4.9 *The system described in (4.42) with initial conditions* $z(n_0) = \begin{bmatrix} x(n_0) \\ \overline{x}(n_0) \\ 0 \end{bmatrix} = z_0$, *has the following response*

$$z(n) = \begin{cases} \Lambda_{Fc}^{(n-n_k)} \Sigma^k z_0, & n \in [n_k, n_k + \tau) \\ \Lambda_{Fo}^{(n-(n_k+\tau))} \Lambda_{Fc}^{\tau} \Sigma^k z_0, & n \in [n_k + \tau, n_{k+1}) \end{cases} \tag{4.43}$$

where $\Sigma = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda_{Fo}^{(h-\tau)} \Lambda_{Fc}^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$,

$$\Lambda_{Fo} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\widetilde{D}K & -\hat{B}K - L\widetilde{D}K \\ LC & L\widetilde{D}K - L\hat{C} & \hat{A} - L\widetilde{D}K \end{bmatrix},$$

$$\Lambda_{Fc} = \begin{bmatrix} A & BK & -BK \\ LC & \hat{A} - L\hat{C} + \hat{B}K + L\widetilde{D}K & -\hat{B}K - L\widetilde{D}K \\ 0 & 0 & 0 \end{bmatrix}.$$

Theorem 4.10 *The system described by (4.42) is globally exponentially stable around the solution* $z = \begin{bmatrix} x \\ \overline{x} \\ e \end{bmatrix} = 0$ *if and only if the eigenvalues of*

$$\Sigma = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \Lambda_{Fo}^{(h-\tau)} \Lambda_{Fc}^{\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

are strictly inside the unit circle, where $\Lambda_{Fo}, \Lambda_{Fc}$ *are as described above.*

Proofs for the last proposition and theorem are similar to their continuous-time counterparts.

4.4 Alternative Conditions for Stability of MB-NCS with Intermittent Feedback

In this section we extend the analysis using the Lifting procedure that was shown in Sect. 2.4.3. In the case of intermittent feedback the input u of the MB-NCS is defined as

$$u(n) = \begin{cases} Kx(n) & h_l \le n \le h_l + \tau_l \\ K\hat{x}(n) & h_l + \tau_l < n < h_{l+1} \end{cases} \tag{4.44}$$

In this section we consider intermittent feedback with constant updates and constant closed-loop times, that is, $h_{l+1} - h_l = h$, which represents how often we close the loop between the sensor and the controller, and $\tau_l = \tau < h$, which represents the constant number of clock ticks that the loop remains closed; h and τ are positive integer numbers.

Theorem 4.11 *The discrete-time MB-NCS with intermittent feedback (4.32) and control input (4.44) is asymptotically stable if only if the eigenvalues of:*

$$\left(A^{h-\tau} + \sum_{j=0}^{h-\tau-1} A^{h-\tau-1-j} BK \left(\hat{A} + \hat{B} K \right)^j \right) (A + BK)^{\tau} \tag{4.45}$$

lie inside the unit circle.

Proof We rewrite the state equation of the lifted LTI system in terms of only the state. To do this, we consider closed-loop and open-loop behaviors. For the interval $h \le n \le h + \tau$ the response of the system at time $nh + \tau$ is given by (closed-loop response):

$$x(nh + \tau) = (A + BK)^{\tau} x(nh). \tag{4.46}$$

The response of the system at time $nh + h$ in terms of the state at time $nh + \tau$ can be shown by following the procedure in Theorem 2.8 in Sect. 2.4.3,

$$x(nh + h) = \left(A^{h-\tau} + \sum_{j=0}^{h-\tau-1} A^{h-\tau-1-j} BK \left(\hat{A} + \hat{B} K \right)^j \right) x(nh + \tau). \tag{4.47}$$

Then, the response of the system at time $(n + 1)h$ in terms of only the state at time nh is given by:

$$x((n + 1)h) = \left(A^{h-\tau} + \sum_{j=0}^{h-\tau-1} A^{h-\tau-1-j} BK \left(\hat{A} + \hat{B} K \right)^j \right) (A + BK)^{\tau} x(nh). \tag{4.48}$$

The resulting system is a LTI system and it is stable when the eigenvalues of (4.45) lie inside the unit circle. ♦

4.4.1 Relation to Previous Results

For the intermittent case, in Sect. 4.3 a necessary and sufficient condition for stability of a discrete-time system is that the eigenvalues of

$$M_{DI}(h) = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \Lambda_{Do}^{(h-\tau)} \Lambda_{Dc}^{\tau} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \tag{4.49}$$

must lie inside the unit circle, where $\Lambda_{Do} = \begin{pmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{pmatrix}$, $\Lambda_{Dc} = \begin{pmatrix} A+BK & -BK \\ 0 & 0 \end{pmatrix}$. We consider the similarity transformation:

$$\overline{\Lambda}_{Do} = P\Lambda_{Do}P^{-1} = \begin{pmatrix} A & BK \\ 0 & \hat{A}+\hat{B}K \end{pmatrix}. \tag{4.50}$$

We assume a fixed value for the closed-loop interval τ, and find the Z-transform of $M_{DI}(n)$, $n = 0, 1, 2, \ldots$, namely

$$Z\{M_{DI}(n)\} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} Z\{\Lambda_{Do}^{n}\} \Lambda_{Dc}^{n} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.51}$$

Notice that:

$$\Lambda_{Dc}^{\tau} = \begin{pmatrix} (A+BK)^{\tau} & -(A+BK)^{\tau-1}BK \\ 0 & 0 \end{pmatrix}. \tag{4.52}$$

Then we obtain the following:

$$\begin{aligned} Z\{M_D(n)\} &= \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P^{-1}\left(zI - \overline{\Lambda}_{Do}\right)^{-1} P \begin{pmatrix} (A+BK)^{\tau} & 0 \\ 0 & 0 \end{pmatrix} z \\ &= \begin{pmatrix} \left((zI-A)^{-1} + (zI-A)^{-1}BK\left(zI - \left(\hat{A}+\hat{B}K\right)\right)^{-1}\right)(A+BK)^{\tau} & 0 \\ 0 & 0 \end{pmatrix} z. \end{aligned} \tag{4.53}$$

We obtain the inverse Z-transform of the upper left sub-matrix which contains the eigenvalues of interest:

$$\begin{aligned} &Z^{-1}\left\{\left((zI-A)^{-1}z + (zI-A)^{-1}BK\left(zI - \left(\hat{A}+\hat{B}K\right)\right)^{-1}z\right)(A+BK)^{\tau}\right\} \\ &= \left(A^{n} + \sum_{j=0}^{n-1} A^{n-1-j}BK\left(\hat{A}+\hat{B}K\right)^{j}\right)(A+BK)^{\tau}. \end{aligned} \tag{4.54}$$

Substitute $n = h\text{-}\tau$ in the above equation to obtain:

$$\left(A^{h-\tau} + \sum_{j=0}^{h-\tau-1} A^{h-\tau-1-j}BK\left(\hat{A}+\hat{B}K\right)^{j}\right)(A+BK)^{\tau}. \tag{4.55}$$

Equation (4.55) is exactly the same condition as (4.45) obtained in Theorem 4.11, i.e., the nontrivial eigenvalues (those different than zero) of $M_{DI}(h)$ correspond to the eigenvalues of (4.45).

4.5 Notes and References

The work presented in Sect. 4.1 and Sect. 4.3 appeared first in [68–70]. Results obtained in Sect. 4.2 were first published in [91]. The first part of Sect. 4.4 was first published in [82] and Sect. 4.4.1 represents complementary results.

The notion of intermittent feedback in MB-NCS can be extended to consider more complex scenarios such as network induced delays, plants with nonlinear dynamics, and time-varying update intervals. The case of network delays was considered in [67]. Nonlinear systems were addressed in [70]. Linear systems with time-varying feedback update intervals were investigated in [70] as well.

The concept of intermittent feedback has been applied in other areas of study. For example, in applications to chemical engineering processes, in intermittent feedback is rather prominent. See, for example [133], where intermittent feedback is used to address turbulence.

Oldroyd [206] addresses the issue of "intermittent distillation," using intermittent feedback to address product removal. Most of the articles in this area are very application-oriented and focus on processes such as manufacturing.

In the field of psychology, the use of intermittent feedback is widespread. The corresponding term often used in psychology papers is "intermittent reinforcement" and often arises in the literature regarding education, learning, and child-rearing. The main idea is that human behavior, in itself, follows this intermittent nature. This does not just apply to physical processes such as motor control, but to psychological pulls to practices, such as work and gambling [235]. The learning process is another area where intermittent feedback arises very naturally, and where methods based on it have proven to be very effective [38]. Intermittent feedback is also used in regards to motor control, such as controlling seizures, epilepsy, etc. [218, 223]. The main idea in terms of psychological aspects of motion control is that while, initially, continuous control is applied, as the human being learns, and there is growth in cognitive and associative skills, there is a shift to intermittent feedback and a more automatic nature of motion control.

Intermittent feedback has also been used to some extent in robotics and mechanical engineering. This makes sense due to the fact that the visual component of robots is often designed to follow a biologically inspired analogous process. Thus, intermittent feedback arises naturally. For example, in [98, 219, 220] the authors use intermittent feedback to address a conceptual, and practical difficulty in robotics: by replacing the continuously moving horizon by an intermittently moving horizon, they solve a continuous-time generalized predictive controller. The work in [134]

also address intermittent visual feedback in robotics and study how to compensate for the effects of delays, while Leonard and Krishnaprasad [144] use intermittent feedback in dealing with motion control of robots, leading to nonlinear control with fewer state variables. Also, because the concept of "learning" in robots is closely tied in with the learning process in human beings, the application of intermittent feedback here makes sense as well, as has been dealt with in [171, 221].

Chapter 5
Time-Varying and Stochastic Feedback Updates

In this chapter we are interested in the stochastic stability properties of MB-NCS when the update time intervals h are time-varying. The statistical properties of h may or may not be known. We will concentrate primarily on the stochastic stability of the networked system when the packet exchange time intervals are time-varying and have some known probability distribution. Conditions are derived for almost sure and mean square stability for independent, identically distributed exchange or update time intervals and for Markov chain-driven update time intervals. However, we will also treat the case of time-varying h when additional properties are available.

We focus our analysis on the Model-Based Networked Control System that was introduced in Chap. 2, specifically the case where the plant is continuous and the states are available (full state feedback). Other MB-NCS (e.g., output feedback, discrete plants) can be analyzed in a similar fashion.

The state feedback model-based networked control system architecture was shown in Fig. 2.1. This control architecture has as main objective the reduction of the data packets transmitted over the network by a networked control system.

The packets transmitted by the sensor contain the measured value of the plant state and are used to update the plant model on the actuator/controller node. These packets are transmitted at times t_k. We define the update time intervals $h(k)$ as the time intervals between transmissions or model updates: $h(k) = t_{k+1}\text{-}t_k$. We have assumed in previous chapters that the update time intervals $h(k)$ are constant. This might not always be the case in many applications. The transmission times of data packets from the plant output to the controller/actuator might not be completely periodic due to network contention and the usual non-deterministic nature of the transmitter task execution scheduler. Soft real time constraints provide a way to enforce the execution of tasks in the transmitter microprocessor. This allows the task of periodically transmitting the plant information to the controller/actuator to be executed at times t_k that can vary according to certain probability distribution function. This translates into an update time interval $h(k)$ that can acquire a certain value according to a probability distribution function. The system designer

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_5,

can determine this probability distribution function by analyzing the transmitter microprocessor scheduler structure, the timing of the possible tasks that can be posted to this scheduler, the network contention scheme, and the network behavior.

We will present different stability criteria that can be applied when the update time intervals $h(k)$ vary with time. The first case is when the probability distribution of h is unknown. The only information available about the update time intervals $h(k)$ will be its maximum and minimum values, and the actual update time intervals observed may jitter on the range $[h_{min}, h_{max}]$. This criterion may be used to provide a first result on the stability properties of a system perhaps for comparison purposes. We will try to calculate the values for h_{min} and h_{max}. This is the strongest and most conservative stability criterion. This stability type is based on the classical Lyapunov function approach and is presented in Sect. 5.1.

Next, two types of stochastic stability are discussed, namely Almost Sure or Probability-1 Asymptotic Stability in Sect. 5.2 and Mean Square or Quadratic Asymptotic Stability in Sect. 5.3. The first one is the one that mostly resembles deterministic stability [136]. Mean square stability is attractive since it is related to optimal control problems such as the LQR.

Two different types of time-varying transmission times are considered for each case. The first assumes that the transmission times are independent identically distributed with probability distribution function that may have support for infinite update time intervals. The second type of stochastic update time interval assumes that the transmission times are driven by a finite Markov chain. Both models are common ways of representing the behavior of network transmission and scheduler execution times.

5.1 Lyapunov Analysis of MB-NCS

The dynamics of the system shown in Fig. 2.1 are given by:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix}; \begin{bmatrix} x(t_k) \\ e(t_k) \end{bmatrix} = \begin{bmatrix} x(t_k^-) \\ 0 \end{bmatrix}, \quad \forall t \in [t_k, t_{k+1}), \quad with \quad t_{k+1} - t_k = h(k) \tag{5.1}$$

where $e(t) = x(t) - \hat{x}(t)$ represents the state error between the plant state and the plant model state, A and B are the matrices of the actual plant state-space representation, $\hat{A}$ and $\hat{B}$ are the matrices of the model state-space representation, while $\widetilde{A} = A - \hat{A}$ and $\widetilde{B} = B - \hat{B}$ represent the modeling error matrices. Define $z(t) = [\, x(t) \quad e(t) \,]^T$, and $\Lambda = \begin{bmatrix} A+BK & -BK \\ \widetilde{A}+\widetilde{B}K & \hat{A}-\widetilde{B}K \end{bmatrix}$ so that (5.1) can be rewritten as $\dot{z} = \Lambda z$ for $t \in [t_k, t_{k+1})$.

Proposition 5.1 *The system described by (5.1) with initial conditions* $z(t_0) = [x(t_0) \quad 0]^T = z_0$, *has the following response*:

$$z(t) = e^{\Lambda(t-t_k)} \left(\prod_{j=1}^{k} M(j) \right) z_0 \tag{5.2}$$

for $t \in [t_k, t_{k+1}), t_{k+1} - t_k = h(k)$, *where* $M(j) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(j)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$.

Proof The proof is similar to the corresponding proof for constant h in Chap. 2 and it is included here for completeness. On the interval $t \in [t_k, t_{k+1})$, the system response is:

$$z(t) = \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = e^{\Lambda(t-t_k)} \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix} = e^{\Lambda(t-t_k)} z(t_k). \tag{5.3}$$

Now, note that at times t_k, $z(t_k) = \begin{bmatrix} x(t_k) \\ 0 \end{bmatrix}$, that is, the error $e(t)$ is reset to 0. We can represent this by:

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z\left(t_k^-\right). \tag{5.4}$$

Using (5.3) we can calculate $z(t_k^-)$:

$$z(t_k) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(k)} z(t_{k-1}). \tag{5.5}$$

In view of (5.3) we have that if at time $t = t_0$, $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ is the initial condition then

$$\begin{aligned} z(t) &= e^{\Lambda(t-t_k)} z\left(t_k\right) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(k)} z\left(t_{k-1}\right) \\ &= e^{\Lambda(t-t_k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(k)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(k-1)} z\left(t_{k-2}\right) \\ &= e^{\Lambda(t-t_k)} \left(\prod_{j=1}^{k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(j)} \right) z_0. \end{aligned} \tag{5.6}$$

Now we know that $\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(j)}$ is of the form $\begin{bmatrix} S_j & P_j \\ 0 & 0 \end{bmatrix}$ and so $\left(\prod_{j=1}^{k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(j)} \right)$ has the form $\begin{bmatrix} \left(\prod_{j=1}^{k} S_j \right) & \cdot \\ 0 & 0 \end{bmatrix}$.

Additionally we note the special form of the initial condition $z(t_0) = z_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$ so that

$$\begin{aligned} \left(\prod_{j=1}^{k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(j)} \right) \begin{bmatrix} x_0 \\ 0 \end{bmatrix} &= \begin{bmatrix} \left(\prod_{j=1}^{k} S_j \right) x_0 & 0 \\ 0 & 0 \end{bmatrix} \\ &= \left(\prod_{j=1}^{k} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h(j)} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x_0 \\ 0 \end{bmatrix}. \end{aligned} \tag{5.7}$$

In view of (5.7) it is clear that we can represent the system response as in (5.2). ♦

Note that in (5.2) the response is given by a product of matrices that share similar structure but are in general different. All $M(j)$ are the same in the case where the update time intervals $h(k)$ are equal. The special case when h is constant was studied in Chap. 2 where necessary and sufficient conditions for stability were derived in terms of the eigenvalues of a matrix. Note that in the present case of varying $h(k)$, stability cannot be guaranteed even if all matrices in the product have their eigenvalues inside the unit circle.

5.1.1 Lyapunov Stability

Let the update time interval be $h(k)$, which is time-varying. The stability criterion derived in this section is the strongest and most conservative stability criterion. It is based on the well-known Lyapunov second method for determining the stability of a system. We will assume that the properties of $h(k)$ are unknown but $h(k)$ is contained within some interval. This stability condition is not stochastic but provides a first approach to stability for time-varying transmission times NCS.

Definition 5.2 The equilibrium $z = 0$ of a system described by $\dot{z} = f(t, z)$ with initial condition $z(t_0) = z_0$ is Lyapunov asymptotically stable at large (or globally) if for any $\varepsilon > 0$ there exists $\beta > 0$ such that the solution of $\dot{z} = f(t, z)$ satisfies

$$\|z(t, z_0, t_0)\| < \varepsilon, \forall t > t_0 \text{ and } \lim_{t \to \infty} \|z(t, z_0, t_0)\| = 0 \tag{5.8}$$

whenever $\|z_0\| < \beta$.

Theorem 5.3 *The system described by (5.1) is Lyapunov Asymptotically Stable for* $h \in [h_{\min},\ h_{\max}]$ *if there exists a symmetric positive definite matrix X such that* $Q = X - MXM^T$ *is positive definite for all* $h \in [h_{\min},\ h_{\max}]$, *where*

$$M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.9}$$

Proof Note that the output norm can be bounded by

$$\begin{aligned} \left\| e^{\Lambda(t-t_k)} \left(\prod_{j=1}^{k} M(j) \right) z_0 \right\| &\leq \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \prod_{j=1}^{k} M(j) \right\| \cdot \|z_0\| \\ &\leq e^{\bar{\sigma}(\Lambda) h_{\max}} \cdot \left\| \prod_{j=1}^{k} M(j) \right\| \cdot \|z_0\|. \end{aligned} \tag{5.10}$$

That is, since $e^{\Lambda(t-t_k)}$ has finite growth rate and will grow until h_{max} at most. Then convergence of the product of matrices $M(j)$ to 0 ensures the stability of the system. Such convergence to 0 is guaranteed by the existence of a symmetric positive definite matrix X in the Lyapunov equation. ♦

Theorem 5.3 may be used to derive an interval $[h_{\min},\ h_{\max}]$ for h for which stability is guaranteed. It is clear that the range for h, that is the interval $[h_{\min},\ h_{\max}]$, will vary with the choice of X. Another observation is that the interval obtained this way will always be contained in the set of constant update time intervals for which the system is stable (as derived using Theorem 2.3). That is, a constant update time h interval contained in $[h_{\min},\ h_{\max}]$ will always be a stable constant update time interval.

Several ways of obtaining the values for $h_{\min}$ and $h_{\max}$ can be used. One is to first fix the value of Q, obtain the solution X of the Lyapunov equation in Theorem 5.3 for a value of h known to be stable. Then, using this value of X, the expression $X - MXM^T$ can be evaluated for positive definiteness. This can be repeated for all the values of h known to stabilize the system to obtain the widest interval $[h_{\min},\ h_{\max}]$.

Example 5.1 We use an unstable plant with dynamics given by

$$A = \begin{bmatrix} 0.22 & 1.31 \\ -0.07 & 0.16 \end{bmatrix}, \quad B = \begin{bmatrix} 0.04 \\ 0.87 \end{bmatrix}.$$

Let our plant model be:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u, \ \ \hat{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

which behaves as a double integrator. Let our feedback law be given by $u = K\hat{x}$ with $K = [\,-0.776 \quad -1.765\,]$. When the update time intervals are constant, stability can

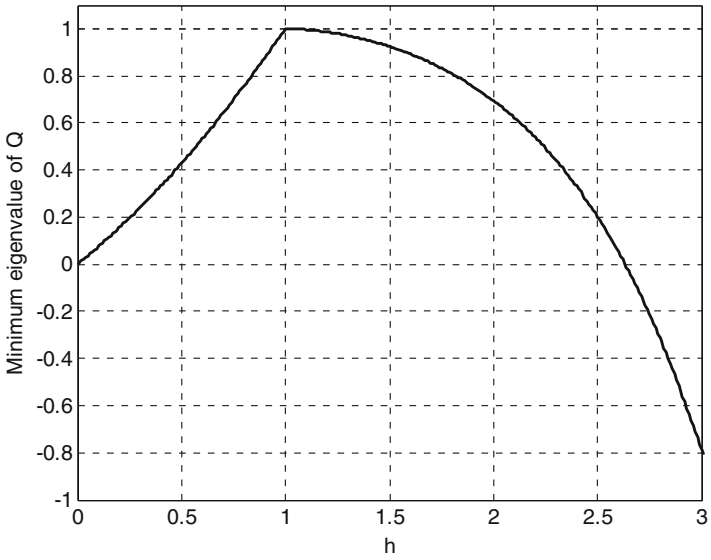

Fig. 5.1 Minimum eigenvalue of Q as a function of the update time interval h

be guaranteed if and only if the eigenvalues of the test matrix M in (5.9) are inside of the unit circle. Using this condition, it is possible to verify that, for stability, the constant update time intervals can take values between 0 and 4.97 s. We now use Theorem 5.3 to find an interval of time-varying update time intervals for which the NCS will remain stable. Note that the update time intervals can vary along this interval. We will set a nominal update time interval of 1 s and set Q to be the identity matrix; then by solving the Lyapunov equation given in Theorem 5.3 we can find a positive definite nominal X. We then obtain the stability interval by searching for update time intervals around 1 s, for which the Q in the Lyapunov equation obtained with the nominal X is positive definite.

Figure 5.1 shows the plot of the minimum eigenvalue of Q as a function of the update time intervals, from which it can be seen that the stability interval is now $[h_{\min},\ h_{\max}] = [0, 2.63]$. Note that for constant h the system remains stable for any $h < 4.97$. Then, it is possible that by using a different nominal update time interval and different nominal Q's less conservative results may be obtained.

5.2 Almost Sure Asymptotic Stability

We will use the definition of almost sure asymptotic stability [136] that provides a stability criterion based on the sample path. This stability definition resembles more the deterministic stability definition [170], and is of practical importance.

Since the stability condition has been relaxed, we expect to see less conservative results than those obtained using the Lyapunov stability considered in the previous section. We now define Almost Sure or Probability-1 Asymptotic stability.

Definition 5.4 The equilibrium $z=0$ of a system described by $\dot{z}=f(t,z)$ with initial condition $z(t_0)=z_0$ is almost sure (or with probability-1) asymptotically stable at large (or globally) if for any $\beta>0$ and $\varepsilon>0$ the solution of $\dot{z}=f(t,z)$ satisfies

$$\lim_{\delta\to\infty} P\left\{\sup_{t\geq\delta}\|z(t,z_0,t_0)\|>\varepsilon\right\}=0 \text{ whenever } \|z_0\|<\beta. \tag{5.11}$$

This definition is similar to the one presented for deterministic systems in Definition 5.2. We will examine the conditions under which the full state feedback continuous networked system is Almost Sure stable for two cases: The case when the update time intervals are independent identically distributed and the case when the update time intervals are driven by a Markov chain.

5.2.1 MB-NCS with Independent Identically Distributed Transmission Times

Here we will assume that the update time intervals $h(k)$ are independent identically distributed (i.i.d.) with probability distribution function $F(h)$. We now present the conditions under which the system described by (5.1) with i.i.d. update time intervals is asymptotically stable with probability-1. We will use a technique similar to lifting [42] to obtain a discrete linear time invariant representation of the system. Lifting techniques were also used in Chaps. 2 and 4 for similar purposes. It can be observed that the system will be described by:

$$\xi_{k+1}=\Omega_k\xi_k, \text{with } \xi_k\in L_{2e} \text{ and } \xi_k=z(t+t_k), t\in\left[0,h_k\right). \tag{5.12}$$

Here L_{2e} stands for the extended L_2. It is easy to obtain from (5.1) and (5.12) the operator Ω_k as:

$$(\Omega_k\nu)(t)=e^{\Lambda t}\begin{bmatrix} I & 0\\ 0 & 0\end{bmatrix}\int_0^{h(k)}\delta(\tau-h(k))\nu(\tau)d\tau. \tag{5.13}$$

Now we can restate the definition on almost sure stability or probability-1 stability given in Definition 5.4 to better fit the equivalent system representation (5.12).

Definition 5.5 The system represented by (5.1) is almost sure stable or stable with probability-1 if for any $\beta>0$ and $\varepsilon>0$ if the solution of $\xi_{k+1}=\Omega_k\xi_k$ satisfies:

$$\lim_{\widetilde{\delta}\to\infty} P\left\{\sup_{k\geq\widetilde{\delta}} \|\xi_k(t_0,z_0)\|_{2,[0,t_k]} > \varepsilon\right\} = 0 \tag{5.14}$$

whenever $\|z_0\| < \beta$. The norm $\|\bullet\|_{2,[0,h(k)]}$ is given by

$$\|\xi_k\|_{2,[0,h(k)]} = \left(\int_0^{h(k)} \|\xi_k(\tau)\|^2 d\tau\right)^{1/2}. \tag{5.15}$$

This definition allows us to study almost sure stability of systems such as (5.12) when the probability distribution function for update time intervals $h(k)$ has infinite support. Based on this definition the following result can now be shown.

Theorem 5.6 *The system described by (5.1), with update time intervals $h(j)$ independent identically distributed random variable with probability distribution $F(h)$ is globally almost sure (or with probability-1) asymptotically stable around the solution $z = \begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ if*

$$N = E\left[\left(e^{2\overline{\sigma}(\Lambda)h} - 1\right)^{1/2}\right] < \infty \tag{5.16}$$

and the expected value of the maximum singular value of the test matrix M,

$$E[\|M\|] = E[\overline{\sigma}_M] < 1, \tag{5.17}$$

where $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$.

Proof Assume that the supremum of the norm bracketed is achieved at $k^* \geq \widetilde{\delta}$, that is $\sup_{k>\widetilde{\delta}} \|\xi_k\| = \|\xi_{k^*}\|$. So now we can use the Chebyshev bound for positive random variables [189] to bound the probability in our definition:

$$P\left\{\sup_{k\geq\widetilde{\delta}} \|\xi_k\| > \varepsilon\right\} = P\{\|\xi_{k^*}\| > \varepsilon\} \leq \frac{E[\|\xi_{k^*}\|]}{\varepsilon}. \tag{5.18}$$

Using (5.2) and basic norm properties, we proceed to bound the expectation on the right hand side

$$
\begin{aligned}
&E\left[\left(\int_0^{h(k^*)} \|\xi_{k^*}(\tau)\|^2 d\tau\right)^{1/2}\right] \\
&\leq E\left[\left(\int_0^{h(k^*)} \left\|e^{\Lambda\tau}\right\|^2 \left\|\prod_{j=1}^{k^*-1} M(j)\right\|^2 \|z_0\|^2 d\tau\right)^{1/2}\right] \\
&\leq E\left[\left(\int_0^{h(k^*)} \left(e^{\overline{\sigma}(\Lambda)\tau}\right)^2 d\tau\right)^{1/2} \left\|\prod_{j=1}^{k^*-1} M(j)\right\| \|z_0\|\right] \\
&= E\left[\left(\int_0^{h(k^*)} \left(e^{\overline{\sigma}(\Lambda)\tau}\right)^2 d\tau\right)^{1/2}\right] E\left[\left\|\prod_{j=1}^{k^*-1} M(j)\right\|\right] \|z_0\|.
\end{aligned}
\tag{5.19}
$$

The last equation follows from the independence of the update time intervals $h(j)$. Analyzing the first term on the last equality we see that is bounded for the trivial case where $\Lambda = 0$. When $\Lambda \neq 0$, the integral can be solved, and it can be shown to be equal to $\sqrt{\frac{1}{2\overline{\sigma}(\Lambda)}} E\left[\left(e^{2\overline{\sigma}(\Lambda)h(k^*)} - 1\right)^{1/2}\right]$ which by assumption is bounded. The second term can also be bounded by using the independency property of the update time intervals $h(j)$.

$$
E\left[\left\|\prod_{j=1}^{k^*-1} M(j)\right\|\right] \leq E\left[\prod_{j=1}^{k^*-1} \|M(j)\|\right] = (E[\|M\|])^{k^*-1}. \tag{5.20}
$$

We can now evaluate the limit over the obtained expression.

$$
\begin{aligned}
\lim_{\widetilde{\delta}\to\infty} P\left\{\sup_{k\geq\widetilde{\delta}} \|\xi_k\| > \varepsilon\right\} &\leq \lim_{\widetilde{\delta}\to\infty} \frac{E[\|\xi_{k^*}\|]}{\varepsilon} \\
&\leq \lim_{\widetilde{\delta}\to\infty} \frac{\sqrt{\frac{1}{2\overline{\sigma}(\Lambda)}} E\left[\left(e^{2\overline{\sigma}(\Lambda)h(k^*)} - 1\right)^{1/2}\right] (E[\|M\|])^{k^*-1} \|z_0\|}{\varepsilon}.
\end{aligned}
\tag{5.21}
$$

It is obvious that the right hand side of the expression will be identically 0 (note that $k* \geq \widetilde{\delta}$) if the averaged maximum singular value $E[\overline{\sigma}_M] = E[\|M\|] < 1$. ♦

Note that the condition may give conservative results if applied directly over the test matrix. To avoid this problem and make the condition tighter we may apply a similarity transformation over the test matrix M.

The condition on the matrix N ensures that the probability distribution function for the update time intervals $F(h)$ assigns smaller occurrence probabilities to increasingly long update time intervals, that is $F(h)$ decays rapidly. In particular we observe that N can always be bounded if there exists h_m such that $F(h) = 0$ for h larger than h_m. We can also bound the expression inside the expectation to obtain $E[(e^{2\sigma(\Lambda)h} - 1)^{1/2}] < E[e^{\sigma(\Lambda)h}]$ and formulate the following corollary.

Corollary 5.7 *The system described by (5.1), with update time intervals $h(j)$ that are independent identically distributed random variable with probability distribution $F(h)$ is globally almost sure (or with probability-1) asymptotically stable around the solution $z = [x \quad e]^T = [0 \quad 0]^T$ if*

$$T = E\left[e^{\sigma(\Lambda)h}\right] < \infty \tag{5.22}$$

and the expected value of the maximum singular value of the test matrix M, $E[\|M\|] = E[\bar{\sigma}_M]$, is strictly less than one, where $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$.

Note that condition $T = E[e^{\sigma(\Lambda)h}] < \infty$ is automatically satisfied if the probability distribution function $F(h)$ does not have infinite support. It otherwise indicates that $F(h)$ should roll off fast enough as to counteract the growth of M's maximum singular value as h increases.

Example 5.2 Consider the following unstable plant dynamics

$$A = \begin{bmatrix} 0.21 & 1.04 \\ 0.63 & -0.46 \end{bmatrix}, \quad B = \begin{bmatrix} -0.52 \\ 0.98 \end{bmatrix}.$$

Let our plant model be:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u, \quad \hat{A} = \begin{bmatrix} 0.3 & 1 \\ 0.5 & -0.5 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix}.$$

Let our feedback law be given by $u = K\hat{x}$ with $K = [-2.1105 \quad -1.5053]$. The initial conditions of the system are $x_0 = [1 \ -1]^T$ while the model is initialized at 0. We now assume that $h(k)$ is a random variable with a uniform probability distribution function U[1, $h_{\max}$]. The plot of the expected maximum singular value of a similarity transformation of the original test matrix M is shown in Fig. 5.2. The similarity transformation used here was one that diagonalizes the matrix M for $h = 1$.

Figure 5.3 shows a simulation example using the plant and the model dynamics shown in this example where the update intervals are time-varying and have a probability distribution U[1, $h_{\max}$] where $h_{\max} = 2.5$ s. This figure shows the states of the system and it also shows the realization of the time update instants for this particular run, where it can be seen that the update time intervals (the difference between any two consecutive update time instants) are time-varying and take values between 1 and 2.5.

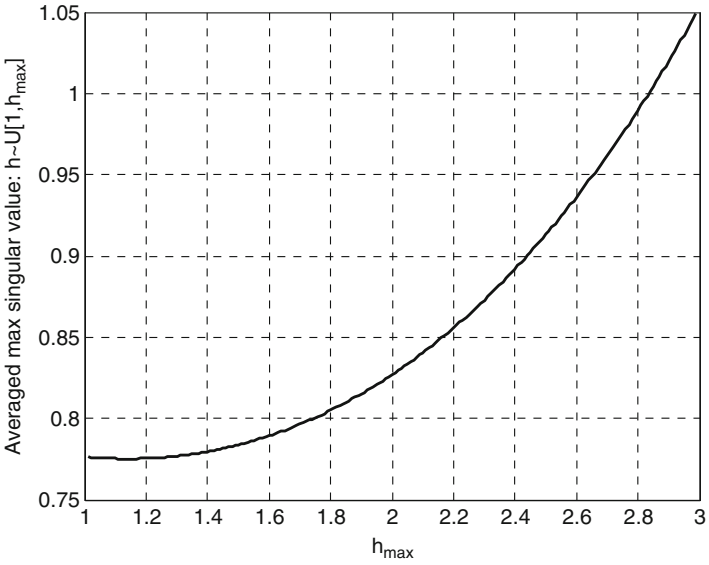

Fig. 5.2 Average maximum singular value for $\overline{\overline{\sigma}}_M = E[\|M\|]$ for $h \sim \mathrm{U}[1,h_{max}]$ as a function of h_{max}

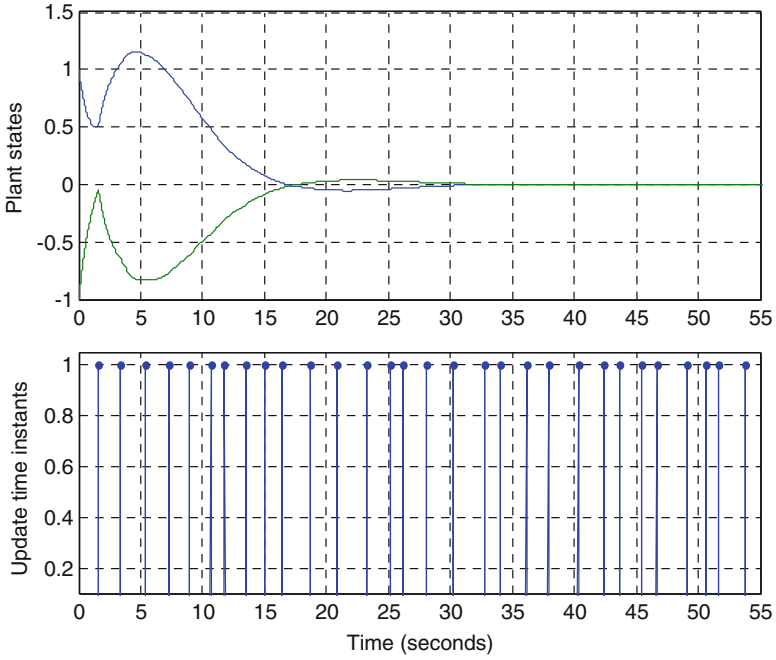

Fig. 5.3 *Top*: Response of system for $h \sim \mathrm{U}[0.5,h_{max}]$. *Bottom*: Time-varying update instants with uniform probability distribution

5.2.2 MB-NCS with Markov Chain-Driven Transmission Times

In certain cases it is appropriate to represent the dynamics of the update time intervals as driven by a Markov chain. A good example of this is when the network experiences traffic congestion or has queues for message forwarding. We now present a stability criterion for the model-based control system in which the update time intervals $h(k)$ are driven by a finite state Markov chain. Assume that the update time intervals can take a value from a finite set:

$$h_i \neq \infty, \forall i \in [1, N]. \tag{5.23}$$

Let us represent the Markov chain process by $\{\omega_k\}$ with state space $\{1, 2, \ldots, N\}$ and transition probability matrix Γ, an $N \times N$ matrix with elements $p_{i,j}$ and initial state probability distribution $\Pi_0 = [\pi_1 \quad \pi_2 \quad \ldots \quad \pi_N]^T$. The transition probability matrix entries are defined as $p_{i,j} = \mathrm{P}\{\omega_{k+1} = j | \omega_k = i\}$. We can now represent the update time intervals more appropriately as $h(k) = h_{\omega_k}$.

A sufficient condition for the almost sure stability of the system under Markovian jumps is given in the following theorem.

Theorem 5.8 *The system described by (5.1), with update time intervals* $h(k) = h_{\omega_k} \neq \infty$ *driven by a finite state Markov chain* $\{\omega_k\}$ *with state space* $\{1, 2, \ldots, N\}$ *and transition probability matrix* Γ *with elements* $p_{i,j}$ *and initial state probability distribution* $\Pi_0 = [\pi_1 \quad \pi_2 \quad \ldots \quad \pi_N]^T$ *is globally almost sure asymptotically stable around the solution* $z = [x \quad e]^T = [0 \quad 0]^T$ *if the matrix* T *has all its eigenvalues inside of the unit circle, where*:

$$\begin{aligned} T &= \begin{bmatrix} \left\| M\big|_{h=h_1} \right\| & 0 & 0 & 0 \\ 0 & \left\| M\big|_{h=h_2} \right\| & 0 & 0 \\ 0 & 0 & .. & 0 \\ 0 & 0 & 0 & \left\| M\big|_{h=h_N} \right\| \end{bmatrix} \Gamma^T \text{ and} \\ M\big|_{h=h_i} &= \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h_i} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{5.24}$$

Proof It is clear that since the Markov chain has a finite number of states, the update time intervals are bounded. Then using the same argument in Theorem 5.3 we can bound the output by:

$$\begin{aligned} \left\| e^{\Lambda(t-t_k)} \left(\prod_{j=1}^{k} M(j) \right) z_0 \right\| &\leq \left\| e^{\Lambda(t-t_k)} \right\| \cdot \left\| \prod_{j=1}^{k} M(j) \right\| \cdot \|z_0\| \\ &\leq e^{\overline{\sigma}(\Lambda) h_{\max}} \cdot \left\| \prod_{j=1}^{k} M(j) \right\| \cdot \|z_0\|. \end{aligned} \tag{5.25}$$

Therefore we can ensure stability by studying stability of the term $\left\|\prod_{j=1}^{k} M(j)\right\|$. We can now use a similar procedure as the one used for Theorem 5.6. For almost sure stability we will require that:

$$\lim_{\widetilde{\delta}\to\infty} P\left\{\sup_{k\geq\widetilde{\delta}}\left\|\prod_{j=1}^{k} M(j)\right\| > \varepsilon\right\} = 0. \tag{5.26}$$

We will assume that the supremum of the norm bracketed is achieved at $k* \geq \widetilde{\delta}$. Using the Chevyshev inequality we obtain.

$$\begin{aligned} P\left\{\sup_{k\geq\widetilde{\delta}}\left\|\prod_{j=1}^{k} M(j)\right\| > \varepsilon\right\} &= P\left\{\left\|\prod_{j=1}^{k_*} M(j)\right\| > \varepsilon\right\} \\ &\leq \frac{E\left[\left\|\prod_{j=1}^{k_*} M(j)\right\|\right]}{\varepsilon} \leq \frac{E\left[\prod_{j=1}^{k_*}\|M(j)\|\right]}{\varepsilon}. \end{aligned} \tag{5.27}$$

Evaluating the expectation yields:

$$\begin{aligned} &E\left[\prod_{j=1}^{k_*}\|M(j)\|\right] \\ &= \sum_{\forall[i_0,i_1,..,i_{k_*-1}]}\left\{\left(\prod_{j=1}^{k_*}\|M(j)\|\right)\times P\left\{[\omega_0 = i_0, \omega_1 = i_1, .., \omega_{k_*-1} = i_{k_*-1}]\right\}\right\} \\ &= \sum_{\forall[i_0,i_1,..,i_{k_*-1}]}\left(\prod_{j=2}^{k_*}\|M(j)\|p_{i_{j-1},i_{j-2}}\right)\|M(1)\|\pi_{i_0} \\ &= [1,1,\ldots,1]\left(\begin{bmatrix} \left\|M\big|_{h=h_1}\right\| & 0 & 0 & 0 \\ 0 & \left\|M\big|_{h=h_2}\right\| & 0 & 0 \\ 0 & 0 & .. & 0 \\ 0 & 0 & 0 & \left\|M\big|_{h=h_N}\right\| \end{bmatrix}\Gamma^T\right)^{k_*-1} \times \begin{bmatrix} \left\|M\big|_{h=h_1}\right\|\pi_1 \\ \left\|M\big|_{h=h_2}\right\|\pi_1 \\ \ldots \\ \left\|M\big|_{h=h_N}\right\|\pi_1 \end{bmatrix} \end{aligned} \tag{5.28}$$

where each $i_n \in \{1,2,..,N\}$. Therefore the right hand side part of (5.27) will converge to 0 if T has all its eigenvalues inside the unit circle where:

$$T = \begin{bmatrix} \left\|M\big|_{h=h_1}\right\| & 0 & 0 & 0 \\ 0 & \left\|M\big|_{h=h_2}\right\| & 0 & 0 \\ 0 & 0 & .. & 0 \\ 0 & 0 & 0 & \left\|M\big|_{h=h_N}\right\| \end{bmatrix}\Gamma^T \tag{5.29}$$

If Γ is irreducible it follows that, since $\|M\|$ is non-negative, T is also irreducible. Then it can be shown using the Perron-Frobenius theorem as in [42], that

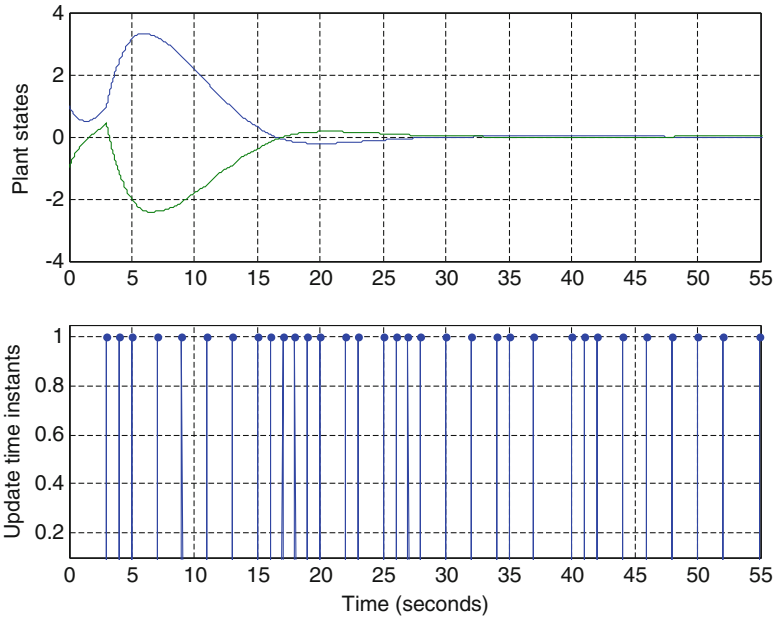

Fig. 5.4 *Top*: Response of system for time-varying h given by a Markov chain. *Bottom*: Time-varying update instants that follow a Markov chain

T's maximum magnitude eigenvalue is real; it is sometimes referred to as the Perron-Frobenius eigenvalue.

Example 5.3 Consider the same plant, model, and controller as in Example 5.2. Now the update time intervals take values from a finite set $h = \{1,2,3\}$. The transition probabilities are given by:

$$\Gamma = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.4 & 0.5 & 0.1 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}.$$

The eigenvalues of T in (5.24) lie inside the unit circle, where a similarity transformation is used to diagonalized M. Figure 5.4 shows a simulation example using the plant and the model dynamics shown in this example where the update intervals are time-varying and follow the transition probabilities given by Γ above. This figure shows the states of the system and it also shows the realization of the time update instants for this particular run. Now, it can be seen that the update time intervals (the difference between any two consecutive update time instants) are time-varying and take the specific values 1, 2, or 3.

5.3 Mean Square Asymptotic Stability

We now define a different type of stability, namely Mean Square Asymptotic Stability.

Definition 5.9 The equilibrium $z = 0$ of a system described by $\dot{z} = f\ (t,\ z)$ with initial condition $z(t_0) = z_0$ is mean square asymptotically stable at large (or globally) if the solution of $\dot{z} = f(t, z)$ satisfies

$$\lim_{t\to\infty} E\left[\|z(t, z_0, t_0)\|^2\right] = 0. \tag{5.30}$$

A system that is mean square stable will have the expectation of system states converging to 0 with time in the mean square sense. This definition of stability is attractive since many optimal control problems use the squared norm in their formulations. We will study the two cases studied in the previous section under this new stability criterion.

5.3.1 MB-NCS with Independent Identically Distributed Transmission Times

We present the conditions under which the networked control system described in (5.1) is mean square stable. We also discuss how these conditions relate to the ones for probability-1 stability.

Theorem 5.10 *The system described by (5.1), with update time intervals h(j) independent identically distributed random variable with probability distribution F(h) is globally mean square asymptotically stable around the solution* $z = [0 \quad 0]^T$ *if*

$$K = E\left[\left(e^{\overline{\sigma}(\Lambda)h}\right)^2\right] < \infty \tag{5.31}$$

and the maximum singular value of the expected value of M^TM,

$$\left\|E\left[M^TM\right]\right\| = \overline{\sigma}\left(E\left[M^TM\right]\right) < 1 \tag{5.32}$$

where $M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} e^{\Lambda h} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$.

Proof Let us start by evaluating the expectation of the squared norm of the output of the system described by (5.1).

$$
\begin{aligned}
&E\left[\left\| e^{\Lambda(t-t_k)}\left(\prod_{j=1}^{k} M(j)\right) z_0 \right\|^2\right] \\
&\quad = E\left[{z_0}^T \left(\prod_{j=1}^{k} M(j)\right)^T \left(e^{\Lambda(t-t_k)}\right)^T e^{\Lambda(t-t_k)} \left(\prod_{j=1}^{k} M(j)\right) z_0 \right] \\
&\quad \leq E\left[\overline{\sigma}\left(\left(e^{\Lambda(t-t_k)}\right)^T e^{\Lambda(t-t_k)} \right) {z_0}^T \left(\prod_{j=1}^{k} M(j)\right)^T \left(\prod_{j=1}^{k} M(j)\right) z_0 \right] \\
&\quad \leq E\left[\left(e^{\overline{\sigma}(\Lambda)h(k+1)}\right)^2 {z_0}^T \left(\prod_{j=1}^{k} M(j)\right)^T \left(\prod_{j=1}^{k} M(j)\right) z_0 \right].
\end{aligned} \tag{5.33}
$$

Now that the expectation is all in terms of the update time intervals, we can use the independently identically distributed property of the update time intervals and the assumption that K is bounded:

$$
\begin{aligned}
&E\left[\left(e^{\overline{\sigma}(\Lambda)h(k+1)}\right)^2 {z_0}^T \left(\prod_{j=1}^{k} M(j)\right)^T \left(\prod_{j=1}^{k} M(j)\right) z_0 \right] \\
&\quad = K \cdot {z_0}^T E\left[\left(\prod_{j=1}^{k-1} M(j)\right)^T M(k)^T M(k) \left(\prod_{j=1}^{k-1} M(j)\right) \right] z_0 \\
&\quad = K \cdot {z_0}^T E\left[\left(\prod_{j=1}^{k-1} M(j)\right)^T E\left[M^T M\right] \left(\prod_{j=1}^{k-1} M(j)\right) \right] z_0 \\
&\quad \leq K \cdot \overline{\sigma}\left(E\left[M^T M\right]\right) {z_0}^T E\left[\left(\prod_{j=1}^{k-1} M(j)\right)^T \left(\prod_{j=1}^{k-1} M(j)\right) \right] z_0.
\end{aligned} \tag{5.34}
$$

We can repeat the last three steps recursively to finally obtain

$$
\begin{aligned}
&E\left[\left\| e^{\Lambda(t-t_k)}\left(\prod_{j=1}^{k} M(j)\right) z_0 \right\|^2\right] \\
&\quad \leq K\left(\overline{\sigma}\left(E\left[M^T M\right]\right)\right)^k {z_0}^T z_0.
\end{aligned} \tag{5.35}
$$

So now it is easy to see that if $\left\|E\left[M^T M\right]\right\| = \overline{\sigma}\left(E\left[M^T M\right]\right) < 1$ then the limit of the expectation as time approaches infinity is 0, which concludes the proof. ♦

Note the similarity between the conditions given by Theorems 5.6 and 5.10. In Theorem 5.6 we require the expectation of the maximum singular value of the test matrix to be less than 1. For the stability in Theorem 5.10 the maximum singular value of the expectation of $M^T M$ should be less than 1.

5.3.2 MB-NCS with Markov Chain-Driven Transmission Times

We now present a sufficient condition for the mean square stability of the MB-NCS with Markov chain-driven update time intervals.

Theorem 5.11 *The system described by (5.1), with update time intervals* $h(k) = h_{\omega_k} \neq \infty$ *driven by a finite state Markov chain* $\{\omega_k\}$ *with state space* $\{1, 2, \ldots, N\}$ *and transition probability matrix* Γ *with elements* $p_{i,j}$ *is globally mean square asymptotically stable around the solution* $z = [x \quad e]^T = [0 \quad 0]^T$ *if there exists positive definite matrices P(1), P(2), …, P(N) such that*

$$\left(\sum_{j=1}^{N} p_{i,j}\left(H(i)^T P(j) H(i)\right) - P(i)\right) < 0 \tag{5.36}$$

$\forall\, i, j = 1, \ldots, N$ *with* $H(i) = e^{\Lambda h_i}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$.

Proof Using the same argument used in Theorem 5.8, since the update time intervals are bounded, we can analyze the system's stability by sampling it at a certain time between each update time interval. For this, we evaluate the response of the system described by (5.1) at times t_k^-:

$$z\left(t_{k+1}^-\right) = e^{\Lambda h_{k+1}}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} z\left(t_k^-\right). \tag{5.37}$$

Let us define $\varsigma(k) = z(t_{k-1}^-)$ and $H(\omega_k) = e^{\Lambda h_{\omega_k}}\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$. Now we can represent the sampled networked control system as:

$$\varsigma(k+1) = H(\omega_k)\varsigma(k). \tag{5.38}$$

To ensure mean square stability we will make use of a Lyapunov function of quadratic form and analyze the expected value of its difference between two consecutive samples. We will use the following Lyapunov function:

$$V(\varsigma(k), \omega_k) = \varsigma(k)^T P(\omega_k)\varsigma(k). \tag{5.39}$$

The expected value of the difference is:

$$
\begin{aligned}
&E\left[\Delta V\middle|\varsigma, i\right] \\
&= E\left[V(\varsigma(k+1), \omega_{k+1}) - V(\varsigma(k), \omega_k)\middle|\varsigma(k) = \varsigma, \omega_k = i\right] \\
&= E\left[\varsigma(k+1)^T P(\omega_{k+1})\varsigma(k+1)\middle|\varsigma(k) = \varsigma, \omega_k = i\right] - \varsigma^T P(i)\varsigma \\
&= E\left[\varsigma^T H(\omega_k)^T P(\omega_{k+1}) H(\omega_k)\varsigma\middle|\omega_k = i\right] - \varsigma^T P(i)\varsigma \\
&= \sum_{j=1}^{N} p_{i,j}\left(\varsigma^T H(i)^T P(j) H(i)\varsigma\right) - \varsigma^T P(i)\varsigma \\
&= \varsigma^T \left(\sum_{j=1}^{N} p_{i,j}\left(H(i)^T P(j) H(i)\right) - P(i)\right)\varsigma.
\end{aligned}
\tag{5.40}
$$

From this last equality is it obvious that to ensure mean square stability we need to have:

$$
\left(\sum_{j=1}^{N} p_{i,j}\left(H(i)^T P(j) H(i)\right) - P(i)\right) < 0. \tag{5.41}
$$

♦

Remark This type of stability criteria depends on our ability to find appropriate $P(i)$ matrices. Several other results in jump system stability [49, 201] can be extended to obtain other conditions on stability of networked control systems. Note though, that most of the results available in the literature deal with similar but not identical type of systems.

5.4 Notes and References

Most work on networked control systems assumes deterministic communication rates [117, 124]. Several authors have addressed time-varying rates [37, 60, 109, 174, 180, 198, 222, 245, 260, 286]. Most of these papers focus on characterizing the Maximum Allowable Transfer Interval (MATI) for stability. Other authors have concentrated on characterizing stability or performance on a networked control system under time-varying, stochastic communication (see [6, 7, 13, 156, 157]).

Additionally, there exist a large number of event-triggered and self-triggered approaches for networked control systems where the transmission intervals are not periodic but defined by the occurrence of certain events related to the current response of the system; see Chap. 6 for related references.

In previous chapters we considered the MB-NCS framework using periodic communication. The results in this chapter extend the MB-NCS framework to consider

time-varying update intervals with and without statistical information about the update intervals. This work appeared first in [189]. In Chap. 6 event-triggered control techniques will be implemented within the MB-NCS architecture. Event-triggered strategies provide a different alternative to control networked systems using time-varying transmission intervals while also reducing the frequency at which measurement updates need to be sent from sensor to controller.

Chapter 6
Event-Triggered Feedback Updates

In Chaps. 2–4 it has been assumed that the update interval h is constant. In Chap. 5, extensions have considered time-varying update intervals that reflect characteristics of the underlying digital network. The main goal in the present chapter is to adjust the update interval based on the current state of the plant. This new way of updating the state of the model is based on a simple idea, namely, to send a measurement update only when it is necessary to do so according to some criterion. The main question in this regard is how to determine when the sensor must send a measurement to update the model state in the MB-NCS setup.

One possible answer to this question is to implement an event-triggered strategy in which the controller updates are triggered by the size of the state error. In typical event-triggered control schemes the controller generates a piecewise input by holding the received measurements constant. The error is defined as $e(t) = x(t_i) - x(t)$, that is, the difference between the last measurement that was used to update the controller and the current state of the system.

In the case of MB-NCS the approach is very similar but the state error that will determine the update instants is the one that we have been using in previous chapters, the difference between the state of the model and the state of the plant. The application of event-triggered control to MB-NCS produces many advantages compared to the periodic implementation studied in previous chapters. For instance, nonlinearities and inaccuracies that affect the system and are difficult or impossible to model and may change over time or under different physical characteristics (temperature, different load in a motor, etc) may be handled more efficiently by tracking the state error than by updating the state at a constant rate. Also, the method presented here is robust under random packet loss. If the sensor sends data but the model state is not updated because of a packet dropout, the state error will grow rapidly above the threshold and the sensor will transmit once again the current information needed to update the model. However, under a fixed transmission rate, if a packet is lost the model will need to wait until the next update time to receive the feedback data, thus compromising the stability of the system.

The implementation of an event-based rule in MB-NCS represents a very intuitive way of saving network resources. It also considers the performance of the

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_6,

closed loop real system. The implementation of the model to generate an estimate of the plant state and using that estimate to control the plant, results in significant savings in bandwidth as we saw in previous chapters. It is also clear that the accuracy of the estimate depends on factors, such as the size of the model uncertainties. One of the results in the present chapter is that these error based updates provide more independence from model uncertainties when performing this estimation. When the error is small, which means that the state of the model is an accurate estimation of the real state, then we save energy, effort, and bandwidth by electing not to send measurements for all the time intervals in which the error remains small.

The combined Model-Based Event-Triggered (MB-ET) framework can be seen as a way of providing "virtual feedback" to control the physical system when no real measurements can be obtained at the controller node due to communication constraints or, more precisely, due to the strategies that we follow in order to reduce network communication. The idea of "virtual feedback" can be realized by using the nominal model of the system to generate an estimate of the real state in joint operation with an event-triggering strategy in the sensor node that determines the communication instants based on the size of the state error. When a measurement of the real state is received at the controller node, it is used to update the state of the model in order to reset the accumulated state error over the previous time interval since the last measurement update took place. The combination of the nominal model at the controller and the event-triggering strategy provides a "virtual feedback" by generating an estimate of the state that is kept close to the real state by maintaining a small state error. In order to obtain the same model state $\hat{x}$ at the sensor node we implement a second identical model at the sensor node that is updated using the measurements $x(t_k)$ only at those instants that a measurement is sent through the network.

Due to the advantages it provides, event-triggered control will be used in several sections of this book. In Sect. 7.2, control of dissipative nonlinear systems is studied using a fixed threshold event-based approach. Section 8.2 follows the same approach of Sect. 6.3 in this chapter to consider systems with quantization and delays. Tracking of reference inputs using event-triggered control is discussed in Sect. 11.3. Event-triggered control will be used to deal with different problems in Chaps. 9, 12, and 14; these problems include: adaptive stabilization, design of optimal controllers and schedulers, and stabilization of distributed systems.

The present chapter is organized as follows. In Sect. 6.1 the MB-ET control architecture is presented in addition to initial approaches and strategies. In Sect. 6.2 we present more formal stabilizing event-triggered strategies. In Sect. 6.3 network delays are considered and notes and references are given in Sect. 6.4.

6.1 Model-Based Event-Triggered Architecture

In event-triggered broadcasting [10, 11, 246, 247], a subsystem sends information about its local state to the network only when it is necessary, that is, only when a measure of the local subsystem state error is above a specified threshold. Event-

triggered control schemes offer a new point of view, with respect to conventional time-driven strategies, on how information could be sampled for control purposes. The author of [247] showed the stabilizing properties of the event-triggered control strategy; a triggering condition was presented based on the norms of the state and the state error (the error is obtained by subtracting the current state of the system from the last state that was used to update the controller). This means that the measurement received at the controller node is held constant until a new measurement arrives. When this happens, the error is set equal to 0 and starts growing until it triggers a new execution or measurement update. In the case that delays are not negligible the control task should be executed before the regular (no delay) execution condition takes place in order to account for those time delays, but the control task should not be executed too soon and provoke accumulation points, i.e., the generation of infinite number of events during a finite period of time.

In previous work on event-triggered control it is assumed that the parameters of the system and/or subsystems are known exactly. Our combined MB-ET control framework offers two important advantages with respect to previous work in event-triggered control that uses only a ZOH model, that is, the updates remain constant between update intervals. The implementation of this strategy using MB-NCS accounts for the unavoidable existence of model uncertainties in the stability analysis and it affects directly the estimated threshold values that aim to ensure stability of the system. The model-based implementation also generalizes the ZOH one by using the model dynamics to generate an estimate of the real plant state between measurement updates.

The architecture that we use in the present chapter is presented in Fig. 6.1 and it is similar to the one presented in Chap. 2 where periodic updates were used. The difference is that a copy of the model is implemented at the sensor node to obtain the state of the model at the right side of the control loop which is used to compute the state error and decide the appropriate time instants to send measurements. This implementation is very similar to the one in Sect. 3.1, which studies state observers, where a copy of the model and a copy of the controller are also needed at

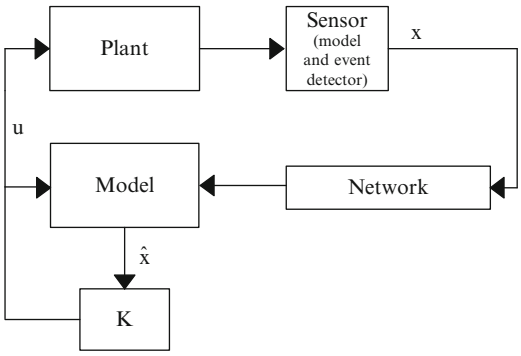

Fig. 6.1 MB-ET architecture

the sensor node. The difference in that case is that the copies of the model and the control gain were needed to generate the control input u in order to feed the state observer, while the updates take place in a periodic fashion. Here, the model at the sensor location is used to generate the state of the model and evaluate the state error.

6.1.1 A Preliminary Update Approach

The general idea in this chapter is to find implementable strategies that can be used to readjust the update intervals h in MB-NCS according to the current operating conditions of the networked system. The simple approach described next illustrates this idea and it is based only on the computation of the state error which is obtained by measuring the states of the plant and the model.

Let us consider a continuous time plant and its model interconnected as in Fig. 6.1 with dynamics given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{6.1}$$

and

$$\dot{\hat{x}}(t) = \hat{A}\,\hat{x}(t) + \hat{B}\,u(t) \tag{6.2}$$

respectively. Let $u = K\hat{x}$. In this setup, every time the controller node receives an update, it sends an acknowledgment message to the sensor indicating when it should send the next state measurement; that is, the controller readjusts the value of the update interval h. h is time-varying and depends on its own previous value and the current response of the system. In order to find the next appropriate value for h we apply the following conditions on the current and the previous measured error:

$$h(t_{i+1}) = a(t_i)h(t_i) \tag{6.3}$$

where the integer index $i \geq 1$ represents the update instants, i.e., the ith time that an update is sent through the network and the adjustment factor $a(t_i)$ is obtained according to the following rules:

$$\begin{aligned}
a(t_i) &< 1 \quad if \quad \{|e(t_i)| > |e(t_{i-1})|\} \& \{|e(t_i)| > \varepsilon\} \\
a(t_i) &= 1 \quad if \quad \{|e(t_i)| > |e(t_{i-1})|\} \& \{|e(t_i)| < \varepsilon\} \\
a(t_i) &= 1 \quad if \quad \{|e(t_i)| < e(t_{i-1})\} \& \{|e(t_i)| > \varepsilon\} \\
a(t_i) &> 1 \quad if \quad \{|e(t_i)| < |e(t_{i-1})|\} \& \{|e(t_i)| < \varepsilon\}
\end{aligned} \tag{6.4}$$

where ε is a small error threshold value. The state error is defined as follows

$$e(t) = \hat{x}(t) - x(t). \tag{6.5}$$

To simplify the implementation for this case, the last measurement sent to the controller can be used instead of the state of the model in order to compute (6.5), which means that for this particular rule we do not necessarily need to implement a copy of the model in the sensor node.

This simple computational approach describes an intuitive idea: if the networked system is behaving well according to our imposed standard, ε, we keep the same update interval (second and third lines). If the system is behaving very well then we can wait a longer interval of time without updating the model. We do this by increasing the update interval by some $a > 1$ (fourth line). But, if the networked system is not performing as desired (first line), then we should update at a higher rate by decreasing the update interval by some $a < 1$. This simple approach is illustrated through an example.

Example 6.1 Consider the system, model, and controller described by:

$$A = \begin{pmatrix} 0.8 & 0.3 \\ -0.2 & 1.9 \end{pmatrix} \quad B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\hat{A} = \begin{pmatrix} 0.6981 & 0.4379 \\ -0.1078 & 1.7031 \end{pmatrix} \quad \hat{B} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$K = \begin{pmatrix} 11.5358 & -17.5158 \end{pmatrix}$$

Figure 6.2 shows the response of the plant and the model for initial conditions: $x(0) = [0.2 \quad 0.5]^T$, $\hat{x}(0) = [0 \quad 0]^T$.

The initial update interval is $h(0) = 0.1$ s. Figure 6.3 shows the state error and the communication in the network due to the updates from the sensor to the controller and it illustrates how the update interval varies with time and according to the state error.

Unfortunately, it is hard to state something concrete about the stability properties of this type of implementation. By limiting ourselves to study this problem based only on information about the measured error we obtained a simple computational strategy but its theoretical justification is challenging.

In order to show the stability properties of the framework used in this chapter we will make use of the configuration shown in Fig. 6.1 but we are going to include different features in the sensor node that will provide a policy on how to update the state of the model. In this scheme the sensor has different functions to perform. The sensor contains a copy of the model and the controller gain so it can have access to the model state. It continuously measures the actual state and computes the model-plant state error (6.5). The sensor also compares the norm of the error to a predefined threshold, and it broadcasts the plant state to update the model state if the error is greater than the threshold.

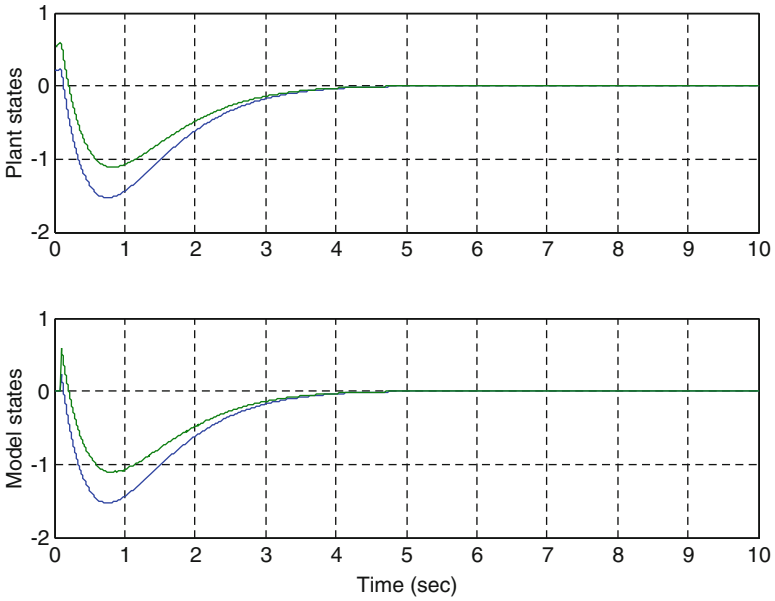

Fig. 6.2 Response of plant and model in Example 3.1

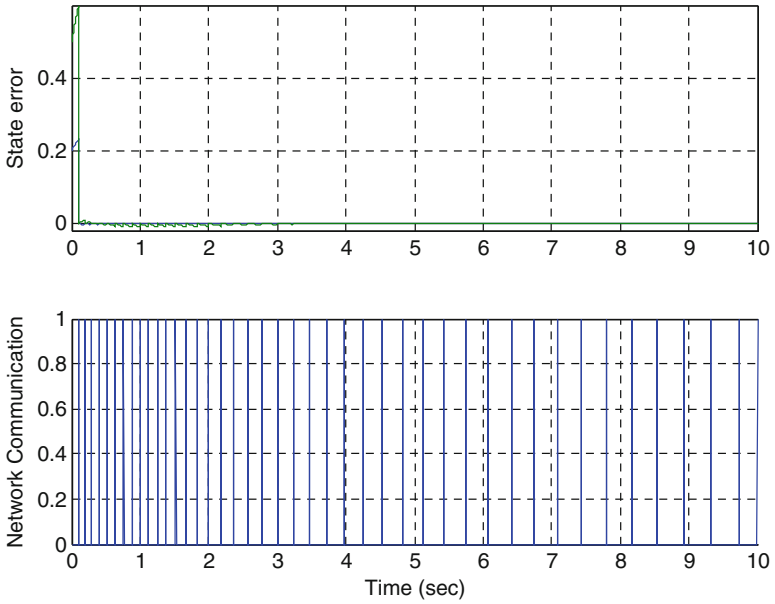

Fig. 6.3 State error and communication in the network due to the measurement updates

The purpose of the following sections is to develop a more formal approach based on previous work on event-triggered control keeping a fixed threshold value for the state error that guarantees bounded stability. The use of time-varying thresholds is also analyzed and asymptotic stability is shown. The use of time-varying thresholds for systems subject to network induced delays enhances the results presented in this chapter.

6.2 Event-Triggered Control Strategies

The main advantage the event-triggered feedback strategy offers compared with the common periodic-update implementation, is that the time interval between updates can be considerably increased, especially when the state of the system is close to its equilibrium point, thus releasing the communication network for other tasks. We will assume in this section that the communication delay is negligible. Instead of adjusting the update intervals explicitly as in Example 6.1 the approach in this section is to compute the norm of the state error and compare it to a positive threshold in order to decide if an update of the state of the model is needed. When the model is updated the error is equal to 0; when it becomes greater than the threshold the next update is sent.

Two distinct types of thresholds are considered in this section: fixed or constant thresholds and relative or time-varying thresholds. Let us start by considering a simple constant threshold with the update rule that the sensor sends the current state value when the norm of the state error (6.5) is greater than a fixed threshold that is denoted by α.

6.2.1 Fixed Threshold Strategy

While $|e| \leq \alpha$ no update is sent and the plant is operating in open loop mode based on the input generated by the model state $\hat{x}$. After substituting the input $u = K\hat{x}$ in (6.1) and using the definition of the error we can write:

$$\dot{x} = (A + BK)x + BKe. \tag{6.6}$$

In the case of the model, after substituting the input u we have a state space description of the form:

$$\dot{\hat{x}} = \left(\hat{A} + \hat{B}K\right)\hat{x}, \quad \text{for} \;\; t_i \leq t < t_{i+1}. \tag{6.7}$$

At the update times t_i, $i \in Z^+$, the state of the model is updated with the measurement obtained from the plant. The update intervals are non-periodic in general and are triggered by the size of the state error.

Let us first make use of a fixed threshold α, that is, an update is sent only when $|e| > \alpha$. By using this simple choice the state of the system can be bounded as shown in next theorem.

Theorem 6.1 *For* $|x(0)| \leq \beta_1, \quad 0 < \beta_1 < \infty$ *the system described by (6.6) with state feedback based on error events is bounded-input bounded-state stable with respect to the measurement error if the eigenvalues of the closed loop matrix* $A + BK$ *have negative real parts.*

Proof The response of the plant (6.6) with $t(0) = 0$ and Hurwitz matrix $A + BK$ at any given time $t \geq 0$ is given by:

$$x(t) = \mathrm{e}^{(A+BK)t}x(0) + \int_0^t \mathrm{e}^{(A+BK)(t-\tau)}BKe(\tau)d\tau \tag{6.8}$$

where $e(t)$ is a piecewise continuous input bounded by $|e| \leq \alpha$. We can show that the state of the plant is bounded by evaluating its norm which is done next:

$$\begin{aligned} |x(t)| &= \left| \mathrm{e}^{(A+BK)t}x(0) + \int_0^t \mathrm{e}^{(A+BK)(t-\tau)}BKe(\tau)d\tau \right| \\ &\leq \left|\mathrm{e}^{(A+BK)t}\right||x(0)| + \int_0^t \left|\mathrm{e}^{(A+BK)(t-\tau)}\right||BK||e(\tau)|d\tau. \end{aligned} \tag{6.9}$$

By the assumption on the initial condition and the triggering condition, and using the bound $|\mathrm{e}^{(A+BK)t}| \leq k_1\mathrm{e}^{-\lambda t}$, for $k_1, \lambda > 0$, we can write:

$$\begin{aligned} |x(t)| &\leq \beta_1 k_1 \mathrm{e}^{-\lambda t} + \alpha k_1 |BK| \int_0^t \mathrm{e}^{-\lambda(t-\tau)}d\tau \\ &= \left(\beta_1 k_1 - \frac{\alpha k_1 |BK|}{\lambda}\right)\mathrm{e}^{-\lambda t} + \frac{\alpha k_1 |BK|}{\lambda}. \end{aligned} \tag{6.10}$$

Also note that:

$$\lim_{t\to\infty} |x(t)| \leq \frac{\alpha k_1 |BK|}{\lambda}.$$

Using a stabilizing controller K, since the closed loop plant poles are in the left hand side of the complex plane, ensures that the first term in the right hand side of (6.10) decreases exponentially with time and the second term is bounded for all time $t > 0$.

Note also that by considering $y = x$, then (y,e) is BIBO stable when $A + BK$ is asymptotically stable. If (A,B) is controllable then the relation is if and only if. Then we need to ensure that the error is bounded by updating the model when $|e| \leq \alpha$ is not satisfied. ♦

Next, we show that the transmission intervals, which are time-varying, never become too close to each other i.e., we find a positive number that represents a lower bound on the update intervals. This is an important feature that every event-based control strategy should posses in order to avoid accumulation points, that is, the sensor should never attempt to send an infinite number of measurements during a finite period of time.

Theorem 6.2 *The model state update intervals for the MB-ET framework using a fixed threshold strategy are lower bounded by $t_\alpha > 0$ where t_α is the minimum time $t > 0$ such that the following holds*:

$$\left| \int_0^{t_\alpha} \mathrm{e}^{\left(\hat{A} - \widetilde{A}\right)(t_\alpha - \tau)} \left(\widetilde{A} + \widetilde{B}K\right) \mathrm{e}^{\left(\hat{A} + \hat{B}K\right)\tau} d\tau \right| |x(t_i)| = \alpha \tag{6.11}$$

for $i = 0,1,2,\ldots$

Proof First, we express the dynamics of the state error (6.5) in the following form:

$$\dot{e} = \left(\hat{A} - \widetilde{A}\right) e + \left(\widetilde{A} + \widetilde{B}K\right) \hat{x} \tag{6.12}$$

where $\widetilde{A} = A - \hat{A}$, $\widetilde{B} = B - \hat{B}$.

Next, we find the response, for $t > t_i$, of the error with initial time t_i, at the time of the model update with initial conditions $e(t_i) = 0$. The error at time t_i is equal to 0 since at that time an update event has taken place and the model state is made equal to the plant state.

$$e(t) = \int_{t_i}^{t} \mathrm{e}^{\left(\hat{A} - \widetilde{A}\right)(t - t_i - \tau)} \left(\widetilde{A} + \widetilde{B}K\right) \hat{x}(\tau) d\tau. \tag{6.13}$$

Consider, without loss of generality, $t_i = 0$ and evaluate the norm of (6.13) as follows:

$$\begin{aligned} |e(t)| &= \left| \int_0^{t} \mathrm{e}^{\left(\hat{A} - \widetilde{A}\right)(t - \tau)} \left(\widetilde{A} + \widetilde{B}K\right) \mathrm{e}^{\left(\hat{A} + \hat{B}K\right)\tau} d\tau \cdot x_0 \right| \\ &\leq \left| \int_0^{t} \mathrm{e}^{\left(\hat{A} - \widetilde{A}\right)(t - \tau)} \left(\widetilde{A} + \widetilde{B}K\right) \mathrm{e}^{\left(\hat{A} + \hat{B}K\right)\tau} d\tau \right| |x_0| \end{aligned} \tag{6.14}$$

where $\hat{x}(t) = e^{(\hat{A}+\hat{B}K)t}x_0$ was used and the quantity $x_0 = x(t_0)$ is the state of the system at the time of the latest update instant. By making the right-hand term in (6.14) equal to α we ensure that $|e| \leq \alpha$. It is clear that for any positive threshold α the minimum time t such that the right-hand term in (6.14) is equal to α is positive, i.e., the time elapsed between successive update intervals is always greater than 0. ♦

Remark In general we can see from (6.11) that for large uncertainties and large values of the plant state at the measurement update instants the model update intervals are very small. This can be better shown trough the following example.

Example 6.2 Consider the following model of a second order system:

$$\hat{A} = \begin{bmatrix} 1 & 1.3 \\ 0.7 & 0.5 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For simplicity, let us consider a system that contains uncertainty only in the first element of the state matrix A as follows:

$$A = \begin{bmatrix} 1 + \widetilde{a}_{11} & 1.3 \\ 0.7 & 0.5 \end{bmatrix}$$

The controller and the fixed threshold are given by:

$$K = [-1.6455 \quad -1.3545], \quad \alpha = 0.1$$

Figure 6.4 shows the lower bound on the transmission intervals for different values of $|x(t_i)|$ and $\widetilde{a}_{11}$. We can see that the transmission intervals are longer for smaller values of the measurements and the uncertainties and vice versa.

Figure 6.5 shows the response of the system for $\widetilde{a}_{11} = 0.1$ and the initial conditions $x(0) = [1 \quad -0.5]^T$. It can be seen in this figure that the transmission intervals get longer as the norm of the state gets smaller.

Remark From Example 6.2 one can ask the following questions: what would happen if the initial state is very large? Can we still guarantee that the model state update intervals are lower bounded by a positive number?

The short answer for the second question is "Yes"; from (6.11) we can see that t_α is always greater than 0 for any finite initial condition.

With respect to the first question, it is clear that the lower bound on the inter-event times will become smaller if the initial state is larger. From (6.11) we can see that the inter-event times will never become 0 but they could dangerously approach 0 for a large initial state compared to the size of the threshold α. A simple solution to avoid very small inter-event times is to, precisely, increase the value of the threshold. The drawback is that the ultimate bound for the plant state will be large as well. A much better approach is to start with a large threshold value which can be later reduced in order to reduce the norm of the state. This is one of the motivations to implement a time-varying threshold which is discussed in the next section.

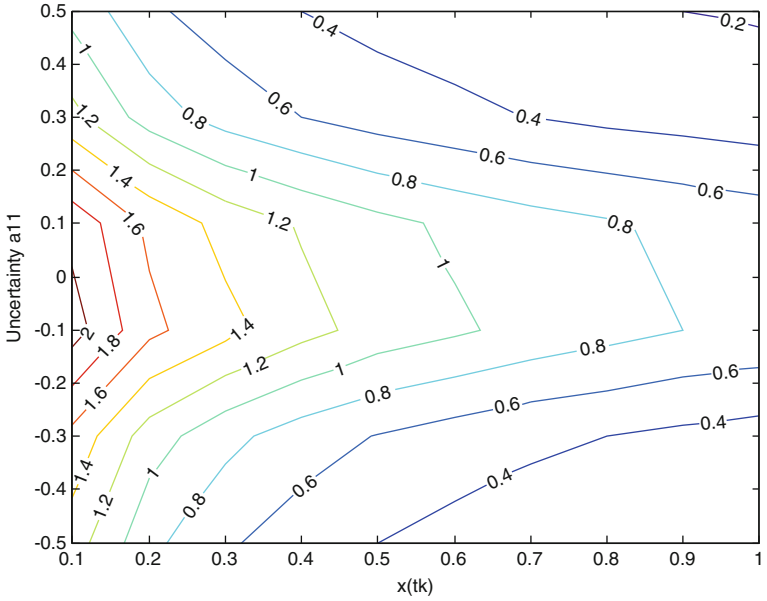

Fig. 6.4 Lower bound on the model update intervals for the system model pair in Example 6.2 for different values of the uncertainty and the value of the plant state

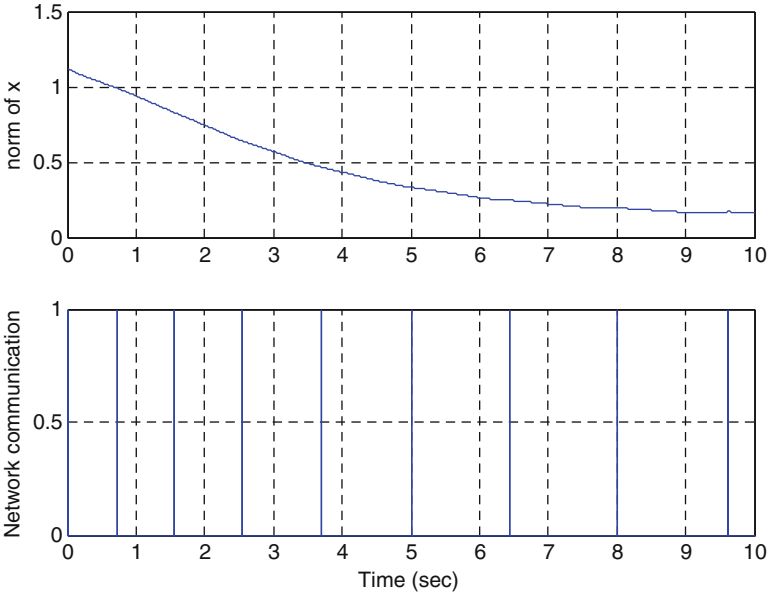

Fig. 6.5 Response of the MB-NCS in Example 6.2 for $\widetilde{a}_{11} = 0.1$

6.2.2 Relative Threshold Strategy

In many different applications it is desirable to stabilize a system asymptotically. Equation (6.11) relates to the intuitive idea that by varying the magnitude of the threshold value we can obtain longer update intervals or a smaller output size. This type of tradeoff leads us to consider a time-varying or relative threshold that becomes large when $|x(t_i)|$ is large and small when $|x(t_i)|$ is small. Furthermore, by making the threshold to be a function of the current norm of the plant state we are also in the position to achieve asymptotic stability and not only boundedness of the plant state.

The work in [247] follows this approach by comparing the norm of the state error to a function of the norm of the state of the plant; in this way, the threshold value is not fixed anymore, and, in particular, it can be reduced as we approach the equilibrium point of the system, assuming that the zero state is the equilibrium of the system. Traditional event-triggered control techniques [246, 247, 261, 262] consider systems controlled by static gains that generate piecewise constant inputs due to the fact that the update is held constant in the controller. The main difference in this section is that we use a Model-Based controller i.e., a model of the system and a static gain; the model provides an estimate of the state between updates and the model/gain controller provides a control input for the plant that does not remain constant between measurement updates.

Consider again the plant and model described by (6.1) and (6.2). Using the control input $u = K\hat{x}$ we obtain the description (6.6) for the plant. Assume that the control input u renders the model (6.2) Input-to-State Stable (ISS) with respect to the measurement error e. For the definition of ISS we use the following [2].

Definition 6.3 A smooth function $V : \mathbb{R}^n \to \mathbb{R}_0^+$ is said to be an ISS Lyapunov function for the dynamical system $\dot{x} = f(x,u),\ x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, t \in \mathbb{R}_0^+$ if there exist class $\mathcal{K}_\infty$ functions $\alpha_1, \alpha_2, \alpha_3$ and γ satisfying:

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \tag{6.15}$$

$$\frac{\partial V}{\partial x} f(x,u) \leq -\alpha_3(|x|) + \gamma(|u|). \tag{6.16}$$

The system $\dot{x} = f(x,u)$ is said to be ISS with respect to the input u if and only if there exists an ISS Lyapunov function for that system.

In our case, we choose a control law $u = K\hat{x}$ that renders the closed loop *model* (6.7) globally asymptotically stable. Any such K also renders the closed loop model ISS with respect to the measurement errors. We proceed to choose a quadratic ISS Lyapunov function, $V = x^T P x$ where P is symmetric positive definite and is the solution of the closed loop model Lyapunov equation:

$$\left(\hat{A} + \hat{B}K\right)^T P + P\left(\hat{A} + \hat{B}K\right) = -Q \tag{6.17}$$

where Q is a symmetric positive definite matrix.

Let us first analyze the case when $\hat{B} = B$ for simplicity and define the uncertainty $\widetilde{A} = A - \hat{A}$; also assume that the next bound on the uncertainty $\left|\widetilde{A}^T P + P\widetilde{A}\right| \leq \Delta < q$ holds where $q = \underline{\sigma}(Q)$, the smallest singular value of Q in the model Lyapunov equation (6.17). This bound can be seen as a measure of how close A and $\hat{A}$ should be. It can be seen from (6.17) that the solution P depends on the choice of Q. One way to obtain a small P and large q is to make $Q = -\left(\hat{A} + \hat{B}K\right)$ and design K such that this closed loop model matrix has eigenvalues with large negative real part. Unfortunately, the predefined location of the eigenvalues of $\hat{A} + \hat{B}K$ does not ensure, in general, a particular selection of the singular values. A particular case when this can be easily achieved is when the number of inputs is equal or greater than the number of states. In such a case, we can obtain a closed loop model matrix that is diagonal with desired eigenvalues, and with the previous choice of Q, the solution of (6.17) is always $P = 0.5 * I_{n \times n}$. Since $\hat{A} + \hat{B}K$ is diagonal its singular values are equal to the absolute value of its eigenvalues. Therefore, we can easily manipulate q while P remains the same.

The next theorem provides conditions on the error and its threshold value so the networked system is asymptotically stable. The error threshold is defined as a function of the norm of the state and Δ which is a bound on the uncertainty in the state matrix A. Similarly, the occurrence of an error event leads the sensor to send the current measurement of the state of the plant that is used in the controller to update the state of the model.

Theorem 6.4 *Consider system (6.1) with input $u = K\hat{x}$. Let the feedback be based on error events, and the relation:*

$$|e| > \frac{\sigma(q - \Delta)}{b}|x| \tag{6.18}$$

where $b = \left|K^T \hat{B}^T P + P\hat{B}K\right|$, $0 < \sigma < 1$, $\left|\widetilde{A}^T P + P\widetilde{A}\right| \leq \Delta < q$, and $q = \underline{\sigma}(Q)$. Let the model be updated when (6.18) is first satisfied. Then the system is globally asymptotically stable.

Proof In order to prove this theorem we will set a bound on the derivative of $V = x^T Px$ along the trajectories of the system (6.6) which is equal to (6.1) when the input $u = K\hat{x}$ has already been substituted and expressed in terms of the state error, then we can easily show that this bound can be appropriately tuned by the choice of the threshold on the error.

$$\begin{aligned}
\dot{V} &= x^T\left[(A + BK)^T P + P(A + BK)\right]x + e^T K^T B^T Px + x^T PBKe \\
&= x^T\left[\left(\hat{A} + \widetilde{A} + \hat{B}K\right)^T P + P\left(\hat{A} + \widetilde{A} + \hat{B}K\right)\right]x + e^T K^T \hat{B}^T Px + x^T P\hat{B}Ke \\
&= -x^T Qx + x^T\left(\widetilde{A}^T P + P\widetilde{A}\right)x + e^T K^T \hat{B}^T Px + x^T P\hat{B}Ke.
\end{aligned}$$

We have just expressed $\dot{V}$ in terms of the model parameters and the uncertainty of the state matrix A. We now proceed to evaluate the contributions of each, the model, the uncertainty, and the error.

$$\begin{aligned}\dot{V} &\leq -q|x|^2 + \left|\widetilde{A}^T P + P\widetilde{A}\right||x|^2 + \left|K^T \hat{B}^T P + P\hat{B}K\right||e||x| \\ &\leq (-q + \Delta)|x|^2 + b|e||x|.\end{aligned}$$

By updating the model when (6.18) is first satisfied we reset the error and we also have that $|e| \leq \sigma(q - \Delta)|x|/b$ holds. Then we can finally write:

$$\dot{V} \leq (\sigma - 1)(q - \Delta)|x|^2. \tag{6.19}$$

Then V is guaranteed to decrease for any σ such $0 < \sigma < 1$ and updating the state of the model every time the error satisfies the condition imposed in (6.18). ♦

Remark In comparison to usual strategies in MB-NCS, an important advantage of this approach is that we define the controller only in terms of the model parameters. The stabilizing threshold is designed using the model parameters $\left(\hat{A}, \hat{B}\right)$ and some bounds on the uncertainty $\widetilde{A}$, quantities that are known a priori.

The extension to the case of $\hat{A} \neq A$ and $\hat{B} \neq B$ is straightforward by assuming that the next bounds on the uncertainty matrices hold:

$$\left|\left(\widetilde{A} + \widetilde{B}K\right)^T P + P\left(\widetilde{A} + \widetilde{B}K\right)\right| \leq \Delta < q \tag{6.20}$$

$$\left|\widetilde{B}\right| \leq \beta \tag{6.21}$$

where $\widetilde{B} = B - \hat{B}$. In order to obtain the bound (6.19) the error is set to satisfy (triggering an update otherwise):

$$|e| \leq \frac{\sigma(q - \Delta)}{\overline{b}}|x| \tag{6.22}$$

where $\overline{b} = 2\left|P\hat{B}K\right| + 2\beta|PK|$. We consider the same Lyapunov function as in Theorem 6.4. In this case the derivative of the Lyapunov function along the trajectories of the system can be expressed by:

$$\begin{aligned}\dot{V} &= -x^T Q x + x^T\left[\left(\widetilde{A} + \widetilde{B}K\right)^T P + P\left(\widetilde{A} + \widetilde{B}K\right)\right]x + 2x^T P\left(\hat{B} + \widetilde{B}\right)Ke. \\ &\leq -q|x|^2 + \left|\left(\widetilde{A} + \widetilde{B}K\right)^T P + P\left(\widetilde{A} + \widetilde{B}K\right)\right||x|^2 + 2\left(\left|P\hat{B}K\right| + \left|P\widetilde{B}K\right|\right)|e||x| \\ &\leq (-q + \Delta)|x|^2 + \overline{b}|e||x|.\end{aligned}$$

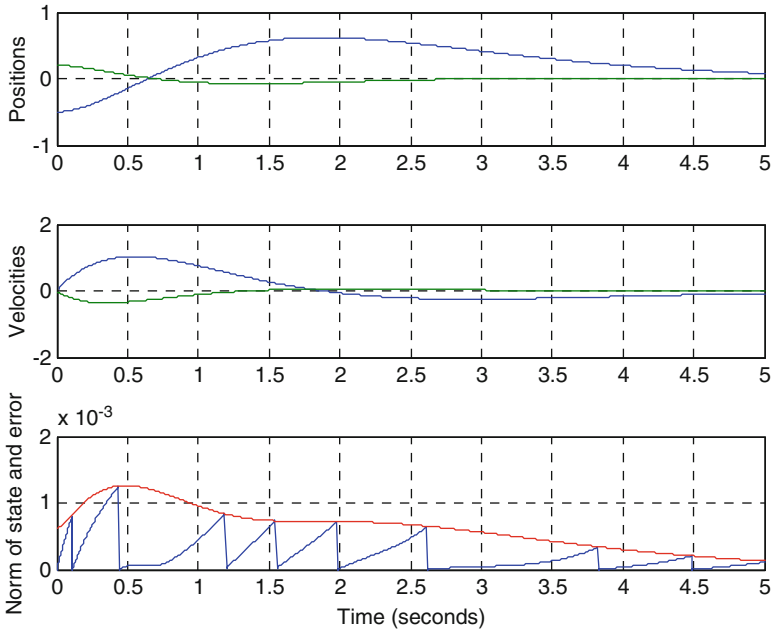

Fig. 6.6 Response of the system using relative threshold. *Top*: positions. *Middle*: velocities. *Bottom*: $|e(t)|$ (piecewise continuous) and $\frac{\sigma(q-\Delta)}{\overline{b}|x(t)|}$ (continuous)

By updating the model according to (6.22) we ensure that the error becomes 0 at the update instant and it satisfies $|e| \leq \sigma(q-\Delta)|x|/\overline{b}$ until a new update is generated and $\dot{V}$ satisfies the inequality (6.19).

Example 6.3 Consider the inverted pendulum on a moving cart dynamics described in Example 3.2. In this case the nominal parameters are given by: $\hat{m} = 0.1$, $\hat{M} = 1$, $\hat{l} = 1$. The values of the real parameters are given by: $m = 0.09978$, $M = 1.0016$, $l = 0.9994$. We also have that $\hat{g} = g = 9.8$. In this case we use the following control gain:

$$K = [0.5379 \quad 25.0942 \quad 1.4200 \quad 7.4812].$$

Figure 6.6 shows the response of the system; it also shows the norm of the system and the norm of the error. It can be seen that the error is reset at the update instants.

6.3 Systems with Network Induced Delays

Although the MB-NCS framework may help reduce network induced delays by limiting the number of packets that attempt to use the network to reach their destination, we should be prepared for situations in which peak conditions on the network produce considerable time delays between nodes. These conditions may be due to a high network load, i.e., many systems and nodes have been added to the network or, in the case of many systems being controlled by event-triggered strategies, due to several of them operating in a region that requires a higher communication rate. The solutions provided in previous sections assumed negligible time delays but it can be shown that the event-triggered control strategy is able to compensate for delays: if some delay characteristics are known (a bound or even the exact time delay when using time-stamped messages) the next update should be scheduled before the regular one (the update when no delay is present) in such a way that stability is never compromised. In this section we take this approach in order to determine the best time to update in the presence of time delays. Two advantages are obtained by using the MB-NCS framework and event-triggered controller. The first one is the known property of generating an estimate of the state when operating in open loop mode to get longer update intervals. The second advantage is that the model is able to produce almost instantaneously an estimate of the current plant state based on the delayed measurement. We can use this estimate instead of using the delayed measurement to update the model in the controller.

Next theorem provides conditions for asymptotic stability in the presence of network induced delays. In the case of delays it is also important to guarantee that the inter-execution update times never become too close to each other causing a model update in the controller when the previous execution has not been finished due to time delays or even resulting in a Zeno behavior. To show that this will never occur is a nontrivial task, since the execution time intervals are only implicitly defined by (6.18); and this is also shown in the next theorem.

Theorem 6.5 *Let* (6.1) *be a control system with control input based on the state of the nominal model* $u = K\hat{x}$ *and assume that*: *there exists a symmetric positive definite solution* P *for the model Lyapunov equation* (6.17), $B = \hat{B}$, *and the next bounds are satisfied*: $\left|\widetilde{A}\right| \leq \Delta_A$ *and* $\left|\widetilde{A}^T P + P\widetilde{A}\right| \leq \Delta < q$. *Then there exists an* $\varepsilon > 0$ *such that for all network delays* $\tau_N \in [0, \varepsilon]$ *the system is asymptotically stable. Furthermore, there exists a time* $\tau > 0$ *such that for any initial condition the inter-execution times* $\{t_{i+1} - t_i\}$ *implicitly defined by* (6.18) *with* $\sigma < 1$ *are lower bounded by* τ, *i.e.*, $t_{i+1} - t_i \geq \tau \ \forall\, i \in Z^+$.

Proof In order to show asymptotic stability for the nonzero network delay case and to bound the inter-execution times let us look at the dynamics of $|e|/|x|$:

$$\begin{aligned}\frac{d}{dt}\frac{|e|}{|x|} &= \frac{d}{dt}\frac{(e^Te)^{1/2}}{(x^Tx)^{1/2}} = \frac{(x^Tx)^{1/2}(e^Te)^{-1/2}e^T\dot{e} - (e^Te)^{1/2}(x^Tx)^{-1/2}x^T\dot{x}}{x^Tx} \\ &= \frac{e^T\dot{e}}{(x^Tx)^{1/2}(e^Te)^{1/2}} - \frac{(e^Te)^{1/2}x^T\dot{x}}{(x^Tx)^{3/2}} \\ &= \frac{e^T\left(\hat{A}\,e - \widetilde{A}x\right)}{|x||e|} - \frac{x^T\left[\left(\hat{A} + \widetilde{A} + BK\right)x + BKe\right]}{|x||x|}\frac{|e|}{|x|} \\ &\leq \left|\widetilde{A}\right| + \left|\hat{A}\right|\frac{|e|}{|x|} + \left|\hat{A} + \widetilde{A} + BK\right|\frac{|e|}{|x|} + |BK|\left(\frac{|e|}{|x|}\right)^2 \\ &\leq \Delta_A + \left(\Delta_A + \left|2\hat{A} + BK\right|\right)\frac{|e|}{|x|} + |BK|\left(\frac{|e|}{|x|}\right)^2. \end{aligned} \tag{6.23}$$

Let us denote the term $|e|/|x|$ by θ so we have the estimate:

$$\begin{aligned}\dot{\theta} &\leq \Delta_A + \left(\Delta_A + \left|2\hat{A} + BK\right|\right)\theta + |BK|\theta^2 \\ &\leq \Delta_A + \left|2\hat{A}\right| + \left(\Delta_A + \left|2\hat{A}\right| + |BK|\right)\theta + |BK|\theta^2\end{aligned} \tag{6.24}$$

and consider the differential equation:

$$\dot{\phi} = \Delta_A + \left|2\hat{A}\right| + \left(\Delta_A + \left|2\hat{A}\right| + |BK|\right)\phi + |BK|\phi^2 \tag{6.25}$$

then we can conclude that $\theta(t) \leq \phi(t, \phi_0)$, where $\phi(t, \phi_0)$ is the solution of (6.25) satisfying $\phi(0, \phi_0) = \phi_0$.

For the case when $\tau_N = 0$, the inter-execution times are bounded by the time it takes for ϕ to evolve from 0 to $\sigma(q - \Delta)/b$, i.e., the solution $\tau \in \mathbb{R}^+$ of $\phi(\tau, 0) = \sigma(q - \Delta)/b$. An estimate of that time can be obtained by solving (6.25). Such solution is given by:

$$\phi(t, 0) = \frac{-e^{dt(c-1)} + 1}{e^{dt(c-1)}/c - 1} \tag{6.26}$$

for $c \neq 1$. Let $y = \sigma(q - \Delta)/b = \phi(\tau, 0)$, then

$$\tau = \left(\ln(y + 1) - \ln\left(\frac{y}{c} + 1\right)\right)\frac{1}{d(c - 1)} \tag{6.27}$$

where $d = |BK|$ and $c = \left(\Delta_A + \left|2\hat{A}\right|\right)/d$. In the analysis if we have the case $c = 1$ we can easily avoid it by increasing the bound on the uncertainty by a very small amount. It can also be verified that $\tau > 0$ for any $y > 0$. The last statement can be shown by analyzing directly the two factors in (6.27). Note that c and d are

nonnegative and we consider two cases and avoiding the case $c = 1$ as we discussed previously. First, for $c > 1$ the second factor in (6.27) is positive, then, in order to obtain $\tau > 0$, we need the condition:

$$\frac{c(y+1)}{(y+c)} > 1$$

which is equivalent to the condition $y > 0$. For the case $0 < c < 1$ the second term is negative and we need the first factor to be negative in order to obtain a strictly positive value for τ. We can ensure that the first factor is negative by satisfying the following condition:

$$\frac{c(y+1)}{(y+c)} < 1$$

which is equivalent to $y > 0$.

In addition we can find the range of values for τ for any positive value of the threshold y is given by $\tau \in (0, \tau_m)$, where:

$$\tau_m = \lim_{y \to \infty} \tau = \frac{\ln(c)}{d(c-1)}. \tag{6.28}$$

For $\tau_N > 0$, we choose some σ' such that the next is satisfied $\sigma < \sigma' < 1$, and let $0 < \varepsilon_1 < \tau_m$ satisfy the solution $\phi(\varepsilon_1, y) = y' = \sigma'(q - \Delta)/b$, such that ε_1 always exists since ϕ is continuous in the range $\tau \in [0, \tau_m)$ that covers all positive thresholds $0 < y$, $y' < \infty$, also $\dot{\phi} > 0$ and $y < y'$ since $\sigma < \sigma'$. Then, by sending the state measurement at time t_i in order to update the model in the controller, this execution is released by the condition $|e| = y|x|$, we guarantee that for $t \in [t_i, t_i + \varepsilon_1]$ we have $|e| \leq y'|x|$, and since $\sigma' < 1$ asymptotic stability is still guaranteed. The inter-execution times are now bounded by $\tau_N + \tau$, where τ is the time it takes ϕ to evolve from $|e(t_i + \tau_N)|/|x(t_i + \tau_N)| = |\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)|/|x(t_i + \tau_N)|$ to y, then the admissible delays τ_N need to satisfy $|e(t_i + \tau_N)|/|x(t_i + \tau_N)| < y$ since $\dot{\phi} > 0$. From continuity of $|\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)|/x(t_i + \tau_N)$ with respect of τ_N there exists an $\varepsilon_2 > 0$ such that for any $0 \leq \tau_N \leq \varepsilon_2$ we have $|\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)|/|x(t_i + \tau_N)| < y$. The term $|\hat{x}(t_i + \tau_N) - x(t_i + \tau_N)|/|x(t_i + \tau_N)|$ is continuous due to the fact that $|x(t_i + \tau_N)|$ is never 0 since the closed loop system is asymptotically stable and never reaches 0 in finite time. We complete the proof by defining $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$. ♦

6.3.1 *Updating the Model State Using Delayed Measurements*

The stability and performance of the networked system is affected by the use of delayed measurements for control since the difference between the delayed measurement and the current state of the system produces a large state error when the

delayed data is used to update the model. A smaller state error can be obtained by estimating the current plant state based on the old measurements. This procedure represents another advantage compared to traditional ZOH event-triggered implementations. By computing a quantity that reflects more accurately the current state of the plant than the delayed measurement does, it is possible to execute a better control action over the next open loop interval, that is, the next event will be triggered later in time than by using the old data directly.

Constant delays. Since we need to implement the model of the plant in both the controller and the sensor node, in order to compute the control input and compute the state error respectively, we have to use the delayed information received by the controller effectively so a good estimate of the current plant state is obtained to update the model in the controller and compute a better control input for the plant. In the case that the network delays are constant we can implement the next strategy: the sensor decides to send a feedback measurement to the controller at time t_i so it updates its own state but keeps using the old input, i.e., the input generated by the same model in the case that no update has taken place, similar to the plant being fed by the model/controller that has not been updated yet. Notice that if the sensor knows the magnitude of the constant network delay τ_N then it will switch to closed loop mode at the end of the known delay. By using this strategy we need to implement a second model in the sensor node, but this is physically possible since we are considering operations performed by a single processor; that is, if we are able to implement the computations needed to measure and compute the state error and threshold comparisons then, in general, we could be able to implement a second closed loop model that only works for short intervals $[t_i,\ t_i+\tau_N]$. When the controller receives the measurement $x(t_i)$ at time $t_i+\tau_N$ it uses this measurement to immediately estimate the state of the model in the sensor by computing the next:

$$\hat{x}_c(t_i+\tau_N) = e^{\hat{A}\,\tau_N}x(t_i) + \int_0^{\tau} e^{\hat{A}\,(\tau-s)}Bu_c(s)ds \tag{6.29}$$

which can be made arbitrarily accurate by storing the sequence of inputs over the previous delay interval, i.e., $[t_i, t_i+\tau_N]$ in the controller node and since the parameters in both models are exactly the same. The subscript c indicates the quantities belonging to or available at the controller node. The result of the operation in (6.29) is used to update the state of the model in the controller.

Time-varying bounded delays. A more general situation in many networked systems is that the network induced delays are time-varying (and bounded) as discussed in the previous section. In this case the sensor does not know the current value of the delay, but by time-stamping the measurement sent over the network the controller node does know the size of the delay for every packet containing a feedback measurement. A simple strategy in this case is to let the model in the sensor remain working in closed loop after measuring and updating its state. When the controller

receives the delayed measurement it simply computes the following, which is used to update the model in the controller node.

$$\hat{x}_c(t_i + \tau_N) = e^{(\hat{A}+BK)\tau_N} x(t_i) \tag{6.30}$$

A slightly different strategy can be implemented in this case that, in general, results in better performance, i.e., longer broadcast intervals, by realizing that the states of both models do not need to be the same, as long as the model in the controller produces a smaller state error than the model in the sensor. This is basically a combination of the two strategies above. The sensor updates its state and continues working in closed loop mode but the controller uses the quantity obtained by (6.29) in order to obtain a better estimate of the current plant state not of the current sensor model state based on the delayed measurement.

Example 6.4 Consider the following networked system implemented as in Fig. 6.1, where the system to be controlled is unstable and is represented by the parameters:

$$A = \begin{bmatrix} 0.55 & -0.4 \\ 0.3 & -0.7 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Let the model be obtained by altering the physical parameters by 10 %,

$$\hat{A} = \begin{bmatrix} 0.495 & -0.360 \\ 0.270 & -0.630 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Let the controller, found using the model parameters, be given by:

$$K = [\,-1.3268 \quad 0.4618\,]$$

and the uncertainty bounds are given by: $\Delta = 1.05$, $\Delta_A = 0.1$. By choosing the following parameters: $q = 5$ and $\sigma = 0.5$ we can find the threshold $y = 0.1382$, then, by using the results of previous section we get $\varepsilon = 0.065$ s. Results of simulations are shown in Figs. 6.7 and 6.8. Figure 6.7 shows the response of the norm of the state of the plant and the norm of the error for a constant delay of 0.06 s. The discrete variations on the error correspond to the events generated at the sensor node i.e., when the sensor decides to transmit the current measurement and updates its internal model, resetting the error as measured by the sensor. Similarly, Fig. 6.8 shows the response of the norms of the state of the plant and the error for time-varying delays bounded by 0.06 s.

In both cases we are able to asymptotically stabilize the system in the presence of delays and using feedback measurements sent through the network at well separated instants of time, i.e., significantly reducing the traffic in the network.

In order to draw a comparison to the case when a ZOH model is used in the controller node, that is, the received measurement is held constant until a new measurement arrives we execute similar simulations using the same parameters, controller gains, and time delays that were used in the executions shown in Figs. 6.7

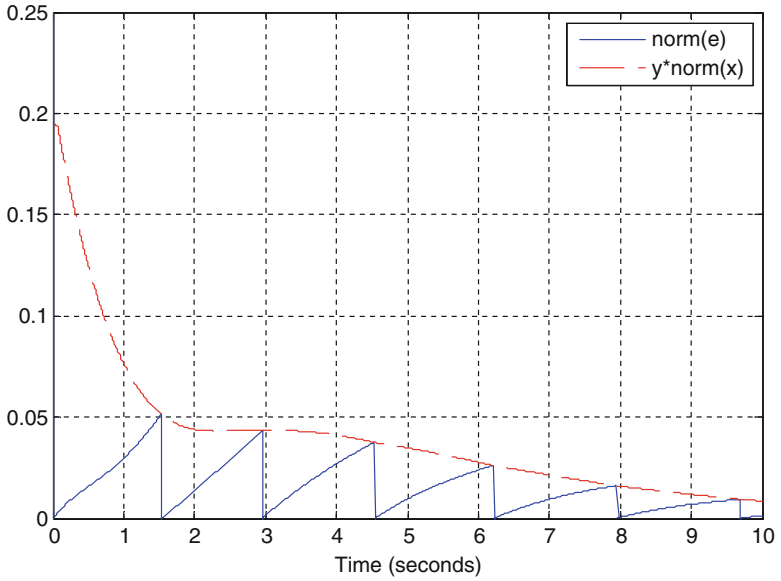

Fig. 6.7 Response of $|e(t)|$ and $y|x(t)|$ for constant delays $\tau_N = 0.06$ S

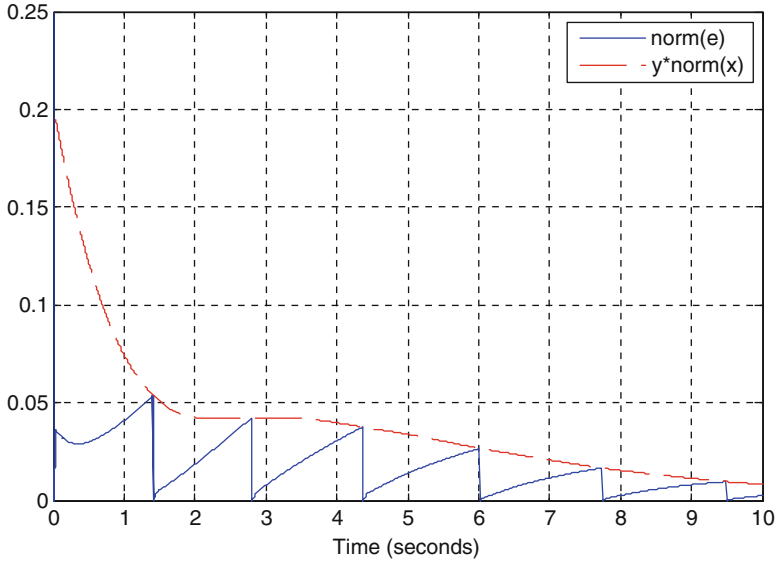

Fig. 6.8 Response of $|e(t)|$ and $y|x(t)|$ for the case of time-varying delays bounded by $\varepsilon = 0.06$ S

and 6.8 but now the measurements are held constant. Figures 6.9 and 6.10 show the simulation results. It can be seen that error events are triggered more frequently in both cases, when the delays are constant in Fig. 6.9, and when the delays are time-varying and bounded in Fig. 6.10.

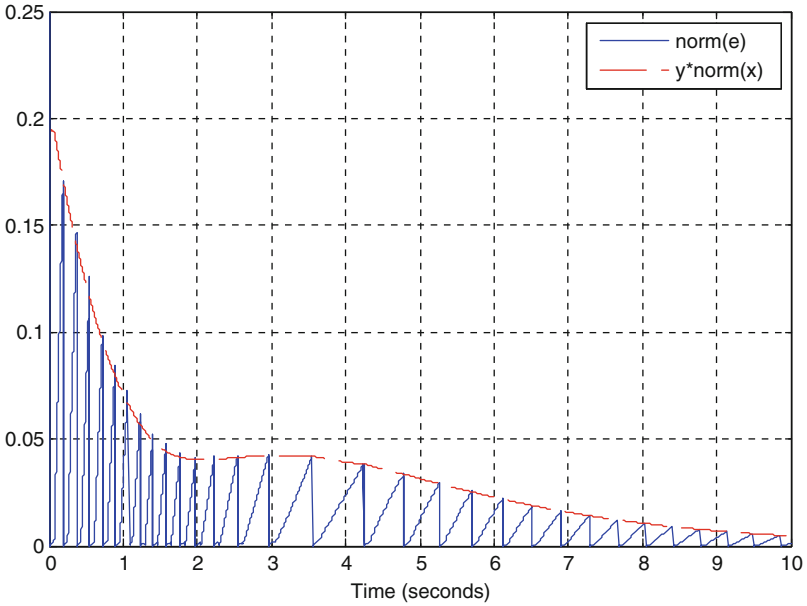

Fig. 6.9 Response of $|e(t)|$ and $y|x(t)|$ for the ZOH model case and constant delay

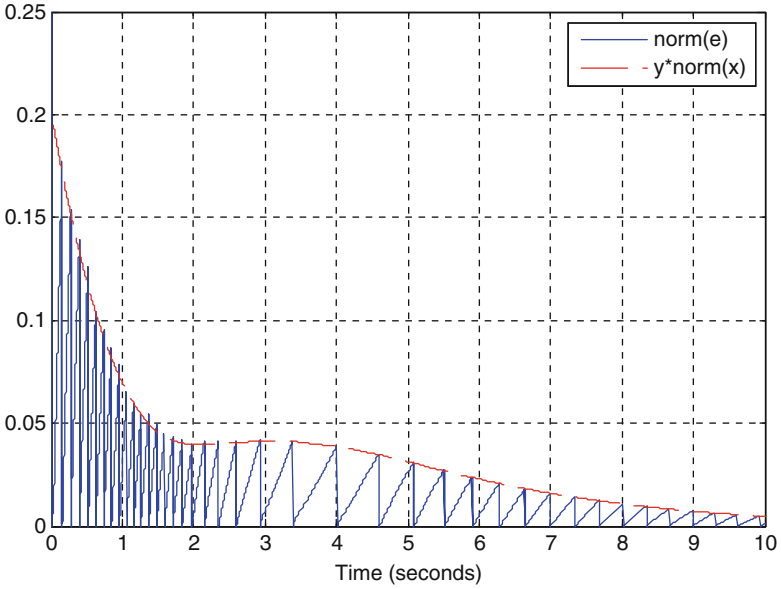

Fig. 6.10 Response of $|e(t)|$ and $y|x(t)|$ for the ZOH model case and time-varying bounded delays

6.4 Notes and References

The work presented in this chapter is an extended version of [84]. Parts of this chapter were also published in [87].

Event-triggered control techniques are commonly used in control systems [9, 10, 40, 58, 59, 99, 105, 108, 140–142, 146, 176–178, 246, 247, 250, 261, 262, 264, 265, 280]. All these references point out the advantage in reducing sampling rate by allowing the sensor to transmit measurements according to the current response of the system. A traditional approach followed by many authors is to implement a Zero-Order-Hold (ZOH) approach which means that the measurements received at the controller node are held constant until a new measurement arrives. The event-triggered control approach is also more robust to unknown disturbances than periodic approaches. Disturbances that affect the system can be compensated faster since the event-detectors will update the controller as soon as an erratic behavior is measured.

The use of event-triggered control techniques requires constant measurement of the state, calculation of the state errors, and comparison to the designed threshold. These requirements impose harder computations than typical periodic schemes. The use of event-triggered control in NCS is valuable from the point of view of communication savings, that is, communication costs are usually higher than computation costs.

In order to reduce the continuous sensing requirement imposed by event-triggered control techniques several authors have developed self-triggered control strategies [1, 2, 175, 176, 205, 263, 282]. In this case, the system dynamics and the last measurement transmitted are used to estimate the next time that the sensor has to measure the state of the system and transmit this measurement.

A similar model-based approach to the one presented in this chapter has been developed by Lunze and Lehman [165]. In their approach, the model is assumed to match the dynamics of the system exactly; however, the system is subject to unknown input disturbances. The main idea of the approach in [165] is the same as in the ideas of this book, that is, to use the nominal model to generate estimates of the current state of the system. Since the system is subject to unknown disturbances and the model is executed with zero input disturbance, then a difference between the states is expected and the sensor updates transmitted over a digital communication network are used to reset this difference between the states of the plant and of the model. The same authors have extended this approach to consider the output feedback, quantization, and network delay cases.

Event-triggered control strategies, in combination to the MB-NCS framework, will be used frequently in subsequent chapters. These techniques will be of great advantage to address different problems and control architectures that appear very often in networked systems.

Chapter 7
Model-Based Nonlinear Control Systems

The use of linear models in the MB-NCS framework not only provides a simple approach for controller and update intervals design, but the overall setup can also be used for nonlinear systems as well. By considering robustness to model uncertainties in the MB-NCS framework with linear systems, it is clear that we can implement a linearized model of the system to control a nonlinear plant. This is a very common approach, namely, controller design based on linearization. The difference with respect to linearization control techniques is that in the MB-NCS approach, the linearized model is not only used to design the control gain but it is also implemented in the actuator/controller node in order to obtain an estimate of the real state between measurement updates.

We expect the mismatch between the nonlinear plant and the linearized model to grow as the state leaves the linearization region. The result is that, in general, we expect only local stability of the model-based network interconnected system compared to the global stability property that can be obtained in the case the plant is linear as it was discussed in previous chapters. In the case of linear time-invariant (LTI) plants and models we saw that the model uncertainties were constant no matter the operating region of the plant state.

In this chapter we provide a more formal analysis of nonlinear MB-NCS using nonlinear models. In Sect. 7.1 we focus on the case when the measurement update intervals are constant, that is, we update the state of the model using the measured state of the system in a periodic fashion. In Sect. 7.2 we study nonlinear dissipative systems using output feedback and event-based updates. Notes and references are provided in Sect. 7.3.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_7,

7.1 Constant Update Time Intervals

In this section we present results for two common classes of nonlinear MB-NCS. The sufficient conditions obtained below relate the stability of the nonlinear MB-NCS to the value of a function that depends on the Lipschitz constants of the plant and model as well as the stability properties of the compensated non-networked model. The results are obtained by studying the worst-case behavior of the norm of the plant state and the error. Each subsection is augmented with an example where the results are illustrated.

7.1.1 Stability of a Class of Nonlinear MB-NCS

In this section we consider systems that can be represented by the following equations:

$$\dot{x} = f(x) + g(u). \tag{7.1}$$

A nominal nonlinear model of the system will be used in the actuator side to estimate the actual state of the plant. The controller will be assumed to be a nonlinear state feedback controller. The plant model state will be used together with the controller to generate the control signal u. The plant state sensor will send the real value of the plant state through the network in order to update the model every h seconds. Upon arrival of the sensor information, the model state is updated or reset to the actual value of the state of the plant. The dynamics of the nominal model are given by:

$$\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(u) \tag{7.2}$$

and the controller has the following form:

$$u = \hat{h}(\hat{x}). \tag{7.3}$$

Also define $e = x - \hat{x}$ as the error between the plant state and the nominal model state. Combining (7.1) and (7.2) with (7.3) we obtain:

$$\begin{aligned} \dot{x} &= f(x) + g\left(\hat{h}(\hat{x})\right) = f(x) + m(\hat{x}) \\ \dot{\hat{x}} &= \hat{f}(\hat{x}) + \hat{g}\left(\hat{h}(\hat{x})\right) = \hat{f}(\hat{x}) + \hat{m}(\hat{x}). \end{aligned} \tag{7.4}$$

We will also assume that the plant model dynamics differ from the actual plant dynamics in an additive fashion:

$$\begin{aligned}\hat{f}(\varsigma) &= f(\varsigma) + \delta_f(\varsigma)\\ \hat{m}(\varsigma) &= m(\varsigma) + \delta_m(\varsigma).\end{aligned} \tag{7.5}$$

So we can rewrite (7.4) as:

$$\begin{aligned}\dot{x} &= f(x) + m(\hat{x})\\ \dot{\hat{x}} &= f(\hat{x}) + m(\hat{x}) + \delta_f(\hat{x}) + \delta_m(\hat{x}) = f(\hat{x}) + m(\hat{x}) + \delta(\hat{x}).\end{aligned} \tag{7.6}$$

We will now assume that f and δ satisfy the following local Lipschitz conditions for $x, y \in B_L$ with B_L a ball centered on the origin:

$$\begin{aligned}\|f(x) - f(y)\| &\leq K_f\|x - y\|\\ \|\delta(x) - \delta(y)\| &\leq K_\delta\|x - y\|.\end{aligned} \tag{7.7}$$

At this point it is to be noted that if the plant model is accurate the Lipschitz constant K_δ will be small.

We will assume that the non-networked compensated plant model is exponentially stable (see Definition 2.2), that is, the controller (7.3) is designed based on the available information as provided by the nominal model dynamics and it is also designed in such a way that the closed-loop nominal model, second equation in (7.4), is exponentially stable when $\hat{x}(t_0) \in B_S$, with $\hat{x}(t) \in B_h$ for $t \in [t_0, t_0 + h)$ with B_S and B_h balls centered on the origin.

$$\|\hat{x}(t)\| \leq \alpha\|\hat{x}(t_0)\|e^{-\beta(t-t_0)}, \text{ with } \alpha, \beta > 0. \tag{7.8}$$

Theorem 7.1 *The nonlinear MB-NCS with dynamics described by (7.1), (7.2), and (7.3) that satisfies the Lipschitz conditions described by (7.7) and with exponentially stable compensated plant model satisfying (7.8) is asymptotically stable if:*

$$\left(1 - \alpha\left(e^{-\beta h} + \left(e^{K_f h} - e^{-\beta h}\right)\left(\frac{K_\delta}{K_f + \beta}\right)\right)\right) > 0. \tag{7.9}$$

Proof We will now analyze the behavior of the plant state norm in between updates. The stability of the system can be guaranteed if $\|x(t)\|$ decreases such that $\|x(t_k)\| > \|x(t_{k+1}^-)\|$, where t_k and t_{k+1} are update times with $t_{k+1} - t_k = h$. Figure 7.1 recreates two possible scenarios. In both of them it is assumed that the closed-loop model is exponentially stable. The difference is that for larger transmission intervals we expect, in general, a growth on the norm of the state that could lead to an unstable response as shown in Fig. 7.1b. Although the closed-loop model

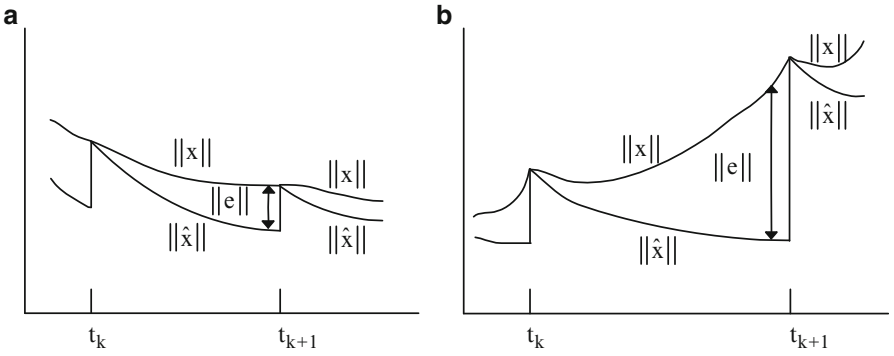

Fig. 7.1 Two different MB-NCS responses according to different update intervals h. (**a**) A plant model stable response. (**b**) A plant model unstable response

is stable the measurement updates make the model unstable when it is used in the MB-NCS setup due to the long time intervals that the system operates without sending measurements to update the model. For a given model, system, and uncertainty parameters, which in this case take the form of the closed-loop model parameters α and β and the system and uncertainty Lipschitz constants K_f and K_δ, we aim to find an estimate of the range of h that results in a stable model-based network interconnected system.

In general we see that in any period of time $[t_k, t_{k+1})$ the following relations hold:

$$\begin{aligned} \|x\| &= \|\hat{x} + e\| \leq \|\hat{x}\| + \|e\|, \\ \|e(t_k)\| &= 0, \\ \|x(t_k)\| &= \|\hat{x}(t_k)\|. \end{aligned} \tag{7.10}$$

So we can guarantee that the norm of the plant state $\|x\|$ will decrease over the period $[t_k,t_{k+1})$ if $\|\hat{x}\| + \|e\|$ decrease. We will now establish bounds over the norm of the error as a function of the update time h. We know that:

$$\dot{e} = \dot{x} - \dot{\hat{x}} = f(x) - f(\hat{x}) - \delta(\hat{x}). \tag{7.11}$$

Therefore:

$$\begin{aligned} e(t) &= e(t_k) + \int_{t_k}^{t} (f(x(s)) - f(\hat{x}(s)) - \delta(\hat{x}(s)))ds \\ &= \int_{t_k}^{t} (f(x(s)) - f(\hat{x}(s)) - \delta(\hat{x}(s)))ds, \ \forall t \in [t_k, t_{k+1}). \end{aligned} \tag{7.12}$$

The last equality holds since at t_k the plant model state is updated and the error is zeroed. We will now use the Lipschitz condition of the functions involved in (7.12) to place a bound on the norm of the error.

$$\begin{aligned}\|e(t)\| &\leq \int_{t_k}^{t} \left(\left\|f(x(s)) - \hat{f}\left(\hat{x}\left(s\right)\right)\right\| + \|\delta(\hat{x}\left(s\right))\|\right) ds \\ &\leq \int_{t_k}^{t} \left(K_f \|x(s) - \hat{x}\left(s\right)\| + K_\delta \|\hat{x}\left(s\right)\|\right) ds \\ &= K_f \int_{t_k}^{t} \|x(s) - \hat{x}\left(s\right)\| ds + K_\delta \int_{t_k}^{t} \|\hat{x}\left(s\right)\| ds \\ &= K_f \int_{t_k}^{t} \|e(s)\| ds + K_\delta \int_{t_k}^{t} \|\hat{x}\left(s\right)\| ds, \ \ \forall t \in \left[t_k, t_{k+1}\right).\end{aligned} \tag{7.13}$$

From (7.8) we have that:

$$\begin{aligned}\|e(t)\| &\leq K_f \int_{t_k}^{t} \|e(s)\| ds + K_\delta \int_{t_k}^{t} \|\hat{x}\left(s\right)\| ds \\ &= K_\delta \int_{t_k}^{t} \alpha \|\hat{x}\left(t_k\right)\| e^{-\beta(s-t_k)} ds + K_f \int_{t_k}^{t} \|e(s)\| ds \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)}\right) + K_f \int_{t_k}^{t} \|e(s)\| ds, \ \ \forall t \in \left[t_k, t_{k+1}\right).\end{aligned} \tag{7.14}$$

The Gronwall-Bellman Inequality [131] states that if a continuous real-valued function $y(t)$ satisfies: $y(t) \leq \lambda(t) + \int_a^t \mu(s) y(s) ds$ with $\lambda(t)$ and $\mu(t)$ continuous real-valued functions and $\mu(t)$ non-negative for $t \in [a, b]$, then: $y(t) \leq \lambda(t) + \int_a^t \lambda(s)\mu(s) e^{\int_s^t \mu(\tau) d\tau} ds$ over the same interval. So by assigning $y(t) = \|e(t)\|$, $\lambda(t) = K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)}\right)$, and $\mu(t) = K_f$ we obtain:

$$\begin{aligned}\|e(t)\| &\leq K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)}\right) + \int_{t_k}^{t} K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(s-t_k)}\right) K_f e^{K_f(t-s)} ds \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)} + \int_{t_k}^{t} \left(1 - e^{-\beta(s-t_k)}\right) K_f e^{K_f(t-s)} ds\right) \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)} + K_f \int_{t_k}^{t} \left(e^{K_f(t-s)} - e^{K_f(t-s)} e^{-\beta(s-t_k)}\right) ds\right) \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)} + K_f \int_{t_k}^{t} \left(e^{K_f(t-s)} - e^{K_f t - K_f s - \beta s + \beta t_k}\right) ds\right) \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)} + K_f \left(\frac{-1}{K_f}\left(1 - e^{K_f(t-t_k)}\right) + \frac{1}{K_f + \beta}\left(e^{-\beta(t-t_k)} - e^{K_f(t-t_k)}\right)\right)\right) \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(1 - e^{-\beta(t-t_k)} - 1 + e^{K_f(t-t_k)} + \frac{K_f}{K_f + \beta}\left(e^{-\beta(t-t_k)} - e^{K_f(t-t_k)}\right)\right) \\ &= K_\delta \frac{\alpha}{\beta} \|\hat{x}\left(t_k\right)\| \left(e^{K_f(t-t_k)} - e^{-\beta(t-t_k)}\right) \left(1 - \frac{K_f}{K_f + \beta}\right) \\ &= K_\delta \|\hat{x}\left(t_k\right)\| \left(e^{K_f(t-t_k)} - e^{-\beta(t-t_k)}\right) \left(\frac{\alpha}{K_f + \beta}\right), \ \ \forall t \in \left[t_k, t_{k+1}\right).\end{aligned} \tag{7.15}$$

From (7.15) we note that the error signal will be 0 if the update time $h = t_{k+1}$-t_k is 0 and also if the plant model and the plant dynamics are the same. With this bound over the system's error signal we can proceed to calculate the bound over the plant state.

$$\begin{aligned}\|x(t)\| &\leq \|\hat{x}(t)\| + \|e(t)\| \\ &\leq \alpha\|\hat{x}(t_k)\|e^{-\beta(t-t_k)} + K_\delta\|\hat{x}(t_k)\|\left(e^{K_f(t-t_k)} - e^{-\beta(t-t_k)}\right)\left(\frac{\alpha}{K_f+\beta}\right) \\ &= \alpha\|\hat{x}(t_k)\|\left(e^{-\beta(t-t_k)} + \left(e^{K_f(t-t_k)} - e^{-\beta(t-t_k)}\right)\left(\frac{K_\delta}{K_f+\beta}\right)\right), \quad \forall t \in [t_k, t_{k+1}).\end{aligned} \tag{7.16}$$

In order to ensure stability of the system we need that $\|x(t_k)\| > \|x(t_{k+1}^-)\|$, therefore we require:

$$\begin{aligned}&\|x(t_k)\| - \alpha\|\hat{x}(t_k)\|\left(e^{-\beta h} + \left(e^{K_f h} - e^{-\beta h}\right)\left(\frac{K_\delta}{K_f+\beta}\right)\right) > 0 \\ &\|x(t_k)\|\left(1 - \alpha\left(e^{-\beta h} + \left(e^{K_f h} - e^{-\beta h}\right)\left(\frac{K_\delta}{K_f+\beta}\right)\right)\right) > 0 \\ &\left(1 - \alpha\left(e^{-\beta h} + \left(e^{K_f h} - e^{-\beta h}\right)\left(\frac{K_\delta}{K_f+\beta}\right)\right)\right) > 0.\end{aligned} \tag{7.17}$$

◆

Example 7.1 We now present an example of an inverted pendulum in Fig. 7.2. The inverted pendulum has length L, mass m, friction coefficient k, and is driven by a torque τ.

The state-space dynamics for the inverted pendulum are:

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-x_2 \\ -\frac{g}{L}\sin(x_1) - \frac{k}{m}x_2\end{bmatrix} + \begin{bmatrix}0 \\ \frac{1}{mL^2}\end{bmatrix}\tau. \tag{7.18}$$

The controller is a gain state feedback found by linearizing the model around the equilibrium point $x_1 = 0$ and $x_2 = 0$, where x_1 is the pendulum angle and x_2 is the angular velocity. The parameters for the plant are given by: $g = 10$, $L = 9.98$, $k = 0.099$, $m = 1.03$. The available model parameters are $\hat{g} = 10$, $\hat{L} = 10$, $\hat{k} = 0.1$, $\hat{m} = 1$. The controller is given by

$$\tau = K\hat{x}, \text{with } K = [316 \quad -316]. \tag{7.19}$$

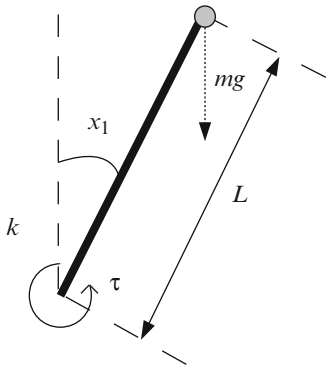

Fig. 7.2 Inverted pendulum

To satisfy the conditions on the plant, on the error, and on the exponential stability of the controlled model we use the following parameters: $K_f = 1.0502$, $K_\delta = 0.1382, \alpha = 1.1$, and $\beta = 0.5$. In Fig. 7.3 we plot the condition given by (7.9) as a function of the update time.

Note that the system stability can be guaranteed for update times approximately between 0.29 and 1.66 s which has been highlighted in Fig. 7.3. Figure 7.4 shows plots of the pendulum and model angle for some update times. The plant is initialized at $x_1 = 0.025, x_2 = 0$ while the model is initialized to 0.

7.1.2 *Stability for a More General Class of Nonlinear MB-NCS*

In this subsection we extend our results to consider MB-NCS with nonlinear systems that are represented as follows:

$$\dot{x} = f(x) + g(x, u). \tag{7.20}$$

The nonlinear model and the controller are given by:

$$\begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x}) + \hat{g}(\hat{x}, u) \\ u &= k(\hat{x}). \end{aligned} \tag{7.21}$$

Inserting the controller into the plant and model equations we get:

$$\begin{aligned} \dot{x} &= f(x) + g(x, k(\hat{x})) = f(x) + m(x, \hat{x}) \\ \dot{\hat{x}} &= \hat{f}(\hat{x}) + \hat{g}(\hat{x}, k(\hat{x})) = \hat{f}(\hat{x}) + \hat{m}(\hat{x}, \hat{x}). \end{aligned} \tag{7.22}$$

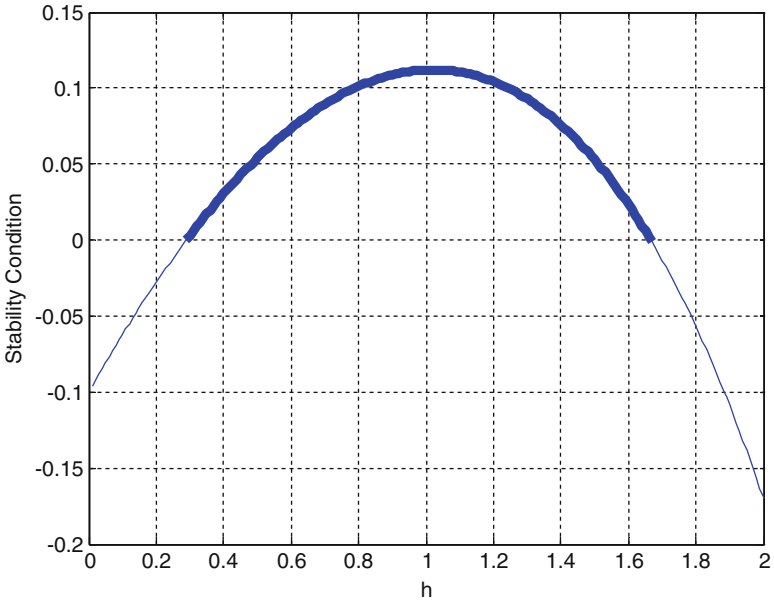

Fig. 7.3 Stability condition versus update time

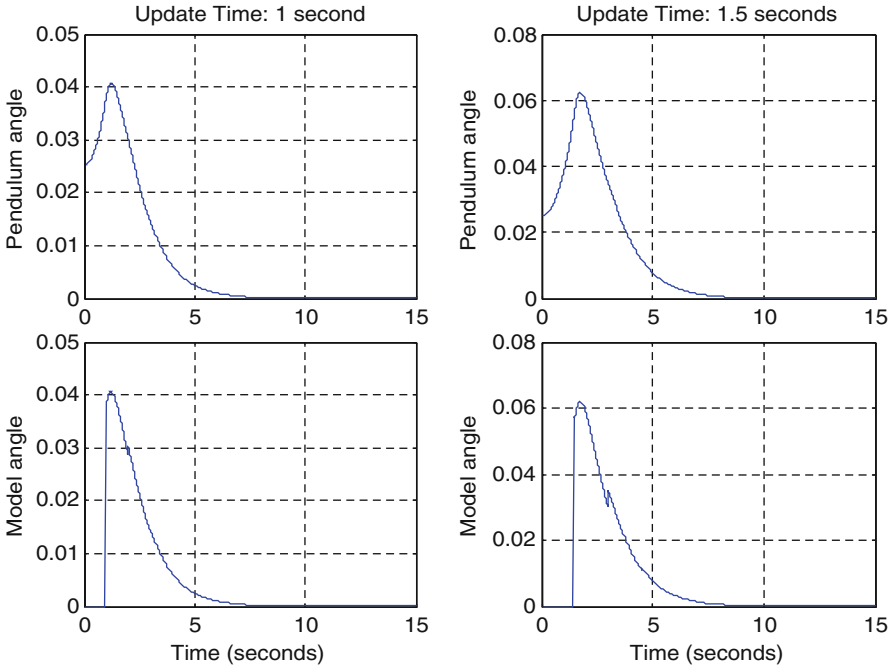

Fig. 7.4 Plant and model state for different update times

We again assume that the uncertainty between the model and the plant is given in an additive fashion as shown below:

$$\begin{aligned}\hat{f}(\varsigma) &= f(\varsigma) + \delta_f(\varsigma)\\ \hat{m}(\varsigma,\varsigma) &= m(\varsigma,\varsigma) + \delta_m(\varsigma).\end{aligned} \tag{7.23}$$

Then the error dynamics between the plant and the model can be expressed as:

$$\dot{e} = f(x) - f(\hat{x}) - \delta_f(\hat{x}) + m(x,\hat{x}) - m(\hat{x},\hat{x}) - \delta_m(\hat{x}) \tag{7.24}$$

and we assume that the following Lipschitz conditions hold:

$$\begin{aligned}&\|f(x) - f(y)\| \le K_f\|x-y\|\\ &\|m(x,s) - m(y,s)\| \le K_m(s)\|x-y\|\\ &\left\|\delta_f(x) - \delta_f(y)\right\| \le K_{\delta_f}\|x-y\|\\ &\|\delta_m(x) - \delta_m(y)\| \le K_{\delta_m}\|x-y\|.\end{aligned} \tag{7.25}$$

Define also $K_{m,\max} = \max_{s\in B_S}(K_m(s))$ for B_S a ball centered in the origin. Finally assume that the compensated system is exponentially stable when $\hat{x}(t_0) \in B_S$, with $\hat{x}(t) \in B_h$ for $t \in [t_0, t_0+h)$ with B_S and B_h balls centered on the origin.

$$\|\hat{x}(t)\| \le \alpha\|\hat{x}(t_0)\|e^{-\beta(t-t_0)}, \text{with } \alpha,\beta > 0. \tag{7.26}$$

We now state a sufficient condition for stability:

Theorem 7.2 *The nonlinear MB-NCS with dynamics described by (7.20), and (7.21) that satisfies the Lipschitz conditions described by (7.25) and with exponentially stable compensated plant model satisfying (7.26) is asymptotically stable if:*

$$\left(1 - \alpha\left(e^{-\beta h} + \left(e^{(K_f+K_{m,\max})h} - e^{-\beta h}\right)\left(\frac{K_{\delta_f}+K_{\delta_m}}{K_f + K_{m,\max}+\beta}\right)\right)\right) > 0. \tag{7.27}$$

Proof Note the error can be bounded as follows:

$$\|e(t)\| \le \int_{t_k}^{t}\left(\left(K_f + K_{m,\max}\right)\|x(s) - \hat{x}(s)\| + \left(K_{\delta_f} + K_{\delta_m}\right)\|\hat{x}(s)\|\right)ds,\ \forall t \in \left[t_k, t_{k+1}\right). \tag{7.28}$$

The rest of the proof is along the same lines as the proof of Theorem 7.1. ♦

Note that this theorem may yield similarly conservative results as those obtained with Theorem 7.1.

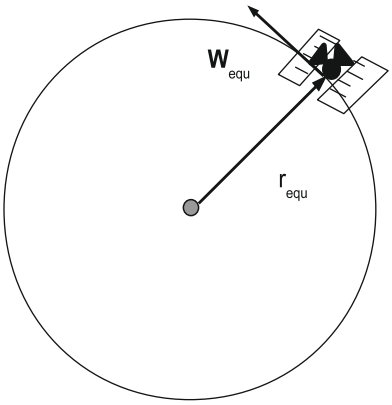

Fig. 7.5 Satellite orbiting planet

Example 7.2 We now consider the control of a satellite orbiting a planet [19]. The satellite has unit mass and the gravitational force is inversely proportional to the square of the distance between the planet and the satellite. The gravitational force constant is g. The satellite can thrust in the radial and tangential directions with thrusts u_1 and u_2, respectively. See Fig. 7.5.

The equations that govern this system are given by:

$$\begin{aligned} \ddot{r} &= \dot{r}\dot{\theta}^2 - \frac{g}{r^2} + u_1 \\ \ddot{\theta} &= \frac{-2\dot{\theta}\dot{r}}{r} + \frac{1}{r}u_2. \end{aligned} \tag{7.29}$$

If we let $x_1 = r$, $x_2 = \dot{r}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$, then the motion equations (7.29) become:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 x_4^2 - \frac{g}{x_1^2} + u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{2x_2 x_4}{x_1} + \frac{u_2}{x_1}. \end{aligned} \tag{7.30}$$

It is desired that the satellite orbit around the planet at a radius $r_{equ} = 1$. The satellite will orbit at this radius with $u_1 = u_2 = 0$ if $\dot{\theta} = \omega_{equ} = \left(g/r_{equ}^3\right)^{1/2}$.

To obtain a suitable controller we first linearize equation (7.30) about the solution:

$$x = [\, r_{equ} \quad 0 \quad \omega_{equ} t + \theta_0 \quad \omega_{equ} \,]^T, \;\; u = 0 \tag{7.31}$$

to obtain:

$$\dot{z} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega_{equ}^2 & 0 & 0 & 2r_{equ}\omega_{equ} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{-2\omega_{equ}}{r_{equ}} & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & \dfrac{1}{r_{equ}} \end{bmatrix} v. \tag{7.32}$$

We obtain a controller by solving a linear quadratic regulator problem with the linearized plant (7.32), state penalty matrix $Q = diag(10, 1, 1, 10)$, and control penalty matrix $R = I$. The obtained linear state feedback controller is given by:

$$K = \begin{bmatrix} 6.1618 & 3.4648 & -0.6884 & 0.3063 \\ 3.0005 & 0.3063 & 0.7254 & 3.5471 \end{bmatrix}. \tag{7.33}$$

The plant model considers uncertainties in the gravitational constant. We will study the stability of the satellite control system with different values of gravitational constant. We first transform the plant model dynamics (7.30) so to have the desired equilibrium at the origin. We obtain:

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \left(y_1 + r_{equ}\right)\left(y_4 + \omega_{equ}\right)^2 - \frac{g}{\left(y_1 + r_{equ}\right)^2} + w_1 \\ \dot{y}_3 &= y_4 \\ \dot{y}_4 &= -\frac{2y_2\left(y_4 + \omega_{equ}\right)}{\left(y_1 + r_{equ}\right)} + \frac{w_2}{\left(y_1 + r_{equ}\right)}. \end{aligned} \tag{7.34}$$

The Lipschitz constants are found over a linear transformation of (7.34) given by:

$$y = \begin{bmatrix} 0 & 4 & -1.5 & -1.5 \\ 0 & 0 & -1.5 & 1.5 \\ -6 & 0 & 1.5 & -1.5 \\ 0 & -6 & 3 & 3 \end{bmatrix} s. \tag{7.35}$$

This linear transformation was found by making the A matrix of equation (7.32) block diagonal, thus reducing the values of K_f and α. The resulting parameters found for a small neighborhood around the origin $\|y\| \leq 0.0001$ are the following:

$$\begin{aligned} K_f &= 3.6503 \quad \alpha = 1.5 \\ K_{\delta_f} &= 0.3155 \quad \beta = 0.9 \\ K_{m,\max} &= 0.0010 \\ K_{\delta_m} &= 0 \end{aligned} \tag{7.36}$$

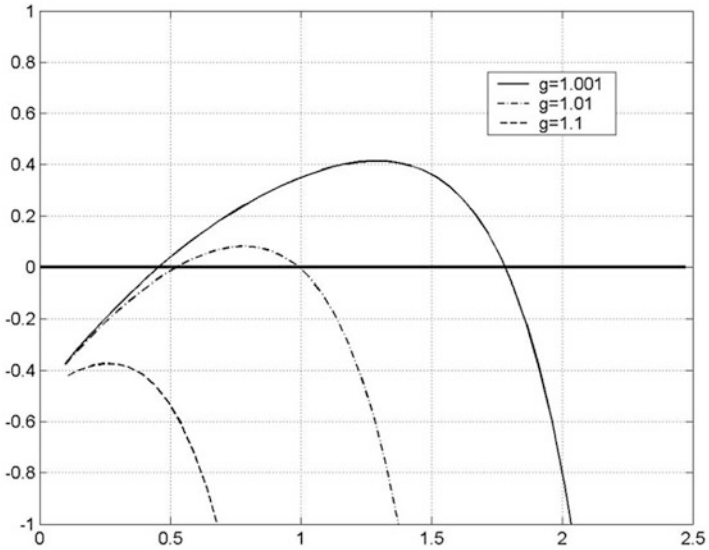

Fig. 7.6 Stability condition for satellite with MB-NCS under different gravitational force constants

Figure 7.6 shows the plot of the stability condition in Theorem 7.2 for different values of gravitational force constants, namely we considered $g = 1.001$, $g = 1.01$, and $g = 1.1$.

We note that the MB-NCS for the orbiting satellite is stable for $g = 1.001$ for update times up to 1.75 time units. For $g = 1.01$, stability is guaranteed for update times up to 1 s, while for $g = 1.1$ no stability is guaranteed at all. As a matter of fact the satellite control system with $g = 1.1$ is stable for $h = 1$ but is unstable for $h = 2$ as shown in Figs. 7.7 and 7.8.

7.2 Dissipative Nonlinear Discrete-Time Systems

In contrast to previous work in MB-NCS, the work in this section does not assume that the entire state vector is available for measurement but only the output of the system. Instead of implementing a state estimator using uncertain parameters as in Sect. 3.1 we use a model of the input-output dynamics of the system which can be updated using the plant output measurements directly.

In this section we consider Single-Input Single-Output (SISO) uncertain and possibly unstable nonlinear discrete-time systems that can be described by:

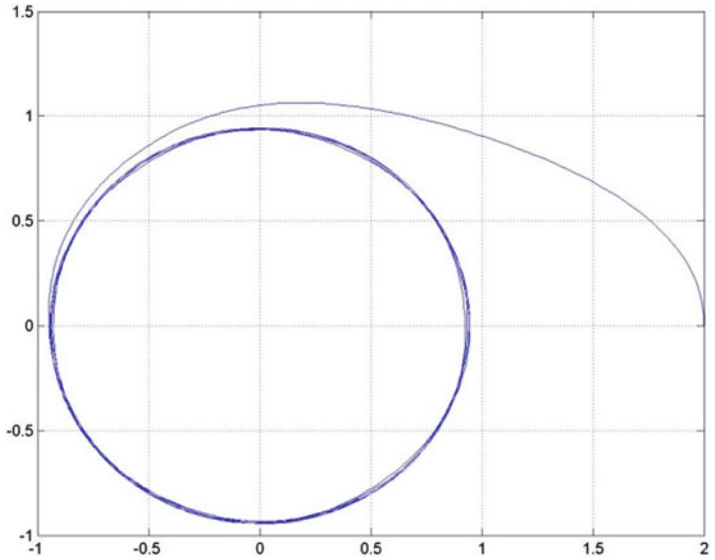

Fig. 7.7 Satellite trajectory with MB-NCS with $g = 1.1$ and update time of 1 s

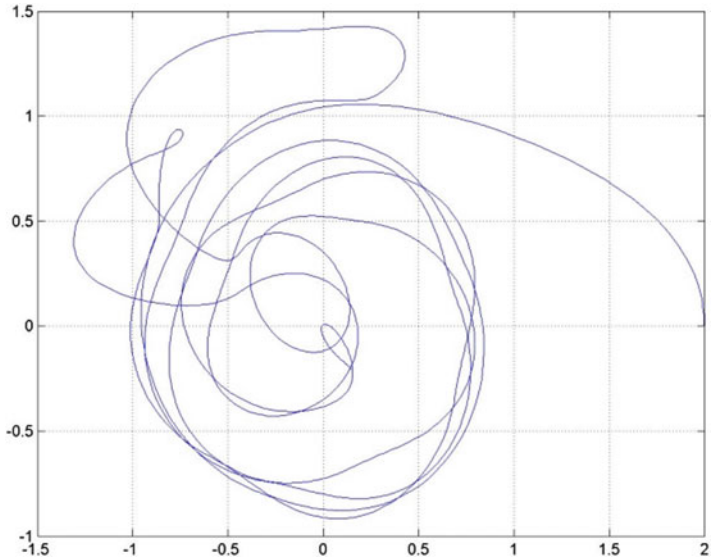

Fig. 7.8 Satellite trajectory with MB-NCS with $g = 1.1$ and update time of 2 s

$$y(k) = f_{io}(y(k-1), \ldots, y(k-n), u(k), \ldots, u(k-m)). \tag{7.37}$$

The dynamics of the model are given by:

$$\hat{y}(k) = \hat{f}_{io}(\hat{y}(k-1), \ldots, \hat{y}(k-n), u(k), \ldots, u(k-m)) \tag{7.38}$$

where the nonlinear function $\hat{f}_{io}(\bullet)$ represents the available model of the system function $f_{io}(\bullet)$.

The aim using this configuration is to operate in open-loop mode for as long as possible while maintaining desirable boundedness properties. This is done by using the estimated outputs $\hat{y}(k), \ldots, \hat{y}(k-n)$ provided by the model to generate the control input u. The system output measurements are used directly to update the current and past output variables of the model without need of implementing a state observer.

The work in this section discards the periodicity assumption for updating the model. Instead, a non-periodic approach is used that is based on events as in Chap. 6. The estimate of the output given by the model of the plant is compared with the actual output. The sensor then transmits the current output of the plant and previous n output measurements in a single packet if the error is above some predefined tolerance. These measurements are used to update the internal variables of the model in the controller. At the same time the sensor uses exactly the same measurements to update its own copy of the model. The sensor contains a copy of the model and the controller so it can have access to the model output. It continuously measures the actual output and computes the model-plant output error, defined by:

$$e(k) = \hat{y}(k) - y(k). \tag{7.39}$$

The sensor also compares the norm of the error to a predefined threshold α, and it broadcasts the plant output to update the model state if the error is greater than the threshold. It is clear that while $\|e\| \leq \alpha$ the plant is operating in open-loop mode based on the model outputs.

7.2.1 Dissipative System Theory

The approach for nonlinear MB-NCS used in this section relies on dissipativity theory. Dissipativity is an energy-based property of dynamical systems. This property relates energy stored in a system to the energy supplied to the system. Dissipativity can be seen as an extension of Lyapunov stability theory to systems with an input-output representation. The energy stored in the system is defined by a positive definite energy storage function. The energy supplied to the system is a function of the system input u and output y. A system can be considered dissipative if it only stores and dissipates energy with respect to the specific energy supply rate and does not generate energy on its own. Let us consider nonlinear discrete-time systems represented by:

$$\begin{aligned} x(k+1) &= f\big(x(k), u(k)\big) \\ y(k) &= h\big(x(k), u(k)\big). \end{aligned} \tag{7.40}$$

Definition 7.3 Consider a nonlinear discrete-time system in the form (7.40). This system is dissipative with respect to the energy supply rate *ω(y,u)* if there exists a positive definite energy storage function *V(x)* such that the following inequality holds, for all times k_1 and k_2 such that $k_1 \leq k_2$,

$$\sum_{k=k_1}^{k_2} \omega(y,u) \geq V(x(k_2)) - V(x(k_1)). \tag{7.41}$$

It is useful to restrict the supply rate to be quadratic. This is the case in QSR dissipativity.

Definition 7.4 A nonlinear discrete-time system (7.40) is QSR dissipative if it is dissipative with respect to the supply rate

$$\omega(y,u) = \begin{bmatrix} y \\ u \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} \tag{7.42}$$

where $Q = Q^T$ and $R = R^T$.

The QSR dissipative framework generalizes many well-known areas of nonlinear systems analysis. The property of passivity can be captured when $Q = R = 0$ and $S = 1/2I$, where I is the identity matrix. Systems that are finite-gain l_2 stable can be represented by $S = 0$, $Q = 1/\gamma I$, and $R = \gamma I$ where γ is the gain of the system. The following theorems give stability results for single QSR dissipative systems as well as feedback interconnections of dissipative systems.

Theorem 7.5 *A discrete-time system is finite-gain l_2 stable if it is QSR dissipative with $Q < 0$.*

Theorem 7.6 *Consider the feedback interconnection of two QSR dissipative nonlinear systems. System G_1 is dissipative with respect to Q_1, S_1, R_1 and system G_2 is dissipative with respect to Q_2, S_2, R_2. The feedback interconnection of these two systems is l_2 stable if there exists a positive constant a such that the following matrix is negative definite*,

$$\begin{bmatrix} Q_1 + aR_2 & aS_2^T - S_1 \\ aS_2 - S_1^T & R_1 + aQ_2 \end{bmatrix} < 0. \tag{7.43}$$

While this section considers dissipativity for general nonlinear systems, the LTI case is important in many applications. Let a discrete-time LTI system be given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bx(k) \\ y(k) &= Cx(k) + Dx(k). \end{aligned} \tag{7.44}$$

An LTI system is dissipative if and only if there exists a quadratic storage function,

$$V(x) = x^T P x \tag{7.45}$$

where $P = P^T > 0$, to satisfy the dissipativity inequality. The matrix P can be found computationally using linear matrix inequality (LMI) methods. An optimization problem can be formulated to find a positive definite matrix P. The problem is feasible when P can be found to satisfy the following matrix inequality,

$$\begin{bmatrix} A^T PA - P - C^T QC & A^T PB - C^T QD - C^T S \\ B^T PA - D^T QC - S^T C & B^T PB - D^T QD - D^T S - S^T D - R \end{bmatrix} \leq 0. \tag{7.46}$$

For general nonlinear systems, there does not exist a computational method of finding storage functions or proving dissipativity.

7.2.2 *Bounded Stability of Output Feedback Dissipative Systems*

The first goal in this section is to reconcile the two models discussed up to this point. The models used are the nonlinear input-output representation of a system and the nonlinear state-space model. Then the notion of boundedness is discussed and used; boundedness is also proved under appropriate assumptions.

Proposed Equivalent Representation of a MB-NCS. The type of nonlinear systems represented by (7.37) can also be described by the state representation (7.40). A straightforward way to obtain a state-space representation of the input-output system (7.37) is as follows:

Define the state-space variables:

$$\begin{aligned} x_1(k) &= y(k-1) \\ x_2(k) &= y(k-2) \\ &\vdots \\ x_n(k) &= y(k-n) \\ x_{n+1}(k) &= u(k-1) \\ x_{n+2}(k) &= u(k-2) \\ &\vdots \\ x_{n+m}(k) &= u(k-m) \end{aligned} \tag{7.47}$$

and the state-space vector

$$x(k) = [x_1(k) \quad x_2(k) \quad \ldots \quad x_{n+m}(k)]^T. \tag{7.48}$$

Then a system of the form (7.37) can be represented as a state-space dynamical system as follows:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \\ x_{n+1}(k+1) \\ x_{n+2}(k+1) \\ \vdots \\ x_{n+m}(k+1) \end{bmatrix} = \begin{bmatrix} f_{io}(x_1(k), \ldots, x_n(k), u(k), x_{n+1}(k) \ldots, x_{n+m}(k)) \\ x_1(k) \\ \vdots \\ x_{n-1}(k) \\ u(k) \\ x_{n+1}(k) \\ \vdots \\ x_{n+m-1}(k) \end{bmatrix} \\ &= f(x(k), u(k)) \\ y(k) &= f_{io}(x_1(k), \ldots, x_n(k), u(k), x_{n+1}(k) \ldots, x_{n+m}(k)) = h(x(k), u(k)). \end{aligned} \tag{7.49}$$

Following a similar procedure a state-space representation of the model is:

$$\begin{aligned} \hat{x}(k+1) &= \hat{f}(\hat{x}(k), u(k)) \\ \hat{y}(k) &= \hat{h}(\hat{x}(k), u(k)). \end{aligned} \tag{7.50}$$

The structure of the state-space model is the same as that of the real plant. The uncertainty in the plant is only due to the output equation and to the first term in the state equation of (7.49), which is repeated in the output equation as well; those expressions contain the nonlinear uncertain dynamics given by (7.37) with respect to the nominal model (7.38).

By using the model parameters in (7.50) it is possible to use the QSR dissipative analysis described in the previous section in order to design QSR dissipative and stabilizing controllers.

Now, the Model-Based Event-Triggered (MB-ET) architecture described in Chap. 6 can be represented in the negative feedback interconnection suitable for dissipativity analysis that was shown in Fig. 7.9.

Figure 7.10 shows an equivalent representation of a MB-ET control system in which the updates of the state of the model are implicit in the model/controller block. From (7.39) it is clear that

$$\hat{y}(k) = y(k) + e(k). \tag{7.51}$$

The input to the controller is the output of the model $\hat{y}(k)$; this can be easily represented by (7.51) as in Fig. 7.10 where the output of the model is the result of the contribution of two terms: the output of the system and the output error. Although

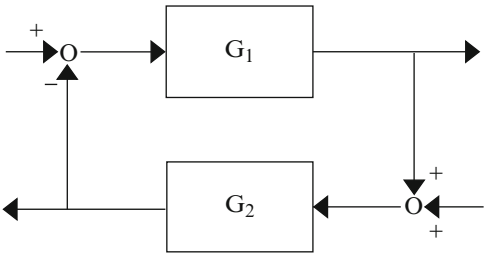

Fig. 7.9 The negative feedback interconnection of two nonlinear systems G_1 and G_2

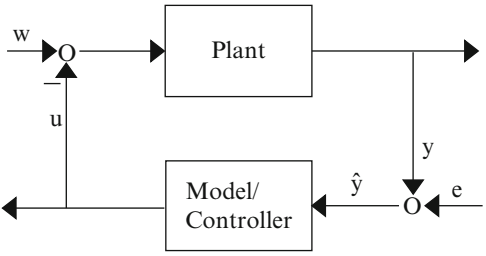

Fig. 7.10 Equivalent feedback loop for the model-based event-triggered control system used for analysis

the actual implementation of the control system is not in this form, this form is useful for the analysis and design of stabilizing controllers using dissipative techniques.

By using the MB-ET implementation the output error can be bounded by the appropriate design of a stabilizing threshold. The error (7.39) can be seen as piecewise bounded external disturbance and at the communication update instants k_i, we have that $e(k_i) = 0$ because the model and plant output variables are equal.

Stabilization of networked discrete-time systems using output feedback event-triggered measurements. This section considers discrete-time nonlinear systems and models of the form (7.37) and (7.38) that are interconnected using a model-based implementation as described above. The analysis and design of the stabilizing controller can be performed as shown in the previous section using the equivalent representation (7.49) and (7.50). Since it is only possible to measure the output of the system and not the whole state it is not possible to implement a state-space representation of the model directly. Instead, the model in both the controller and the sensor nodes is implemented as an equivalent nonlinear difference equation which represents the same input-output behavior as the state-space model. When the sensor decides that a measurement update needs to be sent according to the current output error, then it sends the current and n past output measurements which are used to update the model in the controller. At the same time the sensor uses exactly the same measurements to update its own copy of the model.

In order to make a decision as to whether or not it is necessary to send a measurement to update the model, the absolute value of the output error (7.39) is compared to a fixed positive threshold α. When the relation $\|e(k)\| > \alpha$ holds, the sensor transmits a measurement update. Assuming negligible delay the model is updated using the current and past n system output values at the same update instant k_i. At this point, the output error (7.39) is set to 0, since the model output is made equal to the real output of the system at time k. Therefore, the output error is bounded by:

$$\|e(k)\| \leq \alpha. \tag{7.52}$$

Before the main results are presented, the notion of stability must be clarified. For uncertain systems that are perturbed by an external disturbance and operate open loop for some time periods, the notion of stability must be relaxed. While asymptotic stability is appealing, it simply is not attainable. When the system runs open loop, the state may diverge due to unstable dynamics that are not known exactly. A more reasonable notion of stability is in a form of boundedness such as uniform ultimate boundedness [131]. For systems that are ultimately bounded, as time goes to infinity the state is bounded by a known constant.

Instead of internal state stability, input-output stability is considered in this section. For input-output stability, a notion such as l_2 stability must be relaxed to a bound on the output as time goes to infinity. In general, a uniform bound on the output may not be tight. The output can be a function of the internal states and the system input. Since the input may not be predictable, a uniform bound on the output may be quite large. Instead a tighter bound on the average output amplitude may be found. In this work the notion of *average output squared boundedness* is considered.

Definition 7.7 A nonlinear system is average output squared bounded if there exist a time $\overline{k}$ and a constant b such that the following bound on the output holds for all times k_1 and k_2 such that $\overline{k} \leq k_1 \leq k_2$,

$$\frac{1}{(k_2 - k_1)} \sum_{k=k_1}^{k_2-1} y^T(k) y(k) \leq b. \tag{7.53}$$

This form of boundedness is a practical form of stability on the system output. While the output does not necessarily converge to 0, it is bounded on average with a known bound as time goes to infinity. Although this concept may not be useful for an arbitrarily large bound b, the concept is very informative for a small bound. The notion should be restricted to being used in the case when the bound is constructive and preferably when the bound can be made arbitrarily small by adjusting system parameters.

For the following theorem, assume that the plant and the model of the plant are QSR dissipative with respect to parameters Q_p, S_p, and R_p. Although the plant

dynamics are not known exactly, experimental testing can be done to verify that the dissipative rate is at least a bound on the actual dissipative behavior of the system. The data taken to verify that the parameters hold to a specified level of certainty is similar data taken to identify system parameters. It is also assumed that a model-stabilizing dissipative controller has been designed such (7.43) is satisfied with Q_c, S_c, and R_c representing the QSR parameters of the controller. Next, we provide conditions under which we are able to stabilize uncertain unstable systems with limited feedback.

Theorem 7.8 *Consider the networked system (7.37) with uncertain dynamics, event-triggered updates, and model-based output error (7.39). This feedback system, from external disturbance w to output y, is average output bounded stable if there exists a positive constant a such that the following matrix* $\widetilde{Q}$ *is negative definite*,

$$\widetilde{Q} = \begin{bmatrix} Q_p + aR_c & aS_c^T - S_p \\ aS_c - S_p^T & R_p + aQ_c \end{bmatrix} < 0. \tag{7.54}$$

Proof The plant being QSR dissipative implies the existence of a positive definite storage function V_p, bounded above and below by class-$\mathcal{K}$ functions,

$$\underline{\alpha}_p(|x_p|) \leq V_p(x_p) \leq \overline{\alpha}_p(|x_p|) \tag{7.55}$$

such that the following inequality holds,

$$\Delta V_p(x_p) \leq \begin{bmatrix} y \\ u \end{bmatrix}^T \begin{bmatrix} Q_p & S_p \\ S_p^T & R_p \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix}. \tag{7.56}$$

The same applies for the controller being QSR dissipative, i.e.,

$$\underline{\alpha}_c(|x_c|) \leq V_c(x_c) \leq \overline{\alpha}_c(|x_c|) \tag{7.57}$$

and

$$\Delta V_c(x_c) \leq \begin{bmatrix} u_c \\ \hat{y} \end{bmatrix}^T \begin{bmatrix} Q_c & S_c \\ S_c{}^T & R_c \end{bmatrix} \begin{bmatrix} u_c \\ \hat{y} \end{bmatrix}. \tag{7.58}$$

A total energy storage function can be defined, $V(x) = V_p(x_p) + aV_c(x_c)$, where $x = \begin{bmatrix} x_p \\ x_c \end{bmatrix}$. The total energy storage function has the dissipative property,

$$\Delta V(x) \leq \begin{bmatrix} y \\ u_c \\ w \\ e \end{bmatrix}^T \begin{bmatrix} \widetilde{Q} & \widetilde{S} \\ \widetilde{S}^T & \widetilde{R} \end{bmatrix} \begin{bmatrix} y \\ u_c \\ w \\ e \end{bmatrix}. \tag{7.59}$$

where

$$\widetilde{Q} = \begin{bmatrix} Q_p + aR_c & aS_c{}^T - S_p \\ aS_c - S_p{}^T & R_p + aQ_c \end{bmatrix}, \widetilde{S} = \begin{bmatrix} S_p & aR_c \\ -R_p & aS_c \end{bmatrix}, \widetilde{R} = \begin{bmatrix} R_p & 0 \\ 0 & aR_c \end{bmatrix}. \tag{7.60}$$

By assumption, $\widetilde{Q}$ is negative definite and can be bounded above by a constant q, $\widetilde{Q} \leq -qI$. The other two matrices can be bounded from above, $\widetilde{S} \leq sI$ and $\widetilde{R} \leq rI$. This yields the following bound on ΔV

$$\Delta V(x) \leq -q\left[y^T y + u_c{}^T u_c\right] + 2s\left[y^T w + u_c{}^T e\right] + r\left[w^T w + e^T e\right]. \tag{7.61}$$

A completing the square approach can be applied to remove the cross term leaving the following bound

$$\Delta V(x) \leq -\frac{q}{2}\left[y^T y + u_c{}^T u_c\right] + \frac{(4s^2 + 2qr)}{2q}\left[w^T w + e^T e\right]. \tag{7.62}$$

Summing this inequality over a time interval from k_1 to k_2 yields the following evolution of the storage function.

$$V(x(k_2)) \leq V(x(k_1)) - \frac{q}{2}\sum_{k=k_1}^{k_2-1}\left(y^T y + u_c{}^T u_c\right) + \frac{(4s^2 + 2qr)}{2q}\sum_{k=k_1}^{k_2-1}\left(w^T w + e^T e\right). \tag{7.63}$$

The effect of the continuous l_2 disturbance w can be bounded by some value $\varepsilon_w > 0$ after some time $\overline{k}$,

$$\sqrt{\frac{4s^2 + 2qr}{2q}}|w(k)| \leq \varepsilon_w. \tag{7.64}$$

Using the previous two equations, the following bound on the squared output can be found

$$\sum_{k=k_1}^{k_2-1} y^T y \leq \frac{2}{q}\left[V(x(k_1)) + \varepsilon_w^2\right] + \frac{(4s^2 + 2qr)}{q^2}\sum_{k=k_1}^{k_2-1} e^T e. \tag{7.65}$$

This bound is made up of two quantities that are constant and the final summation that may increase with time.

The rest of this proof is done by studying two cases. Fixing the time $k_1 \geq \overline{k}$ for each time k_2, one of the following is true.

First, in the time range of k_1 to k_2 and for δ in the range $0 < \delta < 1$, the squared output is on average bounded by the following expression,

$$\frac{1}{(k_2 - k_1)} \sum_{k=k_1}^{k_2-1} y^T y \leq \frac{(4s^2 + 2qr)\alpha^2}{q^2(1-\delta)}. \tag{7.66}$$

Second, if the previous bound does not hold, the following holds:

$$\frac{1}{(k_2 - k_1)} \sum_{k=k_1}^{k_2-1} y^T y > \frac{(4s^2 + 2qr)\alpha^2}{q^2(1-\delta)}. \tag{7.67}$$

This quantity can be used to bound the squared error accumulated over time. It can be shown that the following bound on the squared output holds.

$$\sum_{k=k_1}^{k_2-1} y^T y \leq \frac{2}{\delta q} \left[\varepsilon_w^2 + V(x(k_1))\right]. \tag{7.68}$$

With this bound on the total of the output squared, it is clear that the average value of the squared output is bounded. Since this bound is independent of time, it is fixed for arbitrarily large k_2 and any δ. This means that this bound would imply that the average of the output squared goes to 0 as k_2 goes to infinity.

For either of the above cases, the squared output is bounded on average by a constant bound that is independent of time. Since both bounds hold, the maximum of the two is always at least a loose bound on the average of the squared output. Since the second average bound goes to 0 over time, the first bound is the more relevant one on the infinite time horizon. ♦

Remark One important aspect of the proof is that the average squared output is bounded by a constructive bound. These bounds can be made smaller by adjusting the values of control parameters. The bounds depend on q, s, and r which involve parameters of the plant that cannot be changed and parameters of the controller which may change. The first bound, the more relevant one, depends on the value of the state error threshold α. The output squared bound can be made arbitrarily small by making the error threshold smaller. Lastly, the bounds depend on δ and ε_w, which are constructed to analyze the behavior of the disturbance after time $\overline{k}$. The bounds can be made tighter by considering larger $\overline{k}$. Additionally, the parameter δ may be changed to adjust the relative magnitude of the two bounds. By picking an appropriate δ for each time k_2 the bounds may be chosen to be tighter.

Remark The selection of the constant threshold α is made by considering the following tradeoff. A small threshold results in a smaller bound on the system's average output but, in general, increases the communication rate by sending measurement updates more frequently. A reduction on network usage can be achieved by increasing the threshold at the cost of a larger average output of the system.

In the case of linear systems it is possible to estimate the set of admissible uncertain plants that can be stabilized given a model and a controller. Using the same QSR parameters Q_p, S_p, and R_p that were used for the model and assuming that the real parameters contain additive uncertainties with respect to the nominal model parameters, i.e., $A = \hat{A} + \Delta_A,\ B = \hat{B} + \Delta_B,\ C = \hat{C} + \Delta_C,\ D = \hat{D} + \Delta_D$, the following problem

$$\begin{bmatrix} X & W \\ W^T & Y \end{bmatrix} \leq 0 \tag{7.69}$$

where $X = (\hat{A} + \Delta_A)^T P(\hat{A} + \Delta_A) - P - (\hat{C} + \Delta_C)^T Q(\hat{C} + \Delta_C)$,

$$\begin{aligned} Y = {} & (\hat{B} + \Delta_B)^T P(\hat{B} + \Delta_B) - (\hat{D} + \Delta_D)^T Q(\hat{D} + \Delta_D) - (\hat{D} + \Delta_D)^T S \\ & - S^T(\hat{D} + \Delta_D) - R, \end{aligned}$$

and $W = (\hat{A} + \Delta_A)^T P(\hat{B} + \Delta_B) - (\hat{C} + \Delta_C)^T Q(\hat{D} + \Delta_D) - (\hat{C} + \Delta_C)^T S$ can be solved for P and using different values of $\Delta_A,\ \Delta_B, \Delta_C,\ \Delta_D$,

When the above problem is feasible for given choice of uncertainties $\Delta_A,\ \Delta_B$, $\Delta_C,\ \Delta_D$ then the uncertain system A, B, C, D is an element of the set of admissible uncertain plants.

Example 7.3 Consider a model of an unstable system given by:

$$\hat{A} = \begin{bmatrix} -0.81 & 0.37 \\ 0.88 & 0.21 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \hat{C} = [1 \quad 2], \quad \hat{D} = 1. \tag{7.70}$$

It can be shown that the model is QSR dissipative with respect to $Q_P = 0.5$, $S_P = 0.5$, and $R_P = 0.1$, by using the storage function:

$$\hat{V}(\hat{x}) = \hat{x}^T \begin{bmatrix} 0.8 & 0.87 \\ 0.87 & 1.28 \end{bmatrix} \hat{x}. \tag{7.71}$$

A stabilizing controller is given by

$$A_c = 0.5, \quad B_c = 0.3, \quad C_c = 1, \quad D_c = 1. \tag{7.72}$$

This controller is passive and QSR dissipative with respect to $Q_C = -0.2, S_C = 0.5$, and $R_C = -0.6$ which can be shown using the storage function $V_c(x_u) = 1.23x_u^2$.

The controller can be shown to stabilize the model by evaluating (7.43).

$$\widetilde{Q} = \begin{bmatrix} -0.1 & 0 \\ 0 & -0.1 \end{bmatrix} < 0. \tag{7.73}$$

For this example, an actual uncertain plant is given by:

$$A = \begin{bmatrix} -0.71 & 0.55 \\ 0.95 & 0.35 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0.75 \quad 2.3], \quad D = 1.1. \tag{7.74}$$

The plant is also dissipative with respect to the same choice of QSR parameters $Q_P = 0.5$, $S_P = 0.5$, and $R_P = 0.1$; this can be verified using the storage function

$$V(x) = x^T \begin{bmatrix} 1.21 & 1.13 \\ 1.13 & 1.78 \end{bmatrix} x. \tag{7.75}$$

Since the QSR parameters for the plant and the model are the same, the controller (7.72) also stabilizes the plant and satisfies the inequality (7.43) with (7.73). Simulations of the model-based networked system that is also affected by an l_2 external disturbance $w(k)$ are shown in Fig. 7.11 using a threshold value $\alpha = 0.02$. The network communication signal $n_c(k)$ in Fig. 7.12 represents the time instants at which output measurements are sent from the sensor node to the controller node. The rest of the time the networked system operates in open-loop mode.

$$n_c(k) = \begin{cases} 1 & \text{if measurements are sent at time } k \\ 0 & \text{if measurements are not sent at time } k \end{cases} \tag{7.76}$$

Figure 7.11 shows that the outputs of the model-based networked uncertain system are bounded, as expected. Although the parameters of the nominal model differ significantly from those of the real plant, it is still possible to stabilize the system. A larger reduction of network communication can be achieved by using a more accurate model that is QSR dissipative using the same choice of QSR parameters and that it also reflects more accurately the dynamics of the plant.

Example 7.4 Consider the same plant dynamics (7.74) and the following model parameters:

$$\hat{A} = \begin{bmatrix} -0.7 & 0.52 \\ 0.88 & 0.4 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \hat{C} = [0.73 \quad 2.2], \quad \hat{D} = 1.2. \tag{7.77}$$

The results of simulations for the same external disturbance using the same threshold and the new model are shown in Figs. 7.13 and 7.14.

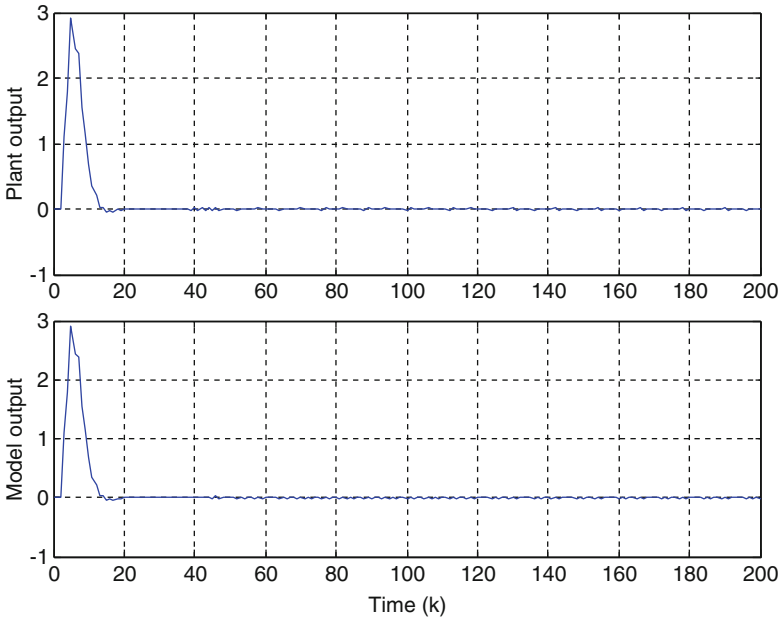

Fig. 7.11 Outputs of the plant (*top*) and the model (*bottom*) for example 1

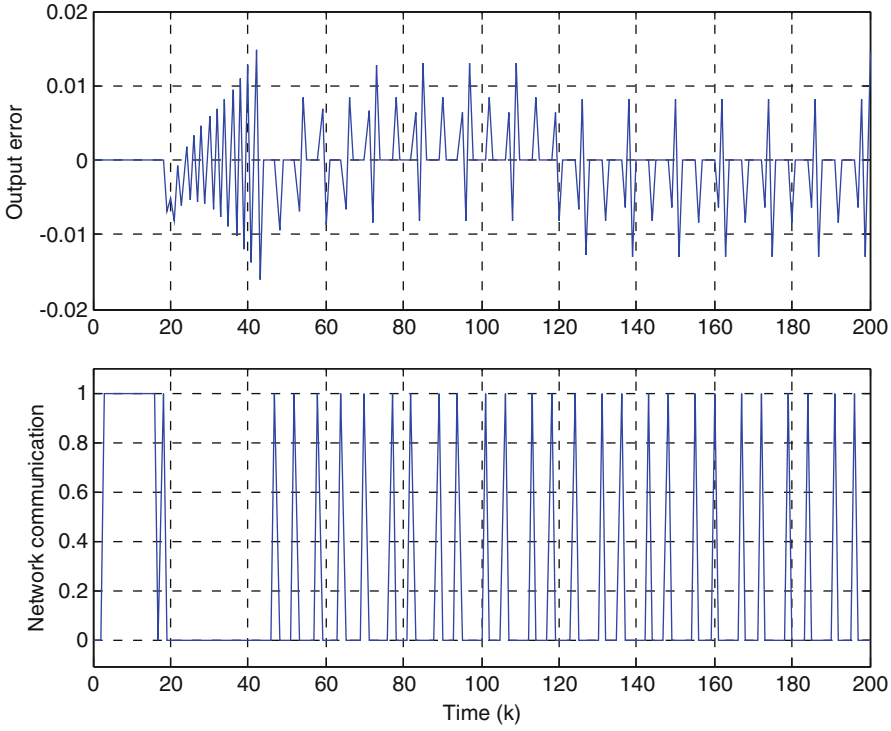

Fig. 7.12 Output error (*top*) and network communication instants for Example 7.3

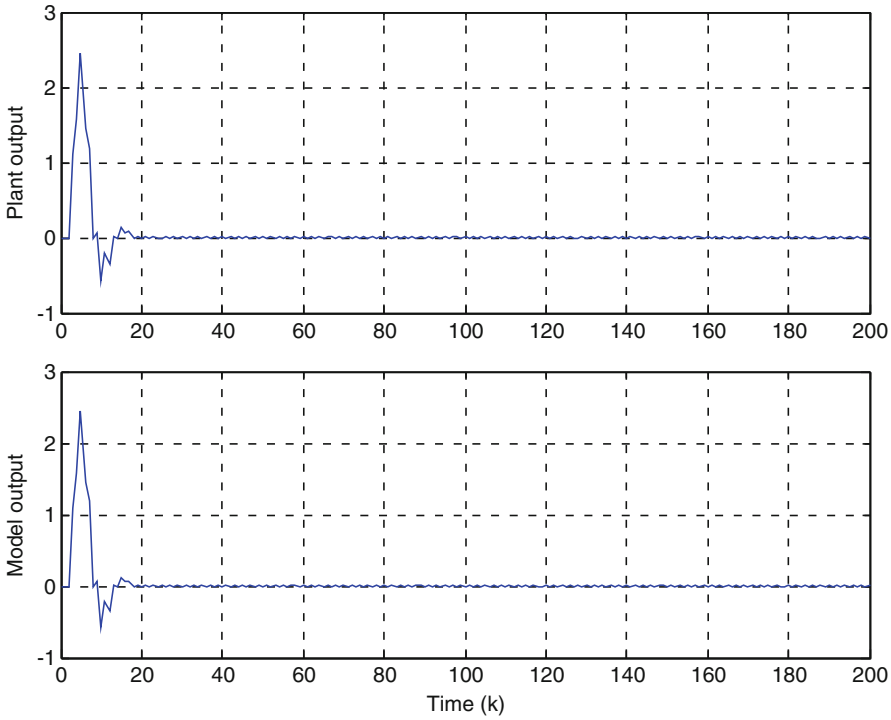

Fig. 7.13 Outputs of the plant (*top*) and the model (*bottom*) for Example 7.4

7.3 Notes and References

In this chapter, stability of networked nonlinear systems was addressed using the MB-NCS framework that reduces the rate at which systems transmit information through the network. Model uncertainties were considered as in the linear case. The first important result of this chapter was given in Theorem 7.1 where a sufficient condition for asymptotic stability of nonlinear continuous-time systems was presented. Periodic updates were used in this case.

The second result in this chapter is contained in Theorem 7.8; it provides sufficient conditions for bounded stability of nonlinear discrete-time dissipative systems. In this case event-based updates were used and the stability conditions and bounds depend mainly on the chosen error threshold instead of the update interval values. Additionally, the work in Sect. 7.2 considered the output feedback scenario and an input-output model of the system was used both for controller design and for the implementation of the MB-NCS setup.

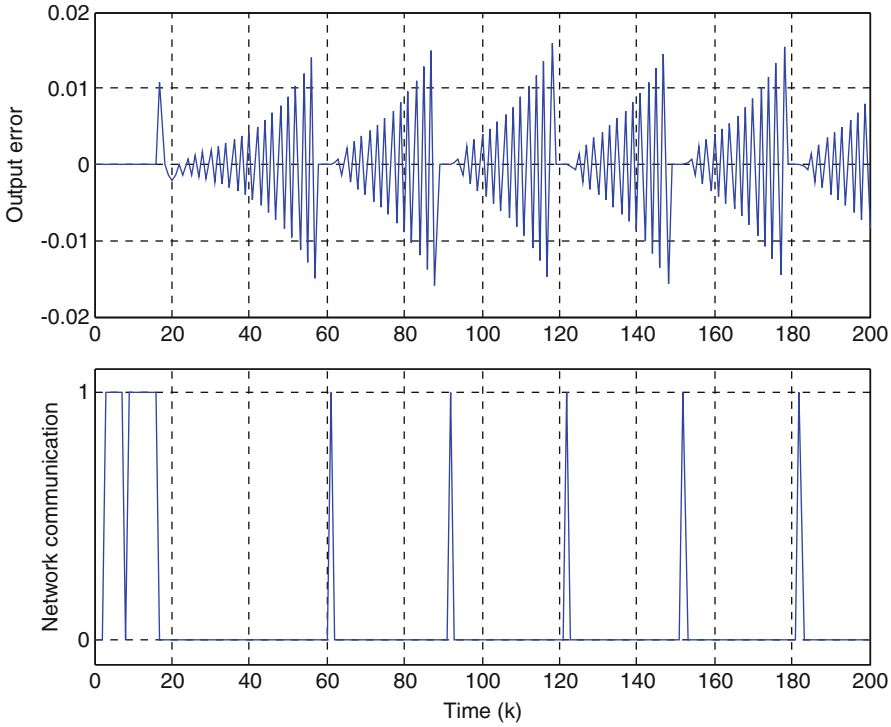

Fig. 7.14 Output error (*top*) and network communication instants for Example 7.4

The results obtained in Sect. 7.1 represent the worst-case behavior of the norm of the plant state and the error. This yields conservative results, but tighter conditions can be obtained by performing linear transformations over the plant and model dynamics. Even less conservative results can be obtained by using Lyapunov functions and stability results over jump systems such as the ones outlined in [277]. The procedure though follows closely the one presented, namely it considers the Lyapunov function as norm instead of using the Euclidean norm used here.

In Sect. 7.2 it is important to note the reduction in network traffic by using the MB-ET approach compared to the case in which a measurement of the current output $y(k)$ is sent at every sampling instant even in the case that n is large compared to the inter-update intervals. This case requires the transmission of n measurements of the output at every update instant, the current one and the past n-1 measurements. It has been shown that packet-based control [100, 212] is able to significantly reduce data transmission by more efficiently using the packet structure, that is, reduction of communication is obtained by sending packets of information using all data bits available (excluding overhead) in the structure of the packet. The work in [100, 212] focuses in the transmission of control input sequences. In Sect. 7.2 a similar approach is taken but it is applied to the transmission of output measurements. Instead of sending a single output value in one data packet at every sampling

instant, the packet structure is used more efficiently in order to include past measurements in the same packet as well. In general, it is possible to decrease network traffic by reducing the number of packets sent by the sensor node since a high percentage of bits transmitted over the network is related to the large number of bits that are used as the packets overheads.

The results in Sect. 7.1 are based on the work by Montestruque [185]. Section 7.2 also appeared in [179]. The concept of dissipativity used in Sect. 7.2 was formalized in [268, 269]. Specifically the notion of QSR dissipativity [114, 115] was used in this chapter.

Related work considering model-based approaches for control of nonlinear networked systems can be found in [48, 160, 210, 282]. Polushin et al. [210] considered a sampled version of a nonlinear continuous-time-varying system controlled by an approximate discrete-time model and proposed a communication protocol that considers network induced communication constraints such as irregularity of transfer intervals, existence of time-varying communication delays, and possibility of packet dropouts. Stability results were given that depend on the integration step parameter; that is, it is assumed that the mismatch between plant and model arises only from the approximate integration of the nonlinear dynamics and by making the parameter small (close to 0) we can recover an exact model of the system. The work by Liu [160] also provided stability results for continuous-time nonlinear systems using the MB-NCS framework for a more general class of nonlinear systems. With respect to the network properties, the author focuses on local area control networks with high data rate and considers random but bounded time delays. For vanishing perturbations this reference provides conditions for exponential stability and conditions for uniformly ultimately boundedness in the case of non-vanishing perturbations. In [282] a combined event-triggered and self-triggered framework is presented for nonlinear systems using a model-based approach. Updates from sensor to controller are generated using a self-triggered technique and updates from the controller to the actuator are generated using an event-triggered control strategy.

Chapter 8
Quantization Analysis and Design

The results provided in previous chapters assumed that the network is capable of transporting infinite precision data. For example, for the state feedback MB-NCS it is assumed that the sensor sends the exact value of the state over the network to the controller/actuator. This is of course not possible with digital networks since the length of each data packet is finite and so many bits can be used to represent the data. It was claimed that since a large portion of standard industrial networks implement a large number of bits available to represent data, the error between the quantized value and the actual value was negligible. Even when this is so, we want to study the effect of these quantization errors on system stability.

In the present chapter stability conditions for MB-NCS under popular quantization schemes are derived. The objective is to reduce the number of bits needed to transmit feedback measurements to stabilize uncertain systems. When using periodic updates within the MB-NCS framework, there are two main parameters that affect the amount of data that is being transmitted. These are: the update interval that dictates how often is necessary to update the state of the model and the quantization parameter that defines the number of quantization levels and, consequently, the number of bits needed to represent every measurement. In the present chapter a quantization parameter refers to the scalar parameters that defines the maximum quantization error, that is, the difference between the quantized (output of quantizer) and the non-quantized (input to quantizer) variables.

It is desired to increase the update intervals, as studied in previous chapters, to reduce the frequency at which we need to send feedback measurements to the controller. It is also desired to increase the quantization parameter in order to use fewer bits to represent each measurement. However, for the stability results presented in this chapter, there is a tradeoff between these two parameters, a small update interval allows us to increase the quantization parameter, and a large update interval requires (for stability) a small quantization parameter.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_8,

In the present chapter we also consider event-based updates. There are two design parameters that affect stability as well: The quantization parameter and the threshold value that is used to generate events. Similarly to the periodic update case, there is a tradeoff between these two parameters. A large threshold requires, in general, a small quantization parameter and vice versa.

The analysis method that we employ in this chapter is as follows: first a stable MB-NCS is designed using the results previously outlined and then the effect of quantization is assessed using the conditions in this section. In this way the designer has a number of parameters to select which include the packet transmission times and the number of bits used for each packet for the periodic update case, and the threshold value when using event-triggered control.

The present chapter is organized as follows: Static Quantizers are discussed in Sect. 8.1. Static quantizers have quantization schemes that do not vary with time, that is the error between the quantized value and the real value does not depend on time. Two quantizers of this type are considered: the Uniform Quantizer with a constant maximum quantization error; and the Logarithmic Quantizer with a maximum quantization error that is proportional to the norm of the quantized value. In Sect. 8.2 we recall the Model-Based Event-Triggered (MB-ET) control strategies from Chap. 6 to study the stabilization properties of model-based networked systems subject to both, quantization and network induced delays; static quantizers are used in this section as well. Finally Dynamic Quantizers are discussed in Sect. 8.3; these quantizers dynamically adjust their quantization regions to compensate for uncertainties while generating a quantization error that shrinks with time.

8.1 Systems Using Static Quantizers

In this section we address the stability analysis of a state feedback MB-NCS using a static quantizer. Static quantizers have defined quantization regions that do not change with time. They are an important class of quantizers since they are simple to implement in both hardware and software and are not computationally expensive as their dynamic counterparts. Two types of quantizers are analyzed here, namely uniform quantizers and logarithmic quantizers. Each quantizer is associated with two popular data representations. The uniform quantizer is associated with the fixed-point data representation. Indeed, fixed-point numbers have a constant maximum error regardless of how close is the actual number to the origin. Logarithmic quantizers, on the other hand, are associated with floating-point numbers; this type of quantizers allows the maximum error to decrease when the actual number is close to the origin.

8.1.1 Uniform Quantizers

We define the uniform quantizer as a function $q:\mathbb{R}^n \to \mathbb{R}^n$ with the following property:

$$\|z - q(z)\| \leq \delta, z \in \mathbb{R}^n, \delta > 0. \tag{8.1}$$

The following theorem provides bounded state stability using the quantizer (8.1) and periodic updates.

Theorem 8.1 *Assume that the networked system without quantization is stable and there exists a symmetric and positive definite matrix P that solves*

$$\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)^T h} + \Delta(h)^T\right) P \left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta(h)\right) - P = -Q_D \tag{8.2}$$

where Q_D is a symmetric and positive definite matrix and $\Delta(h) = \int_0^h \mathrm{e}^{A(h-\tau)}\left(\widetilde{A}+\widetilde{B}K\right)\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau$. Then when using the uniform quantizer defined by (8.1), the state feedback MB-NCS plant state will enter and remain in the region $\|x\| \leq R$ where

$$R = \left(\mathrm{e}^{\overline{\sigma}\left(\hat{A}+\hat{B}K\right)h} + \Delta_{\max}(h)\right) r + \left(\mathrm{e}^{\overline{\sigma}(A)h} + \Delta_{\max}(h)\right)\delta \tag{8.3}$$

with $r = \sqrt{\frac{\lambda_{\max}\left(\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)\right)\delta^2}{\lambda_{\min}(Q_D)}}$, $\Delta_{\max}(h) = \int_0^h \mathrm{e}^{\overline{\sigma}(A)(h-\tau)}\overline{\sigma}\left(\widetilde{A}+\widetilde{B}K\right)$ $\mathrm{e}^{\overline{\sigma}\left(\hat{A}+\hat{B}K\right)\tau}d\tau$, *and δ is the quantization parameter in (8.1).*

Proof The response for the error is given now by:

$$\begin{aligned} e(t) &= \mathrm{e}^{A(t-t_k)}e(t_k) + \Delta(t-t_k)\hat{x}\left(t_k^+\right) \\ &= \mathrm{e}^{A(t-t_k)}e(t_k) + \Delta(t-t_k)(x_k - e(t_k)) \\ &= \left(\mathrm{e}^{A(t-t_k)} - \Delta(t-t_k)\right)e(t_k) + \Delta(t-t_k)x_k \end{aligned} \tag{8.4}$$

where $\Delta(t-t_k) = \int_0^{t-t_k} \mathrm{e}^{A(t-t_k-\tau)}\left(\widetilde{A}+\widetilde{B}K\right)\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau$.

Note that the initial value for the error $e(t_k)$ is no longer zero as it was assumed in previous sections. Moreover, the contribution to the error due to this initial

value will grow exponentially with time and with a rate that corresponds to the uncompensated plant dynamics. So at time $t \in [t_k, t_{k+1}]$ the plant state is:

$$\begin{aligned} x(t) &= \hat{x}(t) + e(t) \\ &= \mathrm{e}^{\left(\hat{A}+\hat{B}K\right)(t-t_k)}x_k + \left(\mathrm{e}^{A(t-t_k)} - \Delta(t-t_k)\right)e(t_k) + \Delta(t-t_k)x_k. \end{aligned} \tag{8.5}$$

We can therefore evaluate the Lyapunov function along $x(t)$ at any instant in time $t \in [t_k, t_{k+1}]$. Uniformly exponential stability is obtained if we can show that the following is satisfied [277]:

$$\frac{1}{h}\left(V(x(t_{k+1})) - V(x(t_k))\right) \leq -c\left(\|x(t_k)\|^2\right), c \in \mathbb{R}^+. \tag{8.6}$$

We are interested in its value at t_{k+1}:

$$\begin{aligned} &V(x(t_{k+1})) = x(t_{k+1})^T P x(t_{k+1}) \\ &= \left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h}x_k + \left(\mathrm{e}^{Ah} - \Delta(h)\right)e_k + \Delta(h)x_k\right)^T \\ &\quad P\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h}x_k + \left(\mathrm{e}^{Ah} - \Delta(h)\right)e_k + \Delta(h)x_k\right) \\ &= {x_k}^T\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta(h)\right)^T P\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta(h)\right)x_k \\ &\quad + {e_k}^T\left(\mathrm{e}^{Ah} - \Delta(h)\right)^T P\left(\mathrm{e}^{Ah} - \Delta(h)\right)e_k \end{aligned} \tag{8.7}$$

where $h = h_k = t_{k+1} - t_k > 0, e_k = e(t_k)$.

So we obtain:

$$\begin{aligned} &V(x(t_{k+1})) - V(x(t_k)) \\ &= {x_k}^T\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta(h)\right)^T P\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta(h)\right)x_k \\ &\quad + {e_k}^T\left(\mathrm{e}^{Ah} - \Delta(h)\right)^T P\left(\mathrm{e}^{Ah} - \Delta(h)\right)e_k - {x_k}^T P x_k \\ &= {e_k}^T\left(\mathrm{e}^{Ah} - \Delta(h)\right)^T P\left(\mathrm{e}^{Ah} - \Delta(h)\right)e_k - {x_k}^T Q_D x_k. \end{aligned} \tag{8.8}$$

Note that we can compute $\mathrm{e}^{Ah} - \Delta(h)$ as follows:

$$\mathrm{e}^{Ah} - \Delta(h) = \begin{bmatrix} I & 0 \end{bmatrix}\left(\mathrm{e}^{\begin{bmatrix} A & \tilde{A}+\tilde{B}K \\ 0 & \hat{A}+\hat{B}K \end{bmatrix}(t-t_k)}\right)\begin{bmatrix} I \\ -I \end{bmatrix}. \tag{8.9}$$

We can bound (8.8) by:

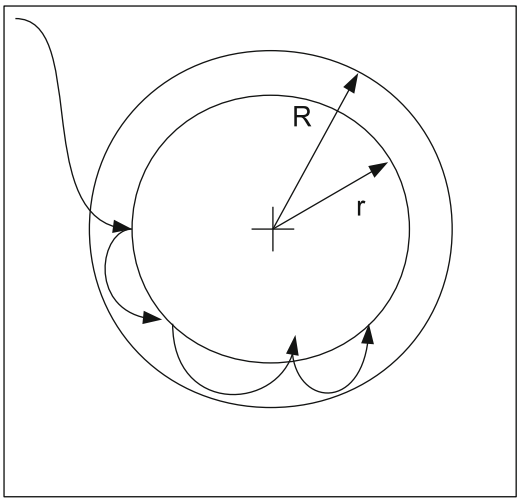

Fig. 8.1 Plant state trajectory

$$\begin{aligned} &e_k{}^T\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)e_k - x_k{}^T Q_D x_k \\ &\le \lambda_{\max}\left(\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)\right)\delta^2 - \lambda_{\min}(Q_D)\|x_k\|^2. \end{aligned} \tag{8.10}$$

The sampled value of the state of the plant at the update times will enter the region $\|x\| \le r$ where:

$$r = \sqrt{\frac{\lambda_{\max}\left((\mathrm{e}^{Ah}-\Delta(h))^T P(\mathrm{e}^{Ah}-\Delta(h))\right)\delta^2}{\lambda_{\min}(Q_D)}}. \tag{8.11}$$

The plant state vector might exit this region between samples; this is pictured in Fig. 8.1. The maximum magnitude the state plant can reach between samples after reaching the sphere $\|x\| \le R$ is given by:

$$\begin{aligned} &\|x(t)\| = \left\|\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)(t-t_k)}+\Delta(t-t_k)\right)x_k + \left(\mathrm{e}^{A(t-t_k)}-\Delta(t-t_k)\right)e_k\right\| \\ &\le \left(\mathrm{e}^{\overline{\sigma}\left(\hat{A}+\hat{B}K\right)h}+\Delta_{\max}(h)\right)r + \left(\mathrm{e}^{\overline{\sigma}(A)h}+\Delta_{\max}(h)\right)\delta \\ &\text{where } \Delta_{\max}(h) = \int_0^h \mathrm{e}^{\overline{\sigma}(A)(h-\tau)}\overline{\sigma}\left(\widetilde{A}+\widetilde{B}K\right)\mathrm{e}^{\overline{\sigma}\left(\hat{A}+\hat{B}K\right)\tau}d\tau. \end{aligned} \tag{8.12}$$

Therefore the plant state will enter and remain in the region $\|x\| \le R$ defined by (8.3):

$$R = \left(\mathrm{e}^{\overline{\sigma}\left(\hat{A}+\hat{B}K\right)h} + \Delta_{\max}(h)\right)r + \left(\mathrm{e}^{\overline{\sigma}(A)h} + \Delta_{\max}(h)\right)\delta$$

where $r = \sqrt{\dfrac{\lambda_{\max}\left((\mathrm{e}^{Ah} - \Delta(h))^T P(\mathrm{e}^{Ah} - \Delta(h))\right)\delta^2}{\lambda_{\min}(Q_D)}}$. ◆

Remark Note that (8.2) is always possible to satisfy when the MB-NCS is asymptotically stable and assuming infinite precision quantization, that is, when the system is stable according to Theorem 2.3 in Chap. 2. The stability results in Theorem 8.1 suggest that a stabilizing update period h should be found first and then proceed to find the greatest quantization parameter δ for a desired stability region.

8.1.2 Logarithmic Quantizers

We define the logarithmic quantizer as a function $q : \mathbb{R}^n \to \mathbb{R}^n$ with the following property:

$$\|z - q(z)\| \leq \delta\|z\|, z \in \mathbb{R}^n, \delta > 0. \tag{8.13}$$

Theorem 8.2 *Assume that the networked system without quantization is stable and there exists a symmetric and positive definite matrix P that solves*

$$\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)^T h} + \Delta(h)^T\right)P\left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta(h)\right) - P = -Q_D \tag{8.14}$$

where Q_D is a symmetric and positive definite matrix and $\Delta(h) = \int_0^h \mathrm{e}^{A(h-\tau)}\left(\widetilde{A} + \widetilde{B}K\right)\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau$. Then when using the logarithmic quantizer defined by (8.13), the state feedback MB-NCS is exponentially stable if:

$$\delta < \sqrt{\frac{\lambda_{\min}(Q_D)}{\lambda_{\max}\left((\mathrm{e}^{Ah} - \Delta(h))^T P(\mathrm{e}^{Ah} - \Delta(h))\right)}}.$$

Proof The difference between the values of the plant's state Lyapunov function at two consecutive update times is given by:

$$V(x(t_{k+1})) - V(x(t_k)) = {e_k}^T\left(\mathrm{e}^{Ah} - \Delta(h)\right)^T P\left(\mathrm{e}^{Ah} - \Delta(h)\right)e_k - {x_k}^T Q_D x_k. \tag{8.15}$$

We can now bound (8.15) using the quantizer property given in (8.13) by:

$$\begin{aligned} & e_k{}^T\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)e_k - x_k{}^T Q_D x_k \\ & \leq \lambda_{\max}\left(\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)\right)\delta^2\|x_k\|^2 - \lambda_{\min}(Q_D)\|x_k\|^2. \end{aligned} \tag{8.16}$$

This allows us to ensure exponential stability as in (8.6) if:

$$\lambda_{\max}\left(\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)\right)\delta^2 - \lambda_{\min}(Q_D) < 0 \tag{8.17}$$

or equivalently (assuming $(\mathrm{e}^{Ah}-\Delta(h))^T P(\mathrm{e}^{Ah}-\Delta(h)) \neq 0$):

$$\delta < \sqrt{\frac{\lambda_{\min}(Q_D)}{\lambda_{\max}\left(\left(\mathrm{e}^{Ah}-\Delta(h)\right)^T P\left(\mathrm{e}^{Ah}-\Delta(h)\right)\right)}}. \tag{8.18}$$

◆

Example 8.1 Consider the following unstable plant:

$$\dot{x} = \begin{bmatrix} 1.2638 & -0.5206 \\ 0.9164 & -0.7470 \end{bmatrix} x + \begin{bmatrix} 1.0012 \\ -0.9967 \end{bmatrix} u. \tag{8.19}$$

Let the available model be

$$\dot{\hat{x}} = \begin{bmatrix} 1.2 & -0.5 \\ 0.9 & -0.8 \end{bmatrix} \hat{x} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u. \tag{8.20}$$

The controller, which is designed using the plant model, is:

$$u = \begin{bmatrix} -2.3515 & 0.4985 \end{bmatrix} \hat{x}. \tag{8.21}$$

We obtain a stable NCS without quantization for update time intervals less than 5.3 s.

First we will study the effects of uniform quantization. The quantizer function for one variable is depicted in Fig. 8.2. The maximum absolute error between the real value of the state and the quantized values for the quantizer shown in Fig. 8.2 is calculated to be $\delta = 0.25$. By using an update time of $h = 0.3$ s and $Q_D = 0.5I$ and equation (8.2), we obtain a suitable P. We then proceed with (8.3) to obtain r and R. The radius of r and R are calculated to be 1.0131 and 3.5251, respectively. Figure 8.3 shows the regions defined by r and R and also the evolution of the plant state when the system is started with an initial condition of $[2\ 3.5]^T$. Figure 8.4 pictures the plant and model state as a function of time.

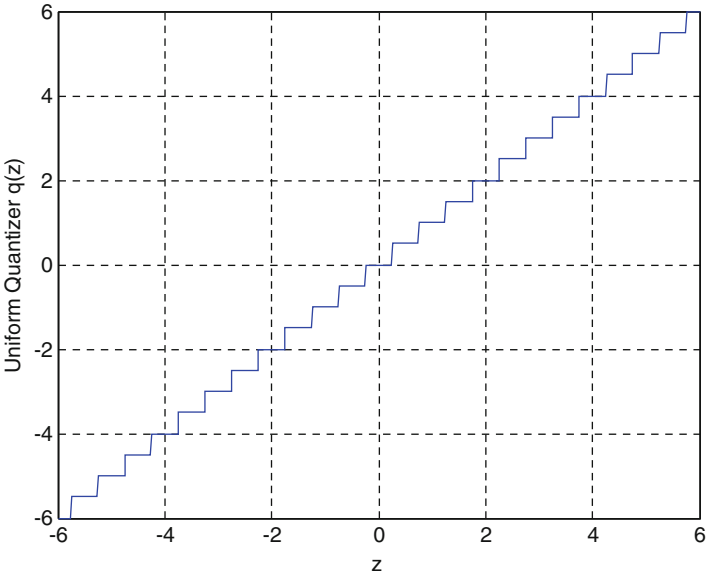

Fig. 8.2 Uniform quantizer function

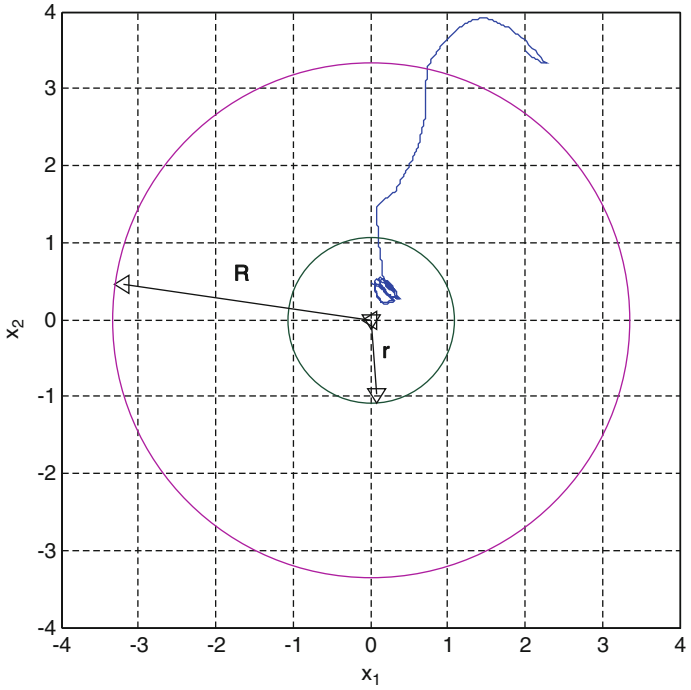

Fig. 8.3 Regions r and R and plant state evolution

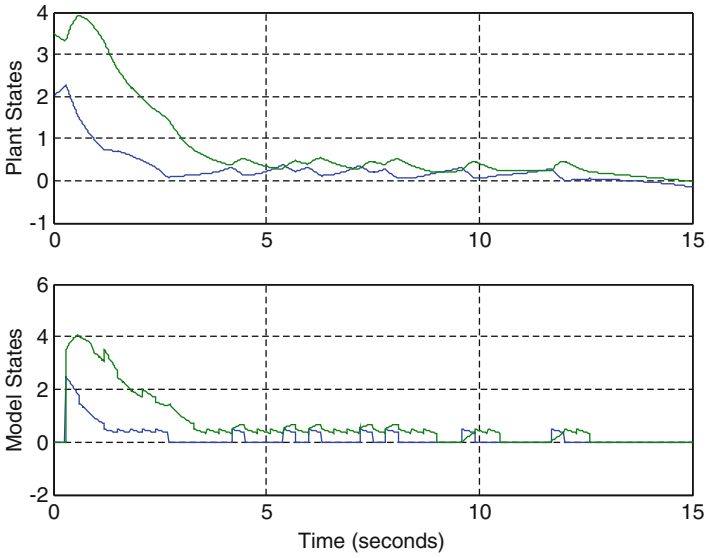

Fig. 8.4 Plant and model states using a uniform quantizer

We note that the actual region R is smaller than the calculated one; this shows that the result is conservative.

We now use the same plant with a logarithmic quantizer and an update time of $h = 2$ s. With this selection of update interval we obtain $\delta < 0.0875$. The logarithmic quantizer shown in Fig. 8.5 is implemented which has a quantization parameter $\delta = 0.06$. The response of the system and the model using the same initial conditions and the logarithmic quantizer is shown in Fig. 8.6.

8.2 Static Quantization, Event-Triggered Control, and Network Induced Delays

In this section we recall the event-triggered strategy that was studied in Chap. 6 and we design stabilizing thresholds taking into account the availability not of the real variables but only of quantized measurements. Additionally, we design stabilizing thresholds using the model-based event-triggered framework for networked systems affected by both quantization and time delays.

As it was mentioned in the last section the measured variables have to be quantized in order to be represented by a finite number of bits, so to be used in processor operations and carried over a digital communication network. It becomes necessary to study the effects of quantization error on networked systems and on any computer implemented control application because of the reasons just mentioned. In addition, we want to emphasize two important implications of quantization in event-triggered control. First, an important step in event-triggered control strategies

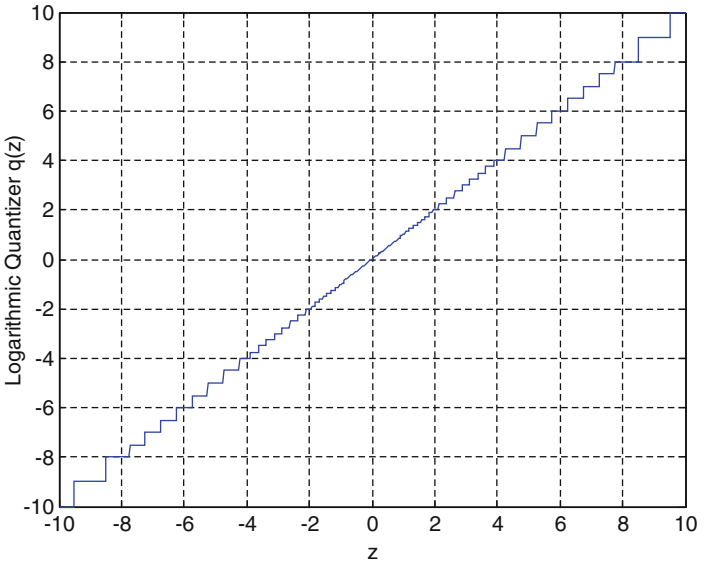

Fig. 8.5 Logarithmic quantizer function

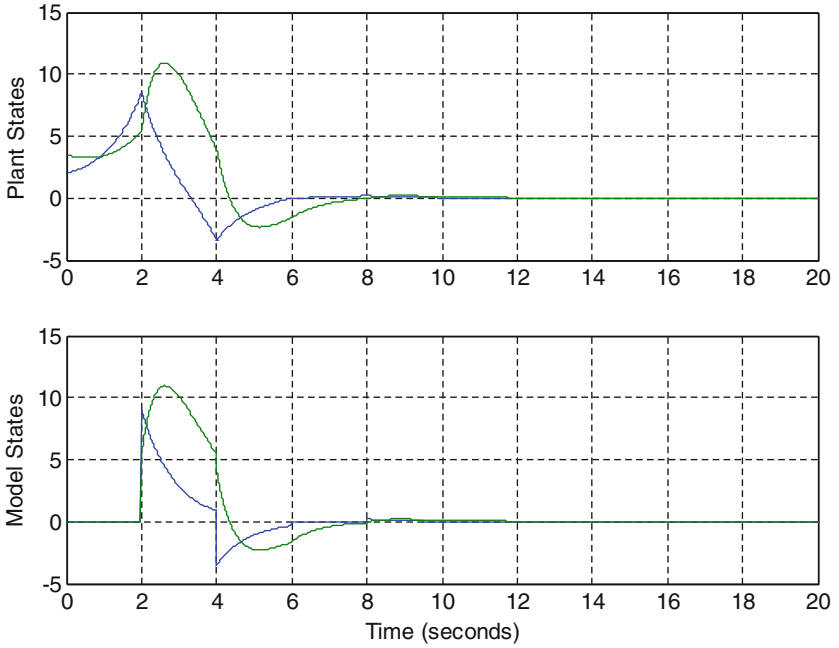

Fig. 8.6 Plant and model states using a logarithmic quantizer

is that the model-plant state error is set to zero at the update instants. When using quantization this is not the case any longer since we use the quantized measurement of the plant state to update the state of the model and this measurement is not, in general, the same as the real state of the plant. Second, in traditional even-triggered control techniques, the updates are triggered by comparing the norm of the state, which is not exactly available due to quantization errors, to the norm of the state error, which is not exactly available since it is a function of the real state of the plant. The problem in those approaches is that stability of the system is directly related to non-quantized measurements that are assumed to be known with certainty.

The aim in this section is to find triggering conditions based on the available quantized variables that also ensure asymptotic stability in the presence of quantization errors.

The type of quantizer that we are going to use in this section is the logarithmic quantizer. It is associated with floating-point data representations. We also associate this quantizer to the relative threshold strategy described in Sect. 6.2 in order to design stabilizing thresholds using the quantized measurements of the state of the plant.

Using the logarithmic quantizer defined in (8.13) we have that at the update instants t_i, $i \in Z^+$ the state of the model is updated using the quantized measurement:

$$q(x(t_i)) \rightarrow \hat{x}(t_i). \tag{8.22}$$

Define the quantized model-plant state error:

$$e_q(t) = \hat{x}(t) - q(x(t)) \tag{8.23}$$

where $q(x(t))$ is the quantized value of $x(t)$ at any time $t \geq 0$ using the logarithmic quantizer (8.13). Note that $q(x)$ and e_q are the available variables that can be used to compute the triggering condition. Also note that $e_q(t_i) = 0$, that is, the quantized model-plant state error is set to zero at the update instants according to the update (8.22).

Consider a stable closed loop nominal model and define the Lyapunov function $V = x^T P x$ where P is a symmetric and positive definite matrix and is the solution of the closed loop model Lyapunov equation:

$$\left(\hat{A} + \hat{B}K\right)^T P + P\left(\hat{A} + \hat{B}K\right) = -Q \tag{8.24}$$

where Q is a symmetric and positive definite matrix. Also consider the following bounds on the uncertainty matrices:

$$\left|\left(\widetilde{A} + \widetilde{B}K\right)^T P + P\left(\widetilde{A} + \widetilde{B}K\right)\right| \leq \Delta < \underline{q} \tag{8.25}$$

$$\left|\widetilde{B}\right| \leq \beta \tag{8.26}$$

where $\underline{q} = \underline{\sigma}(q)$, the smallest singular value of Q in (8.24).

Theorem 8.3 *Consider a linear MB-NCS with control input $u = K\hat{x}$ based on the nominal model. Assume that there exists a symmetric positive definite solution P for the model Lyapunov equation (8.24) and that the bounds in (8.25) and (8.26) are satisfied. Consider the relation*

$$|e_q| > \frac{\sigma\eta}{\delta+1}|q(x)| \tag{8.27}$$

where $\eta = \left(\underline{q} - \Delta\right)/\overline{b}$, $0 < \sigma < \sigma' < 1$ and let the model be updated when (8.27) holds. Then,

$$|e| \leq \sigma^{'}\eta|x| \tag{8.28}$$

is always satisfied and the system is asymptotically stable when,

$$\delta \leq \left(\sigma^{'} - \sigma\right)\eta. \tag{8.29}$$

Proof First observe that for the logarithmic quantizer we have:

$$\begin{aligned} |e| &= |\hat{x} - x + q(x) - q(x)| \leq |q(x) - x| + |e_q| \\ &\leq \delta|x| + |e_q|. \end{aligned} \tag{8.30}$$

Similarly,

$$\begin{aligned} |q(x)| &= |q(x) + x - x| \leq |q(x) - x| + |x| \\ &\leq \delta|x| + |x| \end{aligned}$$

$$\Rightarrow \frac{|q(x)|}{\delta+1} \leq |x|. \tag{8.31}$$

From (8.31) and applying (8.29) we can see that:

$$\sigma\eta\frac{|q(x)|}{\delta+1} + \delta|x| \leq (\sigma\eta + \delta)|x| \leq \sigma^{'}\eta|x|. \tag{8.32}$$

Now, from (8.30), (8.27), and (8.32) we obtain:

$$|e| \leq \delta|x| + |e_q| \leq \sigma\eta\frac{|q(x)|}{\delta+1} + \delta|x| \leq \sigma^{'}\eta|x| \tag{8.33}$$

then the time derivative of the Lyapunov function is bounded by:

$$\dot{V} \leq \left(\sigma^{'} - 1\right)\left(\underline{q} - \Delta\right)|x|^2 \tag{8.34}$$

and since $\sigma' < 1$, the networked system with quantization and with updates based on the error events triggered by (8.27) and using quantized feedback measurements to update the model is asymptotically stable. ◆

Based on the results of Sect. 6.3 and of Theorem 8.3 in this section we now introduce stability thresholds that consider quantization and time delays.

The quantized model-plant state error was defined in (8.23). Consider also the non-quantized model-plant state error $e(t) = \hat{x}(t) - x(t)$. At the update instants t_i we update the model in the sensor node using the quantized measurement of the state. At this instant we have $e_q(t_i) = 0$, at the sensor node. When considering network delays we can reset the quantized model-state error only at the sensor node. The model-plant state error at the update instants is given by:

$$e(t_i) = \hat{x}(t_i) - x(t_i) = q(x(t_i)) - x(t_i). \tag{8.35}$$

It is clear that this error cannot be set to zero at the update instants as the quantized model-plant state error, due to the existence of quantization errors when measuring the state of the plant. Using the logarithmic quantizer (8.13) we have that:

$$|e(t_i)| = |q(x(t_i)) - x(t_i)| \leq \delta |x(t_i)|. \tag{8.36}$$

The same conclusion can be reached by evaluating the expression in (8.30) at time instants t_i and by setting $|e_q(t_i)| = 0$.

The next theorem provides conditions for asymptotic stability of the control system using quantization in the presence of network induced delays. In this case, the admissible delays are also a function of the quantization parameter δ. That is, if we are able to quantize more finely, the system is still stable in the presence of longer delays.

Theorem 8.4 *Consider a linear MB-NCS with control input based on the state $\hat{x}$ of the model, $u = K\hat{x}$. The event-triggering condition is computed using quantized data and using error events according to (8.27). The model is updated using quantized measurements of the state of the plant. Assume that there exists a symmetric positive definite solution P for the model Lyapunov equation (8.24) and a small enough δ, $0 < \delta < 1$ such that $2\delta/(1-\delta) < \sigma\eta/(\delta+1)$. Assume also that $B = \hat{B}$ and the following bounds are satisfied: $\left|\widetilde{A}\right| \leq \Delta_A$ and $\left|\widetilde{A}^T P + P\widetilde{A}\right| \leq \Delta < \underline{q}$. Then there exists an $\varepsilon(\delta) > 0$ such that for all network delays $\tau_N \in [0, \varepsilon]$ the system is asymptotically stable. Furthermore, there exists a time $\tau > 0$ such that for any initial condition the inter-execution times $\{t_{i+1} - t_i\}$ implicitly defined by (8.27) with $\sigma < 1$ are lower bounded by τ, i.e., $t_{i+1} - t_i \geq \tau \ \forall\, i \in Z^+$.*

Proof Following the approach described in Sect. 6.3 for the non-zero delay case without quantization, let us look at the dynamics of the term $|e_q|/|q(x)|$, which

contains the available variables for processing and broadcasting after quantization has taken place. Let us first note that:

$$\begin{aligned} |e_q| &= |\hat{x} - q(x) + x - x| \leq |x - q(x)| + |\hat{x} - x| \\ &\leq \delta|x| + |e|. \end{aligned} \tag{8.37}$$

In addition and, since $0 < \delta < 1$, we have that:

$$\begin{aligned} |x| &= |x - q(x) + q(x)| \leq |x - q(x)| + |q(x)| \\ &\leq \delta|x| + |q(x)| \\ &\Rightarrow |q(x)| \geq (1 - \delta)|x|. \end{aligned} \tag{8.38}$$

The evolution of the term $|e_q|/|q(x)|$ can be bounded as follows:

$$\frac{|e_q|}{|q(x)|} \leq \frac{\delta|x| + |e|}{|q(x)|} \leq \frac{\delta|x| + |e|}{(1-\delta)|x|} = \frac{\delta}{1-\delta} + \frac{1}{1-\delta}\frac{|e|}{|x|}. \tag{8.39}$$

Let us denote the term $|e|/|x|$ by θ and, additionally, denote the term $|e_q|/|q(x)|$ by ψ so we have the estimate:

$$\psi(t) \leq \frac{\delta}{1-\delta} + \frac{1}{1-\delta}\theta(t) \leq \frac{\delta}{1-\delta} + \frac{1}{1-\delta}\phi(t,\delta) \tag{8.40}$$

where $\phi(t,\delta)$ is the solution of (6.25) in Sect. 6.3 satisfying $\phi(0,\delta) = \delta$. The initial condition in the solution of the differential equation (6.25) that we are using for the quantization case is the worst case initial error in the term $\theta(t)$ given by (8.36). The solution $\phi(t,\delta)$ is given by:

$$\phi(t,\delta) = \frac{-c(\delta+1)e^{dt(c-1)}/(\delta+c) + 1}{(\delta+1)e^{dt(c-1)}/(\delta+c) - 1} \tag{8.41}$$

then the evolution of $|e_q|/|q(x)| = \psi$ is bounded by the following expression:

$$\psi(t) \leq \xi(t) = \frac{\delta}{1-\delta} + \frac{1}{1-\delta}\frac{-c(\delta+1)e^{dt(c-1)}/(\delta+c) + 1}{(\delta+1)e^{dt(c-1)}/(\delta+c) - 1}. \tag{8.42}$$

For the case when $\tau_N = 0$, the inter-execution times for the system with quantization measurements are bounded by the time it takes for ξ to evolve from $\xi(0) = 2\delta/(1-\delta)$ to $\sigma\eta/(\delta+1)$, *i.e.*, the solution $\tau \in \mathbb{R}^+$ of $\xi(\tau) = \sigma\eta/(\delta+1)$. An estimate of that time can be obtained in a two-step process using (8.41) and (8.42) that also provides some insight into the tradeoff between the selection of the quantization parameter δ and the admissible network delays τ_N.

First, for a given $y > \delta$, solve for the time variable t in (8.41), that is, find the solution of $\phi(t, \delta) = y$. Such solution is given by:

$$\tau = \left(\ln(y+1) - \ln\left(\frac{\delta+1}{\delta+c}(y+c) \right) \right) \frac{1}{d(c-1)}. \tag{8.43}$$

For $y > \delta$, we always have $\tau > 0$, since ϕ is continuous in the range $\tau \in [0, \tau_m)$ that covers all thresholds $y > \delta$ and $\dot{\phi} > 0$. The last statement can be shown by analyzing directly the two factors in (8.43). We consider two cases avoiding the case $c = 1$. First, for $c > 1$ the second factor in (8.43) is positive. Then, in order to obtain $\tau > 0$, we need the condition:

$$\frac{(y+1)(\delta+c)}{(y+c)(\delta+1)} > 1$$

which is equivalent to the condition $y > \delta$. For the case $0 < c < 1$ the second term is negative and we need the first factor to be negative in order to obtain a strictly positive value for τ. We can ensure that the first factor is negative by satisfying the following condition:

$$\frac{(y+1)(\delta+c)}{(y+c)(\delta+1)} < 1$$

which is equivalent to $y > \delta$.

Note that (6.27) in Sect. 6.3 is a special case of (8.43), when $\delta = 0$. Then $\tau > 0$ for any $y > 0$. In this, more general case, the range of values for τ for any value of the threshold $y > \delta$ is given by $\tau \in (0, \tau_m)$, where

$$\tau_m = \lim_{y \to \infty} \tau = \frac{1}{d(c-1)} \ln\left(\frac{\delta+c}{\delta+1} \right). \tag{8.44}$$

Second, using the result in (8.43) we can obtain a solution $\tau \in \mathbb{R}^+$ of $\xi(\tau) = \sigma\eta/(\delta+1)$, *i.e.*, the solution $\tau \in \mathbb{R}^+$ of:

$$\xi\left(\tau, \frac{2\delta}{1-\delta} \right) = \frac{\delta}{1-\delta} + \frac{1}{1-\delta} \phi(\tau, \delta) = y_q \tag{8.45}$$

where

$$y_q = \sigma\eta/(\delta+1). \tag{8.46}$$

The solution is given by (8.43) with

$$y = (1-\delta)\left(y_q - \frac{\delta}{1-\delta}\right). \tag{8.47}$$

By assumption, we have that:

$$y_q > \frac{2\delta}{1-\delta} \tag{8.48}$$

which results in $y > \delta$ being satisfied, which means that $\tau > 0$. The inter-execution times are bounded by τ and strictly away from zero.

Now, for the case $\tau_N > 0$, choose σ, σ', σ^τ such that $0 < \sigma < \sigma' < \sigma^\tau < 1$ is satisfied. The last choice results in the following relation $0 < y_q < y' < y^\tau$, where y_q is defined in (8.46), $y' = \sigma'\eta$ and $y^\tau = \sigma^\tau \eta$.

Let an execution be triggered at time t_i by the condition (8.27) that in turn enforces (8.28) at time t_i and let ε_1, $0 < \varepsilon_1 < \tau_m$ satisfy the solution $\xi(\varepsilon_1, y') = y^\tau$ such ε_1 always exists since ξ is continuous in the range $\tau \in [0, \tau_m)$ that covers all thresholds $2\delta/(1-\delta) < y', y^\tau < \infty$; also $\dot{\xi} > 0$ and $y' < y^\tau$ by the previous choice of parameters. Then, by sending the state measurement at time t_i in order to update the model in the controller, we guarantee that for $t \in [t_i, t_i + \varepsilon_1]$ we have $|e| \le y^\tau |x|$, and since $\sigma^\tau < 1$ asymptotic stability is still guaranteed, i.e., (8.34) is satisfied with $\sigma^\tau < 1$. The inter-execution times are now bounded by $\tau_N + \tau$, where τ is the time it takes ξ to evolve from $\left|e_q(t_i + \tau_N)\right| / |q(x(t_i + \tau_N))| = |\hat{x}(t_i + \tau_N) - q(x(t_i + \tau_N))| / |q(x(t_i + \tau_N))|$ to $y_q = \sigma\eta/(\delta + 1)$. Then the admissible delays τ_N need to satisfy $|e_q(t_i + \tau_N)|/|q(x(t_i + \tau_N))| < y_q$ since $\dot{\xi} > 0$. From continuity of ξ with respect of τ_N there exists an $\varepsilon_2 > 0$ such that for any $0 \le \tau_N \le \varepsilon_2$

$$\xi\left(\tau_N, \frac{2\delta}{1-\delta}\right) < y_q \tag{8.49}$$

since $y_q > 2\delta/(1-\delta)$. We complete the proof by defining $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$. ◆

Remark Note that by selecting a smaller parameter δ, we increase the gap between the initial value $\xi(0) = 2\delta/(1-\delta)$ and the threshold y_q. This selection allows for longer admissible delays, i.e., a larger value for the solution (8.43) with y defined in (8.47). This corresponds to our intuition in a closed-loop system in the presence of quantization and time delays. If we quantize more finely, more accurate feedback information is available which permits to admit longer time delays while preserving asymptotic stability.

Computation of the quantization parameter δ for a fixed admissible delay ε. In developing the previous proof we first selected a quantization parameter δ (by selecting σ and σ') that ensures stability in the case of zero delay. By fixing δ and for a given choice of σ^τ we were able to estimate the longest admissible delay for which stability is still guaranteed. Now, we would like to be able to estimate the greatest quantization parameter δ for a fixed delay bound ε.

In order to compute the quantization parameter based on the admissible delays we first choose parameters σ and σ' (corresponding to σ' and σ^{τ} in Theorem 8.4), and associated thresholds y and y', and find the admissible delay without quantization, i.e., find ε in Theorem 6.5 in Sect. 6.3. Then select a new smaller delay bound than the one just found, $\varepsilon_n < \varepsilon$. We select a smaller delay since we do not expect a longer or equal admissible delay when using quantization. We proceed to search for the greatest value of δ, $0 < \delta < 1$ and a $\sigma < \sigma'$, such that the next two relations are satisfied:

$$\delta \leq \left(\sigma^{'} - \sigma\right)\eta$$

$$y_q > \frac{2\delta}{1-\delta}.$$

where,

$$y_q = \xi\left(\varepsilon_n, \frac{2\delta}{1-\delta}\right)$$

$$\sigma = \frac{y_q(1+\delta)}{\eta}$$

8.3 Dynamic Quantization: A Different Way of Adapting to System Response

In this subsection we will consider the case of dynamic quantization, where the quantized region and quantization error vary at each transmission time. It has been shown that these types of quantizers can achieve the smallest bit count per packet while maintaining stability [159, 199, 200]. This comes with the price of increased quantizer complexity. While the static quantizers did require a relatively small amount of computations, the dynamic quantizers need to compute new quantization regions and detect the plant state presence within these regions. Yet, dynamic quantizers are an attractive alternative when the number of bits available per transmission is small. Our results extent those already available in the literature extending the stability results to MB-NCS also considering model uncertainty. It will be shown that our results converge to the standard ones when the model uncertainty is zero.

Under a dynamic quantizer scheme an encoder measures the state of the plant at each transmission time and sends a symbol to the decoder collocated with the plant model. To do so, first the encoder and decoder assume that the plant state is contained in a hyper parallelogram R_k. Next, the encoder uses the plant model and plant-model uncertainties to determine the region where the plant state is at the next transmission time. This calculated region will also be a hyper parallelogram

denoted as R_{k+1}^-. The encoder can also calculate R_{k+1}^- since its calculation is based in the plant model dynamics and known uncertainty bounds. Then, the encoder can divide R_{k+1}^- in 2^N smaller equal hyper parallelograms where N is an integer representing the number of bits used to identify each smaller parallelogram. The encoder then sends an N-bit symbol representing the smaller parallelogram R_{k+1} within R_{k+1}^- that contain the plant state. The process then is repeated.

We will assume that the plant model matrix $\hat{A}$ has distinct real unstable eigenvalues. This assumption can be relaxed at the expense of more complex notation and problem geometry. We will also assume that the compensated model is stable. We will use a method that is similar to the Uncertain Set Evolution Method presented in [159]. Namely, at transmission time t_k the encoder partitions the hyper parallelogram R_k^- containing the plant state $x(t_k)$ into 2^N smaller hyper parallelograms and sends the decoder the symbol identifying the partition R_k that contains the plant state. The controller then uses the center c_k of R_k to update the plant model and generates the control signal using the plant model until time t_{k+1}^-. At this point, both encoder and decoder calculate a new hyper parallelogram R_{k+1}^- that should contain the plant state by evolving or propagating forward the initial region R_k. The process is then repeated. Stability will be ensured if the radius and center of the hyper parallelograms converge to zero with time. We will show now how the hyper parallelogram R_{k+1}^- is obtained from R_k.

Assume the plant model matrix $\hat{A} \in \mathbb{R}^{nxn}$ has n distinct unstable eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ with n corresponding linearly independent normalized eigenvectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$. We will also assume that at $t = 0$ both encoder and decoder agree in a hyper parallelogram R_0 containing the initial state of the plant. Denote a hyper parallelogram as the $(n+1)$-tuple where c is the center of the hyper parallelogram and η_i are the coordinates of its axis. In particular:

$$R(c, \eta_1, \eta_2, \ldots, \eta_n) = \left\{ x \in \mathbb{R}^n, \sum_{i=1}^{n} \alpha_i \eta_i = x - c, \text{where } \eta_i \in \mathbb{R}^n, \alpha_i \in [-1, 1], \text{and } c \in \mathbb{R}^n \right\}.$$

Let each hyper parallelogram R_k with center c_k be defined as follows:

$$R_K = R\left(c_k, \eta_{k,1}, \eta_{k,2}, \ldots, \eta_{k,n}\right) \text{ where } \eta_{k,i} = b_{k,i} v_i \text{ and } b_{k,i} \in \mathbb{R}. \tag{8.50}$$

Therefore, it can be easily verified that according to the plant dynamics the region R_k evolves into a hyper parallelogram R_{k+1}^p defined by:

$$\begin{aligned} R_{k+1}^p &= R\left(c_{k+1}^p, \eta_{k+1,1}^p, \eta_{k+1,2}^p, \ldots, \eta_{k+1,n}^p\right) \\ &\text{with } \eta_{k+1,i}^p = e^{Ah} \eta_{k,i} \text{ and } c_{k+1}^p = \left(e^{Ah} + \int_0^h e^{A(h-s)} BK e^{\left(\hat{A} + \hat{B}K\right)s} ds \right) c_k. \end{aligned} \tag{8.51}$$

Correspondingly, according to the plant model dynamics the hyper parallelogram R_k should evolve into a different hyper parallelogram R_{k+1}^m:

$$\begin{aligned} R_{k+1}^m &= R\left(c_{k+1}^m, \eta_{k+1,1}^m, \eta_{k+1,2}^m, \ldots, \eta_{k+1,n}^m\right) \\ &\text{with } \eta_{k+1,i}^m = e^{\lambda_i h}\eta_{k,i}, \text{ and } c_{k+1}^m = e^{\left(\hat{A}+\hat{B}K\right)h}c_k. \end{aligned} \tag{8.52}$$

According to equation (8.52) the hyper parallelogram R_{k+1}^m has edges that are parallel to those of the original hyper parallelogram R_k but are longer by a factor of $e^{\lambda_i h}$ for each corresponding edge. Also the center of the parallelogram has shifted. Note that the hyper parallelogram R_{k+1}^m doesn't necessarily contain the plant state. We will now express R_{k+1}^p in terms of the parameters of R_{k+1}^m. By replacing h by t and using Laplace transforms the expressions in (8.51) can be easily manipulated:

$$\begin{aligned} e^{Ah} &\xrightarrow{L} (sI-A)^{-1} = (sI-A)^{-1}\left(sI-\hat{A}\right)\left(sI-\hat{A}\right)^{-1} \\ &= \left(I+(sI-A)^{-1}\widetilde{A}\right)\left(sI-\hat{A}\right)^{-1} \\ &= \left(sI-\hat{A}\right)^{-1} + (sI-A)^{-1}\widetilde{A}\left(sI-\hat{A}\right)^{-1} \xrightarrow{L^{-1}} e^{\hat{A}h} + \int_0^h e^{A(h-s)}\widetilde{A}e^{\hat{A}s}ds \end{aligned}$$

and

$$\begin{aligned} &e^{Ah} + \int_0^h e^{A(h-s)}BKe^{\left(\hat{A}+\hat{B}K\right)s}ds \xrightarrow{L} (sI-A)^{-1} + (sI-A)^{-1}BK\left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1} \\ &= (sI-A)^{-1}\left(sI-\hat{A}+\widetilde{B}K\right)\left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1} \\ &= \left(sI-\left(\hat{A}+\hat{B}K\right)\right)^{-1} + (sI-A)^{-1}\left(\widetilde{A}+\widetilde{B}K\right)\left(sI-\left(\hat{A}-\hat{B}K\right)\right)^{-1} \\ &\xrightarrow{L^{-1}} e^{\left(\hat{A}+\hat{B}K\right)h} + \int_0^h e^{A(h-s)}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)s}ds. \end{aligned} \tag{8.53}$$

Therefore, the parameters of R_{k+1}^p can be expressed in terms of the parameters of R_{k+1}^m:

$$\begin{aligned} \eta_{k+1,i}^p &= e^{Ah}\eta_{k,i} = \left(e^{\hat{A}h} + \int_0^h e^{A(h-s)}\widetilde{A}e^{\hat{A}s}ds\right)\eta_{k,i} \\ &= e^{\hat{A}h}b_{k,i}v_i + \left(\int_0^h e^{A(h-s)}\widetilde{A}e^{\hat{A}s}ds\right)\eta_{k,i} = e^{\lambda_i h}\eta_{k,i} + \Delta_\eta(h)\eta_{k,i} \\ &= \eta_{k+1,i}^m + \Delta_\eta(h)\eta_{k,i} \\ c_{k+1}^p &= \left(e^{Ah} + \int_0^h e^{A(h-s)}BKe^{\left(\hat{A}+\hat{B}K\right)s}ds\right)c_k \\ &= \left(e^{\left(\hat{A}+\hat{B}K\right)h} + \int_0^h e^{A(h-s)}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)s}ds\right)c_k \\ &= e^{\left(\hat{A}+\hat{B}K\right)h}c_k + \left(\int_0^h e^{A(h-s)}\left(\widetilde{A}+\widetilde{B}K\right)e^{\left(\hat{A}+\hat{B}K\right)s}ds\right)c_k = c_{k+1}^m + \Delta_c(h)c_k. \end{aligned} \tag{8.54}$$

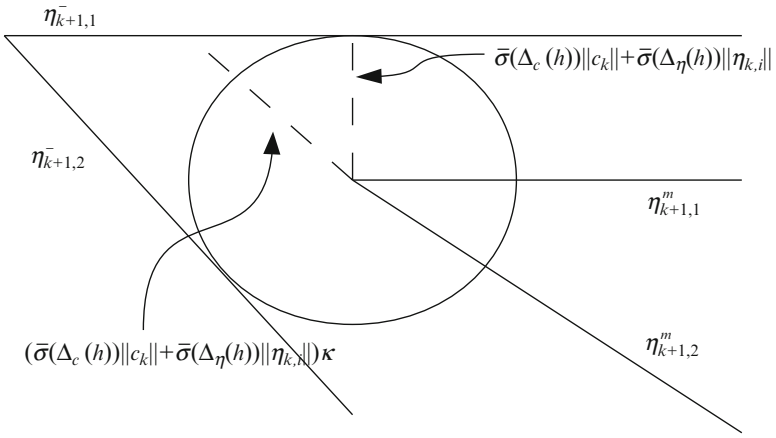

Fig. 8.7 Construction of hyper parallelogram R^{-}_{k+1} from R^{m}_{k+1}

Note that the matrices $\Delta_c(h)$ and $\Delta_\eta(h)$ can be calculated as follows:

$$\begin{aligned}\Delta_c(h) &= [I \quad 0]e^{\left(\begin{bmatrix} A & \widetilde{A}+\widetilde{B}K \\ 0 & \hat{A}+\hat{B}K \end{bmatrix}h\right)}\begin{bmatrix} 0 \\ I \end{bmatrix} \\ \Delta_\eta(h) &= [I \quad 0]e^{\left(\begin{bmatrix} A & \widetilde{A} \\ 0 & \hat{A} \end{bmatrix}h\right)}\begin{bmatrix} 0 \\ I \end{bmatrix}.\end{aligned} \tag{8.55}$$

Since the matrices $\Delta_c(h)$ and $\Delta_\eta(h)$ are unknown, the hyper parallelogram R^{p}_{k+1} cannot be constructed. Instead we will use the expressions in equation (8.54) and the bounds over the norms of $\Delta_c(h)$ and $\Delta_\eta(h)$ to construct a hyper parallelogram that will contain R^{p}_{k+1} the plant state. This is depicted in Fig. 8.7.

Note that

$$\begin{aligned} R^{-}_{k+1} &= R\left(c^{-}_{k+1}, \eta^{-}_{k+1,1}, \eta^{-}_{k+1,2}, \ldots, \eta^{-}_{k+1,n}\right) \\ \text{with } \eta^{-}_{k+1,i} &= \left(1+\overline{\sigma}(\Delta_c(h))\|c_k\|\frac{\kappa}{\left\|\eta^{m}_{k+1,i}\right\|}+\overline{\sigma}\left(\Delta_\eta(h)\right)\left\|\eta_{k,i}\right\|\frac{\kappa}{\left\|\eta^{m}_{k+1,i}\right\|}\right)\eta^{m}_{k+1,i} \\ c^{-}_{k+1} &= c^{m}_{k+1} \quad \text{, where} \quad \kappa = 1/\det([v_1\, v_2 \ldots v_n]), \|v_i\| = 1. \end{aligned} \tag{8.56}$$

Bounds over $\overline{\sigma}(\Delta_c(h))$ and $\overline{\sigma}\left(\Delta_\eta(h)\right)$ can be obtained based on the norms over the error matrices $\widetilde{A}$ and $\widetilde{B}$. Note also that R^{-}_{k+1} is a hyper parallelogram with edges larger but parallel to those of R^{m}_{k+1}. At this time the encoder will divide R^{-}_{k+1} into smaller parallelograms and transmits to the decoder the symbol that identifies the

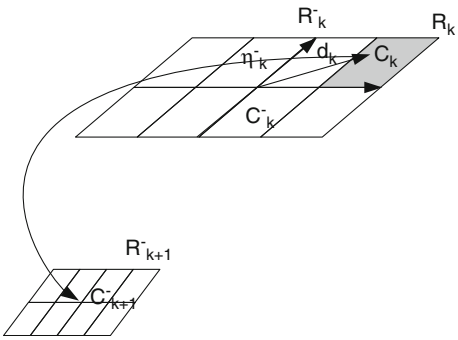

Fig. 8.8 Evolution of quantized regions

one that contains the plant state R_{k+1}. And the process repeats itself again. This process is depicted below in (8.57), also see Fig. 8.8:

$$R_k^- \overset{\text{encoder}}{\longrightarrow} R_k \underset{\text{h seconds}}{\overset{\text{plant}}{\longrightarrow}} R_{k+1}^- \overset{\text{encoder}}{\longrightarrow} R_{k+1}. \tag{8.57}$$

In Fig. 8.8 the term d_k represents the displacement of the center of R_{k+1} with respect to the center of R_{k+1}^-. We will now establish the relationship between the evolution of the hyper parallelograms parameters and stability. Define the radius of the hyper parallelogram R_k with center c_k as:

$$\lambda_{\max}(R_k) = \sup_{x \in R_k} \|x - c_k\|. \tag{8.58}$$

It is clear that in order to ensure stability of the system we require that the center and radius of the hyper parallelograms must converge to zero with time. As a matter of fact for stability only the radius of the hyper parallelograms R_k is relevant. This is stated in the next Theorem.

Theorem 8.5 *Assume that a state feedback NCS without quantization is asymptotically stable. Then the NCS with dynamical quantization is asymptotically stable if and only if:*

$$\lim_{k \to \infty} \lambda_{\max}(R_k) = 0. \tag{8.59}$$

Proof Sufficiency. We know that $\lim_{k\to\infty} \lambda_{\max}(R_k) = 0$ implies that the quantization error at each sampling time also converges to zero: $\lim_{k\to\infty} e(t_k) = 0$. Also, it can be proved, as in equation (8.5), that:

$$\begin{aligned} x(t_{k+1}) = x\left(t_{k+1}^-\right) &= \left(\mathrm{e}^{\left(\hat{A}+\hat{B}K\right)h} + \Delta_c(h)\right)x(t_k) + \left(\mathrm{e}^{Ah} - \Delta_c(h)\right)e(t_k) \\ &= Mx(t_k) + Ne(t_k) \end{aligned} \tag{8.60}$$

Since the NCS without quantization is stable the matrix M is Schur stable, therefore it is clear that if the quantization error converges to zero then the sequence of states $x(t_k)$ also converges to zero. Note that since the plant is an LTI plant, the fact that the sequence $x(t_k)$ converges to zero ensures that plant state will also converge to zero.

Necessity. In order to ensure that there is no non-zero sequence of $e(k)$ that can drive the plant state to zero and keep it there, we just need to prove that the matrix N has full rank. This is readily observed from the way N can be computed, namely:

$$N = \mathrm{e}^{Ah} - \Delta_c(h) = [I \quad 0]\mathrm{e}^{\begin{bmatrix} A & \widetilde{A}+\widetilde{B}K \\ 0 & \hat{A}+\hat{B}K \end{bmatrix}h}\begin{bmatrix} I \\ I \end{bmatrix}. \tag{8.61}$$

From equation (8.61) it can be observed that the left most matrix isolates the two upper blocks of the exponential. Since the exponential matrix has rank $2n$, the isolated matrix (of size $n \times 2n$) should have rank n. Therefore, any non-zero error vector multiplied by N will yield a non-zero vector. ◆

We will now assume that in order to generate the hyper parallelograms R_{k+1} each edge of the hyper parallelogram R^-_{k+1} is divided in Q_i equal parts. Note that all the Q_i must be powers of 2, that is $Q_i = 2^{b_i}$ where b_i represent the number of bits assigned to each axis. The resulting bit rate is $BitRate = (\sum_{i=1}^{n} b_i)/h$. We can now present a sufficient condition for stability of MB-NCS under the described dynamic quantization.

Theorem 8.6 *The state feedback MB-NCS using the dynamic quantization described in (8.57) is globally asymptotically stable if the following conditions are satisfied:*

1. *The non-quantized MB-NCS is stable.*
2. *The test matrix T has all its eigenvalues inside the unit circle, where*

$$\begin{aligned}
T &= \begin{bmatrix} T_{11a}+T_{11b} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \\
\text{with } T_{11a} &= diag\left(\left(\frac{e^{\lambda_1 h}+\overline{\sigma}(\Delta_\eta(h))\kappa}{Q_1}\right),\ldots,\left(\frac{e^{\lambda_n h}+\overline{\sigma}(\Delta_\eta(h))\kappa}{Q_n}\right)\right), \\
T_{11b} &= \begin{bmatrix} \left(\frac{Q_1-1}{Q_1}\right) & \cdots & \left(\frac{Q_n-1}{Q_n}\right) \\ \vdots & & \vdots \\ \left(\frac{Q_1-1}{Q_1}\right) & \cdots & \left(\frac{Q_n-1}{Q_n}\right) \end{bmatrix}\overline{\sigma}(\Delta_c(h))\kappa, \\
T_{12} &= \begin{bmatrix} \overline{\sigma}(\Delta_c(h))\kappa \\ \vdots \\ \overline{\sigma}(\Delta_c(h))\kappa \end{bmatrix}, \\
T_{21} &= \begin{bmatrix} \left(\frac{Q_1-1}{Q_1}\right) & \cdots & \left(\frac{Q_n-1}{Q_n}\right) \end{bmatrix}\overline{\sigma}\left(e^{(\hat{A}+\hat{B}K)h}\right), \\
T_{22} &= \overline{\sigma}\left(e^{(\hat{A}+\hat{B}K)h}\right)
\end{aligned} \tag{8.62}$$

Proof In order to characterize the evolution of the hyper parallelograms it is convenient to establish the relationship between the sizes of edges of R_{k+1}^{-} and the edges of R_k^{-}.

$$\begin{aligned}\left\|\eta_{k+1,i}^{-}\right\| &= \left(\frac{e^{\lambda_i h}+\overline{\sigma}\left(\Delta_\eta(h)\right)\kappa}{Q_i}\right)\left\|\eta_{k,i}^{-}\right\| + \overline{\sigma}(\Delta_c(h))\kappa\|c_k\| \\ &\leq \left(\frac{e^{\lambda_i h}+\overline{\sigma}\left(\Delta_\eta(h)\right)\kappa}{Q_i}\right)\left\|\eta_{k,i}^{-}\right\| + \overline{\sigma}(\Delta_c(h))\kappa\left\|c_k^{-}\right\| + \overline{\sigma}(\Delta_c(h))\kappa\|d_k\|.\end{aligned} \tag{8.63}$$

Equation (8.63) is a scalar discrete linear system. It is dependent on $\|c_k^{-}\|$. The evolution of c_k is given below.

$$c_{k+1}^{-} = e^{\left(\hat{A}+\hat{B}K\right)h}c_k = e^{\left(\hat{A}+\hat{B}K\right)h}c_k^{-} + e^{\left(\hat{A}+\hat{B}K\right)h}d_k. \tag{8.64}$$

The term $\|d_k\|$ is bounded by:

$$\|d_k\| \leq \sum_{i=1}^{N}\left(\left\|\eta_{k+1,i}^{-}\right\|\left(\frac{Q_i-1}{Q_i}\right)\right). \tag{8.65}$$

We will now bound $\|c_k^{-}\|$:

$$\left\|c_{k+1}^{-}\right\| \leq \overline{\sigma}\left(e^{\left(\hat{A}+\hat{B}K\right)h}\right)\left\|c_k^{-}\right\| + \overline{\sigma}\left(e^{\left(\hat{A}+\hat{B}K\right)h}\right)\sum_{i=1}^{N}\left(\left\|\eta_{k+1,i}^{-}\right\|\left(\frac{Q_i-1}{Q_i}\right)\right). \tag{8.66}$$

From (8.63), (8.65), and (8.66) it is clear that stability is guaranteed if T has its eigenvalues inside the unit circle. ◆

Note that if the plant model is exact, then we have that $\widetilde{A}=0$ and $\widetilde{B}=0$, $\Delta_c(h)=0$ and $\Delta_\eta(h)=0$. This implies that if $\overline{\sigma}\left(e^{\left(\hat{A}+\hat{B}K\right)h}\right)<1$ then stability is guaranteed if $\max_i\left(e^{\lambda_i h}/Q_i\right)<1$ which is a well-established result. In order to enforce the condition that $\overline{\sigma}\left(e^{\left(\hat{A}+\hat{B}K\right)h}\right)<1$, it is convenient to apply a similarity transformation that diagonalizes $\hat{A}+\hat{B}K$.

The following example depicts the way a MB-NCS can be designed; first a non-quantized MB-NCS is designed and then a suitable quantization scheme is added and tested for stability.

Example 8.2 Consider the plant represented by the following state space matrices:

$$A = \begin{bmatrix} 0 & 1 \\ a_{21} & 0.5 \end{bmatrix} B = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

where $a_{21} \in [-0.01, 0.01]$ represents the uncertainty in the matrix A. The plant model is defined as the nominal plant, that is:

$$\hat{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0.5 \end{bmatrix} \hat{B} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}.$$

A feedback gain $K = [\,-3.3333 \quad -8.3333\,]$ is selected so to place the eigenvalues of the plant model at $(-0.5, -1)$. An update time of $h = 1$ s is used. The following similarity transformation that diagonalizes $\hat{A} + \hat{B}K$ is applied to the system:

$$x_{new} = Px, \text{where } P = \begin{bmatrix} 1.8856 & 0.4714 \\ 1.3744 & 1.3744 \end{bmatrix}.$$

Finally, the quantized levels are defined as $n_1 = 1$ bit and $n_2 = 2$ bits for the eigenvectors corresponding to the eigenvalues at -0.5 and -1, respectively. The bounds for the norms of uncertainty matrices are calculated in the transformed space by searching along the parameter a_{21} and are as follows:

$$\overline{\sigma}(\Delta_c(h)) \leq 0.1354, \overline{\sigma}\big(\Delta_\eta(h)\big) \leq 0.0961.$$

The maximum eigenvalue for the test matrix T is at 0.9531 indicating that the quantized system is stable. Next a simulation of the system is presented. In this simulation the parameter a_{21} is chosen randomly to be 0.0034, the starting region with center $[\,2 \quad -3\,]^T$, edges with length 1, and the plant state randomly placed within this region. The plots are in the non-transformed original space (Figs. 8.9, 8.10, and 8.11).

8.4 Notes and References

One of the main problems in NCS is the reduction of the needed bandwidth by reducing the amount of information that is transmitted over the network while preserving desired properties of the controlled system, such as stability. Reduction

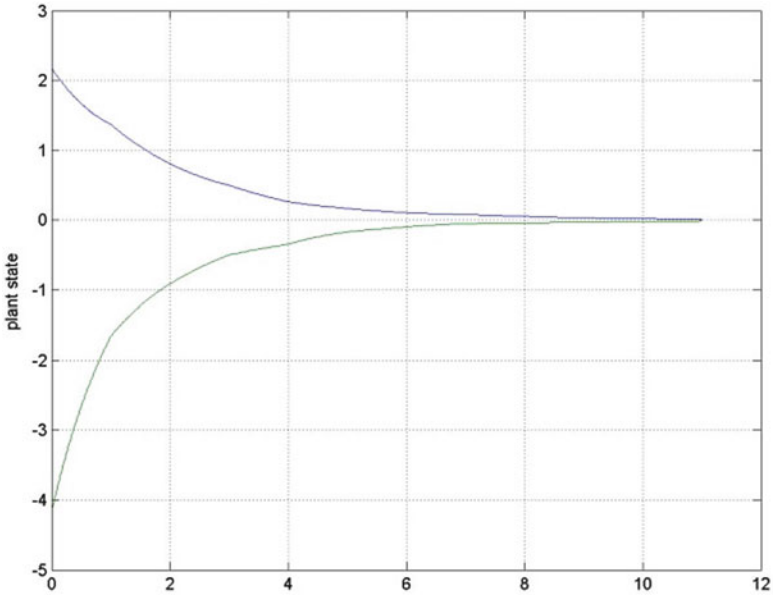

Fig. 8.9 Plant State

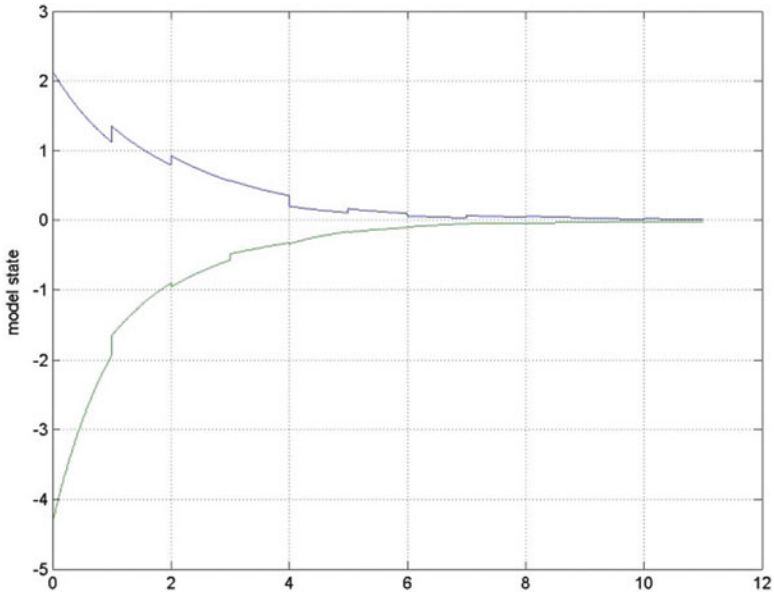

Fig. 8.10 Plant Model State

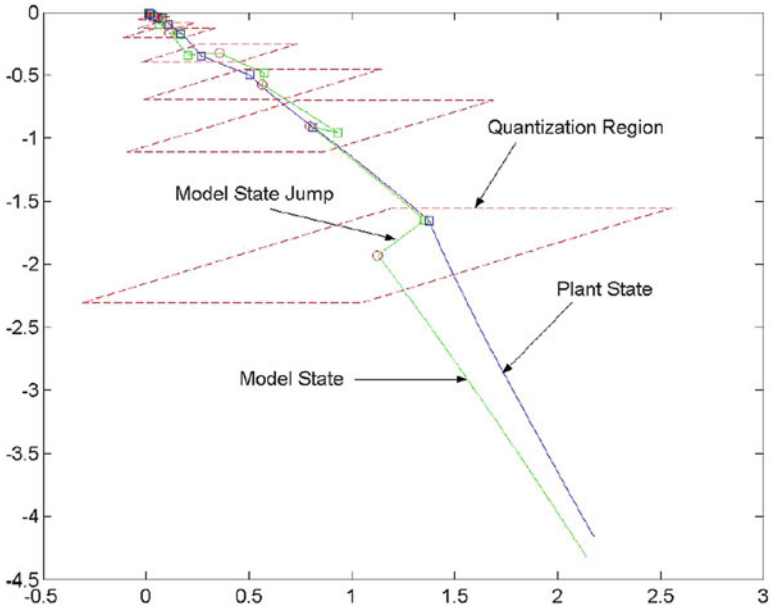

Fig. 8.11 Trajectories for Plant State and Plant Model State showing the evolution of quantized regions

of network traffic is desired in order not to overload the communication network with unnecessary data, so a faster and more reliable service can be provided. The reduction of network bandwidth also facilitates the inclusion of more control systems that can share the same network. In this chapter bandwidth reduction is considered in the MB-NCS framework. The MB-NCS framework using the quantization approaches discussed in this chapter achieve such reduction in two ways: first, by reducing the rate at which packets are sent and, second, by reducing the number of necessary bits used to transmit in each packet. This allows the designer to consider several parameters in a sequential fashion as it was shown in this chapter.

The contents of Sects. 8.1 and 8.3 are based on the work presented in [191]. The event-triggered control approach described in Sect. 8.2 was published in [87].

Several results have been published regarding the issues involved with quantization in NCS and sampled data problems [15, 65, 72, 79, 150–153]. Most results attempt to characterize the stability properties of NCS while limiting the number of bits used by each network packet.

The work in [202] offers a framework that applies to nonlinear systems subject to quantization. In addition, network scheduling is considered using periodic updates in terms of the maximal allowable transfer interval (MATI). Copies of the plant dynamics are used at both ends of the communication channel. Model uncertainties are not considered and the difference between the plant and the copies being that the copies operate using the quantized variables instead of the real ones.

The transmission of quantized measurements at times dictated by event-based strategies has been considered by several authors. The authors of [17] and [182] discuss this approach when considering the most convenient times to sample a signal. The reference [140] also considers event-triggered transmission for stabilization of systems that communicate using a limited bandwidth network and using a similar model-based approach as the one discussed in this book. In [275] the authors address the problem of output feedback control of networked systems subject to quantization and packet dropouts. A similar problem was also discussed in [125]. Dynamic quantizers were used in [43] for stabilization of NCS with nonlinear systems that are modeled using the T-S (Takagi-Sugeno) fuzzy model. Two quantizers are implemented, one in the sensor-controller channel and the other one in the controller-actuator channel. Other frameworks considering quantization issues in NCS can be found in [213, 252, 266].

Similar results to Sect. 8.3 using dynamics quantizers can be found in [159] where the authors calculate the minimum bit rate for NCS under network dropouts. In [111], Hespanha, et al. consider the case of a NCS that incorporates an exact model of the plant. The results in [111] yield the minimum bit rate for stabilizing the NCS under bounded measurement noise and input disturbance.

Part II
Performance

Chapter 9
Optimal Control of Model-Based Event-Triggered Systems

The problems of control and estimation under communication constraints have received increased attention in recent years motivated by the extensive use of digital communication networks with limited bandwidth. The communication channel is shared by different applications and in many instances only a reduced number of nodes are able to send information through the network at some specified time interval. It becomes essential to determine the conditions under which a dynamical system will remain stable and achieve a desired performance in the presence of model uncertainties, disturbances, and limited feedback information.

A common characteristic shared by the control methodologies studied in previous chapters is that they offer conditions for stability based on the design of quantization parameters, transmission rates, or error event thresholds, given the system and the controller. In this chapter we will use the Model-Based Event-Triggered (MB-ET) framework shown in Chap. 6 to maximize the transmission intervals, but here we will also consider the required control effort. In other words, we consider the design of optimal control laws and optimal thresholds for communication in the presence of plant-model mismatch by appropriately weighting the system performance, the control effort, and the communication cost. The approach we follow is to optimize the performance of the nominal system, which can be unstable in general, and to ensure robust stability for a given class of model uncertainties and for lack of feedback for extended intervals of time.

Consider an uncertain discrete-time linear plant of the form:

$$x(k+1) = Ax(k) + Bu(k). \tag{9.1}$$

The available nominal model of the system is represented by:

$$\hat{x}\,(k+1) = \hat{A}\,\hat{x}\,(k) + \hat{B}\,u(k) \tag{9.2}$$

where $x, \hat{x} \in \mathbb{R}^n$. Define the state error as:

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_9,

$$e(k) = \hat{x}(k) - x(k). \tag{9.3}$$

Note that the system and model can be stable or unstable. The aim is to design an optimal controller for the nominal system (9.2) that is robust to model uncertainties and to limited information. The later means that feedback measurements are not always available for control. This is typical, for instance, in Networked Control Systems (NCS), where the communication channel is shared by different users and may not be available for a given system to communicate at every instant [195] or due to the limited communication and processing capabilities at every node or agent within the network [251]. In other applications, it may be desirable not to use the communication channel even if it is available due to energy constraints [8, 24]. Therefore, it becomes essential to establish a trade-off between the performance of the control system and the information that can be transmitted. This trade-off can be defined by solving the next optimization problem:

$$\min_{u,\beta} J = x^T(N)Q_N x(N) + \sum_{k=0}^{N-1} x^T(k)Qx(k) + u^T(k)Ru(k) + S\beta(k) \tag{9.4}$$

where Q and Q_N are real, symmetric, and positive semi-definite matrices, R is a real, symmetric, and positive definite matrix. $\beta_k \in \{0, 1\}$ is a binary decision variable that dictates the communication pattern in the system as follows:

$$\beta_k = \begin{cases} 1 & measurement\ \ x_k\, is\, sent \\ 0 & measurement\ \ x_k\, is\, not\, sent \end{cases} \tag{9.5}$$

and S is a positive weighting factor that penalizes network communication.

In this chapter we follow the MB-ET framework that was studied in Chap. 6. This configuration makes use of an explicit model of the plant which is added to the controller node to compute the control input based on the state of the model rather than on the plant state as represented in Fig. 9.1. The goal is to reduce the communication between nodes by reducing the rate at which feedback information is sent to the controller. The MB-ET framework has the advantage that it can provide "virtual feedback" to control the physical system when no real measurements can be obtained at the controller node due to communication constraints. The idea of "virtual feedback" can be realized by using the nominal model of the system to generate an estimate of the real state in joint operation with an event-triggering strategy in the sensor node that determines the communication instants based on the size of the state error (9.3). When a measurement of the real state is received at the controller node, it is used to update the state of the model in order to reset the accumulated state error over the previous time interval since the last measurement update took place. The combination of the nominal model at the controller and the event-triggering strategy provides a "virtual feedback" by generating an estimate of the state that is kept close to the real state by maintaining a small state error. In order to obtain the same model state $\hat{x}$ at the sensor node we implement a second

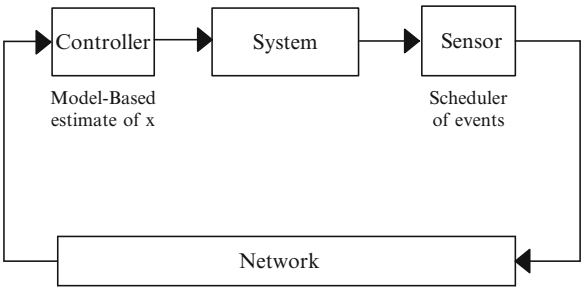

Fig. 9.1 Model-based event-triggered networked architecture

identical model at the sensor node that is updated using the measurements $x(k)$ only when $\beta_k = 1$.

The idea of "virtual feedback" allows for an approximate solution to (9.4) which considers the separation of problem (9.4) into two subproblems. The first one requires the design of the optimal control for the nominal system for the case when feedback measurements are always available to compute the control input. Since we consider model uncertainties and communication constraints, the optimal controller needs to be robust to both model mismatch and lack of real feedback for intervals of time. Sections 9.1 and 9.2 address this problem for the case of infinite horizon optimization problems. The second subproblem aims at minimizing the state error by also considering the cost that needs to be paid every time we decide to send a measurement to update the model and reset the state error. In Sect. 9.3 we use the corresponding solution for the first subproblem for finite horizon optimization problems along with a Dynamic Programming algorithm that provides the optimal times to send updates. Notes and references are given in Sect. 9.4.

9.1 Continuous-Time Robust LQR

The robustness of the continuous-time LQR to model uncertainties has been analyzed by different authors; see Sect. 9.4 at the end of this chapter. In the case of matched uncertainties the LQR guarantees robust stability for any bounded uncertainty of this type. In this section we extend the approach in [154] for matched uncertainties in order to consider state feedback uncertainties as well. The state uncertainty is characterized by the state error (9.8) and is the result of limited communication between sensor and controller. We also make use of the MB-ET approach in order to design error events that will prompt the sensor to send the current measurement of the state of the system.

We consider network interconnected uncertain linear systems of the form:

$$\dot{x}(t) = Ax(t) + Bu(t). \tag{9.6}$$

The nominal model of the system is:

$$\dot{\hat{x}}(t) = \hat{A}\,\hat{x}(t) + \hat{B}\,u(t) \tag{9.7}$$

where $x, \hat{x} \in \mathbb{R}^n$. Define the state error for the continuous-time case as:

$$e(t) = \hat{x}(t) - x(t). \tag{9.8}$$

We also consider the following assumptions:

(a) The nominal system $\left(\hat{A}, \hat{B}\right)$ is stabilizable.
(b) $B = \hat{B}$
(c) We assume matched uncertainty, that is, the uncertainty is in the range of the matrix B. Mathematically we have that there exists a $m \times n$ matrix ϕ such that $\widetilde{A} = A - \hat{A} = B\phi$, and ϕ is bounded in the Euclidean sense.

Using the nominal system parameters (9.7) we design a feedback control law $u = Kx$ that minimizes

$$J = \int_0^\infty \frac{1}{2}\left[x^T(t)Fx(t) + x^T(t)Qx(t) + u^T(t)Ru(t)\right]dt \tag{9.9}$$

where F is defined as

$$F = \inf\left\{F' : F' \geq \phi^T R \phi\right\}. \tag{9.10}$$

Q is a real, symmetric, and positive semi-definite matrix, and R is a real, symmetric, and positive definite matrix.

Theorem 9.1 *System (9.6) with a model-based input $u = K\hat{x}$ and with model updates based on error events is asymptotically stable for all matched uncertainties satisfying (9.10) if the updates are triggered when*

$$|e| \geq \frac{\sigma \underline{q}}{\left|K^T R K\right|}|x| \tag{9.11}$$

where K is the feedback gain obtained by solving the Algebraic Riccati Equation (ARE) using the nominal parameters, $\underline{q} = \underline{\sigma}(Q)$, and $0 < \sigma < 1/2$.

Proof The solution of the optimal control problem that minimizes (9.9) is the LQR which provides the feedback gain:

$$K = -R^{-1}B^T P \tag{9.12}$$

and the matrix P is the solution of the associated ARE:

$$\hat{A}^T P + P\hat{A} - PBR^{-1}B^T P + F + Q = 0. \tag{9.13}$$

Define:

$$V(x_0) = \min_{u \in L^2} \int_0^\infty \frac{1}{2}\left[x^T(t)Fx(t) + x^T(t)Qx(t) + u^T(t)Ru(t)\right]dt = J * (x(t), t). \tag{9.14}$$

$V(x_0)$ is the minimum cost of the optimal control of the nominal system from some initial state x_0.

The Hamiltonian is given by:

$$H = \frac{1}{2}x^T Fx + \frac{1}{2}x^T Qx + \frac{1}{2}u^T Ru + V_x^T\left[\hat{A}x + Bu\right] \tag{9.15}$$

and the HJB equation for the nominal model is given by:

$$\frac{1}{2}x^T Fx + \frac{1}{2}x^T Qx + \frac{1}{2}x^T K^T RKx + V_x^T\left[\hat{A}x + BKx\right] = 0. \tag{9.16}$$

Additionally, the optimality condition results in:

$$x^T K^T R + V_x^T B = 0. \tag{9.17}$$

Next, we will show that $V(x)$ is a Lyapunov function for the *real* system (9.6) affected by model uncertainties and state feedback errors which are caused by the lack of communication from sensor to controller. System (9.6) can be represented by:

$$\dot{x} = Ax + Bu = \hat{A}x + BKx + B\phi x + BKe. \tag{9.18}$$

We can also see that:

$$\begin{aligned} V(x) &> 0, \quad x \neq 0 \\ V(x) &\geq 0, \quad x = 0. \end{aligned}$$

Now we evaluate $\dot{V}(x)$ along the trajectories of (9.18)

$$\begin{aligned}\dot{V}(x) &= V_x^T\left[\hat{A}x + BKx + B\phi x + BKe\right] \\ &= -\frac{1}{2}x^T Fx - \frac{1}{2}x^T Qx - \frac{1}{2}x^T K^T RKx - x^T K^T R\phi x - x^T K^T RKe + \frac{1}{2}x^T\phi^T R\phi x - \frac{1}{2}x^T\phi^T R\phi x \\ &= -\frac{1}{2}x^T\left[F - \phi^T R\phi\right]x - \frac{1}{2}x^T\left[K+\phi\right]^T R\left[K+\phi\right]x - \frac{1}{2}x^T Qx - x^T K^T RKe \\ &\leq -\frac{1}{2}x^T Qx - x^T K^T RKe \leq -\frac{1}{2}\underline{q}|x|^2 + \left|K^T RK\right||e||x|.\end{aligned}$$

Using the threshold (9.11) we ensure that $|e| \leq \frac{\sigma\underline{q}}{\left|K^T RK\right|}|x|$ and then we have:

$$\dot{V}(x) \leq \left[\sigma - \frac{1}{2}\right]\underline{q}|x|^2 < 0 \tag{9.19}$$

♦

9.2 Discrete-Time Robust LQR

The purpose of this section is to extend the results of the previous section to the discrete-time domain. Following a similar analysis but using the discrete-time LQR we arrive at a slightly different set of conditions and event thresholds. In this section we consider the discrete-time system (9.1) with model (9.2) and the state error as defined in (9.3). We also consider assumptions (a)–(c) in Sect. 9.1.

Using the nominal system parameters (9.2) we design a feedback control law $u = Kx$ that minimizes

$$J = \sum_{k=0}^{\infty} x^T(k)Fx(k) + x^T(k)Qx(k) + u^T(k)Ru(k) \tag{9.20}$$

where F is defined as

$$F = \inf\left\{F' : F' \geq \phi^T\left[B^T PB + R\right]\phi\right\}. \tag{9.21}$$

Q is a real, symmetric, and positive semi-definite matrix, and R is a real, symmetric, and positive definite matrix.

Theorem 9.2 *System (9.1) with a model-based input $u = K\hat{x}$ and with model updates based on error events is asymptotically stable for all matched uncertainties satisfying (9.21) if the updates are triggered when*

$$|e| \geq \alpha|x| \tag{9.22}$$

where $\alpha = min(\alpha_1, \alpha_2)$, $\alpha_1 = \sigma\underline{q}/2c_1$, $\alpha_2 = \left[\sigma\underline{q}/2c_2\right]^{1/2}$, $0 < \sigma < 1$, $c_1 = 2\left|\left[\hat{A} + B[K+\phi]\right]^T PBK\right|$, *and* $c_2 = /\mathrm{K^T B^T PBK}/$. *The controller gain K is the feedback gain obtained by solving the discrete-time ARE using the nominal parameters, and* $\underline{q} = \underline{\sigma}(Q)$.

Proof The solution of the optimal control problem that minimizes (9.20) is the discrete-time LQR which provides the feedback gain:

$$K = -\left[B^T PB + R\right]^{-1} B^T P\hat{A} \tag{9.23}$$

and the matrix P is the solution of the associated discrete-time ARE:

$$\hat{A}^T P\hat{A} - P + F + Q - \hat{A}^T PB\left[B^T PB + R\right]^{-1} B^T P\hat{A} = 0. \tag{9.24}$$

By using the following relationship:

$$-\hat{A}^T PB\left[B^T PB + R\right]^{-1} B^T P\hat{A} = \hat{A}^T PBK + K^T B^T P\hat{A} + K\left[B^T PB + R\right]K \tag{9.25}$$

we can rewrite the discrete-time ARE as:

$$\left[\hat{A} + BK\right]^T P\left[\hat{A} + BK\right] + F + Q + K^T RK - P = 0. \tag{9.26}$$

Let us consider the candidate Lyapunov function $V(x) = x^T Px$ and we evaluate the first forward difference of $V(x)$ along the trajectories of the real system not of the model but using the model control input $u = K\hat{x}$

$$\begin{aligned}
\Delta V(x) &= V\big(x(k+1)\big) - V\big(x(k)\big) \\
&= \left[x^T\left[\hat{A} + BK + \widetilde{A}\right]^T + e^T K^T B^T\right] P\left[\left[\hat{A} + BK + \widetilde{A}\right]x + BKe\right] - x^T Px \\
&= x^T\left[\hat{A} + BK + \widetilde{A}\right]^T P\left[\hat{A} + BK + \widetilde{A}\right]x \\
&\quad + 2x^T\left[\hat{A} + BK + \widetilde{A}\right]^T PBKe + e^T K^T B^T PBKe - x^T Px \\
&= -x^T Qx - x^T\left[F - \phi^T\left[B^T PB + R\right]\phi\right]x - x^T\left[K + \phi\right]^T R\left[K + \phi\right]x \\
&\quad + 2x^T\left[\hat{A} + BK + \widetilde{A}\right]^T PBKe + e^T K^T B^T PBKe \\
&\le -x^T Qx + 2x^T\left[\hat{A} + BK + \widetilde{A}\right]^T PBKe + e^T K^T B^T PBKe \\
&\le -\underline{q}|x|^2 + 2\left|\left[\hat{A} + B[K+\phi]\right]^T PBK\right||e||x| + \left|K^T B^T PBK\right||e|^2 \\
&\le -\underline{q}|x|^2 + c_1|e||x| + c_2|e|^2.
\end{aligned}$$

Now, by updating the model (the error is set to zero when we update) following the condition in (9.22), we can bound the error using the term in the right hand side of (9.22) and we can finally write:

$$V(x(k+1)) - V(x(k)) \leq [\sigma - 1]q|x|^2. \tag{9.27}$$

Then V is guaranteed to decrease for any σ such $0 < \sigma < 1$ and updating the state of the model using the threshold in Theorem 9.2. ♦

9.3 Finite Horizon Optimal Control and Optimal Scheduling

Finite horizon optimal control problems correspond to more realistic implementations to real problems. In most cases we are concerned with the operation of a physical system over a finite period of time in an optimal sense. The consideration of infinite horizon problems results, in many cases, in simplified controller design steps. This can be easily observed in the continuous and discrete-time LQR. These simpler controllers can be used in practice, especially, for long or unknown finite horizons. For shorter horizons, we need to formulate the problem differently.

In this section, assuming N is known, we consider our original problem as stated at the beginning of this chapter. The system and model are given by (9.1) and (9.2) respectively and the cost to be minimized is given by the alternative expression of (9.4) using the separate weights F and Q, i.e.,

$$\begin{aligned} \min_{u,\beta} J = {} & x^T(N)Q_N x(N) \\ & + \sum_{k=0}^{N-1} x^T(k)Fx(k) + x^T(k)Qx(k) + u^T(k)Ru(k) + S\beta(k) \end{aligned} \tag{9.28}$$

By following similar ideas as in previous sections we first design the optimal controller for the nominal system assuming that feedback measurements are always available. The optimal control input for this case is given by:

$$u*(N-i) = -\left[B^T P(i-1)B + R\right]^{-1} B^T P(i-1)\hat{A}\, x(N-i) \tag{9.29}$$

where $P(i)$ is recursively computed using:

$$\begin{aligned} P(i) = {} & \left[\hat{A} + BK(N-i)\right]^T P(i-1)\left[\hat{A} + BK(N-i)\right] + F + Q \\ & + K^T(N-i)RK(N-i). \end{aligned} \tag{9.30}$$

Define the Lyapunov function $V(x(k)) = x^T(k)P(k)x(k)$. Using a similar analysis as in Sect. 9.2 we can show the following results.

Corollary 9.3 *System (9.1) with input (9.29) is stable in the Lyapunov sense for all matched uncertainties satisfying:*

$$F = \inf\{F' : F' \geq \phi^T [B^T P(k)B + R]\phi\} \tag{9.31}$$

for all $k = 0,1,\ldots$N. ♦

Note that in order to obtain the optimal control law we need to solve offline the discrete-time LQR, i.e., we need to find $K(k)$ and $P(k)$ before the execution of the system, then it is possible (knowing a bound on the uncertainty) to check (9.31) in advance. When (9.31) holds we know the system is stable and the optimal cost of the form (9.20) with no communication penalty is finite when measurements are available at every time k. Then we can proceed to select an appropriate weight on the communication to restrict measurement updates from the sensor to the controller.

We are now in the position to approach the second problem that was introduced in the beginning of this chapter in a more formal way. In Sects. 9.1 and 9.2 we were able to reduce the communication rate between sensor and controller while using the optimal control law and the estimates generated by the model. However, the communication pattern was not optimal in any sense. Next we use the error nominal dynamics and the selected communication factor S in order to design the optimal update events.

This approach to solve the optimal scheduling problem can be seen as the minimization of the deviation of the system performance from the nominal performance by also considering the cost that needs to be paid by updating the model and resetting the state error.

The error dynamics are given by:

$$\begin{aligned} e(k+1) &= \hat{x}(k+1) - x(k+1) = \hat{A}\hat{x}(k) + Bu(k) - Ax(k) - Bu(k) \\ &= \hat{A}e(k) - \widetilde{A}x(k). \end{aligned} \tag{9.32}$$

Since the uncertainty $\widetilde{A}$ is not known, we use the nominal error dynamics, i.e.,

$$e(k+1) = \hat{A}e(k). \tag{9.33}$$

Furthermore, when the sensor decides to send a measurement update, which makes $\beta_k = 1$, we reset the error to zero. Then the complete nominal error dynamics can be represented by the following equation:

$$e(k+1) = \hat{A}e(k) - \beta_k \hat{A}e(k) = [1 - \beta_k]\hat{A}e(k). \tag{9.34}$$

It is clear that, in the nominal case, once we update the model the state error is equal to zero for the remaining time instants. However, in a real problem the state error

dynamics are disturbed by the state of the real system which is propagated by means of the model uncertainties as expressed in (9.32). Then, using the available model dynamics we implement the optimal control input and the optimal scheduler that results from the following optimization problem:

$$\begin{aligned} \min_{\beta} J_e &= e^T(N)Q_N e(N) + \sum_{k=0}^{N-1} e^T(k)Qe(k) + S\beta_k \\ &\text{subject to} \quad e(k+1) = \left[1-\beta_k\right]\hat{A}\, e\left(k\right) \\ &\beta_k \in \{0,1\}. \end{aligned} \tag{9.35}$$

In order to solve problem (9.35) we use Dynamic Programming in the form of look up tables. The main reason for using Dynamic Programming is that, although the error will be finely quantized, the decision variable β_k only takes two possible values, which reduces the amount of computations performed by the Dynamic Programming algorithm. The sensor operations at time k are reduced to measure the real state, compute and quantize the state error, and determine if the current measurement needs to be transmitted by looking at the corresponding table entries which are computed offline. The table size depends only on the horizon N and the error quantization levels.

Example 9.1 Consider the nominal model of an unstable discrete-time second order system given by:

$$\hat{A} = \begin{bmatrix} 1.5 & 0.3 \\ -0.6 & -0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The unknown dynamics of the unstable real system used in the next simulation example are:

$$A = \begin{bmatrix} 1.25 & 0.48 \\ -0.73 & -0.81 \end{bmatrix}.$$

The model-based controlled system is intended to operate over a finite period of $N = 30$ stages. The parameters in the optimization problem are as follows: $Q = R = I$, $S = 1$, $Q_N = 2I$. The unknown initial conditions of the system are given by $x(0) = [1.9 \;\; -1.4]^T$, since these initial conditions are unknown to the controller, the model is initialized using $\hat{x}(0) = [0 \quad 0]^T$. The computation of the control gains at every time instant and the solution of the Dynamic Programming optimization problem are computed offline and stored accordingly in the controller and sensor nodes. The results of the simulation are shown in Figs. 9.2, 9.3, and 9.4 using linear interpolation to connect the consecutive values in the response of the system and model.

Figure 9.2 shows both the states of the real system and the states of the model that are used in the computation of the control input when feedback measurements

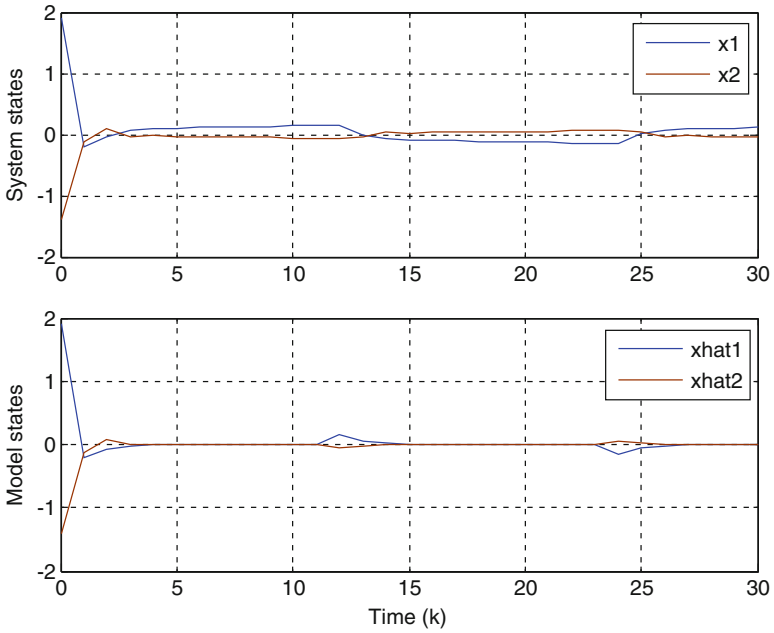

Fig. 9.2 Response of the real uncertain system with limited feedback (*top*). States of the model used to control the system (*bottom*)

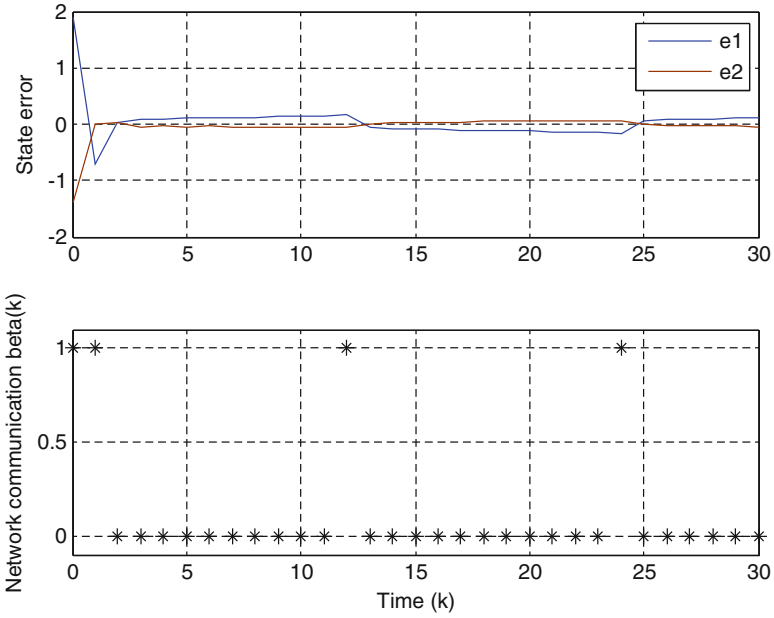

Fig. 9.3 State error used to determine the instants at which a feedback measurement has to be transmitted (*top*). Communication pattern β_k (*bottom*)

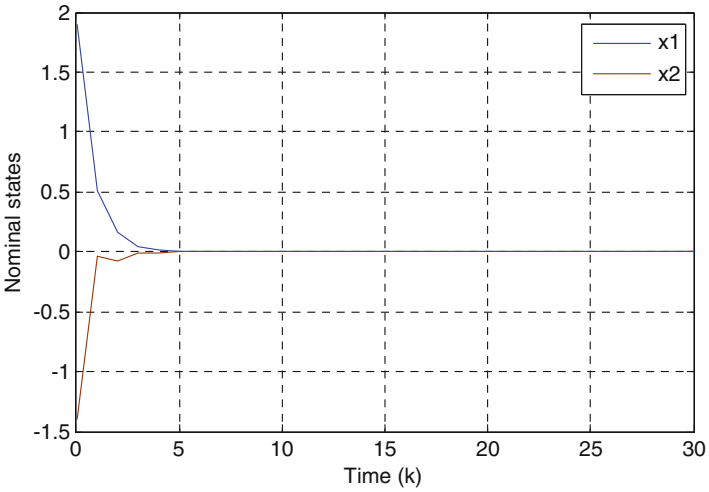

Fig. 9.4 Response of the nominal system receiving feedback measurements at every time k

are unavailable. The top portion of Fig. 9.3 shows the state error. For the different combinations of the state error we find the corresponding entry on the table that contains the solution of the Dynamic Programming optimization problem. At every time instant the sensor decides to send or not the measurement of the state of the system based on the values of the error, those decisions are represented at the bottom of Fig. 9.3. An important difference with respect to the error threshold designed in Sects. 9.1 and 9.2 is that the solution of problem (9.35) considers the dynamics (nominal) of the error in the design of the transmission events. By including a prediction of the behavior of the state error we are also able to predict the consequences, as measured by the computation of the optimal cost, of updating or not the model at a given time instant k. We can see, for instance, that some combinations of state error values will result in different transmission decisions depending at which of the N stages we are operating at that moment.

It is also important to note that in the absence of model uncertainties we obtain the response of the nominal system as if measurements were always available. Figure 9.4 shows the response of the nominal system when feedback measurements are available at every time instant k. The behavior of the overall controlled system which consists on the MB-ET framework with optimal control input and optimal scheduler is comparable to that of the nominal system for which the optimal control law is designed, and the difference is considerably reduced as the uncertainties diminish.

Example 9.2 The approach in this section can also be applied to the case when the real system is continuous. From the available nominal continuous-time model we obtain a discretized nominal system at a low sampling rate. Measurements of the real system are obtained at the same rate in order to compute the state error and to take the decision of whether sending an update or not.

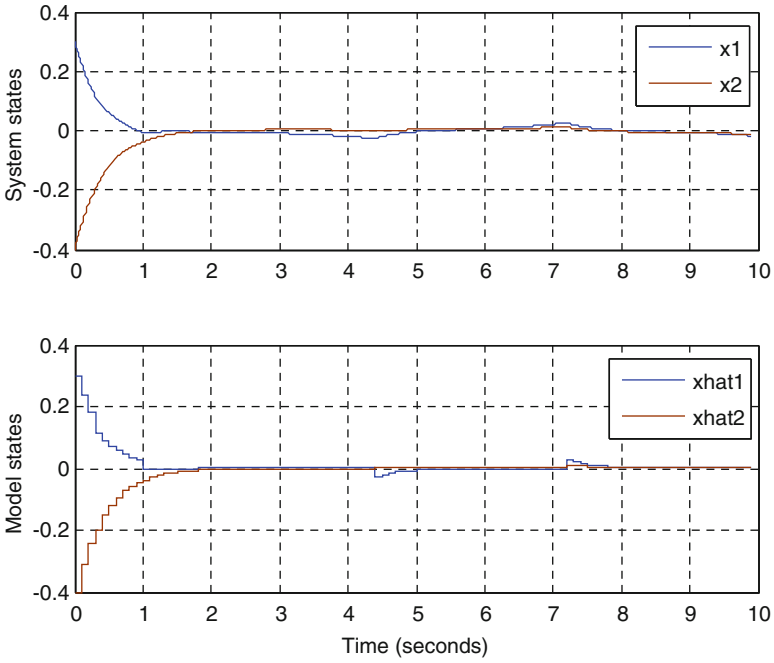

Fig. 9.5 Response of the real continuous-time uncertain system with limited feedback (*top*). States of the discretized model used to control the system (*bottom*)

Consider the unstable nominal continuous-time model given by:

$$\hat{A} = \begin{bmatrix} 1 & 0.2 \\ 0.4 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The unknown dynamics of the unstable continuous-time real system used in the next simulation example are:

$$A = \begin{bmatrix} 0.88 & 0.32 \\ 0.15 & 0.34 \end{bmatrix}.$$

The parameters in the optimization problem are as follows: $S = 0.1$, $Q_N = 10I$, $Q = 5I$, $R = I$. The unknown initial conditions of the system are given by $x(0) = [0.3 \ -0.4]^T$ and the sampling rate used to determine the discretized nominal model is $Ts = 0.1$ s. The results of the simulation are shown in Figs. 9.5, 9.6, and 9.7. The discrete variables have been plotted using zero-order-hold interpolation in order to clearly differentiate them from the continuous states of the system.

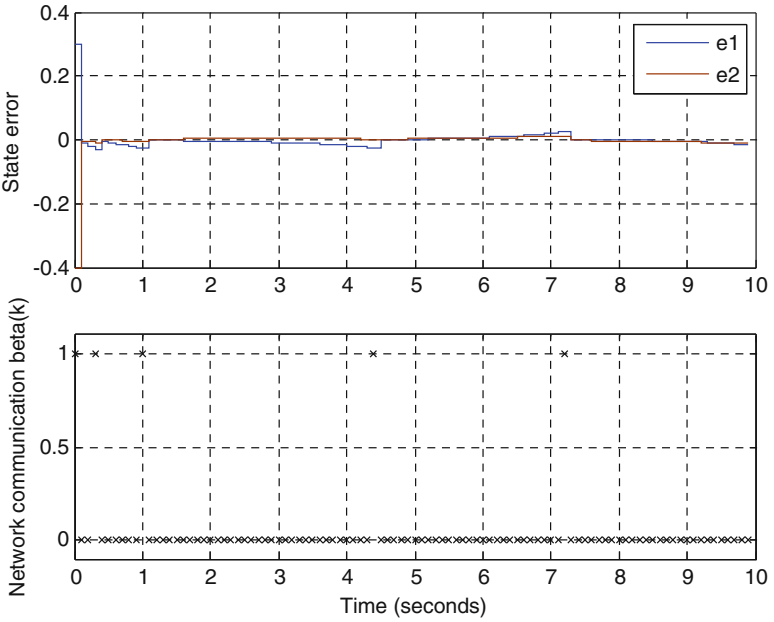

Fig. 9.6 State error used to determine the instants at which a feedback measurement has to be transmitted (*top*). Communication pattern β_k (*bottom*)

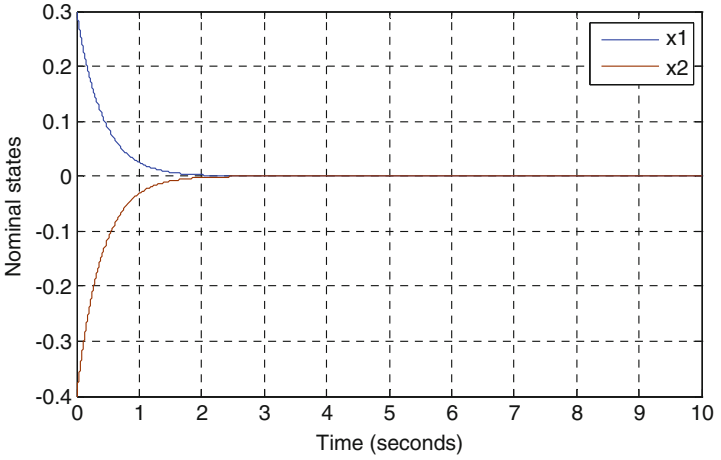

Fig. 9.7 Response of the continuous-time nominal system receiving continuous feedback measurements

9.4 Notes and References

The design of optimal controllers for unknown systems represents a challenging task. An approach that has been studied in the literature is to design an optimal controller for a nominal system and analyze its robustness to a range of uncertainties. In this chapter we considered a more complex version of this problem in which feedback information is limited and, in addition to designing an optimal control input, we also addressed the problem of finding the optimal instants at which we should transmit feedback measurements. The approach considered in this chapter, which consists of model-based estimates of the state and of error events, provides a practical and promising methodology that considers the overall performance of the communication constrained system. Parts of this chapter can be found in [94].

Similar ideas have been considered by different research groups. For instance, Imer and Basar, [120, 121], considered separately the estimation and control problems with limited information when the nominal system is affected by process and measurement noise. In both cases the source node is able to send data through the communication channel only M times of possible N ($M < N$). The aim is to find the optimal control law (minimize the average estimation error) by indirectly penalizing channel uses. Molin and Hirche [184] also considered the trade-off between control performance and resource utilization, i.e., the cost of updating the controller node using current measurements. The authors study linear systems affected by zero-mean Gaussian noise and they assume the system parameters and statistics are known, that is, model uncertainties are not taken into account in their work. The authors of [167, 274, 289] also address similar problems.

The robustness of the continuous-time LQR to model uncertainties has been analyzed by different authors. Douglas and Athans [62] presented an LQR design that is robust to real parametric uncertainty in the state matrix of linear time invariant systems by wisely choosing the state weighting matrix. Misra [183] discussed the LQR design with assignment of the closed loop eigenvalues. Lin and Olbrot [154] showed a robust LQR design method for uncertain continuous-time systems that apply to both matched and unmatched uncertainties.

Chapter 10
Performance Analysis Using Lifting Techniques

The performance characterization of Networked Control Systems under different conditions represents an important and interesting problem. In previous chapters it was shown that the stability of MB-NCS is, in general, a function of the update times, the difference between the dynamics of the plant and the dynamics of the plant model, and of the control law used. The performance of the MB-NCS can be studied using several techniques and considering different scenarios. One promising technique is called Continuous Lifting.

Lifting techniques for discrete-time systems were employed in Chaps. 2 and 4. Lifting represents a transformation of a periodic system to a discrete LTI system. The periodic system could be a continuous-time or a discrete-time system. The main advantage of this approach is that most of the results available for LTI systems are readily applicable to the lifted system. The disadvantage is that the input and output spaces grow in size if the underlying periodic system is a discrete time one, and the input and output spaces become infinite dimensional when the underlying periodic system is a continuous-time system. In the latter case the parameters of the lifted system are operators and not matrices.

The results in this chapter are concerned with the performance analysis of MB-NCS with continuous-time systems and with periodic updates. Corresponding results can also be obtained for discrete-time systems with periodic updates following the similar approaches discussed in this chapter. Section 10.1 describes a continuous-time lifting technique used in this chapter and it also discusses the performance of MB-NCS using H_2 norm characterizations. Section 10.2 addresses the design of optimal controllers for MB-NCS. Notes and references are given in Sect. 10.3.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_10,

10.1 Performance Analysis of MB-NCS Using Continuous-Time Lifting

In this section we introduce two performance measures related to the traditional H_2 performance measure for LTI systems. The first H_2 like performance measurement, called the Extended H_2 norm of the system, is based on the norm of the impulse response of the NCS at time zero. The second performance measure, called the Generalized H_2 norm, it basically replaces the traditional trace norm by the Hilbert-Schmidt norm that is more appropriate for infinite dimensional operators. The Generalized H_2 norm represents the average norm of the impulse response of the NCS for impulse inputs applied at different times.

Before defining the performance measures previously described, a brief summary of the lifting technique is presented. As it was pointed out, lifting can transform a periodic linear system such as a MB-NCS into a discrete linear time invariant system with operator-valued parameters. These parameters are computed for a class of MB-NCS and used throughout the section.

10.1.1 Continuous Lifting Technique

We will give a brief introduction of the Lifting technique for continuous-time periodic systems. We need to define two Hilbert spaces, the first space is defined as:

$$L_2\big[0,h\big) = \left\{ u(t) / \int_0^h u^T(t)u(t)dt < \infty \right\}. \tag{10.1}$$

The second Hilbert space of interest is formed by an infinite sequence of $L_2[0,h)$ spaces and is defined as:

$$\begin{aligned} l_2\big(\mathbb{Z}, L_2\big[0,h\big)\big) &= l_2 \\ &= \left\{ [\ldots, u_{-2}, u_{-1}, u_0, u_1, u_2, \ldots]^T / \sum_{-\infty}^{+\infty} \int_0^h u_j{}^T(t)u_j(t)dt < \infty \right\}. \end{aligned} \tag{10.2}$$

Now we can define the lifting operator L as a mapping from L_{2e} (L_2 extended) to l_2:

$$\mathrm{L} : L_{2e} \to l_2, \;\; \mathrm{L}u(t) = [\ldots, u_{-2}, u_{-1}, u_0, u_1, u_2, \ldots]^T \tag{10.3}$$

where $u_k(\tau) = u(\tau + kh)$, $\tau \in [0,h)$. It can be shown that L preserves inner products and thus is norm preserving [42]. Since L is surjective, it is an isomorphism of Hilbert spaces. So, lifting basically transforms a continuous function into a discrete function where each element of the sequence is a continuous function restricted to $[0,h)$.

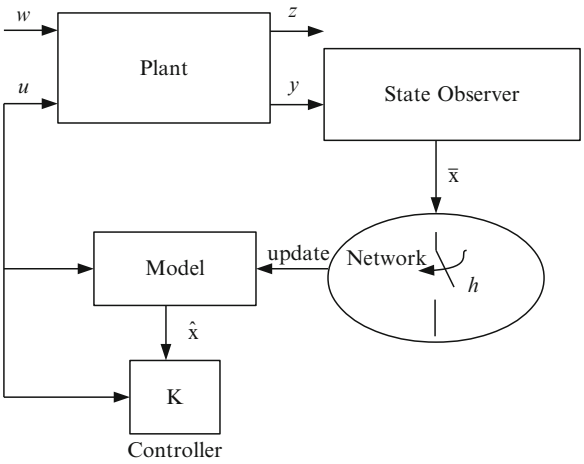

Fig. 10.1 MB-NCS with disturbance input and objective signal output

As an example of the application of this lifting technique we will compute the lifted parameters of a MB-NCS with output feedback. A disturbance signal w and a performance or objective signal z are included in the setup. We will use an observer that will estimate the plant state. This estimate will then be send through a network to the controller. The controller makes use of a plant model, which is updated with the observer estimate, to reconstruct the actual plant state in between updates. The model state is then used to generate the control signal. See Fig. 10.1. We will assume that the plant model is updated at regular intervals h.

We will start by defining the system dynamics:

$$
\begin{aligned}
&\text{Plant Dynamics :}\\
&\qquad \dot{x} = Ax + B_1 w + B_2 u\\
&\qquad z = C_1 x + D_{12} u\\
&\qquad y = C_2 x + D_{21} w + D_{22} u\\
&\text{Observer Dynamics :}\\
&\qquad \dot{\bar{x}} = \left(\hat{A} - L\hat{C}_2\right)\bar{x} + \left(\hat{B}_2 - L\hat{D}_{22}\right)u + Ly\\
&\text{Model Dynamics :}\\
&\qquad \dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}_2 u\\
&\text{Controller :}\\
&\qquad u = K\hat{x}
\end{aligned}
\tag{10.4}
$$

The model state $\hat{x}$ is updated with the observer state $\overline{x}$ every h seconds. The augmented state contains $[x\ \overline{x}\ e]^T$. The augmented system dynamics are given by:

$$
\begin{aligned}
&G_{zw}: \\
&\begin{bmatrix} \dot{x} \\ \dot{\overline{x}} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & B_2K & -B_2K \\ LC_2 & \hat{A} - L\hat{C}_2 + \hat{B}_2K + L\widetilde{D}_{22}K & -\hat{B}_2K - L\widetilde{D}_{22}K \\ LC_2 & L\widetilde{D}_{22}K - L\hat{C}_2 & \hat{A} - L\widetilde{D}_{22}K \end{bmatrix} \begin{bmatrix} x \\ \overline{x} \\ e \end{bmatrix} + \begin{bmatrix} B_1 \\ LD_{21} \\ LD_{21} \end{bmatrix} w \\
&z = \begin{bmatrix} C_1 & D_{12}K & -D_{12}K \end{bmatrix} \begin{bmatrix} x \\ \overline{x} \\ e \end{bmatrix}
\end{aligned}
\tag{10.5}
$$

for $t \in [t_k, t_{k+1})$. At the update times $t = t_{k+1}$ we have $e = \overline{x} - \hat{x} = 0$. We will also use the following definitions:

$$
\begin{aligned}
\varphi(t) &= \begin{bmatrix} x(t) \\ \overline{x}(t) \\ e(t) \end{bmatrix}, \Lambda = \begin{bmatrix} A & B_2K & -B_2K \\ LC_2 & \hat{A} - L\hat{C}_2 + \hat{B}_2K + L\widetilde{D}_{22}K & -\hat{B}_2K - L\widetilde{D}_{22}K \\ LC_2 & L\widetilde{D}_{22}K - L\hat{C}_2 & \hat{A} - L\widetilde{D}_{22}K \end{bmatrix} \\
B_N &= \begin{bmatrix} B_1 \\ LD_{21} \\ LD_{21} \end{bmatrix}, C_N = \begin{bmatrix} C_1 & D_{12}K & -D_{12}K \end{bmatrix}.
\end{aligned}
\tag{10.6}
$$

System (10.5) is clearly h periodic, and therefore, after the lifting procedure, we expect to obtain an LTI system of the form:

$$
\widehat{\varphi}_{k+1} = \widehat{A}\widehat{\varphi}_k + \widehat{B}\widehat{w}_k, \widehat{z}_k = \widehat{C}\widehat{\varphi}_k + \widehat{D}\widehat{w}_k. \tag{10.7}
$$

To obtain the operators we "chop" the time response of the system described in (10.6) and evaluate it at times kh:

$$
\begin{aligned}
\varphi(h) &= \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\varphi(0) + \int_0^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda(h-s)}B_N w(s)ds, \\
\varphi(2h) &= \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}\varphi(h) + \int_h^{2h} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda(2h-s)}B_N w(s)ds, \ldots
\end{aligned}
\tag{10.8}
$$

Then the output response for each interval $[kh,(k+1)h)$ is:

$$\begin{aligned} z(t) &= C_N e^{\Lambda t}\varphi(0) + C_N \int_0^t e^{\Lambda(t-s)} B_N w(s) ds, && t \in [0, h) \\ z(t) &= C_N e^{\Lambda(t-h)}\varphi(h) + C_N \int_h^t e^{\Lambda(t-s-h)} B_N w(s) ds, && t \in [h, 2h) \\ &\dots \end{aligned} \tag{10.9}$$

We can now extract the lifted parameters:

$$\begin{aligned} \widehat{A} &= \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}, \quad \widehat{B}\,\widehat{w} = \int_0^h \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda(h-s)} B_N \widehat{w}(s) ds \\ \widehat{C} &= C_N e^{\Lambda\tau}, \quad \widehat{D}\widehat{w} = C_N \int_0^\tau e^{\Lambda(\tau-s)} B_N \widehat{w}(s) ds. \end{aligned} \tag{10.10}$$

The new lifted system (10.10) is a LTI discrete system. Note the dimension of the state space is left unchanged, but the input and output spaces are now infinite dimensional. Nevertheless, the new representation allows extending the results available for discrete LTI systems to the lifted domain. These tools have been traditionally used to analyze and synthesize sample and hold devices, and digital controllers. It is to be noted, though, that in this case the discrete part embedded in the controller doesn't operate in the same way a typical sampled system does. Here, for instance, the controller gain operates over a continuous signal, as opposed to over a discrete signal as it is customary in sampled data systems.

10.1.2 An H_2 Norm Characterization of a MB-NCS

It is clear that, since the MB-NCS is h-periodic, there is no transfer function in the normal sense whose H_2 norm can be calculated [42]. For LTI systems the H_2 norm can be computed by obtaining the 2-norm of the impulse response of the system at $t = t_0$. We will extend this definition to specify an H_2 norm, or more properly, to define an H_2-like performance index [42]. We will call this performance index Extended H_2 Norm. We will study the extended H_2 norm of the MB-NCS with output feedback studied in the previous section and shown in Fig. 10.1. The Extended H_2 Norm is defined as:

$$\|G_{zw}\|_{xh2} = \left(\sum_i \|G_{zw}\delta(t_0)e_i\|_2^2 \right)^{1/2}. \tag{10.11}$$

The extended H_2 norm is an approach to creating a framework for performance analysis. The extended H_2 norm is defined in (10.11) as the square root of the sum

of the squares of the 2-norms of the impulse response of the system on each input. In this case i represents the index for the inputs, G_{zw} represents the plant dynamics, δ is the impulse operator, and e_i is a vector on the input space with zeros except on the ith input, where it has a 1.

Theorem 10.1 *The Extended H_2 Norm, $\|G\|_{xh2}$, of the Output Feedback MB-NCS described in (10.5) is given by $\|G\|_{xh2} = (trace(B_N{}^T X B_N))^{1/2}$ where X is the solution of the discrete Lyapunov equation $M(h)^T X M(h) - X + W_o(0,h) = 0$ and $W_o(0,h)$ is the observability Gramian $W_o(0,h) = \int_0^h e^{\Lambda^T t} C_N{}^T C_N e^{\Lambda t} dt$.*

Proof We will compute the extended H_2 norm of the system by obtaining the 2-norm of the objective signal z to an impulse input $w = \delta(t - t_0)$. It can be shown that the response of the system to an input $w = \delta(t - t_0)$ (assuming that the input dimension is one) is:

$$\begin{aligned} z(t) &= [C_1 \quad D_{12}K \quad -D_{12}K]\varphi(t), \\ \varphi(t) &= e^{\Lambda(t-t_k)} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h} \right)^k \begin{bmatrix} B_1 \\ LD_{21} \\ LD_{21} \end{bmatrix} \end{aligned} \tag{10.12}$$

with

$$\begin{aligned} \varphi(t) &= \begin{bmatrix} x(t) \\ \overline{x}(t) \\ e(t) \end{bmatrix}, \\ \Lambda &= \begin{bmatrix} A & B_2K & -B_2K \\ LC_2 & \hat{A} - L\hat{C}_2 + \hat{B}_2K + L\widetilde{D}_{22}K & -\hat{B}_2K - L\widetilde{D}_{22}K \\ LC_2 & L\widetilde{D}_{22}K - L\hat{C}_2 & \hat{A} - L\widetilde{D}_{22}K \end{bmatrix}, \\ h &= t_{k+1} - t_k \end{aligned}$$

So we can compute the 2-norm of the output:

$$\begin{aligned} \|z\|_2^2 &= \int_{t_0}^{\infty} z(t)^T z(t) dt \\ &= \int_{t_0}^{\infty} B_N{}^T \left(M(h)^T \right)^k e^{\Lambda^T (t-t_k)} C_N{}^T C_N e^{\Lambda(t-t_k)} (M(h))^k B_N dt \end{aligned} \tag{10.13}$$

where $M(h) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda h}, B_N = \begin{bmatrix} B_1 \\ LD_{21} \\ LD_{21} \end{bmatrix}, C_N = [C_1 \quad D_{12}K \quad -D_{12}K]$.

It is easy to see that the norm of a system with more than one input can be obtained by taking the norm of the integral shown in (10.13). So at this point we can

drop our assumption of working with a one-dimensional input system. We will concentrate now on the integral expression (10.13).

$$\begin{aligned}
\Sigma(h) &= \int_{t_0}^{\infty} {B_N}^T \left(M(h)^T\right)^k e^{\Lambda^T(t-t_k)} {C_N}^T C_N e^{\Lambda(t-t_k)} (M(h))^k B_N dt \\
&= {B_N}^T \left(\int_{t_0}^{\infty} \left(M(h)^T\right)^k e^{\Lambda^T(t-t_k)} {C_N}^T C_N e^{\Lambda(t-t_k)} (M(h))^k dt \right) B_N \\
&= {B_N}^T \left(\sum_{i=0}^{\infty} \int_{t_i}^{t_{i+1}} \left(M(h)^T\right)^i e^{\Lambda^T(t-t_i)} {C_N}^T C_N e^{\Lambda(t-t_i)} (M(h))^i dt \right) B_N \\
&= {B_N}^T \left(\sum_{i=0}^{\infty} \int_{0}^{h} \left(M(h)^T\right)^i e^{\Lambda^T t} {C_N}^T C_N e^{\Lambda t} (M(h))^i dt \right) B_N \\
&= {B_N}^T \left(\sum_{i=0}^{\infty} \left(M(h)^T\right)^i \left(\int_{0}^{h} e^{\Lambda^T t} {C_N}^T C_N e^{\Lambda t} dt \right) (M(h))^i \right) B_N \\
&= {B_N}^T \left(\sum_{i=0}^{\infty} \left(M(h)^T\right)^i W_o(0,h) (M(h))^i \right) B_N
\end{aligned} \tag{10.14}$$

where

$$W_o(0,h) = \int_{0}^{h} e^{\Lambda^T t} {C_N}^T C_N e^{\Lambda t} dt. \tag{10.15}$$

Note that (10.15) has the form of the observability Gramian. Also note that the summation resembles the solution of a discrete Lyapunov equation. This Lyapunov equation can be expressed as:

$$M(h)^T X M(h) - X + W_o(0,h) = 0. \tag{10.16}$$

In this equation we note that $M(h)$ is a stable matrix if and only if the networked system is stable. Also $W_o(0,h)$ is a positive semi-definite matrix. Under these conditions the solution X will be positive semi-definite. ♦

Note that the observability Gramian can be factorized as $W_o(0,h) = C_{aux}^T C_{aux} = \int_{0}^{h} e^{\Lambda^T t} {C_N}^T C_N e^{\Lambda t} dt$. This allows us to compute the norm of the system as the norm of an equivalent discrete LTI system.

Corollary 10.2 *Define* $C_{aux}^T C_{aux} = \int_{0}^{h} e^{\Lambda^T t} {C_N}^T C_N e^{\Lambda t} dt$ *and the auxiliary discrete system* G_{aux} *with parameters:* $A_{aux} = M(h)$, $B_{aux} = B_n$, C_{aux}, *and* $D_{\text{aux}} = 0$. *Then the following holds:*

$$\|G_{zw}\|_{xh2} = \|G_{aux}\|_2 \tag{10.17}$$

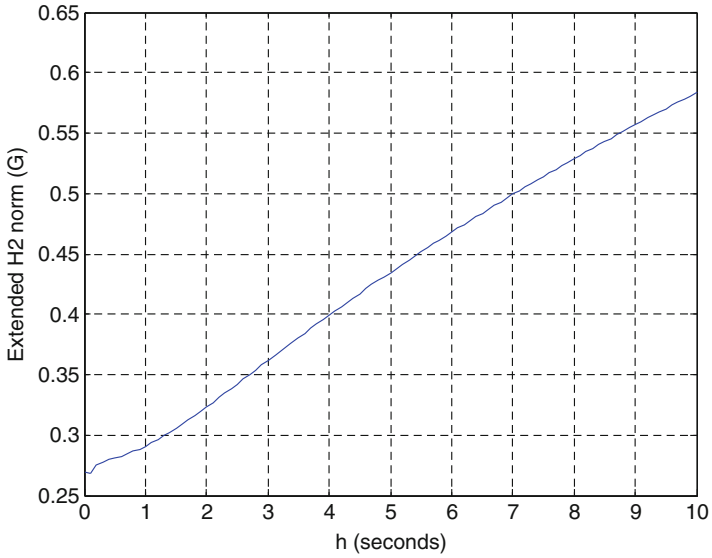

Fig. 10.2 Extended H_2 norm of the instrument servo system as a function of the update time intervals

Example 10.1 Consider the output feedback instrument servo example presented in Example 3.1. Consider the model parameters $\hat{\alpha} = 1$, $\hat{\beta} = 3$ and the real parameters are given by $\alpha = 0.89$, $\beta = 2.79$. In addition we have that $\hat{D}_{22} = 0$ and the plant parameters associated with the disturbance input and the performance output are given by

$$B_1 = \begin{bmatrix} 0.2371 \\ 0.1489 \end{bmatrix}; \quad C_1 = [0.4312 \quad -0.3156]; \quad D_{12} = 0.1452; \quad D_{21} = -0.1974; \quad D_{22} = 0.$$

We will use the state feedback controller $K = [-0.1667, \quad -0.1667]$. The state estimator gain is $L = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$. In Fig. 10.2 we plot the extended H_2 norm of the system as a function of the update time intervals h.

10.1.3 A Generalized H_2 Norm for MB-NCS

In the previous subsection the Extended H_2 Norm was introduced to study the performance of MB-NCS. This norm was defined as the norm of the output of the system when a unit impulse at $t = t_0$ is applied to the input. But since the MB-NCS is a time-varying system it may seem inappropriate to apply this input only at $t = t_0$. By letting the input be $\delta(t - \tau)$ applied at time $t = \tau$ we arrive to an alternate

definition. Since the system is h periodic we only need to consider $\tau \in [t_0, t_0+h)$. We will call this norm the Generalized H_2 Norm and we define it as:

$$\|G_{zw}\|_{gh2} = \left(\frac{1}{h}\int_{t_0}^{t_0+h}\left(\sum_i \|G_{zw}\delta(t-\tau)e_i\|_2^2\right)d\tau\right)^{1/2}. \tag{10.18}$$

A detailed study of this norm can be found in [42]. Note that this norm evaluates the time average of the system response to the impulsive function applied at different times. We will now show some relations concerning this norm.

Let a continuous-time linear transformation $G : L_2[0,h) \to L_2[0,\infty)$ be defined by:

$$(Gu)(t) = \int_0^t g(t,\tau)u(\tau)d\tau. \tag{10.19}$$

where $g(t,\tau)$ is the impulse response of G. Let G be periodic and let its Hilbert-Schmidt norm $\|G\|_{HS}$ be defined as:

$$\|G\|_{HS} = \left(\int_0^h\int_0^\infty trace\left[g(t,\tau)^T g(t,\tau)\right]dtd\tau\right)^{1/2}. \tag{10.20}$$

Then it is clear that:

$$\|G_{zw}\|_{gh2} = \frac{1}{\sqrt{h}}\|G_{zw}\|_{HS}. \tag{10.21}$$

Note the slight abuse of notation: originally G_{zw} was considered a transformation with domain $L_2[0,\infty)$ while the Hilbert-Schmidt norm in (10.20) is defined for transformations with domain on $L_2[0,h)$. Now denote the lifted operator $\widehat{G}_{zw} = LG_{zw}L^{-1}$ where the input-output relation is given by the convolution:

$$\widehat{z}_k = \sum_{l=0}^{k}\widehat{g}_{k-l}\widehat{w}_l \tag{10.22}$$

where $\widehat{g}_k : L_2[0,h) \to L_2[0,h)$ and $\left(\widehat{g}_k u\right)(t) = \int_0^h g(t+kh,\tau)u(\tau)d\tau$. The Hilbert-Schmidt operator for $\widehat{g}_k$ is given by:

$$\left\|\widehat{g}_k\right\|_{HS} = \left(\int_0^h\int_0^h trace\left[g(t+kh,\tau)^T g(t+kh,\tau)\right]dtd\tau\right)^{1/2}. \tag{10.23}$$

Then it is easy to show that:

$$\|G_{zw}\|_{HS}^2 = \sum_{k=0}^{\infty} \left\|\widehat{g}_k\right\|_{HS}^2 = \left\|\widehat{g}\right\|_2^2. \tag{10.24}$$

The last expression shows a relationship between the discrete lifted representation of the system and the Generalized H_2 Norm. Finally we will show the relationship between the Generalized H_2 Norm and the norm of an operator-valued transfer function:

$$\widetilde{g}(\lambda) = \sum_{k=0}^{\infty} \widehat{g}_k \lambda^k \tag{10.25}$$

By defining in a similar way the λ-transform for the input and output of the system we obtain:

$$\widetilde{z}(\lambda) = \widetilde{g}(\lambda)\widetilde{w}(\lambda). \tag{10.26}$$

Note that $\widetilde{w}(\lambda)$ and $\widetilde{z}(\lambda)$, for every λ in their respective regions of convergence, are functions on $[0, h)$; $\widetilde{g}(\lambda)$ is a Hilbert-Schmidt operator. Define the Hardy space $H_2(D, HS)$ with operator-valued functions that are analytic in the open unit disc, boundary functions on ∂D, and with finite norm:

$$\|\widetilde{g}\|_2 = \left[\frac{1}{2\pi}\int_0^{2\pi} \left\|\widetilde{g}\left(e^{j\theta}\right)\right\|_{HS}^2 d\theta\right]^{1/2}. \tag{10.27}$$

Note the norm in $H_2(D, HS)$ is a generalization of the norm in $H_2(D)$ by replacing the trace norm by the Hilbert-Schmidt norm. It can be shown that:

$$\|G_{zw}\|_{HS}^2 = \left\|\widehat{g}\right\|_2^2 = \|\widetilde{g}\|_2^2. \tag{10.28}$$

We will now show how to calculate the Generalized H_2 Norm of the Output Feedback MB-NCS in (10.5). Define the auxiliary discrete LTI system:

$$G_{aux} \stackrel{S}{=} \left[\begin{array}{c|c} \widehat{A} & B_{aux} \\ \hline C_{aux} & 0 \end{array}\right] \tag{10.29}$$

where:

$$\begin{aligned} B_{aux}B_{aux}^T &= \int_0^h \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda\tau} B_N B_N^T e^{\Lambda^T\tau} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) d\tau \\ C_{aux}^T C_{aux} &= \int_0^h \left(e^{\Lambda^T\tau} C_N^T C_N e^{\Lambda\tau} \right) d\tau. \end{aligned} \tag{10.30}$$

Theorem 10.3 *The Generalized H_2 Norm, $\|G_{zw}\|_{gh2}$, of the Output Feedback MB-NCS described in (10.5) is given by* $\|G_{zw}\|_{gh2} = \frac{1}{\sqrt{h}} \left(\left\| \widehat{D} \right\|_{HS}^2 + \|G_{aux}\|_2^2 \right)^{1/2}$.

Proof The transfer function for G_{zw} can be written as:

$$\widetilde{g}(\lambda) = \widehat{D} + \widehat{C}\left(\widetilde{g}_t(\lambda)\right)\widehat{B} \tag{10.31}$$

with $\widetilde{g}_t(\lambda) = \left[\begin{array}{c|c} \widehat{A} & I \\ \hline I & 0 \end{array}\right]$. Note that $\widetilde{g}_t(\lambda)$ is a matrix-valued function and that $\widetilde{g}_t(0) = 0$. Therefore the two functions on the right of (10.31) are orthogonal and:

$$h\|G_{zw}\|_{gh2}^2 = \|\widetilde{g}(\lambda)\|_2^2 = \left\|\widehat{D}\right\|_{HS}^2 + \left\|\widehat{C}\left(\widetilde{g}_t(\lambda)\right)\widehat{B}\right\|_2^2. \tag{10.32}$$

The second norm on the right can be calculated as:

$$\left\|\widehat{C}\left(\widetilde{g}_t(\lambda)\right)\widehat{B}\right\|_2^2 = \frac{1}{2\pi}\int_0^{2\pi} \left\|\widehat{C}\left(\widetilde{g}_t\left(e^{j\theta}\right)\right)\widehat{B}\right\|_{HS}^2 d\theta. \tag{10.33}$$

By fixing θ the integrand $F = \widehat{C}\left(\widetilde{g}_t\left(e^{j\theta}\right)\right)\widehat{B}$ is a Hilbert-Schmidt operator with impulse response:

$$f(t,\tau) = C_N e^{\Lambda t}\widetilde{g}_t\left(e^{j\theta}\right) \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} e^{\Lambda(h-\tau)} B_N. \tag{10.34}$$

Then:

$$\begin{aligned}
\|F\|_{HS}^2 &= trace\left[\int_0^h\int_0^h f(t,\tau)^* f(t,\tau)dtd\tau\right] \\
&= trace\left[\int_0^h\int_0^h B_N^T e^{\Lambda^T(h-\tau)}\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}\widetilde{g}_t\left(e^{j\theta}\right)^* e^{\Lambda^T t}C_N^T C_N e^{\Lambda t}\widetilde{g}_t\left(e^{j\theta}\right)\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}e^{\Lambda(h-\tau)}B_N dtd\tau\right] \\
&= trace\left[\int_0^h B_N^T e^{\Lambda^T(h-\tau)}\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}\widetilde{g}_t\left(e^{j\theta}\right)^* C_{aux}^T C_{aux}\widetilde{g}_t\left(e^{j\theta}\right)\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}e^{\Lambda(h-\tau)}B_N d\tau\right] \\
&= trace\left[\int_0^h\left(\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}e^{\Lambda(h-\tau)}B_N B_N^T e^{\Lambda^T(h-\tau)}\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}\right)d\tau\left(\widetilde{g}_t\left(e^{j\theta}\right)^* C_{aux}^T C_{aux}\widetilde{g}_t\left(e^{j\theta}\right)\right)\right] \\
&= trace\left[B_{aux}B_{aux}^T\widetilde{g}_t\left(e^{j\theta}\right)^* C_{aux}^T C_{aux}\widetilde{g}_t\left(e^{j\theta}\right)\right] = trace\left[B_{aux}^T\widetilde{g}_t\left(e^{j\theta}\right)^* C_{aux}^T C_{aux}\widetilde{g}_t\left(e^{j\theta}\right)B_{aux}\right].
\end{aligned} \tag{10.35}$$

So (10.33) can be calculated as the H_2 norm of $C_{aux}\widetilde{g}_t\left(e^{j\theta}\right)B_{aux}$ which corresponds to the H_2 norm of G_{aux}. ♦

To determine the Generalized H_2 Norm several calculations need to be done, among these are:

$$\begin{aligned}
&\left\|\widehat{D}\right\|_{HS}^2 = trace\left(\int_0^h\int_0^t B_N^T e^{\Lambda^T\tau}C_N^T C_N e^{\Lambda\tau}B_N d\tau dt\right) \\
&B_{aux}B_{aux}^T = \begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}P_{22}^T P_{12}\begin{bmatrix} I & 0 & 0\\ 0 & I & 0\\ 0 & 0 & 0\end{bmatrix}, \text{with } \begin{bmatrix} P_{11} & P_{12}\\ 0 & P_{22}\end{bmatrix} = \exp\left(h\begin{bmatrix} -\Lambda & B_N B_N^T\\ 0 & \Lambda^T\end{bmatrix}\right) \\
&C_{aux}^T C_{aux} = M_{22}^T M_{12}, \text{with } \begin{bmatrix} M_{11} & M_{12}\\ 0 & M_{22}\end{bmatrix} = \exp\left(h\begin{bmatrix} -\Lambda^T & C_N^T C_N\\ 0 & \Lambda\end{bmatrix}\right).
\end{aligned} \tag{10.36}$$

Note that in this particular case it was relatively easy to separate the infinite dimensional components of the system from a finite dimensional core component. This is not always possible. In particular one might be tempted to apply the previous techniques to obtain a finite dimensional auxiliary discrete LTI that can be used to solve an H_2 optimal control problem. The described separation technique cannot be carried out since the controller and observer gains operate over continuous signals. Nevertheless it will be shown later how to address an H_2 optimal control problem using other techniques.

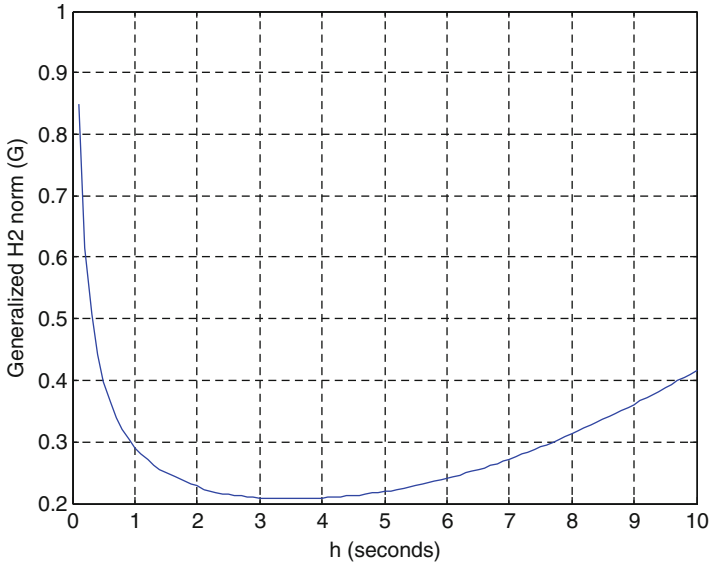

Fig. 10.3 Generalized H_2 norm of the instrument servo system as a function of the update time intervals

Example 10.2 Consider the model and plant parameters used for the instrument servo in Example 10.1. We also use the same controller and observer gains derived for that example. The generalized H_2 norm has been computed for different update time intervals h. The results are shown in Fig. 10.3.

Since both performance measurements are defined in a different manner no real comparison can be made between them. It seems though that the Generalized H_2 Norm is more appropriate since it considers the application of the impulse input at different times. Also its link with a well-defined operator-valued transfer function makes it very attractive. The next subsection presents an alternate parameter representation that overcomes the inconveniencies of dealing with infinite dimensional operators.

10.2 Optimal Controller Design for MB-NCS

In this subsection we address the issue of designing optimal controllers for MB-NCS. We saw previously that lifting can transform a periodic system such as the MB-NCS into a discrete LTI system. Most results for the design of optimal controllers for discrete systems directly apply to the lifted system. Since the parameters of the lifted system are infinite dimensional, computations using the integral representation given in (10.10) can be difficult. This is evident when one

considers operators such as $\left(I - \widehat{D}^* \widehat{D}\right)^{-1}$, which appears for instance in sampled data H_∞ problems.

To circumvent some of the problems associated with optimal control problems, an auxiliary discrete LTI system is obtained so that its optimal controller also optimizes the lifted system. The separation of the infinite dimensionality from the problem is not always guaranteed. In particular we note that the controller for the auxiliary system works in the discrete-time domain while the controller for the lifted system representing the MB-NCS in (10.10) works in continuous time. This means the controller has to be obtained using the lifted parameters directly.

In this subsection we start by giving a brief summary of an alternative representation of the lifted parameters proposed by Mirkin and Palmor [181]. This alternative representation allows performing complex computations using lifted parameters directly. Results on the computation of an optimal sampler, hold, and controller are shown and their equivalence with the components of the output feedback MB-NCS is shown.

The representation of lifted parameters proposed in [181] considers the lifted parameters as continuous LTI systems operating over a finite time interval. The main advantage of such representation lies in the possibility of simplifying operations over the parameters to algebraic manipulations over LTI system with two-point boundary conditions (STPBC). These manipulations can then be performed using well-known state-space machinery.

Consider the following LTI STPBC:

$$\begin{aligned} G: \quad & \dot{x}(t) = Ax(t) + Bu(t), \quad \Omega x(0) + \Upsilon x(h) = 0 \\ & y(t) = Cx(t) + Du(t) \end{aligned} \tag{10.37}$$

Here the square matrices Ω and Υ define the boundary conditions. It is said that the boundary conditions are well posed if $x(t) = 0$ is the only solution to (10.37) when $u(t) = 0$. It can be verified that the STPBC G has well-posed boundary conditions if and only if the matrix:

$$\Xi_G = \Omega + \Upsilon e^{Ah} \tag{10.38}$$

is non-singular. If the STPBC G has well-posed boundary conditions, then its response is uniquely determined by the input $u(t)$ and is given by:

$$y(t) = Du(t) + \int_0^h K_G(t,s)u(s)ds \tag{10.39}$$

where the kernel $K_G(t,s)$ is given by:

$$K_G(t,s) = \begin{cases} Ce^{At}\Xi_G^{-1}\Omega e^{-As}B & \text{if } 0 \le s \le t \le h \\ -Ce^{At}\Xi_G^{-1}\Upsilon e^{A(h-s)}B & \text{if } 0 \le t \le s \le h \end{cases}. \tag{10.40}$$

We will use the following notation to represent (10.37):

$$G = \left(\begin{array}{c|c|c} A & \Omega \rightleftharpoons \Upsilon & B \\ \hline C & & D \end{array}\right). \tag{10.41}$$

The following is a list of manipulations that are used to perform operations over STPBCs.

1. Adjoint System

$$G^* = \left(\begin{array}{c|c|c} -A^T & e^{A^T h}\Upsilon^T\Xi_G^{T-1} \rightleftharpoons \Omega^T\Xi_G^{T-1}e^{A^T h} & C^T \\ \hline -B^T & & D^T \end{array}\right). \tag{10.42}$$

2. Similarity Transformation: (for T and S non-singular)

$$TGT^{-1} = \left(\begin{array}{c|c|c} TAT^{-1} & S\Omega T^{-1} \rightleftharpoons S\Upsilon T^{-1} & TB \\ \hline CT^{-1} & & D \end{array}\right). \tag{10.43}$$

3. Addition

$$G_1 + G_2 = \left(\begin{array}{cc|c|c} A_1 & 0 & \multirow{2}{*}{$\begin{bmatrix}\Omega_1 & 0\\ 0 & \Omega_2\end{bmatrix} \rightleftharpoons \begin{bmatrix}\Upsilon_1 & 0\\ 0 & \Upsilon_2\end{bmatrix}$} & \begin{matrix}B_1\\ B_2\end{matrix} \\ 0 & A_2 & & \\ \hline C_1 & C_2 & & D_1 + D_2 \end{array}\right). \tag{10.44}$$

4. Multiplication

$$G_1G_2 = \left(\begin{array}{cc|c|c} A_1 & B_1C_2 & \multirow{2}{*}{$\begin{bmatrix}\Omega_1 & 0\\ 0 & \Omega_2\end{bmatrix} \rightleftharpoons \begin{bmatrix}\Upsilon_1 & 0\\ 0 & \Upsilon_2\end{bmatrix}$} & \begin{matrix}B_1D_2\\ B_2\end{matrix} \\ 0 & A_2 & & \\ \hline C_1 & D_1C_2 & & D_1D_2 \end{array}\right). \tag{10.45}$$

5. Inversion (exists if and only if $\det(D) \neq 0$ and $\det\left(\Omega + \Upsilon e^{\left(A - BD^{-1}C\right)h}\right) \neq 0$)

$$G^{-1} = \left(\begin{array}{c|c|c} A - BD^{-1}C & \Omega \rightleftharpoons \Upsilon & BD^{-1} \\ \hline -D^{-1}C & & D^{-1} \end{array}\right). \tag{10.46}$$

This representation reduces the complexity of computing operators such as $\left(I-\widehat{D}^*\widehat{D}\right)^{-1}$. Using the integral representation of (10.10) one can get that $\xi=(I-\widehat{D}^*\widehat{D})^{-1}\omega$ if and only if:

$$\omega(t)=\xi(t)+\int_t^h B_N^T e^{-\Lambda^T(t-s)}C_N^T C_N\int_0^s e^{\Lambda(s-\tau)}B_N\xi(\tau)d\tau ds \tag{10.47}$$

It is not clear how to solve this equation. On the other hand using the alternative representation we note that:

$$\widehat{D}=\left(\begin{array}{c|c|c}\Lambda & I\rightleftharpoons 0 & B_N\\ \hline C_N & & 0\end{array}\right). \tag{10.48}$$

Using the properties previously listed we obtain:

$$\begin{aligned}\left(I-\widehat{D}^*\widehat{D}\right)^{-1}&=\left(I-\left(\begin{array}{c|c|c}\Lambda & I\rightleftharpoons 0 & B_N\\ \hline C_N & & 0\end{array}\right)^*\left(\begin{array}{c|c|c}\Lambda & I\rightleftharpoons 0 & B_N\\ \hline C_N & & 0\end{array}\right)\right)^{-1}\\ &=\left(\begin{array}{cc|c|c}-\Lambda^T & C_N^TC_N & & 0\\ -B_NB_N^T & \Lambda & \begin{bmatrix}0&0\\0&I\end{bmatrix}\rightleftharpoons\begin{bmatrix}I&0\\0&0\end{bmatrix} & B_N\\ \hline -B_N^T & 0 & & I\end{array}\right).\end{aligned} \tag{10.49}$$

To be able to represent operators with finite dimension domains or ranges such as $\widehat{B}$ and $\widehat{C}$ two new operators are defined. Given a number $\theta\in[0,h]$, the impulse operator I_θ transforms a vector $\eta\in\mathbb{R}^n$ into a modulated impulse as follows:

$$\varsigma=\mathrm{I}_\theta\eta \ \text{ iff } \ \varsigma(t)=\delta(t-\theta)\eta. \tag{10.50}$$

Also define the sample operator I_θ^*, which transforms a continuous function $\varsigma\in\mathbb{C}_n[0,h]$ into a vector $\eta\in\mathbb{R}^n$ as follows:

$$\eta=\mathrm{I}_\theta^*\varsigma \ \text{ iff } \ \eta=\varsigma(\theta). \tag{10.51}$$

Note that the representation of I_θ^* is as the adjoint of I_θ, even when this is not strictly true, it is easy to see that given an $h\geq\theta$ the following equality holds:

$$\langle \varsigma, \mathrm{I}_\theta \eta \rangle = \int_0^h \varsigma(\tau)^T (\mathrm{I}_\theta \eta)(\tau) d\tau = \varsigma(\theta)^T \eta = \langle \mathrm{I}_\theta^* \varsigma, \eta \rangle \tag{10.52}$$

The following is a summary of technical results extracted from [181] that can be used to address problems involving lifted parameters.

Lemma 10.4 *The parameters of a lifted system* $\widehat{G} = LGL^{-1}$ *of a continuous LTI system* $G = \left[\frac{A}{C}\middle|\frac{B}{D}\right]$ *are given by:*

$$\begin{bmatrix} \widehat{A} & \widehat{B} \\ \widehat{C} & \widehat{D} \end{bmatrix} = \begin{bmatrix} \mathrm{I}_h^* & 0 \\ 0 & I \end{bmatrix} \left[\begin{array}{c|cc} A & I & B \\ \hline I & 0 & 0 \\ C & 0 & 0 \end{array}\right] \begin{bmatrix} \mathrm{I}_0 & 0 \\ 0 & I \end{bmatrix} \tag{10.53}$$

Lemma 10.5 *Given an STPBC, G, on the interval [0, h] with* $D = 0$ *then*

$$(G\mathrm{I}_\theta)^* = \mathrm{I}_\theta^* G^* \quad \text{and} \quad \left(\mathrm{I}_\theta^* G\right)^* = G^* \mathrm{I}_\theta \tag{10.54}$$

for any $\theta \in [0, h]$.

Lemma 10.6 *Let* $G = \left(\begin{array}{c|c|c} A & \Omega \rightleftharpoons \Upsilon & B \\ \hline C & & D \end{array}\right)$, *then:*

$$\begin{aligned} \mathrm{I}_h^* G \mathrm{I}_0 &= Ce^{Ah}\left(\Omega + \Upsilon e^{Ah}\right)^{-1} \Omega B, \\ \mathrm{I}_0^* G \mathrm{I}_h &= -C\left(\Omega + \Upsilon e^{Ah}\right)^{-1} \Upsilon B, \end{aligned} \tag{10.55}$$

and if in addition $CB = 0$, then

$$\begin{aligned} \mathrm{I}_0^* G \mathrm{I}_0 &= C\left(\Omega + \Upsilon e^{Ah}\right)^{-1} \Omega B, \\ \mathrm{I}_h^* G \mathrm{I}_h &= -Ce^{Ah}\left(\Omega + \Upsilon e^{Ah}\right)^{-1} \Upsilon B. \end{aligned} \tag{10.56}$$

Lemma 10.7 *Let*

$$\begin{aligned} G_1 &= \left(\begin{array}{c|c|c} A_1 & \Omega_1 \rightleftharpoons \Upsilon_1 & B_1 \\ \hline C_1 & & D_1 \end{array}\right), \quad G_{C1} = \left(\begin{array}{c|c|c} A_1 & \Omega_1 \rightleftharpoons \Upsilon_1 & I \\ \hline C_1 & & 0 \end{array}\right), \\ G_2 &= \left(\begin{array}{c|c|c} A_2 & \Omega_2 \rightleftharpoons \Upsilon_2 & B_2 \\ \hline C_2 & & D_2 \end{array}\right), \quad G_{B2} = \left(\begin{array}{c|c|c} A_2 & \Omega_2 \rightleftharpoons \Upsilon_2 & B_2 \\ \hline I & & 0 \end{array}\right), \end{aligned} \tag{10.57}$$

and λ_1, λ_2, *be either 0 or 1. Then for any appropriately dimensioned matrix M:*

$$G_1G_2 + G_{C1}\mathrm{I}_{\lambda_1 h} M \mathrm{I}^*_{\lambda_2 h} G_{B2}$$
$$= \left(\begin{array}{cc|c|c} A_1 & B_1C_2 & \multicolumn{1}{c}{\multirow{2}{*}{$\left[\begin{array}{cc} \Omega_1 & (1-\lambda_2)M_1 \\ 0 & \Omega_2 \end{array}\right] \rightleftharpoons \left[\begin{array}{cc} \Upsilon_1 & \lambda_2 M_1 \\ 0 & \Upsilon_2 \end{array}\right]$}} & B_1D_2 \\ 0 & A_2 & & B_2 \\ \hline C_1 & D_1C_2 & & D_1D_2 \end{array}\right) \tag{10.58}$$

where $M_1 = (\lambda_1 \Upsilon_1 - (1-\lambda_2)\Omega_1)M$.

The presented results allow us to make effective use of the impulse and sample operator. Namely the last two lemmas show how to absorb the impulse operators into an STPBC. Now let us present a result that links the solutions of the lifted algebraic discrete Riccati equation and the algebraic continuous Riccati equation for the continuous system for the H_2 control problem.

Lemma 10.8 *Let the lifted algebraic discrete Riccati equation for the lifted system* $\widehat{G} = LGL^{-1}$ *be as follows:*

$$\widehat{A}^T \widehat{X} \widehat{A} - \widehat{X} + \widehat{C}^* \widehat{C} - \left(\widehat{A}^T \widehat{X} \widehat{B} + \widehat{C}^* \widehat{D}\right)\left(\widehat{D}^* \widehat{D} + \widehat{B}^T \widehat{X} \widehat{B}\right)^{-1}\left(\widehat{D}^* \widehat{C} + \widehat{B}^T \widehat{X} \widehat{A}\right) = 0 \tag{10.59}$$

and let the algebraic continuous Riccati equation for G be:

$$A^T X + XA + C^T C - \left(XB + C^T D\right)\left(D^T D\right)^{-1}\left(D^T C + B^T X\right) = 0. \tag{10.60}$$

Then the conditions for existence of a unique stable solution for both Riccati equations are equivalent. Moreover if they exist, then $\widehat{X} = X$.

This implies that in order to solve the optimal control problem we just need to solve the regular continuous Riccati equation. We can for example obtain the optimal H_2 state feedback "gain" given by $\widehat{F} = -\left(\widehat{D}^* \widehat{D} + \widehat{B}^* X \widehat{B}\right)^{-1}\left(\widehat{D}^* \widehat{C} + \widehat{B}^* X \widehat{A}\right)$. It can be shown [181] that:

$$\widehat{F} = \left(\begin{array}{c|c|c} A + BF & I \rightleftharpoons 0 & I \\ \hline F & & 0 \end{array}\right). \tag{10.61}$$

Here F is the H_2 optimal control gain for the continuous system. Note that the expression in (10.61) exactly represents the dynamics of the actuator/controller for the state feedback MB-NCS when the modeling errors are zero and the feedback gain is the H_2 optimal feedback gain. Finally, we present the following result that obtains the H_2 optimal sampler, hold and controller.

Lemma 10.9 *Given the standard assumptions, when the hold device is given by* $(Hu)(kh+\tau) = \phi_H(\tau)u_k, \forall\, \tau \in [0,h]$ *and the sample device is given by* $(Sy)_k = \int_0^h \phi_S(\tau)y(kh-\tau)d\tau$, *the* H_2 *optimal hold, sampler, and discrete controller for the lifted system* $\widehat{G} = LGL^{-1}$ *with* $G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right]$ *are as follows:*

$$\begin{aligned} &\text{Hold}: \quad \phi_H(\tau) = Fe^{(A+B_2F)\tau} \\ &\text{Sampler}: \quad \phi_S(\tau) = -e^{(A+LC_2)\tau}L \\ &\text{Controller}: \quad K_d = \left[\begin{array}{c|c} \Theta & I \\ \hline I & 0 \end{array}\right] \\ &\text{where}: \quad \Theta = e^{(A+B_2F)h} + \int_0^h e^{(A+LC_2)(h-\tau)}LC_2e^{(A+B_2F)\tau}d\tau \end{aligned} \tag{10.62}$$

Remark Note that the H_2 optimization problem solved in [181] is related to the Generalized H_2 norm previously presented. That is, replacing the trace norm with the Hilbert-Schmidt norm.

As it has been observed, there is a strong connection between the H_2 optimal hold of a sampled system and the H_2 optimal controller of the non-sampled system. As pointed out in [181] it is clear that the H_2 optimal hold attempts to recreate the optimal control signal that would have been generated by the H_2 optimal controller in the non-sampled case. That is, the H_2 optimal hold calculated in [181] generates a control signal identical to the one generated by the non-sampled H_2 optimal controller in the absence of noise and disturbances.

Another connection exists between the H_2 optimal sampler, hold, and discrete controller calculated in [181] and the output feedback MB-NCS. It can be readily shown that, under certain conditions, the optimal hold has the same dynamics as the controller/actuator in the output feedback MB-NCS. This is true when the modeling errors are zero and the gain is the optimal H_2 gain. The same equivalence can be shown for the combination of the optimal sampler and discrete controller dynamics and the output feedback MB-NCS observer.

The techniques shown here can be used to solve robust optimal control problems that consider the modeling error. This is possible due to the alternative representation that allows the extension of traditional optimal control synthesis techniques to be used with the infinite dimensional parameters that appear in the lifted domain.

10.3 Notes and References

Optimal performance and optimal controller design represent an important subject in control systems. In Sect. 10.1 performance of the MBNCS framework using periodic update intervals was analyzed based on lifting techniques. The results in

Sect. 10.2 provided a technique to design optimal controllers for MB-NCS with periodic updates. In many scenarios the decision on whether or not to transmit certain measurements is also an important part of the overall performance of control systems that send information through a limited bandwidth communication channel. In Chap. 9 we considered this situation. The main goal is to find the optimal transmission instants which are not periodic in general. Section 10.2 also contains discussion of the techniques described in [181] and their application to optimal control synthesis problems. Specifically, here an alternative representation of the lifted system parameters is introduced. This allows for efficient computation of the optimal gains for the controller and observer.

The work described in this chapter can also be found in [185, 190]. Description and applications of lifting techniques in general can be found in [16, 26, 42, 63, 130, 181].

Chapter 11
Reference Input Tracking

Previous chapters of this book have mainly considered the stabilization problem under different communication constraints. The focus of this chapter is the design of model-based controllers for tracking of external reference inputs and the analysis of performance of the Model-Based Networked Control System (MB-NCS) under the main communication constraint studied in this book which is the absence of feedback measurem/ents for extended periods of time. In general, the exact tracking of a reference input in a Networked Control System (NCS) is a difficult problem mainly because we can measure and use feedback only at discrete-time instants. The difficulty is increased if there is uncertainty in the plant parameters since control methodologies that achieve a zero steady-state error rely on the exact knowledge of these parameters. In this chapter a controller is designed based on the model of the plant. An MB-NCS configuration is used for the analysis of the resulting NCS that includes the model, the controller, and the plant.

In this chapter, two different configurations are proposed for the tracking problem over networks. The first one assumes the whole state can be measured and can be used to update the model state. A periodic communication strategy is implemented which permits the tracking of different types of reference signals. The second configuration is intended for applications where state measurements cannot be obtained and only a system's output can be measured. The model in this case is implemented in a very different form than previously seen. The state-space model is changed to an input-output model of the system. This type of modeling provides a great advantage with respect to the types of uncertainty we are able to deal with. By using this configuration, we are able to control systems with uncertain parameters and also with unknown order dimension of the state space. In previous chapters it was always assumed that the dimension of the system is known. In this chapter we relax the known order constraint and use a model of the system that is not necessarily of the same order as the real plant. Section 11.1 describes the reference input tracking controller using state feedback and using the model-based approach. The analysis of the resulting MB-NCS is presented in Sect. 11.2. Section 11.3 presents a model-based configuration for tracking reference inputs using discrete-time systems and

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_11,

using input-output model representations of the system. This section also includes an extension to consider network induced delays using the described model-based configuration. Notes and references are provided in Sect. 11.4.

11.1 Reference Input Tracking Controller

In this section we consider linear time-invariant systems of the form:

$$\dot{x} = Ax + Bu + Pw \tag{11.1}$$

where $x \in \mathbb{R}^n$ is the state of the plant or physical system to be controlled, $u \in \mathbb{R}^m$ is the control input, and $w \in \mathbb{R}^s$ is the reference input that the controlled system attempts to track or follow. The matrices A, B, and P are of appropriate dimensions. There is no restriction on the stability of the original system. The physical plant may be unstable, i.e., one or more of the eigenvalues of matrix A may have positive real parts. It is assumed that the exogenous signal w is the measurable output of the autonomous linear time-invariant system:

$$\dot{w} = Sw. \tag{11.2}$$

We consider the scenario where the state of the plant is measured by a sensor and broadcasted to the controller through a communication network. This means that the controller does not have access to the state $x(t)$ at all times but only at some instants or periods of time. For our purposes we assume that quantization and delay effects are negligible. We propose an MB-NCS solution in which a model of the physical system is used in the controller node. The model is represented by:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u + \hat{P}w \quad for \;\; t \in [t_k, t_{k+1}) \tag{11.3}$$

where $\hat{x} \in \mathbb{R}^n$ is the state of the model, and at the time instants t_k we have $\hat{x}(t_k) = x(t_k)$. $h = t_{k+1} - t_k$ represents the constant update period. The model parameters $\hat{A}, \hat{B}, \hat{P}$ represent the available model of the physical system. We assume this model is known and we consider model uncertainties resulting from errors in parameter estimation techniques or other possible inaccuracies. $\hat{A}$ does not have to be a stable matrix since it is an estimate of the system matrix A which may be stable or unstable. The state of the model is used by the controller to generate the control input $u = f(\hat{x}, w)$. The controller then has access to the state of the model at all times and when a measurement of the state of the plant arrives at the controller, this measurement is used to update the internal state of the model. A representation of the interconnections within the actuator node in this case is shown in Fig. 11.1.

The dynamics of the model and the plant are similar to those of standard MB-NCS schemes but now we include a reference signal w which is only available

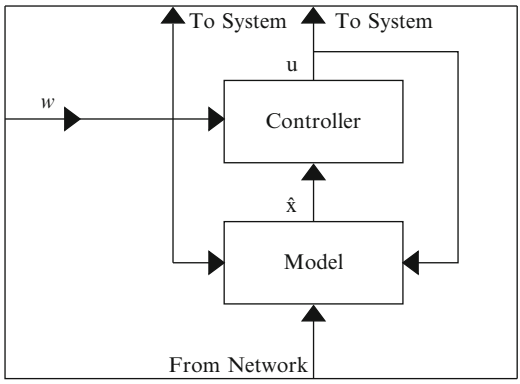

Fig. 11.1 Model-based networked control system actuator/controller node for tracking of a reference input

at the controller node. We follow the procedure in [39] in order to obtain a feedback controller that ensures zero steady-state tracking error for the nominal closed-loop model. Then we study the tracking properties of the networked uncertain system.

The control input is given by:

$$u = \Gamma w + K(\hat{x} - \Pi w). \tag{11.4}$$

Three control parameters are needed in (11.4). The control parameters Π and Γ are obtained by solving the next equations that involve the parameters of the nominal model:

$$\begin{aligned} \Pi S &= \hat{A}\,\Pi + \hat{B}\,\Gamma + \hat{P} \\ 0 &= \hat{C}\,\Pi + Q \end{aligned} \tag{11.5}$$

and the control parameter K is such that the closed-loop matrix $\hat{A} + \hat{B}\,K$ is Hurwitz. The plant tracking error is defined, in general, by:

$$\varepsilon = Cx + Qw. \tag{11.6}$$

This definition allows reference input signals that may not necessarily have the same dimension as the output. A particular case of (11.6) is when each element of the state follows a corresponding element of the reference input. In this case both the state and the reference input need to be of the same dimension.

Before we analyze the MB-NCS using the controller (11.4), let us briefly study the properties of the closed-loop nominal model using the same controller. This is the case in which we only consider the model with control input (11.4) with no model updates taking place. Define the model tracking error:

$$\hat{\varepsilon} = \hat{C}\,\hat{x} + Qw. \tag{11.7}$$

This error characterizes the difference between the model response and the reference input. It is easy to show that (11.4) guarantees zero steady-state model tracking error. To see this we define the auxiliary model tracking error:

$$\hat{\varepsilon}_{ax} = \hat{x} - \Pi w. \tag{11.8}$$

Then, the time derivative of (11.8) is given by:

$$\begin{aligned} \dot{\hat{\varepsilon}}_{ax} &= \dot{\hat{x}} - \Pi \dot{w} \\ &= \hat{A}\hat{x} + \hat{B}(\Gamma w + K\hat{x} - K\Pi w) + \hat{P}w - \left(\hat{A}\Pi + \hat{B}\Gamma + \hat{P}\right)w \\ &= \left(\hat{A} + \hat{B}K\right)\hat{\varepsilon}_{ax}. \end{aligned} \tag{11.9}$$

Also note that $\hat{\varepsilon} = C\hat{\varepsilon}_{ax}$ then the model tracking error is asymptotically stable (zero steady-state error) since $\hat{A} + \hat{B}K$ is Hurwitz.

When we implement this controller in order to stabilize a networked uncertain system and to make that system follow the reference input, we need to consider model uncertainties and unavailability of the state of the plant. This is a more complex problem and the tracking performance of the networked system considerably deteriorates with respect to that of the nominal closed-loop system. The MB-NCS architecture which uses an explicit model of the system to generate an estimate of the state between network update intervals provides a simple approach to improve such performance and to characterize the tracking error of the networked uncertain system.

11.2 Discretized Input Tracking Analysis

In general, an exact model of the physical system is unavailable. The available nominal model (11.3) is used instead to solve the controller equations (11.5). Consider the model tracking error (11.7) and for the moment let $C = \hat{C} = I$ and $Q = -I$, this is the particular case in which each element of the state x has to track or follow the corresponding element of the reference input w.

Since the controller is designed using the model parameters $\hat{A}, \hat{B}, \hat{P}$, the model tracking error time derivative is given by (11.9) with $\hat{\varepsilon} = \hat{\varepsilon}_{ax}$ since $C = I$.

The plant tracking error cannot be expressed directly in terms of itself only, but it follows the more complex dynamics:

$$\dot{\varepsilon} = \dot{x} - \Pi \dot{w} = Ax + B(\Gamma w + K\hat{x} - K\Pi w) + Pw - \left(\hat{A}\Pi + \hat{B}\Gamma + \hat{P}\right)w. \tag{11.10}$$

As it can be seen in (11.10), the parameters of the plant and the model are involved in the dynamics of the error. This is expected because the control input of the plant is given in terms of the state of the model and the controller was designed

using the model parameters. Let us define the error system parameters $\widetilde{A} = A - \hat{A}$, $\widetilde{B} = B - \hat{B}$, $\widetilde{P} = P - \hat{P}$ and use them to express the tracking error dynamics in a more compact way:

$$\dot{\varepsilon} = A\varepsilon + BK\hat{\varepsilon} + \left(\widetilde{A}\Pi + \widetilde{B}\Gamma + \widetilde{P}\right)w. \tag{11.11}$$

Equation (11.11) is a useful result that will give rise to a discrete-time equivalent system as it is shown next.

The aim here is to obtain a discrete-time equivalent description of the dynamics of the model tracking error $\hat{\varepsilon}$ and especially, of the plant tracking error ε. This is done by considering a periodic update interval h, that is, we measure the state of the plant, send this value through the network and use it to update the state of the model every h time units.

Theorem 11.1 *For any bounded reference input the plant tracking error response is bounded if the eigenvalues of F are inside the unit circle, where*

$$F = e^{Ah} + \int_0^h e^{A(h-\tau)}BKe^{\left(\hat{A}+\hat{B}K\right)\tau}d\tau. \tag{11.12}$$

Proof First, we find the response of the model tracking error in the interval $t \in [t_k, t_{k+1})$ with initial condition $\hat{\varepsilon}(t_k)$:

$$\hat{\varepsilon}(t) = e^{\left(\hat{A}+\hat{B}K\right)(t-t_k)}\hat{\varepsilon}(t_k) \tag{11.13}$$

where $t_{k+1} - t_k = h$. Note that at times t_k we update the model $\hat{x}(t_k) = x(t_k)$ which makes $\hat{\varepsilon}(t_k) = \varepsilon(t_k)$ and (11.13) becomes

$$\hat{\varepsilon}(t) = e^{\left(\hat{A}+\hat{B}K\right)(t-t_k)}\varepsilon(t_k). \tag{11.14}$$

Now, the response of the plant tracking error can be found as follows:

$$\varepsilon(t) = e^{A(t-t_k)}\varepsilon(t_k) + \int_{t_k}^{t} e^{A(t-\tau)}BK\hat{\varepsilon}(\tau)d\tau + \int_{t_k}^{t} e^{A(t-\tau)}\left(\widetilde{A}\Pi + \widetilde{B}\Gamma + \widetilde{P}\right)e^{S(\tau-t_k)}d\tau. \tag{11.15}$$

Using (11.14) we can write (11.15) at time t_{k+1} in closed form:

$$\varepsilon(t_{k+1}) = F\varepsilon(t_k) + Gw(t_k) \tag{11.16}$$

where F is given in (11.12) and

$$G = \int_0^h e^{A(h-\tau)} \left(\widetilde{A}\Pi + \widetilde{B}\Gamma + \widetilde{P} \right) e^{S\tau} d\tau. \tag{11.17}$$

Equation (11.16) describes the plant tracking error dynamics at the time instants when the state of the model is updated and it is a function of the values of the tracking error and the bounded reference input at the previous update instant. Then, we can conclude that the error dynamics are bounded for any bounded reference input. By (11.6) we can write for this special case:

$$x(t_k) = \varepsilon(t_k) + w(t_k). \tag{11.18}$$

The state of the plant is the sum of two bounded signals; therefore it is also a bounded signal. ♦

Example 11.1 The system and the model are described respectively by:

$$A = \begin{pmatrix} 1.8634 & 0.8851 \\ -0.1590 & 1.2174 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$
$$\hat{A} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \quad \hat{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \hat{P} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The reference input is a sinusoidal signal which is bounded in magnitude. The reference input matrix is given by:

$$S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

It is assumed that the state can be measured at a constant rate and that every state element is set to follow the corresponding element of the reference input. The constant update interval is given by $h = 0.2$ s.

The controller parameters are given by:

$$K = \begin{pmatrix} -33.6815 & -0.5318 \\ -0.5318 & -32.6509 \end{pmatrix} \quad \Gamma = \begin{pmatrix} -3 & 0 \\ -1 & -2 \end{pmatrix}.$$

The results from the simulations are shown in Figs. 11.2 and 11.3. Figure 11.2 shows the sinusoidal reference input and the response of the system. There is also uncertainty in the initial conditions on the networked system. Before the system starts operating there is no knowledge of the value of the reference input or the

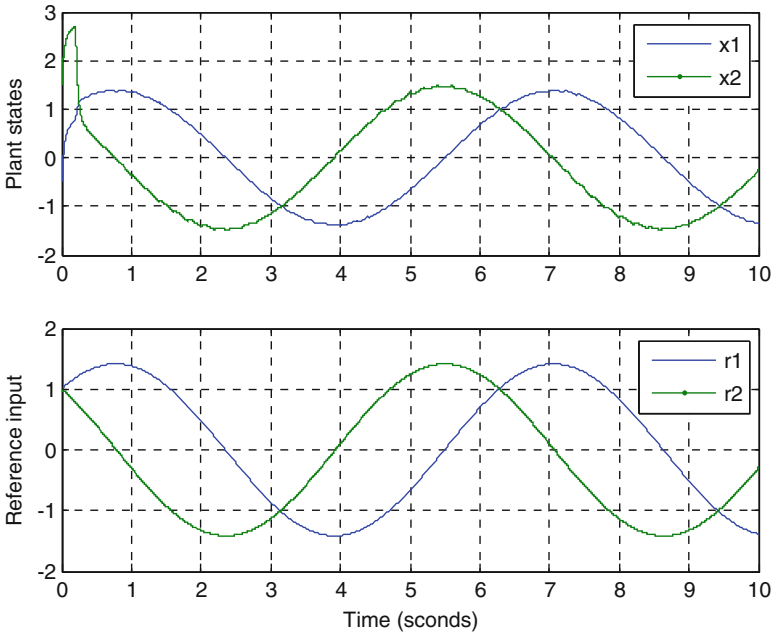

Fig. 11.2 Response of the networked system (*top*) to a sinusoidal reference input (*bottom*)

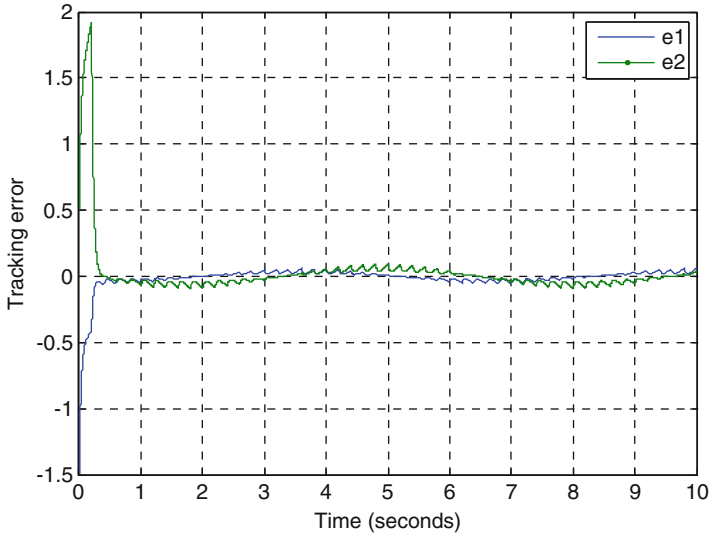

Fig. 11.3 Bounded plant tracking error

initial conditions of the plant. The initial conditions for the reference input and the plant were given respectively by:

$$r_0 = [1 \quad 1]^T, \quad x_0 = [-0.5 \quad 1.5]^T$$

but, since the model does not have any of this information the model initial conditions are set as:

$$\hat{x}_0 = [0 \quad 0]^T.$$

This uncertainty in the initial conditions is observed at the beginning of the simulation. We can see in Fig. 11.3 that the tracking error has significantly larger magnitude before the first few updates of the model take place than in its steady-state response. After a few updates we can see in Fig. 11.2 that each state of the plant follows the corresponding element of the reference input.

It is also worth mentioning that if in this example we employ a ZOH in the controller node instead of the nominal model of the plant then the networked system is unstable for the sampling interval used before, $h = 0.2$ s. According to simulations, for this ZOH case we have to reduce the sampling interval to about $h = 0.055$ s, in order to obtain a stable networked systems and a bounded tracking error.

11.2.1 Inter-Sample Behavior Analysis

Equation (11.16) establishes boundedness of the tracking error at the updating instants as long as the sampled matrix F in (11.12) has all eigenvalues inside the unit circle. The signal w is an external signal for which we have no control but it is assumed bounded at all times. The response of the tracking error (11.16) was simplified at the updating instants by realizing that at those instants $\hat{\varepsilon}(t_k) = \varepsilon(t_k)$, but between updates of the state of the model we have that the tracking error is described by (11.11). In this section we will formally establish a bound for the inter-sample response of the tracking error.

Theorem 11.2 *Assume that* $\left|\widetilde{A}\right| \leq \Delta A$, $\left|\widetilde{B}\right| \leq \Delta B$, *and* $\left|\widetilde{A}\Pi + \widetilde{B}\Gamma + \widetilde{P}\right| \leq \Delta$ *are known bounds on the uncertainties*; *also assume that the sampled-data tracking error is bounded as expressed in Theorem 11.1. Then the growth of the tracking error in the inter-sample intervals* $t \in [t_k, t_{k+1})$ *with constant* $h = t_{k+1} - t_k$ *is bounded*:

$$\begin{aligned} |\varepsilon(h)| \leq & \left(|\varepsilon(t_k)| \left(1 + \frac{\alpha b |K|}{a + \beta} \right) + \frac{\Delta \overline{w}}{a} \right) e^{ah} \\ & - \frac{\alpha b |K|}{a + \beta} |\varepsilon(t_k)| e^{-\beta h} - \frac{\Delta \overline{w}}{a} \end{aligned} \tag{11.19}$$

where $a = \left|\hat{A}\right| + \Delta A$, $b = \left|\hat{B}\right| + \Delta B$, *and* $|w| \leq \overline{w}$. *The parameters* $\alpha, \beta > 0$ *are known constants that represent bounds on the response of the compensated model.*

Proof In order to bound the inter-sample behavior let us look at the dynamics of $|\varepsilon|$:

$$\begin{aligned}\frac{d}{dt}|\varepsilon| &\leq |A||\varepsilon| + |BK||\hat{\varepsilon}| + \left|\widetilde{A}\Pi + \widetilde{B}\Gamma + \widetilde{P}\right||w| \\ &\leq |A||\varepsilon| + |BK|\left|e^{\left(\hat{A}+\hat{B}K\right)(t-t_k)}\right||\varepsilon(t_k)| + \left|\widetilde{A}\Pi + \widetilde{B}\Gamma + \widetilde{P}\right|\overline{w}\end{aligned} \tag{11.20}$$

for $t \in [t_k, t_{k+1})$. Without loss of generality let $t_k = 0$ and using the bound $\left|e^{\left(\hat{A}+\hat{B}K\right)t}\right| \leq \alpha e^{-\beta t}$ we obtain:

$$\frac{d}{dt}|\varepsilon| \leq a|\varepsilon| + \alpha b|K|e^{-\beta t}|\varepsilon_0| + \Delta\overline{w}. \tag{11.21}$$

Consider the differential equation:

$$\dot{\phi} = a\phi + \alpha b|K|\phi_0 e^{-\beta t} + \Delta\overline{w} \tag{11.22}$$

where $\phi_0 = |\varepsilon_0| = |\varepsilon(t_k)|$. Equation (11.22) represents a bound on the rate of growth of the norm of the tracking error between update instants. Then we can conclude that $|\varepsilon(t)| \leq \phi(t, \phi_0)$, where $\phi(t, \phi_0)$ is the solution of (11.22) satisfying $\phi(0, \phi_0) = \phi_0$. Such solution is given by:

$$\phi(t) = \left(\phi_0\left(1 + \frac{\alpha b|K|}{a+\beta}\right) + \frac{\Delta\overline{w}}{a}\right)e^{at} - \frac{\alpha b|K|}{a+\beta}\phi_0 e^{-\beta t} - \frac{\Delta\overline{w}}{a}. \tag{11.23}$$

Note that it is easy to show that $\phi(t) > 0$ for $\phi_0 > 0, \forall\, t \geq 0$, and the typical case in which the compensated model is stable. Now we are able to establish a bound on the growth of the tracking error between updates in the model state, that is, in the intervals $t \in [t_k, t_{k+1})$ which is given by:

$$|\varepsilon(h)| \leq \phi(h) = \left(\phi_0\left(1 + \frac{\alpha b|K|}{a+\beta}\right) + \frac{\Delta\overline{w}}{a}\right)e^{ah} - \frac{\alpha b|K|}{a+\beta}\phi_0 e^{-\beta h} - \frac{\Delta\overline{w}}{a}. \tag{11.24}$$

◆

Remark The tracking error may grow by some amount between updates on the state of the model. This variation in the tracking error depends on the uncertainties and the open-loop response of the system and, especially, on the size of the updating interval h. Intuitively we can reduce the increase of the tracking error between updates by reducing h; this also can be shown by taking the limit of (11.24) as h goes to 0, namely

$$\lim_{h\to 0}\phi(h)=\phi_0\left(1+\frac{ab|K|}{a+\beta}\right)+\frac{\Delta\overline{w}}{a}-\frac{ab|K|}{a+\beta}\phi_0-\frac{\Delta\overline{w}}{a}=\phi_0 \tag{11.25}$$

and $\phi_0=|\varepsilon(t_k)|$ is bounded if F in (11.12) has all its eigenvalues inside the unit circle.

11.2.2 Reference Input Tracking for General System Output

The discrete-time error model obtained in previous sections assuming that $C=\hat{C}=I$ and $Q=-I$ can be generalized to make the output of a system, which is a linear combination of the states, track a linear combination of the reference input signals, as expressed in (11.6) and in (11.7) for the model tracking error.

We can show from (11.7) that the model tracking error can be expressed as:

$$\hat{\varepsilon}_o=\hat{C}\hat{x}+Qw=\hat{C}(\hat{x}-\Pi w)=\hat{C}\hat{\varepsilon}. \tag{11.26}$$

Here, we distinguish between the output model tracking error $\hat{\varepsilon}_o$ and the model tracking error $\hat{\varepsilon}$. More specifically, the former is a linear combination of the components of the later. Since our purpose is to stabilize the error dynamics, the stabilization of the model tracking error corresponds to the stabilization of the output model tracking error. The same approach is considered for the plant tracking error. In view of (11.6) we can express the output plant tracking error as:

$$\varepsilon_o=Cx+Qw=Cx-\hat{C}\Pi w=C(x-\Pi w)+\widetilde{C}\Pi w=C\varepsilon+\widetilde{C}\Pi w \tag{11.27}$$

where $\widetilde{C}=C-\hat{C}$. We obtain the discrete-time model for the plant tracking error signal and for the reference signal. The first is given by (11.16) and the second by:

$$w(t_{k+1})=e^{Sh}w(t_k). \tag{11.28}$$

Then the discretized output plant tracking error is given by:

$$\begin{aligned}\varepsilon(t_{k+1})&=F\varepsilon(t_k)+Gw(t_k)\\ \varepsilon_o(t_k)&=C\varepsilon(t_k)+\widetilde{C}\Pi w(t_k).\end{aligned} \tag{11.29}$$

Corollary 11.3 *The plant tracking error response for the MB-NCS in Fig. 11.1 with general values of C and Q is bounded for any bounded reference input if the eigenvalues of F lie inside the unit circle.* ♦

11.3 Output Feedback Tracking Using Different Dimension Models

The work presented in previous chapters in relation to the MB-NCS framework with model uncertainties assumes that the model is of the same order or dimension as the real plant. The new MB-NCS architecture described in this section not only generalizes to the case of output feedback but it also considers different types of uncertainties, such as multiplicative and additive uncertainties that result in the real system being of different dimension than the available model. The results in this section consider Single-Input Single-Output (SISO) systems using output feedback and dynamic controllers. The objective is to minimize the steady-state plant tracking error for step reference inputs while also considering a reduction on network communication.

In contrast to the previous results in this book, we do not assume that the entire state vector is available for measurement but only a single output of the system. In order to obtain a better tracking performance and to avoid the implementation of state observers using uncertain parameters, we use a transfer function representation for the model and the system. We consider discrete-time systems which are modeled by:

$$\hat{T}(z) = \frac{\hat{Y}(z)}{U(z)} = \frac{b_0 z^m + b_1 z^{m-1} + b_2 z^{m-2} \ldots + b_{m-1} z + b_m}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \ldots + a_{n-1} z + a_n} \tag{11.30}$$

where z is the Z-transform variable and $T(z)$ is the transfer function. We consider strictly proper systems, i.e., $n>m$. The plant model may be unstable i.e., not all poles of the transfer function $\hat{T}(z)$ have magnitude less than 1. We consider uncertain systems that can be described using stable and proper multiplicative or additive uncertainties,

$$T(z) = \hat{T}(z) \cdot \Delta T_M(z), \quad T(z) = \hat{T}(z) + \Delta T_A(z). \tag{11.31}$$

As a result, the model and the system are, in general, of different order. Let $\widetilde{T}(z)$ represent, in general, a multiplicative or an additive uncertainty. If a state-space representation is to be used, we will find that the model and plant state vectors have different dimensions and this type of uncertainty has not been considered yet in the MB-NCS setup described in this book. In order to deal with this dimensionality problem, we implement the discrete-time model as a simple difference equation which represents the time-domain equivalent of (11.30). The output measurements are used directly to update the model without need of implementing a state observer.

In order to find the time instants that the sensor needs to send a measurement to update the model in the controller node we implement an event-triggered

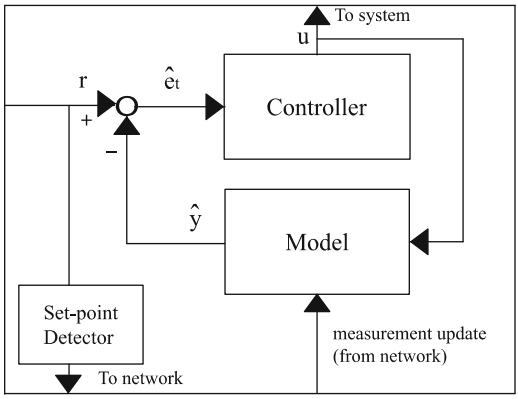

Fig. 11.4 Model-based set-point tracking networked system: actuator/controller node

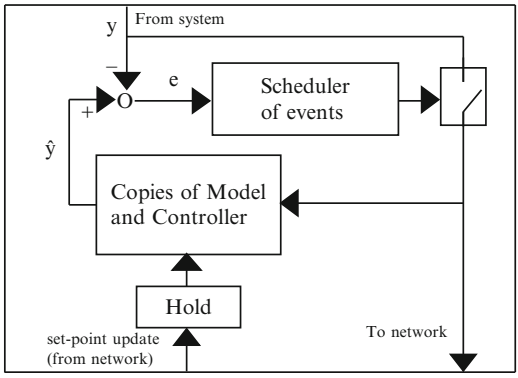

Fig. 11.5 Model-based set-point tracking networked system: sensor node

strategy as in Sect. 6.2 of Chap. 6 instead of using periodic updates. The difference in this case is that we are only able to measure the output of the system. The output error is given by:

$$e(k) = \hat{y}(k) - y(k) \tag{11.32}$$

where $y(k)$ is the output of the system and $\hat{y}(k)$ is the output of the model.

The actuator/controller and sensor node architectures for the set-point tracking model-based problem are shown in Figs. 11.4 and 11.5. Figure 11.4 represents the actuator/controller node and it contains the model of the system $\hat{T}(z)$, the controller $C(z)$, which is designed based on the available model, and the set-point detector. The function of the set-point detector is to determine the time instants at which the reference input, which is only available at the controller node, changes its set-point value in order to transmit this information to the sensor node using the

communication network. Figure 11.5 represents the sensor node. The sensor needs to compute the output error (11.32), and compare its absolute value to a fixed threshold. When the error is greater than the threshold a measurement update is triggered and the current and n-1 past measurements of the plant are sent to the controller node. In order to calculate the output error, the sensor needs the current value of the model output $\hat{y}(k)$ in addition to the plant output measurements. The exact copy of the output of the model can be easily obtained without sending frequent information as follows: copies of the model and controller are implemented in the sensor and at the time instants when the reference input changes values the controller node only needs to transmit the new set-point value. When a reference input value is received at the sensor node, it is held until a new value arrives. In addition, when the event scheduler at the sensor node decides that a measurement packet needs to be sent to update the model in the controller then it also updates the model in the sensor node using the same measurements.

The overall approach can be used for tracking of different types of reference input signals but it will be necessary to obtain an accurate estimation of the external input in the sensor node or to increase the communication rate from the controller to the sensor which is undesirable in networked implementations since other systems and applications may need to communicate information. For this reason, we restrict this framework to track piecewise constant signals which is general enough for many applications [217, 249].

The output feedback problem in MB-NCS has been studied mostly through the use of state observers as in Sects. 3.1 and 3.2. The implementation of a state observer, of course, is obtained using the model parameters. For traditional non-networked systems, that is when continuous feedback is available, and in the absence of model uncertainties the design of observer and controller gains can be done independently from each other, i.e., the separation principle holds. However, in most real-life problems there exists a plant-model mismatch and we need to design a controller that meets the specified requirements and that is also robust to model uncertainties.

The design of a state observer for a non-networked uncertain system is a more difficult task than for a system without uncertainties. Additionally, the results are less satisfactory. In particular, the separation principle does not hold any longer and the state observer error can be generally bounded but it does not asymptotically converge to 0.

Consider a sensor that provides output measurements continuously (not the whole state) and the system is driven by a reference input signal u. The system equations are represented by:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{11.33}$$

and the observer by:

$$\bar{x}(k+1) = \left(\hat{A} - L\hat{C}\right)\bar{x}(k) + \left(\hat{B} - L\hat{D}\right)u(k) + Ly(k) \tag{11.34}$$

where $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ are the available matrices representing the system dynamics. Define the estimation error $\varepsilon = x - \bar{x}$ and the error matrices: $\widetilde{A} = A - \hat{A}$, $\widetilde{B} = B - \hat{B}$, $\widetilde{C} = C - \hat{C}$, and $\widetilde{D} = D - \hat{D}$. In the presence of model uncertainties, because of the plant-model mismatch, it is not possible to perform a perfect estimation of the plant non-zero states as in the case when the observer parameters are the same as those of the plant. In general, if we analyze the estimation error dynamics by explicitly taking into account the difference between the plant and model parameters we have, from (11.33) and (11.34), that the error dynamics are described by the next equation:

$$\begin{aligned} \varepsilon(k+1) &= x(k+1) - \bar{x}(k+1) \\ &= (\hat{A} - L\hat{C})\varepsilon(k) + (\widetilde{B} - L\widetilde{D})u(k) + (\widetilde{A} - L\widetilde{C})x(k). \end{aligned} \tag{11.35}$$

Equation (11.35) shows that in order to obtain asymptotic estimates of a system states using imperfect parameters of that system we need a zero reference input and a state that tends asymptotically to zero as time tends to infinity. The implementation of this type of state observer in MB-NCS works for stabilization as it was used in Chap. 3, but for the tracking problem the input u contains a non-zero component that results in non-zero steady-state estimation error.

In this work the model in both the controller and the sensor nodes is implemented as a difference equation which represents the time-domain equivalent of the transfer function (11.30). The absolute value of the output error is compared to a fixed positive threshold α. When the relation $|e(k)| > \alpha$ holds then the sensor transmits a measurement update. At this point, the output error (11.32) is set to 0, since the model output is equal to the real output of the system. Therefore, the output error is bounded by:

$$|e(k)| \leq \alpha. \tag{11.36}$$

When the sensor decides that a measurement update needs to be sent according to the current output error, then it sends the current and n-1 past output measurements which are used to update the model in the controller. At the same time the sensor uses exactly the same measurements to update its own copy of the model.

Next, we provide conditions under which we are able to stabilize uncertain unstable systems with limited feedback. Furthermore, by using the internal model principle [61, 75] we are also able to bound the steady-state tracking error.

Theorem 11.4 *The plant output tracking error is bounded for any bounded reference step input if*

(a) *The term 1+T(z)C(z) has all its zeros inside the unit circle.*
(b) *The denominator of the controller C(z) contains the factor (z-1).*
(c) *The poles of* $\widetilde{T}(z)$ *have magnitude less than one.*

Proof First, define the plant output tracking error

$$E_t(z) = R(z) - Y(z) \tag{11.37}$$

and the model output tracking error.

$$\hat{E}_t(z) = R(z) - \hat{Y}(z). \tag{11.38}$$

The output of the plant is given by:

$$Y(z) = T(z)C(z)\hat{E}_t(z) = T(z)C(z)\left[R(z) - \hat{Y}(z)\right]$$

and using (11.32) we obtain the following:

$$Y(z) = T_{cl}(z)R(z) - T_{cl}(z)E(z) \tag{11.39}$$

where $T_{cl}(z) = \dfrac{T(z)C(z)}{1 + T(z)C(z)}$.

The output tracking error is given by:

$$E_t(z) = \frac{1}{1 + T(z)C(z)}R(z) + T_{cl}(z)E(z). \tag{11.40}$$

The reference input term in (11.40) is asymptotically stable for constant reference inputs $r(k)$ since the zeros of $1+T(z)C(z)$ are inside the unit circle and at least one of the poles of the controller is at +1. The second term in (11.40) contains the stable closed-loop transfer function $T_{cl}(z)$ but the output error $E(z)$ is not constant. However, the output error is bounded by updating the model (and resetting the output error) every time the error's absolute value is greater than some positive threshold α.

Stability of the networked system can be obtained from Theorem 11.4.

Corollary 11.5 *The networked system* (*2*) *with model* (*1*) *is bounded-input bounded-output stable with respect to the error* (*3*) *if*

(a) *The term 1+T(z)C(z) has all its zeros inside the unit circle.*
(b) *The poles of* $\widetilde{T}(z)$ *have magnitude less than one.*

Proof In the presence of plant-model mismatch and for zero reference input, the conditions (a) and (b) above are sufficient for stability of the networked system. In this case the controller is not required to provide tracking performance and the additional condition in Theorem 1 is not needed. The output of the system is given by

$$Y(z) = -T_{cl}(z)E(z) \tag{11.41}$$

and the output error is bounded as before. ♦

Remark The selection of the constant threshold α is made considering the following trade-off. A small threshold results in a smaller bound on the steady-state tracking error but, in general, it increases communication rate by sending measurement updates more frequently. A reduction on network usage can be achieved by increasing the threshold at the cost of a larger steady-state tracking error.

Remark The controller $C(z)$ is designed in such a way that the closed-loop model is stable and with desired properties by selection of desired closed-loop poles in addition to providing zero steady-state tracking error in the absence of model uncertainties.

Example 11.2 Consider the transfer function representation of the model of an unstable system:

$$\hat{T}(z) = \frac{1}{z^2 - 0.2z - 0.9}.$$

The controller is designed in order to stabilize $\hat{T}(z)$ and to provide zero steady-state model output tracking error to a step input and is given by:

$$C(z) = \frac{2.448z^2 + 0.3834z - 1.881}{z^3 + 0.2z^2 + 0.89z - 2.09}.$$

The real system consists of the model dynamics and the following multiplicative uncertainty:

$$\Delta T_M(z) = \frac{z + 0.61}{z + 0.55}.$$

Then, the dynamics of the real system are given by:

$$T(z) = \frac{z + 0.61}{z^3 + 0.35z^2 - 1.01z - 0.495}.$$

The reference input is shown at the top of Fig. 11.6 along with the output of the real system. The bottom of Fig. 11.6 shows the output tracking error $E_t(z)$. The output error $E(z)$ is shown at the top of Fig. 11.7, where a constant threshold $\alpha = 0.02$ was used. The network communication signal $n_c(k)$ represents the time instants at which output measurements are sent from the sensor node to the controller node.

$$n_c(k) = \begin{cases} 1 & \text{if measurements are sent at time } k \\ 0 & \text{if measurements are not sent at time } k \end{cases} \tag{11.42}$$

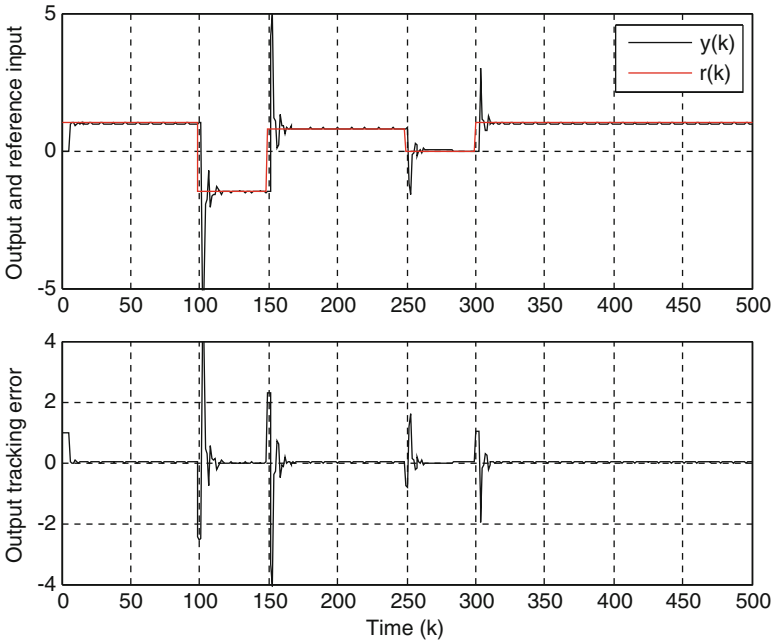

Fig. 11.6 System output and reference input for $\alpha = 0.02$ (*top*). Output tracking error (*bottom*)

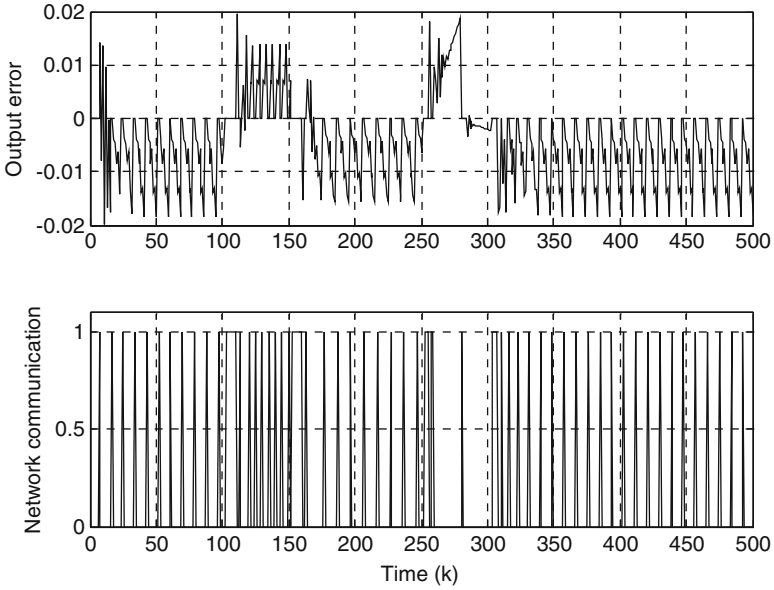

Fig. 11.7 Output error for $\alpha = 0.02$ (*top*). Network communication instants (*bottom*)

11.3.1 Output Feedback Tracking with Delays

The results in the previous sections of this chapter assumed zero delay when information is sent from the sensor to the controller node using the communication network. By the nature of the event-triggered strategies, it may happen that several systems attempt to access the network at the same time in order to update their corresponding controller node. In this case only one node can gain access and the rest need to wait until the network turns into an idle state.

An approach to consider network induced delays within the MB-NCS framework is to propagate the delayed measurements received at the controller, i.e., to estimate the current output of the system based on the delayed measurements using the model parameters; refer to Sect. 3.3. In this section we introduce a different approach for the single-channel networked system that provides better results in terms of system performance and reduced network communication.

Assume that the measurement updates arrive at the controller/actuator node d sampling times later, that is, when a measurement update is received at time k, it corresponds to an event generated at time k-d which contains measurements $y(k\text{-}d)\ldots y(k\text{-}n\text{-}d)$. The advantage of using a transfer function representation in this case is that the network induced delay can be represented by z^{-d}. Assuming a constant delay, we are able to jointly model the dynamics of the system and the delay induced by the network as follows:

$$\hat{T}_d(z) = \hat{T}(z) \cdot z^{-d}. \tag{11.43}$$

The new controller, represented by $C_d(z)$, is designed based on $\hat{T}_d(z)$, that is, the controller stabilizes $\hat{T}_d(z)$ and provides zero steady-state model output tracking error. The model $\hat{T}_d(z)$ is updated using the delayed measurements directly.

Theorem 11.6 *The plant output tracking error corresponding to the networked system* (11.31) *with induced delays and with model* (11.43) *is bounded for any bounded reference step input if*

(a) *The term* $1 + T_d(z)C_d(z)$ *has all its zeros inside the unit circle, where* $T_d(z) = T(z) \cdot z^{-d}$.
(b) *The poles of* $\widetilde{T}(z)$ *have magnitude less than one.*
(c) *The denominator of the controller* $C_d(z)$ *contains the factor* $(z\text{-}1)$.

Proof Let $Y_d(z)$ and $\hat{Y}_d(z)$ represent, respectively, the outputs of the delayed system and model $T_d(z)$ and $\hat{T}_d(z)$. Define the errors $E_d(z) = \hat{Y}_d(z) - Y_d(z)$, $E_t^d(z) = R(z) - Y_d(z)$, $\hat{E}_t^d(z) = R(z) - \hat{Y}_d(z)$. It can be shown that the delayed output tracking error is given by

$$E_t^d(z) = \frac{1}{1 + T_d(z)C_d(z)} R(z) + T_{cl}^d(z) E_d(z) \tag{11.44}$$

where $T_{cl}^{d}(z) = \dfrac{T_d(z)C_d(z)}{1 + T_d(z)C_d(z)}$. The error (11.44) is bounded by updating the model using the real output of the system, that is, when the controller receives delayed measurements it updates the model using those measurements directly, i.e., $\hat{y}_d(k) \doteq y(k-d) = y_d(k)$ which makes $e_d(k) = 0$,

Since $E_t^d(z)$ represents the delayed version of $E_t(z)$ then the output tracking error is bounded as well. ♦

Corollary 11.7 *The networked system* (11.31) *with induced delays and with model* (11.43) *is bounded-input bounded-output stable with respect to the error* (11.32) *if*

(a) *The term* $1 + T_d(z)C_d(z)$ *has all its zeros inside the unit circle.*

(b) *The poles of* $\widetilde{T}(z)$ *have magnitude less than one.*

Proof For a constant delay d, we are able to design a stabilizing controller $C_d(z)$ for the delayed model $\hat{T}_d(z)$ and for zero reference input, the conditions (a) and (b) above are sufficient for stability of the networked system. In this case the controller is not required to provide tracking performance and the additional condition in Theorem 11.6 is not needed. The delayed output of the system is given by

$$Y_d(z) = -T_{cl}^{d}(z)E_d(z) \tag{11.45}$$

and the delayed output error is bounded as before. Since $Y_d(z)$ represents the delayed version of $Y(z)$, then the output of the system with no delay is bounded as well. ♦

Remark One important aspect in the implementation of this approach for the case of network delays is the computation of $e_d(k) = \hat{y}_d(k) - y(k-d)$. This task needs to be accomplished at every sampling time which requires the comparison of the outputs of the real system with no delay (11.31) and the delayed model (11.43). One way to compute this error is to use old system outputs, but since the current system output is available at the sensor node, then it can be used to compute the output error $e(k)$ instead of computing $e_d(k)$. Therefore we compute $e(k) = \hat{y}_d(k+d) - y(k)$ and the quantity $\hat{y}_d(k+d)$ is obtained by executing the model in the sensor node until time $k+d$ in order to obtain an estimate of $y_d(k+d) = y(k)$.

Remark Controller complexity. The cost to be paid by using the approach described in this section compared to the usual prediction using the model with no delay is in the form of a more complex controller. The order of the controller increases since $C_d(z)$ is designed to control the higher order model $\hat{T}_d(z)$.

Example 11.3 (Uncertain system with delays). Consider the unstable model

$$\hat{T}_d(z) = \frac{z+1}{z^2 - 0.2z - 0.9} \cdot \frac{1}{z^3}.$$

which models the dynamics of the system and a constant delay z^{-3} equivalent to a 3-sample delay.

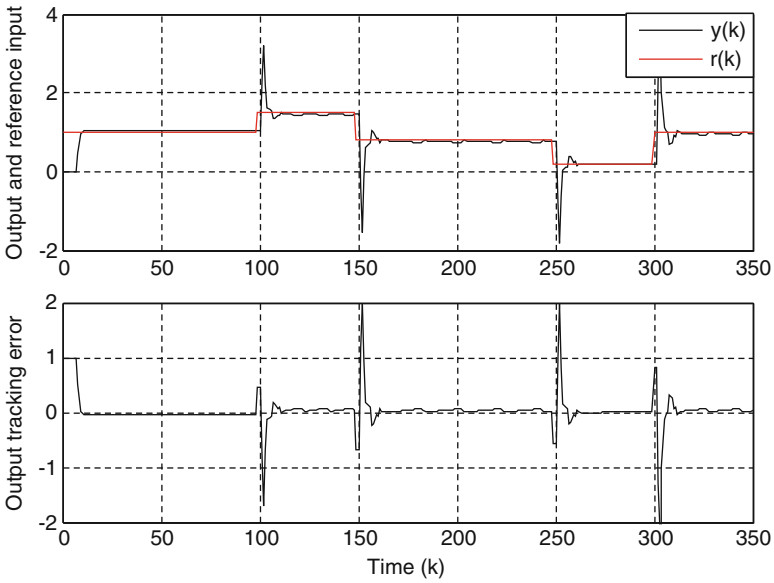

Fig. 11.8 System output and reference input for $\alpha = 0.05$ (*top*). Output tracking error (*bottom*)

The controller is designed in order to stabilize $\hat{T}_d(z)$ and to provide zero steady-state model output tracking error and is given by:

$$C_d(z) = \frac{2.638z^5 - 1.102z^4 - 1.061z^3}{z^6 + 0.2z^5 + 0.8895z^4 + 0.3579z^3 + 0.8726z^2 - 2.141z - 1.179}.$$

As it was mentioned before, the complexity of the controller increases by considering the model $\hat{T}_d(z)$ instead of $\hat{T}(z)$. The real system consists of the model dynamics with zero delay and the following multiplicative uncertainty:

$$\Delta T_M(z) = \frac{z + 0.61}{z + 0.55}$$

then, the dynamics of the real system are given by:

$$T(z) = \frac{z^2 + 1.61z + 0.61}{z^3 + 0.35z^2 - 1.01z - 0.495}.$$

Note that the real system contains no delay since the delays are induced by the network when a feedback measurement is sent from the sensor node to the controller/actuator node. Good performance and reduction of communication are obtained as shown in the simulation results shown in Figs. 11.8 and 11.9. Note that the output error grows for a few sampling instants after an update has been sent since the model is not updated until $d = 3$ samplings later.

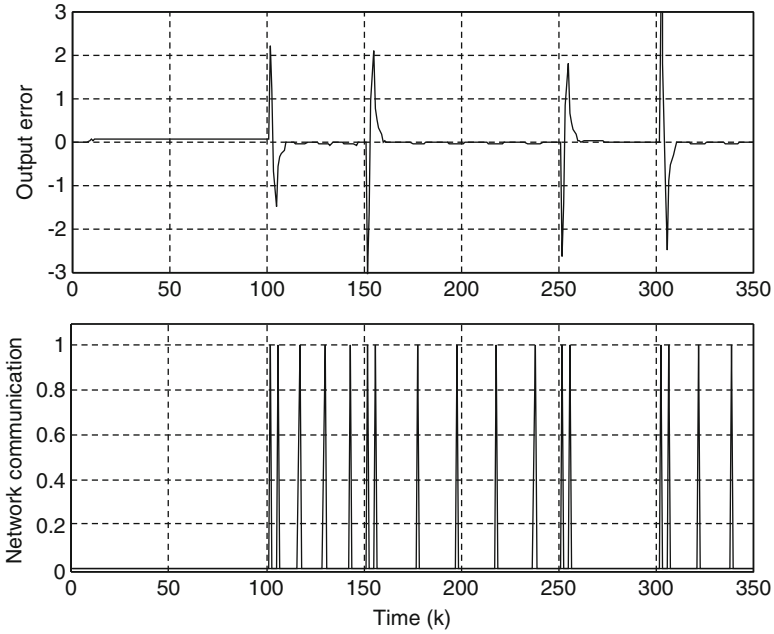

Fig. 11.9 Output error for $\alpha = 0.05$ (*top*). Network communication instants (*bottom*)

The performance of the system using the delayed model and controller is considerably superior to the propagation approach used in Chap. 3. In order to show the improved performance, we simulate the same system and model using the same reference input, delay, and threshold. The difference is that we use the propagation method and the controller is designed for the no-delay model $\hat{T}(z)$. Results of simulation are shown in Fig. 11.10, which shows a poor tracking performance of the system and a significant increase in network communication.

Example 11.4 Stabilization over an additive Gaussian channel. We consider the unstable system:

$$T(z) = \frac{1}{z^2 - 0.2z - 0.9}.$$

The system is assumed to be known. The measurements are transmitted over an additive Gaussian channel modeled by:

$$y_g(k) = H_g y(k) + v(k)$$

where $v(k)$ is zero-mean Gaussian noise with variance σ_v and $y_g(k)$ is the measurement received at the controller node. The parameter H_g is assumed to be unknown and it represents the uncertainty in our approach in addition to the Gaussian noise. Figure 11.11 shows the results of simulation when an impulse

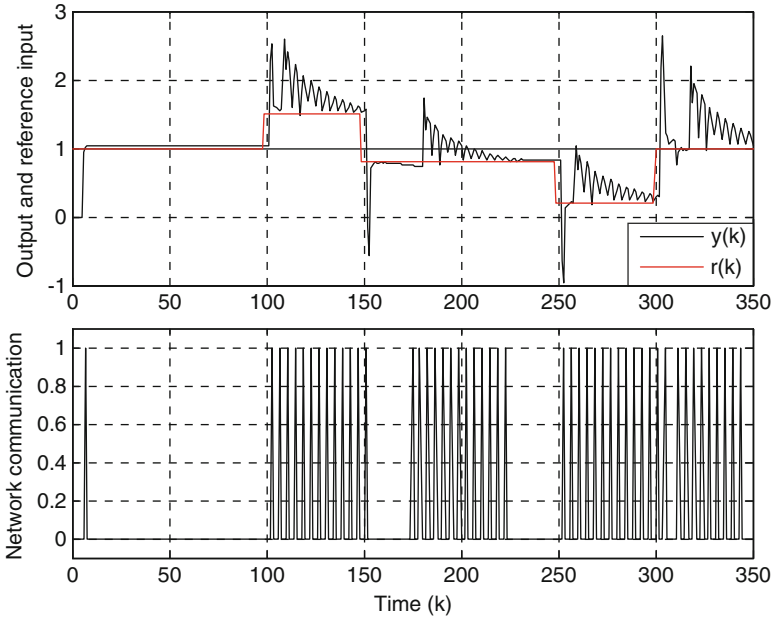

Fig. 11.10 Control of system with network delays using propagation for $\alpha = 0.05$ (*top*). Network communication instants (*bottom*)

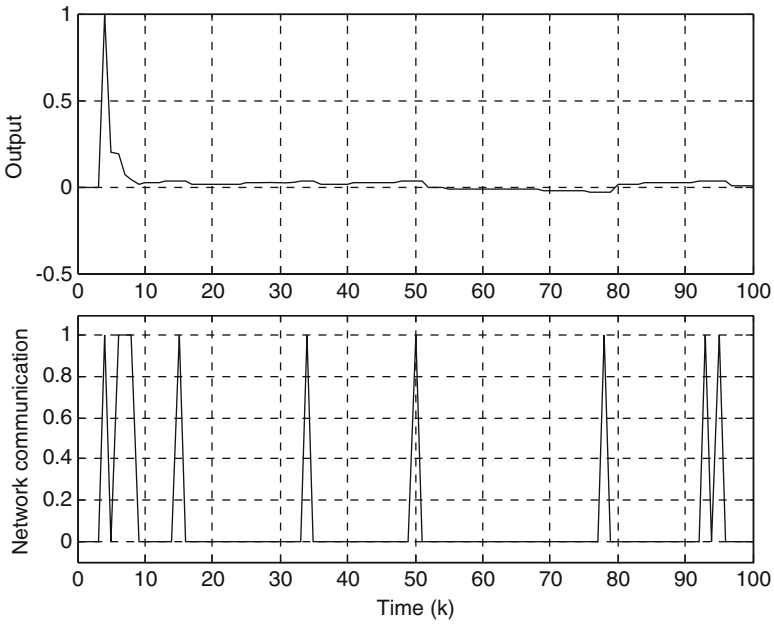

Fig. 11.11 Stabilization of system over an uncertain network. Output of the system for $\alpha = 0.03$ (*top*). Network communication instants (*bottom*)

external disturbance perturbs the system and using $H_g = 0.8$ and $\sigma_v = 0.01$. The system is stable since the conditions in Corollary 11.5 are satisfied for this channel uncertainty.

11.4 Notes and References

Preliminary results for the tracking problem using MB-NCS were discussed in [67, 71]. Two implementation cases were explained in [67] but they do not provide, in general, small tracking errors. This motivates the search of improved architectures that are able to achieve better, close to 0, steady-state tracking errors.

Section 11.1 and parts of Sect. 11.2 were presented in [93]. Section 11.3 was first published in [89]. With respect to the work presented in Sect. 11.3 where output measurements are sent directly instead of implementing state observers, similar to the approach shown in Sect. 7.2, it is important to note the reduction in network traffic by using this approach, as compared to the case in which a measurement of the current output $y(k)$ is sent at every sampling instant, even in the case when n is large compared to the inter-update intervals. This improvement is due to packet structure in NCS.

The problem of reference input tracking in NCS has been considered by different authors. Gao and Chen [97] proposed a new model which is based on the original plant and the model reference system. This new model considers a zero-order-hold (ZOH) in the actuator node and is formulated following a sampled-data approach at the updating instants of the ZOH. Although parameter uncertainties are considered in the controller design step, the nominal model is not used between updates to estimate the state of the plant. Goodwin et al. [101] presented different NCS architectures and compared their properties for typical problems such as disturbance rejection and input tracking. One of these architectures considers a model of the plant and a model of the network channel properties. The objective in [101] is to model the whole NCS at the controller node and generate a nominal output of the system that is compared to the real output and feed the controller using the resulting error. It is shown that under some conditions, this architecture outperforms the basic ones that do not use a model of the system in the control loop. In addition, the authors only consider uncertainty in the model of the channel, i.e., they assume that a perfect model of the physical plant is known. In [271] the authors consider the tracking problem in NCS subject to time-varying sampling intervals and delays. Two approaches for modeling the NCS are proposed and conditions for input-to-state stability of the tracking error are provided. Other approaches to the networked input tracking problem can be found in [44] and [287].

Chapter 12
Adaptive Stabilization of Networked Control Systems

The applicability of Model-Based Networked Control Systems (MB-NCS) is often limited by the inexact knowledge of the dynamics of the system being controlled. The main motivation in this chapter is that the model state updates that we perform at the controller node should not be restricted only to the state, but we can also update the nominal parameters of the model as well. The performance of a MB-NCS depends mostly on the accuracy of the model that we use for both controller design and for implementation in order to estimate the real state between update intervals. In this chapter we focus on applying identification algorithms in the MB-NCS context. The main idea is to estimate the current parameters of the real system since in many problems the dynamics of the system may change over time due to age, use, or nature of the physical plant. In many other situations the initial given model is simply outdated or very inaccurate. A better knowledge of the plant dynamics will provide an improvement in the control action over the network, i.e., we can achieve longer periods of time without need for feedback. At the same time, we overcome a possibly restrictive assumption in MB-NCS; namely that the controller is designed to stabilize the real system. This assumption may be restrictive since our knowledge of the plant dynamics is limited. As we will see, the identification process allows us to update not only the model but the controller itself so it can better respond to the dynamics of the real plant being controlled.

There exist two typical approaches for system identification namely, parametric and nonparametric [76, 163]. A parametric model may take different forms, the most common ones are transfer functions (expressed as a polynomial fraction or in poles and zeros form), state-space representations, and differential equations. In these forms there exist coefficients (parameters) that specify completely the model. Nonparametric models result generally from the data obtained from frequency response methods. In these cases the system is subject to a wide range of inputs in order to find a characteristic curve. A frequency response is difficult to obtain while the system is in normal operation, limiting the use of nonparametric approaches for online identification.

The estimated parameters obtained by means of recursive parameter identification are used to upgrade an explicit model of the plant. The state of this model is

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_12,

used for control when no feedback information from the real plant is available. An explicit model implies that a parametric model is needed. In order to achieve the described goals we will follow the parametric approach in what follows. The upgraded model is also useful to redesign the controller using the new parameters and stabilize the plant in a specific way, e.g., selected pole placement or optimal regulation by choosing appropriate weights. In the examples that concern stabilization of MB-NCS shown in this section we opt to determine a Linear Quadratic Regulator (LQR) problem every time we update the model of the plant. The complete process, identification of parameters and controller recalculation, can be seen as an adaptive stabilization scheme but in contrast to common adaptive techniques we do not redefine the controller at each time instant, but only when new estimated parameters are received in the controller node. The main reason is that we aim to use the network, either for state or parameter updates, as little as possible. When the controller node receives updated model parameters, an LQR controller is calculated solving a discrete-time algebraic Riccati equation using the same weights and the new parameters that were just obtained.

The contents of this chapter are organized as follows. Section 12.1 considers parameter estimation and adaptive stabilization of systems with noiseless measurements. Section 12.2 presents similar results when the Gaussian noise is added to the measurements of the state. Diverse applications are explored in Sect. 12.3 where the adaptive stabilization of MB-NCS is implemented. A different approach based on Luenberger observers and additional measurements is described in Sect. 12.4. We conclude with notes and references in Sect. 12.5.

12.1 Adaptive Stabilization of Deterministic MB-NCS

To start this section we consider plant and model dynamics that are described, respectively, by:

$$x(k+1) = Ax(k) + Bu(k) \tag{12.1}$$

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}u(k) \quad \text{for} k \in [k_i, k_{i+1}) \tag{12.2}$$

where $x, \hat{x} \in \mathbb{R}^n$ are the states of the plant and the model respectively, $u(k) = K\hat{x}(k)$, k_i represents the times at which the state of the model is updated using the current state of the real system for $i \in \mathbb{Z}^+$. In addition to updating the state of the model at times k_i the parameters of the model may or may not be updated at the same time instants depending on whether new estimated parameters are available. The matrices $\hat{A} \in \mathbb{R}^{n\times n}$, $\hat{B} \in \mathbb{R}^{n\times m}$ represent the parameters of the model that correspond to the real system matrices A,B. The plant may be unstable i.e., not all eigenvalues of A have magnitude less than 1.

The model (12.2) represents a linear system but by introducing updates of its state it can be seen as an impulsive system when the same parameters are used and only the state is updated (state jump); this is a typical approach in previous chapters. In the present chapter we update the state of the model and we update the parameters of the model as well. So the model matrices switch when new parameter estimates are obtained. Note that the real plant is a linear system and not a switched one as the model (12.2). In contrast to the model, the plant always operates using the same parameters and its state is never updated or reset.

12.1.1 Stability

In many applications it is convenient to drop the usual periodic update implementation in favor of one based on events, for example, the event that the plant-model state error is equal to or greater than some predetermined threshold. This approach was discussed in Chap. 6 for the case of continuous-time systems. A sensor node within the network broadcasts its local state only when it is necessary, i.e., when a measure of the local subsystem state error is above some predetermined threshold. The error, in this case, is defined as the difference between the state of the model and the state of the plant:

$$e(k) = \hat{x}(k) - x(k). \tag{12.3}$$

When we use an event-based strategy to update the model, both parameters and state, we obtain the implicit advantage of extending the time intervals in which the plant works in open loop. In this case a copy of the model is needed in the sensor node to generate the model state and compute (12.3).

In view of the fact that we need to be able to identify a system in general state-space representation, not necessarily in canonical form as discussed in the literature [12, 138, 204] we will use the Kalman filter for identification of systems of the form (12.1). For the case we receive noisy measurements of the state we will use the extended Kalman filter (EKF), see Sect. 12.2.

The next theorem provides stability bounds of the discrete-time MB-NCS when we update the model based on error events. In the sensor node we measure the state of the plant and compare the norm of the error (12.3) to a predefined fixed threshold $\alpha < \infty$. The plant state is used to update the model state if the error is greater than the threshold, i.e., when $|e(k)| > \alpha$.

Theorem 12.1 *Assume* $|x(0)| \leq \beta_1$, for $0 < \beta_1 < \infty$, *then the networked system described by* (12.1) *with state feedback updates triggered when* $|e(k)| > \alpha$ *has a bounded state if the eigenvalues of A+BK are within the unit circle of the complex plane.*

Proof System (12.1) can be described by:

$$x(k+1) = (A+BK)x(k) + BKe(k) \tag{12.4}$$

after (12.3) and the control input $u = K\hat{x}$ have been used.

The response of the plant with initial time $k_0 = 0$ and stable matrix $A+BK$ at any given time $k \geq 0$ is given by:

$$x(k) = (A+BK)^k x(0) + \sum_{j=0}^{k-1} (A+BK)^{k-(j+1)} BKe(j) \tag{12.5}$$

where $e(k)$ is bounded by $|e(k)| \leq \alpha$ since when $|e(k)| > \alpha$ we send a measurement update and the error becomes $e(k_i) = 0$ at the update instant k_i. We can show that the state of the plant is bounded by evaluating its norm which is done next:

$$\begin{aligned} |x(k)| &= \left| (A+BK)^k x(0) + \sum_{j=0}^{k-1} (A+BK)^{k-(j+1)} BKe(j) \right| \\ &\leq \left| (A+BK)^k \right| |x(0)| + \sum_{j=0}^{k-1} \left| (A+BK)^{k-(j+1)} \right| |BK||e(j)|. \end{aligned}$$

By the assumption on the initial condition of the plant and the triggering condition, and using the bound $|(A+BK)^k| \leq \beta_2 \lambda^k, \lambda \in (0,1),\ \beta_2 > 0$ we can write:

$$\begin{aligned} |x(k)| &\leq \beta_1 \beta_2 \lambda^k + \alpha \beta_2 |BK| \sum_{j=0}^{k-1} \lambda^{k-(j+1)} \\ &\leq \beta_1 \beta_2 \lambda^k + \frac{\alpha \beta_2 |BK| \left(1 - \lambda^{k-1}\right)}{1-\lambda}. \end{aligned} \tag{12.6}$$

Note that: $\lim_{k \to \infty} |x(k)| = \dfrac{\alpha \beta_2 |BK|}{1-\lambda}$. ♦

A choice of a stabilizing controller K, or in other words, the fact that the eigenvalues of $A+BK$ lie strictly inside the unit circle ensures that the first term in the right hand side of (12.6) decreases exponentially with time and the second term is bounded for all time $k>0$. ♦

Remark Note that the design of the stabilizing gain K requires a robust-type controller since we only have the nominal parameters available. Note also that a specific location of the closed-loop eigenvalues is not needed in Theorem 12.1; they only need to have magnitude less than 1. The advantage of the framework in this section is that we can obtain accurate models that significantly reduce the initial uncertainty and stabilizing controllers can be easily obtained based on these estimated parameters.

The advantages of intermittent feedback as discussed in Chap. 4 can be used in the event-triggered case as well. The following theorem extends the results in the previous theorem to this case.

Theorem 12.2 *Assume* $|x(0)| \leq \beta_1,\ 0 < \beta_1 < \infty$, *then the networked system* (12.1) *with intermittent state feedback updates triggered when* $|e(k)| > \alpha$ *has a bounded state if the eigenvalues of A+BK lie strictly inside the unit circle.*

Proof The proof is similar to the one in Theorem 12.1 by noting that $e(k) = \hat{x}(k) - x(k) = 0, \forall k \in [k_i, k_i + \tau)$ i.e., in the closed-loop interval, then $|e(k)| \leq \alpha, \forall k$. ♦

For stability conditions of discrete-time MB-NCS using periodic updates refer to Chap. 2.

Comparing the results in Theorem 12.1 to those in Sect. 2.3 in Chap. 2 in which discrete-time systems with periodic updates are used, we can see that the event-triggered approach only offers a bounded output compared to the asymptotic properties when using periodic updates. This drawback can be addressed by applying a time-varying threshold. It is intuitively clear that by varying the magnitude of the threshold value in Theorem 12.1 we can obtain longer update intervals or a smaller output size as measured by the norm of the state as in (12.6) in Theorem 12.1. The norm of the state can be reduced by using a smaller threshold α. In particular, we opt to reduce the threshold value as we approach the equilibrium point of the system. This is the same idea expressed in Sect. 6.2 where continuous-time systems were considered. The following treatment represents an extension of that approach in order to obtain similar conditions for discrete-time systems.

Consider again the plant and model described by (12.1) and (12.2) and the control input $u = K\hat{x}$ to obtain the description (12.4) for the plant. Assume that the control input u renders the model (12.2) Input-to-State Stable (ISS) with respect to the measurement error e. For the definition of ISS Lyapunov function for nonlinear discrete-time systems we use the following [128]:

Definition 12.3 A continuous function $V : \mathbb{R}^n \to \mathbb{R}_0^+$ is called an ISS Lyapunov function for the dynamical system $x(k+1) = f(x(k), u(k))$, $x(k) \in \mathbb{R}^n, u(k) \in \mathbb{R}^m$, $k \in \mathbb{Z}^+$ if there exist class $\mathbb{K}_\infty$ functions $\alpha_1, \alpha_2, \alpha_3$ and γ satisfying:

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \tag{12.7}$$

$$V(f(x,u)) - V(x) \leq -\alpha_3(|x|) + \gamma(|u|). \tag{12.8}$$

The system $x(k+1) = f(x(k), u(k))$ *is said to be ISS with respect to the input u if and only if there exists an ISS Lyapunov function for that system.*

In this section we focus on linear systems and we choose a control law $u = K\hat{x}$ that renders the closed-loop *model* (12.2) globally asymptotically stable. Any such K also renders the closed-loop model ISS with respect to the measurement errors.

We proceed to choose a quadratic ISS Lyapunov function, $V = x^T P x$ where P is symmetric positive definite and is the solution of the closed-loop model discrete-time Lyapunov equation:

$$\left(\hat{A} + \hat{B}K\right)^T P\left(\hat{A} + \hat{B}K\right) - P = -Q \tag{12.9}$$

where Q is a symmetric positive definite matrix.

Let us first analyze the case when $\hat{B} = B$ for simplicity and define the uncertainty $\widetilde{A} = A - \hat{A}$; also assume that the following bounds on the uncertainty hold:

$$\begin{aligned} \left|\widetilde{A}\right| &\leq \Delta_A \\ \Delta_A |P|\left(\Delta_A + 2\left|\hat{A} + BK\right|\right) &\leq \Delta < q \end{aligned} \tag{12.10}$$

where $q = \underline{\sigma}(Q)$, the smallest singular value of Q in the Lyapunov equation (12.9). The parameters Δ and Δ_A represent given bounds on the norm of the uncertainty matrices. The next theorem provides stability conditions based on the error and the threshold value that is used to trigger a measurement update. The resulting Model-Based Event-Triggered (MB-ET) networked system is asymptotically stable. The error threshold is defined as a function of the norm of the state and the uncertainty bounds Δ and Δ_A. Similarly, the occurrence of an error event leads the sensor to send the current measurement of the state of the plant that is used in the controller to update the state of the model and at these update instants the error is equal to 0 since the model state is made equal to the real state.

Theorem 12.4 *Assume that* (12.10) *holds, then system* (12.1) *with* $u = K\hat{x}$ *and feedback based on error events generated when the relation*:

$$|e| > \alpha_r |x|, \tag{12.11}$$

is first satisfied, is globally asymptotically stable, where $\alpha_r = \min(\alpha_{r1}, \alpha_{r2})$, $\alpha_{r1} = \sigma(q - \Delta)/2c_1$, $\alpha_{r2} = (\sigma(q - \Delta)/2c_2)^{1/2}$, $c_1 = 2|PBK|\left(\Delta_A + \left|\hat{A} + BK\right|\right)$, $c_2 = |K^T B^T P B K|$, *and* $0 < \sigma < 1$.

Proof In order to prove this theorem we will set a bound on the first forward difference of $V = x^T P x$ along the trajectories of the *system* (12.4) which is equal to (12.1) when the input $u = K\hat{x}$ has already been substituted and expressed in terms of the state error. Then we can easily show that this bound can be appropriately tuned by the choice of the threshold on the error.

$$\begin{aligned} &V\left(x(k+1)\right) - V\left(x(k)\right) \\ &= \left[x^T (A + BK)^T + e^T K^T B^T\right] P\left[(A + BK)x + BKe\right] - x^T P x \\ &= \left[x^T \left(\hat{A} + \widetilde{A} + BK\right)^T + e^T K^T B^T\right] P\left[(\hat{A} + \widetilde{A} + BK)x + BKe\right] - x^T P x \\ &= -x^T Q x + x^T \left[\left(\hat{A} + BK\right)^T P \widetilde{A} + \widetilde{A}^T P\left(\hat{A} + BK\right) + \widetilde{A}^T P \widetilde{A}\right] x \\ &\quad + 2x^T \left[\left(\hat{A} + \widetilde{A} + BK\right)^T PBK\right] e + e^T K^T B^T PBKe. \end{aligned}$$

We have just expressed $V(x(k+1)) - V(x(k))$ in terms of the model parameters and the uncertainty of the state matrix $\widetilde{A}$. We now proceed to evaluate the contributions of each, the model, the uncertainty, and the error.

$$\begin{aligned} &V\big(x(k+1)\big) - V\big(x(k)\big) \\ &\leq -q|x|^2 + \left|\big(\hat{A} + BK\big)^T P\widetilde{A} + \widetilde{A}^T P\big(\hat{A} + BK\big) + \widetilde{A}^T P\widetilde{A}\right||x|^2 \\ &\quad + 2\left|\left(\hat{A} + \widetilde{A} + BK\right)^T PBK\right||e||x| + \left|K^T\hat{B}^T P\hat{B}K\right||e|^2 \\ &\leq (-q + \Delta)|x|^2 + c_1|e||x| + c_2|e|^2 \end{aligned}$$

where $e(k)$ is bounded by $|e| \leq \alpha_r|x|$. Then

$$V(x(k+1)) - V(x(k)) \leq (\sigma - 1)(q - \Delta)|x|^2 \tag{12.12}$$

which shows that V is guaranteed to decrease for any σ, $0 < \sigma < 1$. Here the state of the model is updated every time the error satisfies the condition imposed in (12.11).♦

Remark Similarly to the stabilizing conditions for continuous-time systems described in Sect. 6.2, an important advantage of this approach is that we define the controller only in terms of the model parameters. The stabilizing threshold is designed using the model parameters $\left(\hat{A}, \hat{B}\right)$ and some bounds on the uncertainty $\widetilde{A}$, quantities that are known.

Remark Also note that an accurate estimate of the plant parameters results in small values of Δ_A and Δ. It follows that a larger value for the threshold α_r can be obtained and longer update intervals can be achieved which results in significant reduction in network communication.

The conditions in Theorem 12.4 are only sufficient and conservative in general. The main reason being that these conditions are based on bounds on the uncertainty norms and not on the location of eigenvalues as the results shown in Chap. 2. By using uncertainty bounds we can obtain more practical conditions but we need to consider worst case scenarios.

12.1.2 Parameter Estimation

In this subsection we will focus on parameter estimation of deterministic linear systems of the form (12.1) with no particular form of the matrices A and B. This identification problem can be solved using a linear Kalman filter; this implementation provides much better convergence properties than the EKF, as it is shown in Sect. 12.3.

In the special case when the sensors provide noiseless measurements of the state, it is possible to modify the model that will be used for the Kalman filter equations in order to estimate the parameters A and B. In order to show this simple idea let us focus on second order autonomous systems, (the idea can be easily

extended to higher order systems with deterministic inputs) with unknown time-invariant parameters a_{ij}:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \tag{12.13}$$

We do not know the values of the parameters and we only receive measurements of the states $x(0) \ldots x(k)$. At any given step due to the iterative nature of the Kalman filter we only need $x(k)$ and $x(k-1)$. Now we rewrite (12.13) as:

$$\begin{bmatrix} \overline{x}_1(k) \\ \overline{x}_2(k) \end{bmatrix} = \begin{bmatrix} x_1(k-1) & x_2(k-1) & 0 & 0 \\ 0 & 0 & x_1(k-1) & x_2(k-1) \end{bmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \end{bmatrix} = \hat{C}(k)\hat{a}(k) \tag{12.14a}$$

where $\hat{a}_{ij}$ represent the estimated values of the real parameters a_{ij} and $\overline{x}_i$ are the estimates of the state based on the estimated parameters and on measurements of the real plant state x_i. Equation (12.14a) becomes the output equation of our filter. The state equation is described by:

$$\begin{bmatrix} \hat{a}_{11}(k+1) \\ \hat{a}_{12}(k+1) \\ \hat{a}_{21}(k+1) \\ \hat{a}_{22}(k+1) \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \end{bmatrix} = A_f \hat{a}(k). \tag{12.14b}$$

The state matrix of our model A_f is constant and the output matrix $\hat{C}(k)$ is time-varying but the overall model is linear. Therefore we can use a linear filter to obtain estimates of the parameters a_{ij} of the state matrix of the original system. Note that we do not need any external input, only nonzero initial conditions on the state. We can also estimate the elements of both matrices A and B if we receive measurements of the state and the deterministic input $u(k)$. Any common inputs such as steps and sinusoidal inputs can be used for identification purposes; sinusoidal inputs do not need to have any particular frequency i.e., there is no requirement of a persistently exciting input which makes this approach a suitable tool for adaptive stabilization. In this case we need to include estimates of the parameters of B in the state vector of the filter model and the input values in $\hat{C}(k)$. The output equation in this case is given by (using a second order example with single input):

$$\begin{bmatrix} \overline{x}_1(k) \\ \overline{x}_2(k) \end{bmatrix} = \begin{pmatrix} x_1(k-1) & x_2(k-1) & 0 & 0 & u(k-1) & 0 \\ 0 & 0 & x_1(k-1) & x_2(k-1) & 0 & u(k-1) \end{pmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \\ \hat{b}_1(k) \\ \hat{b}_2(k) \end{bmatrix}. \tag{12.15a}$$

The filter state matrix is expanded according to the number of additional states,

$$\begin{bmatrix} \hat{a}_{11}(k+1) \\ \hat{a}_{12}(k+1) \\ \hat{a}_{21}(k+1) \\ \hat{a}_{22}(k+1) \\ \hat{b}_1(k+1) \\ \hat{b}_2(k+1) \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \\ \hat{b}_1(k) \\ \hat{b}_2(k) \end{bmatrix}. \tag{12.15b}$$

Both filter models, (12.14a)–(12.14b) and (12.15a)–(12.15b), can be easily applied to higher order systems by following the structures of (12.14) and (12.15). A clear disadvantage is that the order of the filter is n^2 where n is the dimension of the state of the original system.

Song and Grizzle [236] have shown that the linear time-varying Kalman filter is a global asymptotic observer for the underlying deterministic system. Consider the deterministic system described by (12.14), and the associated noisy system:

$$\begin{aligned} \hat{a}(k+1) &= A_f \hat{a}(k) + N w_p(k) \\ \hat{y}(k) &= \hat{C}(k) \hat{a}(k) + M v_p(k) \end{aligned} \tag{12.16}$$

where the design parameters M, N are chosen to be positive definite matrices and the artificial noise processes w_p, v_p are white, zero-mean, uncorrelated, and have known covariance matrices Q and R respectively. The next theorem states the convergence of the estimation error.

Theorem 12.5 [236] *Consider the deterministic system* (12.14) *and the Kalman filter associated with* (12.16). *Suppose that the deterministic system is uniformly observable and* $A_f(k)$ *is invertible for all* k, *and that* $|A| := \sup\{|A_f(k)| : k \geq 0\}$ *and* $|C| := \sup\{|C_k| : k \geq 0\}$ *are bounded. Then the Kalman filter for the noisy system* (12.16) *is a global, uniform asymptotic observer for the deterministic system* (12.14).

For the derivation of this theorem refer to [236]. We will now focus on the details pertaining to our specific model. From (12.14b) we can see that $A_f(k) = A_f = I$ is constant, bounded, and invertible for all k. The output matrix is built by using the measurements of the deterministic system. For unstable systems it is required that the initial condition of the system is finite. The matrices M, N are simply chosen to be identity matrices of appropriate dimensions. The problem here is that the pair $\left(A_f, \hat{C}(k)\right)$ is not observable. A simple solution is to increase the number of measurements used in the output equation (12.14a) as shown in Sect. 12.4, although this is not a necessary condition. A simulation-based discussion on the convergence of the Kalman filter used as a parameter estimator when the deterministic system is not observable is also presented in that section.

Remark Note that for systems of the form $x(k+1) = Ax(k)$, we do not need any external input in order to successfully identify all elements of the state matrix A; we only need nonzero initial conditions on the state. In MB-NCS we mainly focus on

stabilization of systems of the form given by (12.1), and for this case we can estimate the elements of both matrices A and B if we receive measurements of the state and the deterministic input $u(k)$. As it was expressed before, any common inputs such as steps and sinusoidal inputs can be used for identification purposes. Sinusoidal inputs do not need to include any particular frequency i.e., there is no requirement of a persistently exciting input, which makes this approach a suitable tool for adaptive stabilization. If we try to estimate the parameters online and at the same time try to stabilize the system, then the same input that is used for stabilization can be used for estimation purposes. This relaxation on the need of a persistently exciting input represents a great advantage over traditional parameter estimation methods and it can be intuitively explained by realizing that we are estimating the states, not the parameters, of a modified linear system. In this case we try to meet the conditions for state estimation such as those expressed in Theorem 12.5. In general, we try to find the best state estimate (the estimated parameters in our case) that minimize the second order moment of the difference between the output of the system $x(k)$ and our predicted output $\overline{x}(k)$. It is also important to note that one particular case in which it is not possible to estimate the parameters is when the initial state of the system is 0. Since the main objective is stabilization of unstable linear systems then this scenario does not represent a problem. Now, if a perturbation happens to move the state from its equilibrium point, then we are in a position to estimate the parameters and to compute a stabilizing control input.

Remark Note that we are not performing closed-loop identification since most of the time there is no communication between sensor and controller and the model parameters are not updated until the sensor decides to broadcast a measurement, thus the overall system operates in open-loop mode.

12.2 Adaptive Stabilization of Stochastic MB-NCS

In this section we study the case in which stochastic systems of the form:

$$\begin{aligned} x(k+1) &= Ax\big(k\big) + Bu\big(k\big) + w\big(k\big) \\ y(k) &= x\big(k\big) + v\big(k\big) \end{aligned} \tag{12.17}$$

are implemented using the configurations of Fig. 12.1. The noise processes w and v are white, Gaussian, uncorrelated, zero-mean, and have known covariance matrices Q and R respectively. The model of the system is still given by (12.2) and, since we only measure $y(k)$, the error is now given by:

$$e(k) = \hat{x}\,(k) - y(k). \tag{12.18}$$

Theorem 12.6 *Assume $x(0)$ is a random variable with Gaussian distribution (μ_0, Σ_0). Then the state of the stochastic model-based networked system* (12.17)

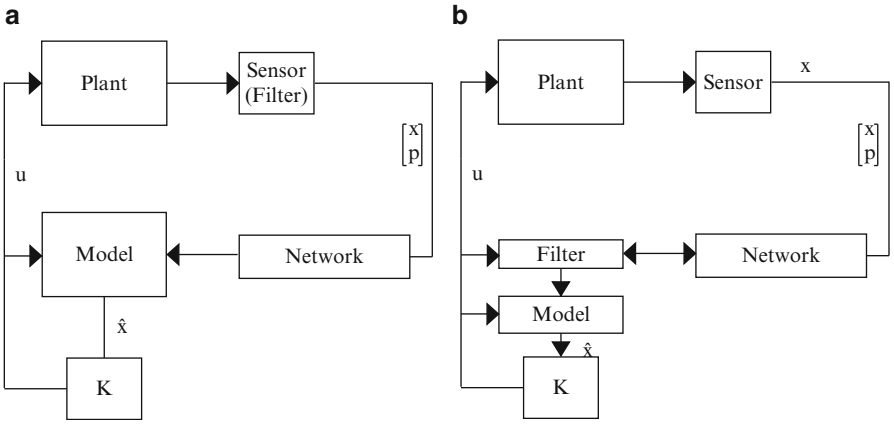

Fig. 12.1 MB-NCS with filter implemented (**a**) in the sensor node, and (**b**) in the controller node

with feedback based on error events $|e(k)| > \alpha$, *with* $e(k)$ *given by* (12.18) *and fixed threshold* α, *has finite mean and covariance for all* k *if the eigenvalues of A+BK are within the unit circle of the complex plane*.

Proof The state of equation (12.17) can be expressed using the linear system:

$$x(k+1) = (A+BK)x(k) + BKe(k) + BKv(k) + w(k). \tag{12.19}$$

Note that the system is a linear system and not a switched one as the model (12.2). In contrast to the model, the system always operates using the same parameters and its state is never updated or reset. Its inputs are $v(k)$, $w(k)$, and $e(k)$. The last one is a bounded input that is set to 0 at the update times. Then we can see that the state $x(k)$ is a Gaussian random variable for all k with mean and covariance given by:

$$\mu_k = (A+BK)^k \mu_0 + \sum_{j=0}^{k-1} (A+BK)^{k-(j+1)} BKe(j) \tag{12.20}$$

$$\begin{aligned}\Sigma_k &= (A+BK)\Sigma_{k-1}\big(A+BK\big)^T + \big(BKR\big(BK\big)^T + Q\big) \\ &= (A+BK)^k \Sigma_0 \big(\big(A+BK\big)^k\big)^T + \sum_{j=0}^{k-1} (A+BK)^j \Big(BKR(BK)^T + Q\Big)\Big((A+BK)^j\Big)^T.\end{aligned} \tag{12.21}$$

In Theorem 12.1 we showed that (12.20) is bounded if the eigenvalues of $A+BK$ lie strictly inside the unit circle. In order for the covariance Σ_k to converge we need the series (12.21) to be summable as $k \to \infty$ and this is obtained by making $(A+BK)^j$

to converge to 0, i.e., if $A+BK$ is stable then the covariance converges to a finite value. For a formal proof, we use the next theorem:

Theorem 12.7 [137] *Suppose $A+BK$ is stable. Then there is a positive semi-definite matrix Σ_∞ such that $\lim_{k\to\infty}\Sigma_k = \Sigma_\infty$. Moreover, Σ_∞ is the unique solution of the linear equation*:

$$\Sigma_\infty = (A+BK)\Sigma_\infty(A+BK)^T + \Big(BKR(BK)^T + Q\Big). \tag{12.22}$$

By using Theorem 12.7, the covariance Σ_k of the state $x(k)$ in (12.17) is finite for all k if the condition in Theorem 12.6 is satisfied. ♦

12.3 Applications of Parameter Estimation and the Model-Based Approach

12.3.1 Examples and Implementation Cases

Whether a linear Kalman filter or an EKF is used we can implement the filter in the MB-NCS framework using one of two approaches.

Filter collocated with sensor In the configuration shown in Fig. 12.1a the filter is implemented in the sensor node. We assume that copies of the model and controller gain are contained in the sensor to generate the state, which is compared to the measured state, and the input, which is needed to construct the filter equations. The parameters are estimated (and the state error is evaluated, in the event-triggered case) in the sensor node at every sampling time of the system. At the update instants, the sensor will transmit the measured state and the new value of the estimated parameters, or it can send a smaller packet containing only the state of the model if no significant variation has been detected in the parameter values. Intermittent feedback (Chap. 4) is not necessary in this case.

Filter collocated with controller Due to several factors, especially computational limitations in the sensor node, it may be necessary to implement the identification algorithm in the controller node. In this configuration (shown in Fig. 12.1b) the filter in the controller receives a set of measurements (intermittent feedback is needed) that are used for estimation of the parameters of interest. When the estimated variables pass a convergence test, the model is updated with the new value of the parameters and the state of the model is updated using the last measurement available. That is, we use intermittent feedback for parameter identification and instantaneous feedback for control. No model of the plant is needed in the sensor node when using periodic updates. The filter updates directly the model in the controller immediately after its estimates have converged since no network exists between filter and model. For the case when we send the measurements based on checking the state error, we need a

Table 12.1 Estimation error order results for the linear time-varying Kalman filter and the extended Kalman filter

Filter	Random initial conditions for filters	Estimation error order in 200 simulations
Linear time-varying Kalman filter	Uniform: [−3,3]	$10^{-7} - 10^{-4}$
	$a_{ij}(0) \pm \alpha$	$10^{-8} - 10^{-4}$
Extended Kalman filter	Uniform: [−3,3]	10^{1}
	$a_{ij}(0) \pm \alpha$	$10^{-1} - 10^{1}$

copy of the model in the sensor node in order to generate the model state. For this scenario we require the controller node to send back to the sensor node the new estimated parameters and the new calculated controller to update the model in the sensor as it does with the model in the controller.

Comparison between Linear Time-Varying Kalman Filter (LTV-KF) and EKF for parameter identification of deterministic systems The EKF can also be used as parameter estimator of deterministic systems of the form (12.1). However, for the special case when noiseless measurements of the state are available there is a significant improvement in the quality of the estimated parameters given by the LTV-KF compared with those obtained using the EKF, particularly for higher order systems.

As explained at the end of this chapter, the EKF will diverge or provide biased estimates due to many factors, especially if the initial estimates are not sufficiently close to the real parameters we try to estimate. We present a comparison between the linear time-varying Kalman filter model we proposed in this section and the EKF, assuming similar conditions.

Example 12.1 A fourth order deterministic system is given by:

$$x_{k+1} = Ax_k$$

where,

$$A = \begin{pmatrix} 1.7209 & -1.1484 & -2.8700 & -1.8609 \\ 0.9510 & 2.9805 & 2.3617 & -0.3365 \\ -2.6334 & -2.4172 & 0.1133 & 2.6316 \\ 2.8223 & 0.5102 & -2.1791 & -2.7730 \end{pmatrix}$$

For both filters we assume we receive noiseless measurements of the state and that the initial estimates of the elements of the matrix A lie somewhere in the range [−3,3]. Results of multiple simulations are shown in Table 12.1. The main difference given by the simulations of the two filters with the same deterministic system is that both converge to some constant value but the EKF tends to provide biased estimates. The first and third row of Table 12.1 show the estimation error for

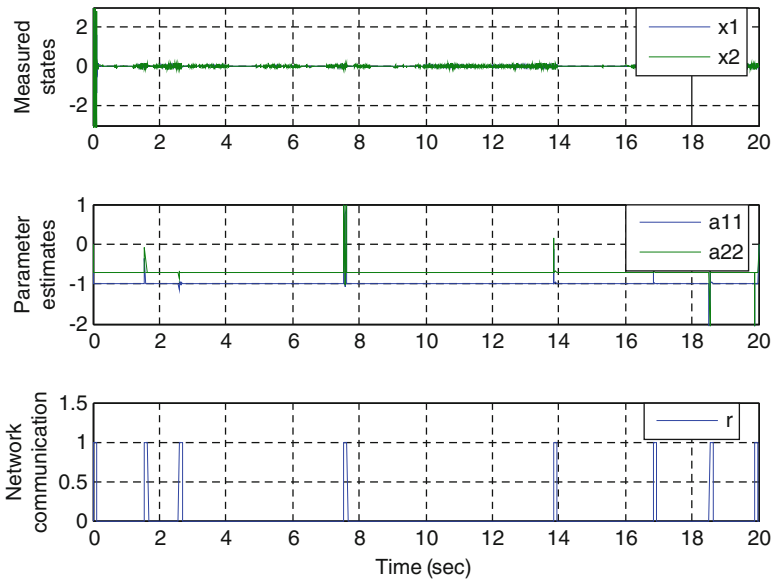

Fig. 12.2 Estimation of two parameters and stabilization of stochastic MB-NCS with intermittent feedback based on events

random initial conditions (uniform $[-3,3]$). The error is given by the difference between the real parameters and their estimates. It can be seen that the error is of several orders greater in the EKF than in the LTV-KF.

In the case we have more accurate previous knowledge of the a priori estimates of the parameters then the results of the EKF show some improvement, emphasizing the dependence of the EKF to the initial estimates $\hat{a}(0)$, where α is a random variable with uniform distribution in [0, 0.1]. The error magnitude is shown in the second and fourth rows of Table 12.1.

Adaptive stabilization examples In both implementation cases shown in Fig. 12.1, sensor and controller implementation of the filter, when the controller node receives or obtains new estimates of the parameters, a discrete-time algebraic Riccati equation is solved using the same weights and the new parameters. Then, we are able to obtain a new stabilizing control law K that reflects the new knowledge about the plant.

Example 12.2 Consider the second order system described by (12.17) with time-invariant but partially unknown parameters and time index $T = 0.01$ s; T is the original time index of the physical system. $B = [1 \quad 1]^T$, the elements $a_{12} = 0.3$, $a_{21} = -1.05$ are known constants, and a_{11}, a_{22} are unknown constants. We implement an EKF in the controller node using intermittent feedback triggered by the state error. Figure 12.2 shows the simulation results. The measured states are shown in the top of the figure. The middle section of the figure shows the estimated values

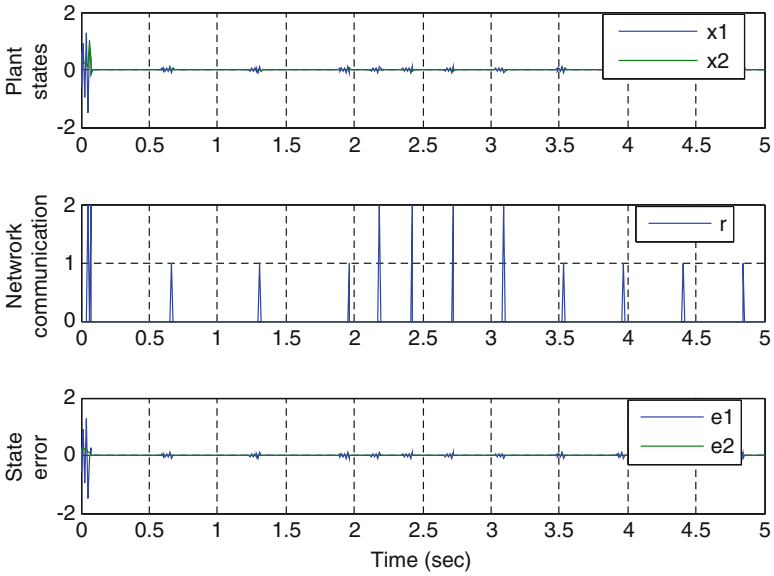

Fig. 12.3 Stabilization of MB-NCS in Example 12.3

of the unknown parameters (for reference, the real values used for the unknown parameters in this example were $a_{11} = -0.985, a_{22} = -0.7$). The update instants are shown in the bottom of the figure. The initial model was inadequate and made the real plant went unstable in the beginning of the execution until we updated the model and recomputed the controller. Every time the controller receives a set of measurements, the filter estimates again the parameters, the rest of the time the system runs in open-loop mode.

Example 12.3 A second order system of the form (12.1) ($T = 0.01$ s) is interconnected to a model-based controller as shown in Fig. 12.1a. All of the elements of the matrices A and B are unknown. The Kalman filter implemented in the sensor location provides estimates of all parameters. We design the filter by using only one previous measurement. When the state error is greater than a predefined threshold the sensor sends the latest measured state to the controller. The most recent parameter estimates are sent only if there is a significant variation with respect to the previous updated parameters. For illustration purposes we construct the communication signal $r(k)$ as:

$$r(k) = \begin{cases} 0 \text{ if no packet is sent} \\ 1 \text{ if only the state is sent} \\ 2 \text{ if both, parameters and state are sent} \end{cases} \tag{12.23}$$

The initial model contained random estimates of the parameters and the control input obtained from that model does not stabilize the real plant. This can be seen at the beginning of the simulation in Fig. 12.3. After a few iterations the sensor is able

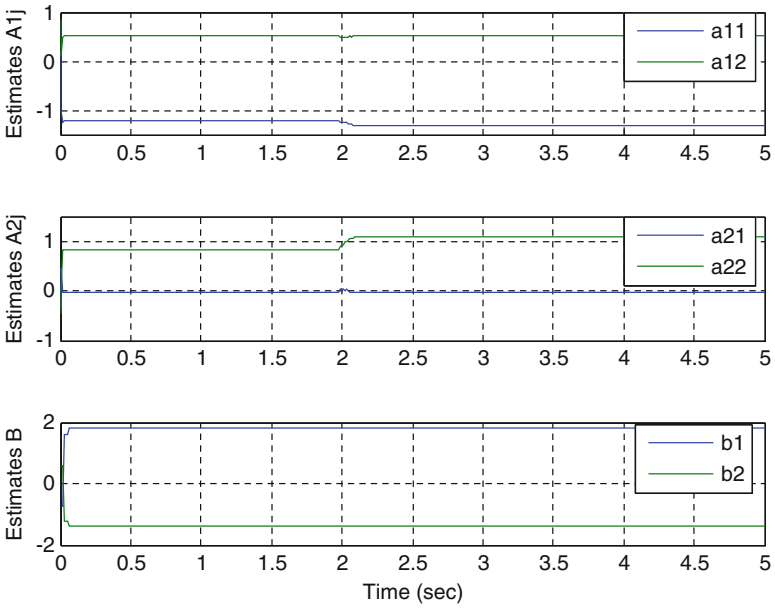

Fig. 12.4 Identification of parameters A and B in Example 12.3

to provide estimates of the parameters, redesign the controller based on the upgraded model, and update the state of the model as well. Variations in the main diagonal components of matrix A of the plant at $t = 2$ s were introduced and successfully identified as shown in Fig. 12.4. The real values used in this example were:

$$a_{11} = -1.231 \rightarrow_{t=2} -1.303 \quad a_{12} = 0.503$$
$$a_{21} = -0.034 \quad a_{22} = 0.820 \rightarrow_{t=2} 1.086$$
$$b_1 = 1.8 \quad b_2 = -1.4$$

In the absence of measurement noise we are able to identify all parameters of the plant with great precision and if we use a linear filter there is no restriction on the initial estimates compared to, for example, the EKF. In Sect. 12.4 we show another practical algorithm that provides global estimates of the states or, in our case, of the parameters of the system, by applying the same ideas of the present section. Its implementation in the MB-NCS framework is also discussed.

12.3.2 Actuator Fault Detection and Reconfiguration

The first application in this section relates to the problem of fault detection and redesign of stabilizing controllers. In the case that a stabilizing controller cannot be found (the fault makes the system not stabilizable) then the control scheme shuts

down the faulty configuration and turns on a healthy backup actuator in order to continue with the operation of the system and maintain closed-loop stability.

Consider an unknown linear discrete-time system represented by:

$$x(k+1) = Ax(k) + B_c u(k) \tag{12.24}$$

with unknown backup actuator matrices:

$$B_i \in \mathrm{B}^{bp} \tag{12.25}$$

where B_c represents the current actuator configuration and B_i represents an element of the backup actuator set B^{bp}. We implement a linear time-varying Kalman filter as in Sect. 12.1. In the case when we can obtain noise-free measurements of the state we can estimate all parameters of the system but here we focus on the elements of B_c. We also need to determine, after some change on the parameters of B_c has been detected, if the system remains stabilizable. A new controller needs to be calculated if the system can be stabilized under the partial fault detected; otherwise shut down the faulty actuator for maintenance and activate and identify the parameters of the new actuator in order to compute the corresponding controller.

Example 12.4 Consider the networked plant be given by:

$$A = \begin{bmatrix} 1.3 & -0.04 & 0.5 \\ -0.62 & -0.7 & 0.37 \\ 0.3 & 0.86 & -1.23 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 & 0.5 \\ 1.5 & 1 \\ 1 & 1 \end{bmatrix}$$

and the backup actuator configurations given by

$$B_2 = \begin{bmatrix} 1.3 & 1 \\ 1 & 0.6 \\ -0.7 & 0.5 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 & 1 \\ 2 & 0.4 \\ 0.1 & 1.5 \end{bmatrix}$$

with $T_s = 0.01$. Figure 12.5 shows the response of the system and the communication signal $r(k)$ as defined in (12.23). A partial fault was detected at 1.5 s. Here the remaining configuration could be stabilized and the new controller was designed for this case. At time 3 s a fault was introduced and successfully detected moments after. The configuration is not stabilizable and the backup configuration B_2 was activated, its parameters were successfully estimated, and the new controller was computed. A similar situation happened at time 6 s and the backup configuration B_3 was activated. The estimated parameters corresponding to the elements of the matrix B_c, that is, of the activated actuator configuration are shown in Fig. 12.6.

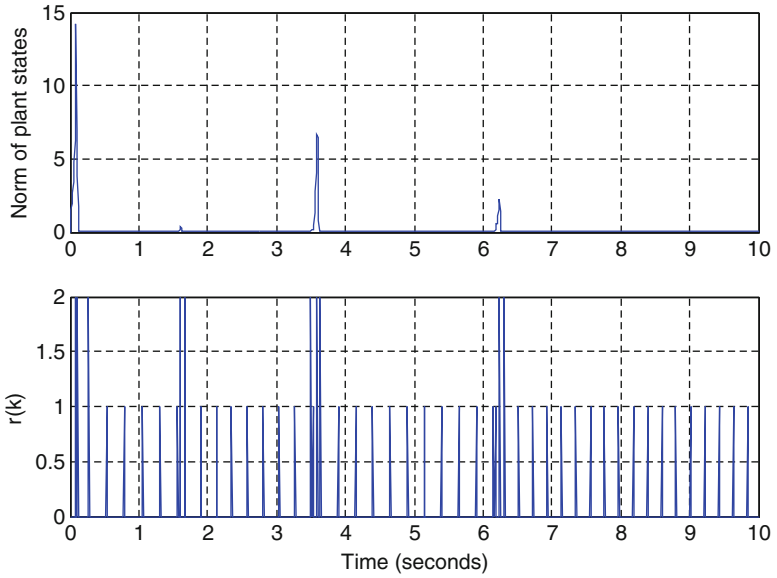

Fig. 12.5 Norm of the states of the system and communication signal

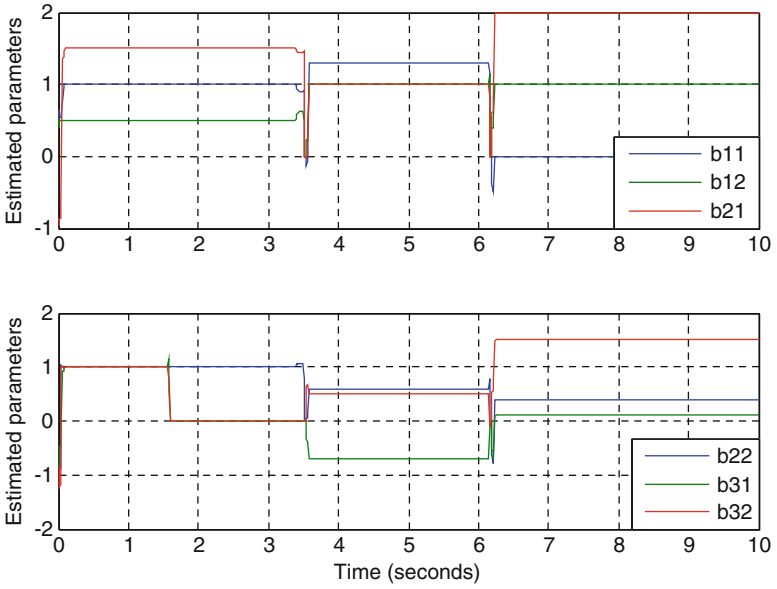

Fig. 12.6 Estimated parameters for the faulty actuator

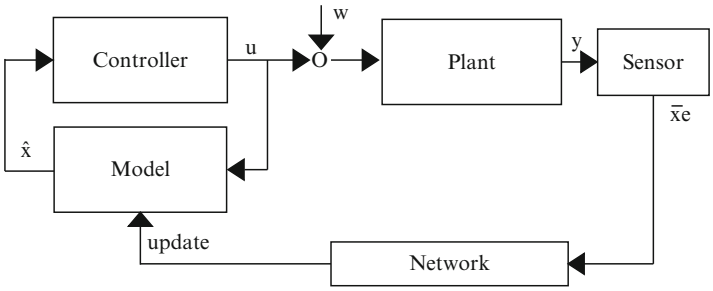

Fig. 12.7 Model-based networked system with external disturbance

12.3.3 Identification and Rejection of Input Disturbances

A very interesting application of the modified Kalman filter that was discussed in Sect. 12.1 is related to the joint estimation of parameters and exogenous disturbances. The aim in this section is to stabilize systems with uncertain parameters that are affected by unknown piece-wise constant external disturbances and as shown in Fig. 12.7. The presence of a communication network does not allow the use of continuous feedback for control and the networked system must operate in open-loop mode for possibly large intervals of time. Due to these communication constraints we use the MB-NCS approach for stabilization and due to uncertainty in the parameters and on the external disturbance we also implement a parameter estimation algorithm in the sensor node.

Consider the following type of uncertain discrete-time system:

$$x(k+1) = Ax(k) + B(u(k) + w(k)) \tag{12.26}$$

where $x \in \mathbb{R}^n$, $u, w \in \mathbb{R}^m$ represent the state of the system, the control input, and the unknown input disturbance, respectively. We consider the special case where B is known in advance but the state matrix A and the input disturbance are unknown. Additionally, we restrict this work to the case when the input disturbances are given by piece-wise constant signals and this represents all the information available about the external disturbance. The time instants at which the external disturbance changes its value are also unknown. Those time instants must be detected by our estimation algorithm.

Consider a second order system with single input for ease of exposition,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} (u(k) + w(k)). \tag{12.27}$$

Note that as in Sect. 12.1 this approach can be easily generalized for any dimensions n and m, as long as the system is of the form (12.26). We rewrite (12.27) in the next form:

$$\begin{bmatrix} \overline{x}_1(k) \\ \overline{x}_2(k) \end{bmatrix} = \begin{pmatrix} x_1(k-1) & x_2(k-1) & 0 & 0 & b_1 \\ 0 & 0 & x_1(k-1) & x_2(k-1) & b_2 \end{pmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(k)$$

$$= \hat{C}(k)\hat{a}(k) + \overline{u}(k). \tag{12.28}$$

Equation (12.28) represents the output equation for the filter. Since we consider LTI systems and piece-wise constant input disturbances the state equation for the filter is given by:

$$\begin{bmatrix} \hat{a}_{11}(k+1) \\ \hat{a}_{12}(k+1) \\ \hat{a}_{21}(k+1) \\ \hat{a}_{22}(k+1) \\ \hat{w}(k+1) \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \\ \hat{w}(k) \end{bmatrix} = \hat{A}\hat{a}(k) \tag{12.29}$$

where the scalars $\hat{a}_{ij}$ represent the estimates of the elements of the matrix A and $\hat{w}$ represents the estimate of the input disturbance.

The overall filter model is a linear time-varying system, similar to the filter model used in Sect. 12.1, and the same convergence properties apply to (12.28) and (12.29).

The control input is now given by

$$u(k) = K\hat{x}(k) - \hat{w}_c(k) \tag{12.30}$$

where $\hat{x} \in \mathbb{R}^n$ represents the state of the model in both the sensor and the controller. A copy of the model is implemented in the controller in order to compute the state error and use an event-triggered control technique. The state of the model is also needed at the sensor node to compute the control signal which in turn is needed by the estimation algorithm. $\hat{w}_c(k)$ represents the value of the last disturbance estimate update. Since the filter in the sensor node may be producing estimates at each instant in order to check for a possible change in the parameters or in the value of the disturbance, $\hat{w}_c(k) = \hat{w}(k)$ holds only at the time of a parameter update while in general they have slightly different values. A large difference between the current estimated disturbance and the previous value that was sent to the controller will also trigger an update event, and $\hat{w}_c(k)$ has to be updated using the current estimate $\hat{w}(k)$.

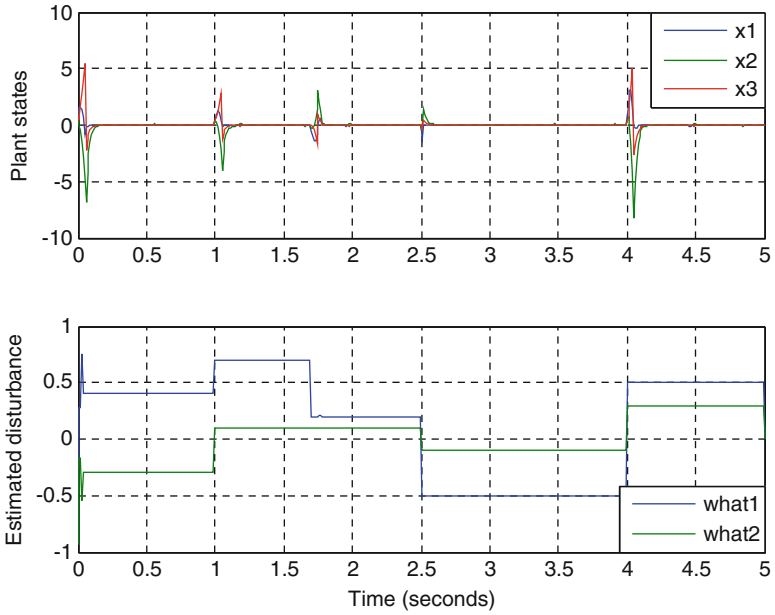

Fig. 12.8 States of the uncertain system (*top*) affected by an external disturbance. Estimates of the external disturbance (*bottom*)

Example 12.5 Consider an unstable and uncertain system given by (12.26) with dimensions $n = 3$, $m = 2$. The goal is to stabilize the system around its equilibrium point $x_e = [0\ 0\ 0]^T$ in the presence of unknown input disturbances. The input matrix is known and given by:

$$B = \begin{pmatrix} 1 & 0 \\ 0.5 & 0.2 \\ 0 & 1 \end{pmatrix}.$$

Figures 12.8, 12.9, and 12.10 show the results of the simulation. Figure 12.10 shows the estimated parameters of the state matrix A. These estimated parameters are used to update the model parameters and to redesign the controller gain. Figure 12.8 (bottom) shows the successful estimation of the input disturbance which is used in (12.30) to reject the real input disturbance $w(k)$. Figure 12.8 (top) shows the response of the system. When the disturbance changes values there is an undesirable effect in the system but after a few iterations we are able to estimate the new value of the disturbances in order to counteract this response and stabilize the system once again. Note that the individual value of the disturbances need not change at the same instant in order to detect and estimate the new disturbance value. At time about 1.7 s only $w_1(k)$ changes its value while $w_2(k)$ remains the same. The state error and the communication instants are shown in Fig. 12.9.

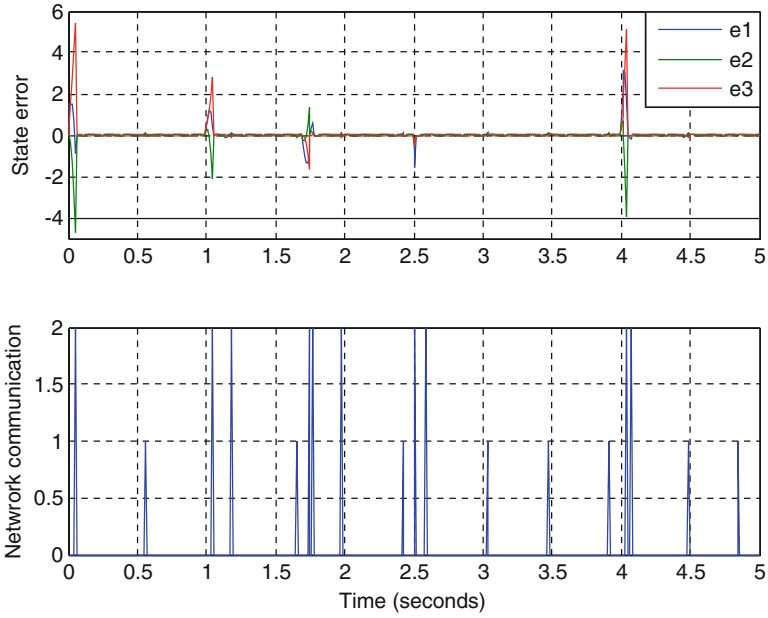

Fig. 12.9 State error and communication signal for Example 12.5

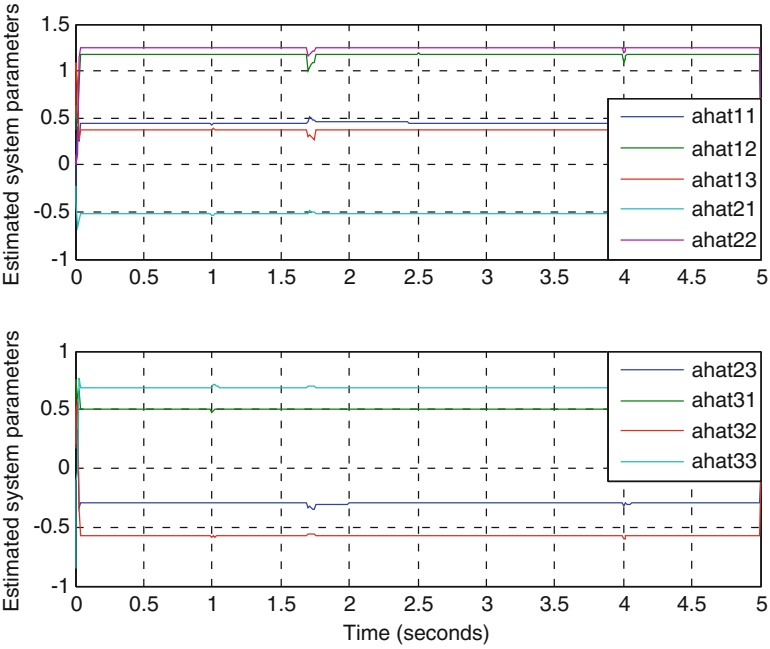

Fig. 12.10 Estimated parameters of the state matrix A in Example 12.5

This example shows that this approach offers good disturbance attenuation properties. In particular we are able to reject the disturbance effect on the system a few iterations after the disturbance changes suddenly.

The overall approach discussed in the present chapter, Model-Based Event-Triggered (MB-ET) control combined with Kalman filter-based parameter and disturbance estimation, is able to react relatively fast to the presence of unknown disturbances and to parameter uncertainties while reducing the number of feedback measurements that need to be sent from sensor to controller.

12.3.4 Identification and Stabilization of Switched Systems with State Jumps

The application presented in the present subsection considers the interesting problem of identification and stabilization of switched systems with state jumps and unknown switching sequences. The implementation of the Kalman filter for identification of the parameters of each operating mode is straightforward, assuming we can obtain noise-free measurements of the states. Consider a discrete-time switched system:

$$x(k+1) = A_i x(k) + B_i u(k) \tag{12.31}$$

for each $i \in \mathrm{N}$, N denotes the set $\{1, 2, \ldots N\}$ of N integers, where each $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m}$ are unknown and each A_i may be unstable. In addition the switching sequence, denoted by times τ_{ij} equal to the times k when the system switches from mode i to mode j for $i, j \in \mathrm{N}$, is also unknown. Finally, when the system switches, the state is not restricted to keep the same values but it can be reset to a different finite value, that is, $x(\tau_{ij}) = f(\tau_{ij})$. The state jumps are dictated, in general, by a time-varying function which can be nonlinear, stochastic, or any other function as long as $f(\tau_{ij}) < \infty$.

In order to identify the switching times we implement an event-error strategy. As in Sect. 12.1, the event-based scheduler will trigger a state update if the state of the system is different by a given amount with respect to the state of the model, that is, it computes the norm of the state error (12.3) and compares it to a fixed threshold. In addition, the sensor node also checks for the difference between the current estimates of the system parameters with respect to the latest parameter updates and it will send a parameter update if the current estimates are significantly different, which corresponds, in general, to a switch in the parameters of the real system. From time to time the sensor may send parameter updates even though the real system has not switched yet, depending of the convergence threshold we implement. This updates are basically refinements of the estimated parameters that were sent previously. The application of the Kalman filter and an event-triggered strategy for the identification and control of this type of switched systems can be implemented for both networked and non-networked systems. In the case of

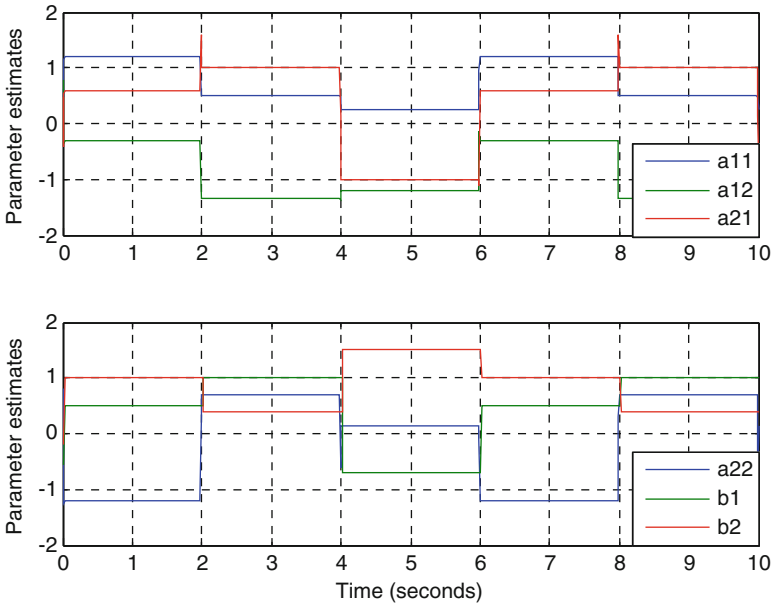

Fig. 12.11 Estimated parameters of the switched system

non-networked systems the controller is able to receive feedback measurements at all times and the identified parameters are used solely for the design of the corresponding stabilizing controller. For the case of networked systems we implement the MB-NCS setup discussed in this and previous chapters.

Example 12.6 Consider a model-based networked switched system with state jumps that operates using the next set of unstable (unknown) modes:

$$A_1 = \begin{bmatrix} 1.2 & -0.3 \\ 0.6 & -1.2 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$
$$A_2 = \begin{bmatrix} 0.5 & -1.35 \\ 1.01 & 0.7 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}$$
$$A_3 = \begin{bmatrix} 0.24 & -1.2 \\ -1 & 0.13 \end{bmatrix}, \quad B_3 = \begin{bmatrix} -0.7 \\ 1.5 \end{bmatrix}$$

Figures 12.11 and 12.12 show the results of simulations. The sampling time of the real system is $T_s = 0.01$ s. Figure 12.11 shows the estimated parameters as a function of time; the figure shows the estimated elements of A and B as they switch over time. Figure 12.12 shows the response of the switched system and the communication signal $r(k)$ as defined in (12.23). This example shows that the switching parameters of the system can be successfully identified and the model can be updated in order to obtain better estimates of the state between feedback updates. Additionally, the switching sequence does not have to be known in

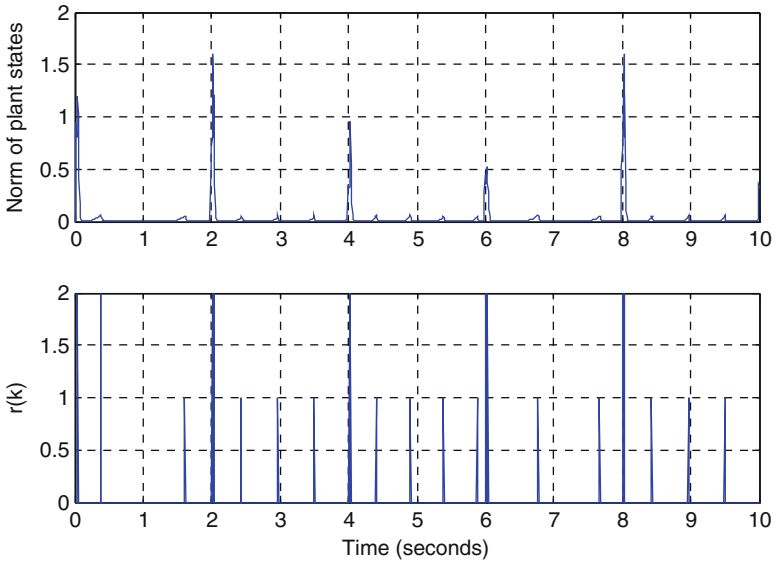

Fig. 12.12 Norm of the states of the system and communication signal for Example 12.6

advance since the sensor is able to detect the switching times and adjust the parameters of the new model. Similarly, a new control gain is designed in order to stabilize the current operating mode of the switched system. The switching sequence that was used here, starting with mode 1, is:

$$\tau_{12} = 2, \;\; \tau_{23} = 4, \;\; \tau_{31} = 6, \;\; \tau_{12} = 8.$$

One drawback of this control approach is that every time that the system switches modes the controller node needs to wait until the new parameter estimates converge within given bounds.

Using previous parameter values, an important extension can be made to this control algorithm in order to allow the overall MB-NCS learn or remember past modes of the switched system as follows. The estimated parameters are stored in the sensor node and used for future switching times. When a state jump is detected based on the state error the sensor uses past "learned" modes in order to predict the next states. The sensor node uses the next predicted error in order to select a past mode and send the corresponding parameters. The identification keeps executing as a backup option or in order to refine the estimated parameters. This type of implementation results, in general, in a better performance as shown in the next example.

Example 12.7 Consider the same unknown system and switching sequence as in Example 12.6. The response of the switched system and the communication signal are shown in Fig. 12.13. Note that at switching times $\tau_{31} = 6$, and $\tau_{12} = 8$ the system

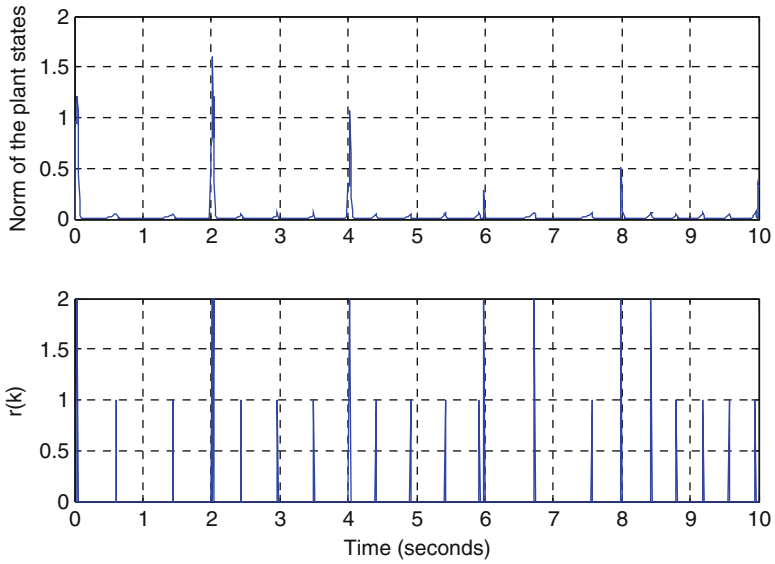

Fig. 12.13 Norm of the states of the system and communication signal for the learning control system

switches to previously identified modes and therefore the sensor is able to update the parameters of the model sooner than in the previous example and the system performs better at those instants.

12.4 Parameter Identification Using Additional Measurements

12.4.1 Parameter Identification Using Luenberger Observers

Following the ideas in Sect. 12.1 we present now a similar modification regarding the model of the system and the implementation of the Luenberger observer that will treat the unknown parameters of the system as the states to be observed. This technique also provides global asymptotic estimates of the parameters i.e., we can provide any initial conditions and the observer will converge to the real values of the parameters. Further, the computations required by this algorithm are only those related to the computation of the observer gain $L(k)$. Similarly to the results in Sect. 12.1, the modified model is linear time-varying.

Suppose we receive perfect measurements of the state of the system described by (12.1) and also assume that we know exactly B and want to identify the unknown elements of A.

The output equation of our model is similar to (12.14a), but since the computation of the observer gain is very sensitive to the observability condition of the new model, we increase the number of measurements used in the model to obtain, for example in the case of a second order system:

$$\begin{bmatrix} \overline{x}_1(k) \\ \overline{x}_2(k) \\ \overline{x}_1(k-1) \\ \overline{x}_2(k-1) \end{bmatrix} = \begin{bmatrix} x_1(k-1) & x_2(k-1) & 0 & 0 \\ 0 & 0 & x_1(k-1) & x_2(k-1) \\ x_1(k-2) & x_2(k-2) & 0 & 0 \\ 0 & 0 & x_1(k-2) & x_2(k-2) \end{bmatrix} \begin{bmatrix} \hat{a}_{11}(k) \\ \hat{a}_{12}(k) \\ \hat{a}_{21}(k) \\ \hat{a}_{22}(k) \end{bmatrix}$$

$$+ \begin{bmatrix} b_1 & 0 \\ b_2 & 0 \\ 0 & b_1 \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \end{bmatrix} = \hat{C}(k)\hat{a}(k) + \hat{D}\begin{bmatrix} u(k) & u(k-1) \end{bmatrix}^T. \tag{12.32}$$

The state equation for our model is given by (12.14b). The observer, in general, is expressed by:

$$\begin{aligned} \hat{a}(k+1) = \big(I - L(k)\hat{C}(k)\big)\hat{a}(k) + L\begin{bmatrix} x(k) & x(k-1) & \dots \end{bmatrix}^T \\ - L\hat{D}\begin{bmatrix} u(k) & u(k-1) & \dots \end{bmatrix}^T \end{aligned} \tag{12.33}$$

where $\hat{a}(k) = [\hat{a}_{11}(k) \quad \hat{a}_{12}(k) \quad \hat{a}_{21}(k) \quad \hat{a}_{22}(k)]^T$ is the state of the observer and represents the estimation of the elements of A in system (12.1). The matrices in the output equation and the inputs for the observer are increased with an additional measurement for every single increase in the order of (12.1).

Theorem 12.8 *The observer* (12.33) *with augmented measurements and the pair* $\big(I, \hat{C}(k)\big)$ *observable, converges globally to the real parameters of A in* (12.1) *if the eigenvalues of* $I - L(k)\hat{C}(k)$ *are within the unit circle of the complex plane.*

Proof Define the estimation error:

$$e(k) = a(k) - \hat{a}(k) \tag{12.34}$$

where $a(k) = vec(a_{ij})$ is a column vector that contains the true values of A in (12.1). Now, we compute:

$$\begin{aligned} e(k+1) &= Ia(k) - \big(I - L(k)\hat{C}(k)\big)\hat{a}(k) \\ &\quad - L\begin{bmatrix} x(k) & x(k-1) & \dots \end{bmatrix}^T + L\hat{D}\begin{bmatrix} u(k) & u(k-1) & \dots \end{bmatrix}^T \\ &= Ie(k) + L(k)\hat{C}(k)\hat{a}(k) - L(k)\hat{C}(k)a(k) \\ &\quad - L\hat{D}\begin{bmatrix} u(k) & u(k-1) & \dots \end{bmatrix}^T + L\hat{D}\begin{bmatrix} u(k) & u(k-1) & \dots \end{bmatrix}^T \\ &= \big(I - L(k)\hat{C}(k)\big)e(k), \end{aligned}$$

where relation (12.32) was used involving the real parameters. ♦

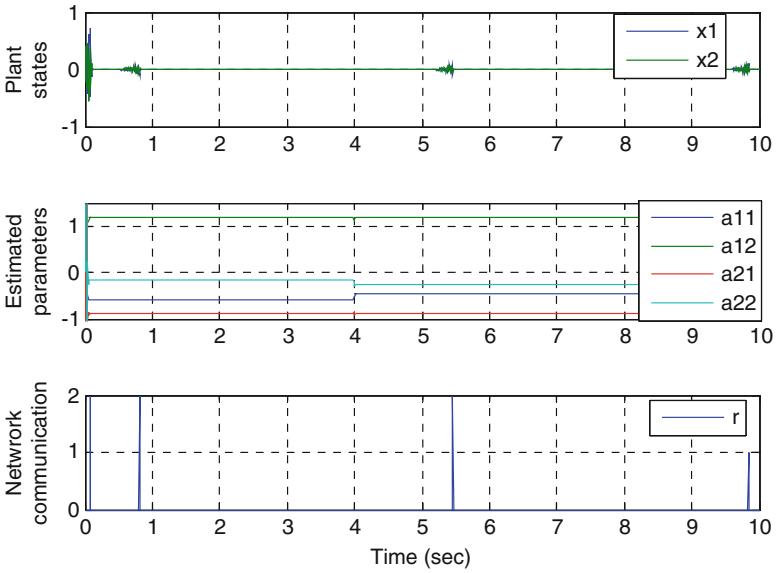

Fig. 12.14 Stabilization of MB-NCS in Example 12.8

Remark Note that given the increased number of measurements available and the form of the observer state matrix (identity matrix) we could simply find the inverse of the built C matrix to obtain an estimation of the parameters. The advantage therefore is in the freedom of choice of the desired pole locations in order to compute the gain $L(k)$ compared to the computation of a matrix inverse.

Example 12.8 Consider a second order unstable system described by (12.1) with known $B = (\,1 \quad 0.8\,)^T$ and the unknown, time-varying values of the state matrix are shown below. A state observer is implemented in the sensor node. The estimated parameters are shown in Fig. 12.14. Variations in the parameters were introduced at $t = 4$ s. The same figure shows that, after the model is updated, the states of the plant are bounded and the plant operates in open loop for longer intervals of time. Updates from the sensor to the controller take place when an error measure is greater than some predefined fixed threshold. It can be seen that although the parameters have been successfully identified after the variations the plant keeps working in open loop until an error triggers an update. The real parameters used in this example were:

$$a_{11} = -0.6 \rightarrow_{t=4} -0.45 \quad a_{12} = 1.18$$
$$a_{21} = -0.9 \quad a_{22} = -0.17 \rightarrow_{t=4} -0.27$$

12.4.2 Convergence Properties of the LTV-KF Used for Parameter Identification

The linear time-varying Kalman filter described in Sect. 12.1 has been used for parameter identification of systems in general state-space form, but it also can be seen as an estimator of the states of a modified system model. Typical conditions for convergence of the Kalman filter require the pair (A,C) to be observable [32, 137, 267]. In particular, [236] shows that the linear time-varying Kalman filter is a global asymptotic observer for the underlying deterministic system. As noted in [231], observability is not a necessary condition, i.e., there may be a solution for the Kalman filter equations even if the observability condition does not hold. If the filter is capable of retaining previous measurements (up to n of them) then we could use the increased order model as in Sect. 12.4.1 to obtain the observable pair. This will also increase the complexity of computations. Fortunately this is not required and the measurement of the state at the previous instant k-1 only, such as in the examples in this chapter, is able to provide enough information about the unknown parameters of the plant and even for higher order systems as in Example 12.1. This is of great advantage in order to keep a low order model and especially if the sensor is not able to retain state measurements beyond the previous ones. Moreover, the improvement obtained by using more measurements is minimal as shown in the following example.

Example 12.9 Consider the third order system with random initial conditions described by:

$$x(k+1) = Ax(k)$$

$$A = \begin{pmatrix} 1.1806 & -0.7323 & -1.8800 \\ -1.2073 & 0.6673 & 2.0203 \\ 1.6078 & -0.1910 & -1.7223 \end{pmatrix}.$$

The elements of matrix A are assumed to be unknown but time-invariant. Table 12.2 shows the order in the magnitude of the estimation error using a single measurement to construct the output matrix of the filter model and also using n previous measurements. Notice that an almost unnoticeable increase in the precision of the estimates is gained by increasing the number of measurements.

Table 12.2 Estimation error order results for the linear time-varying Kalman filter using a single measurement and using $n = 3$ measurements

Number of measurements in LTV-KF	Estimation error order in 200 simulations
1	$10^{-6} - 10^{-9}$
n	$10^{-6} - 10^{-10}$

12.5 Notes and References

The present chapter discussed implementations of Kalman filters in order to estimate the parameters of the real plant and upgrade the model parameters. It has been shown that upgraded model implies better state estimates between update time intervals and these intervals can be made longer as well. The results presented in this chapter were published in [83] and [88] with the exception of the work described in Sects. 12.3.2, 12.3.3, and 12.3.4, which represent original results first presented in this book.

The EKF whether it is used for estimation of states of nonlinear systems or combined estimation of states and parameters is prone to divergence. The EKF lacks the robustness and the convergence properties of the linear Kalman filter. Many of the causes for the estimates to be biased or divergent have been illustrated and somewhat successful remedies have been proposed in many papers and books, see for instance [204, 231, 236, 254]. A more rigorous analysis of the convergence properties of the EKF used as a parameter estimator for linear systems may be found in [162].

The most common causes of divergence in the EKF are related to the fact that the EKF is based on linearization about the current estimate, if the a priori state estimates are poor, or if later estimates should take the filter out of the linear region, the estimates often diverge [204]. In general, the Taylor series expansion works well only when the expansion is taken in the neighborhood of $\widetilde{x}_k \approx x_k$ ($\widetilde{x}_0 \approx x_0$), then good initial estimates are needed. Also, if the system is highly nonlinear we run the risk to be taken out of the linearization region. In [32] the authors compare the EKF with the regular linearized Kalman filter; in particular, they address the concern that the EKF is a riskier filter than the linearized version when the initial uncertainty and measurement error are large.

Song and Grizzle [236] state that the EKF is a quasi-local observer i.e., the initial condition should be close to the real parameter that we are trying to estimate. In that paper, strong conditions for convergence of the EKF are postulated. These conditions are related to the observability of the underlying nonlinear system. A somewhat weaker observability rank condition leads to similar results but this condition must be accompanied by a convergence period condition; in other words, uniform asymptotic convergence of the observed error is achieved whenever the nonlinear system satisfies an observability rank condition and the states stay within a convex compact domain.

In practice, several approaches have been proposed to improve the convergence characteristics of the EKF, for instance, if the gain K_k obtained from the linear approximation is not good enough, then use a second, third, ... order approximation. A known approach is the second order EKF, where a second order Taylor series expansion of $f(\bullet)$ and $h(\bullet)$ is performed. Another computational solution is the iterative EKF, in this version one refines the point at which the first order Taylor series expansion of the measurement equation is performed.

Ljung [162] provides a rigorous analysis of convergence of the estimates of the EKF to the true values of the system for the special case when the EKF is used for parameter identification of linear stochastic systems. In that paper, it is expressed that bias estimates come from incorrect noise assumptions associated with the model and not from the EKF-method. An adaptive Kalman filter is used to learn the correct noise covariance, see [32, 168] for some details.

According to Ljung [162] divergence of the EKF as a parameter estimator can be traced to the fact that effects of a change in the parameters on the Kalman gain are not taken care of, i.e., the coupling between the Kalman gain and the parameters to be estimated is not considered. Ljung is able to associate a differential equation in order to analyze the convergence of a recursive, stochastic algorithm in this case the EKF. The differential equation is defined in terms of the equations that characterize the EKF, with a slight modification for the case when the parameter vector to be estimated is kept constant. This differential equation provides insight into convergence points of the estimate and the causes of divergence and biased estimates, and for some examples, it is possible to obtain the domain of attraction based on this differential equation see [162, section V.C]. For the cases when the Kalman gain is independent of the parameter vector the convergence properties are much better. One interesting example of this case is deterministic systems where the covariance of the process noise is always 0. The main idea of this paragraph is to establish a link between the Kalman gain and the vector of parameters. The approach is to improve convergence in the general case (non-deterministic systems, as specified above) by including a term that relates the variations in the parameters to the Kalman gain. This leads to a modified algorithm based on an innovation representation model that has much better convergence properties.

Chapter 13
Multirate Systems

In many applications digital communication networks are used to interconnect control elements corresponding to large scale uncertain systems with multiple outputs and spatially distributed sensing implementation. In this type of implementation the sensors that measure different elements of the state vector can be located at distant positions. A limited-bandwidth communication network is used to send all sensor measurements to the controller. In order to schedule the sampling and sharing of information and to increase the sampling intervals at each sensor node as much as possible, in this chapter we implement a multirate sampling scheme. The results presented provide a simple approach to address robustness to parameter uncertainties using multirate sampling.

In addition, this chapter also considers a different architecture using a similar multirate approach. A two-channel implementation is studied, where not only the path from sensor to controller is implemented using a limited-bandwidth network, but also the path from controller to actuator.

This chapter is organized as follows. In Sect. 13.1 we study the problem where different sensors measure and transmit the states of the same system. In order to reduce network traffic, we implement a multirate approach. Additional results are provided considering a network constraint in the form of single access to the network, that is, at every sampling instant corresponding to the plant transitions at most one node can have access to the network to transmit its measurement. A strict scheduling protocol is designed in order to satisfy this constraint. Two-channel network configurations are discussed in Sect. 13.2. First, we present stabilization results using a similar to Sect. 13.1 multirate approach. Secondly, we derive conditions for bounded tracking error concerning the reference input problem and using an event-triggered control strategy. Section 13.3 provides notes and references.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_13,

13.1 Multirate State Updates Using a Centralized Controller

In this section we use the MB-NCS approach and the lifting procedure to establish stability conditions for discrete-time spatially distributed systems when the plant states are sampled periodically but using different sampling rates. This multirate approach provides good performance in terms of network resource utilization especially when different sensors are used to measure different elements of the system output, see Fig. 13.1.

13.1.1 Model-Based Multirate Approach

When the sensors use the same network to transmit their measurements to the controller node, then the multirate approach brings important benefits to the operation of the overall networked system. By allowing the sensors to transmit their measurements using different update periods, we avoid packet collisions and networked induced delays compared to the case when all of the sensors need to sample and transmit at the same instants. Additionally, we will show that in many cases a further reduction in network communication can be obtained by using different update rates for each sensor. Although the multirate sampling case requires a more complex analysis, the same lifting approach as in Sect. 2.4 can be used in order to find a system representation for the LTI equivalent (lifted) system.

Consider a multi-output system depicted in Fig. 13.1. In this case we do not assume that sensors measure all states at the same time; instead we consider a spatially distributed system for which different sensors measure different elements of the state and send this information to the centralized controller at different rates. The states of the model are partially updated according to the information that is received at any given update instant.

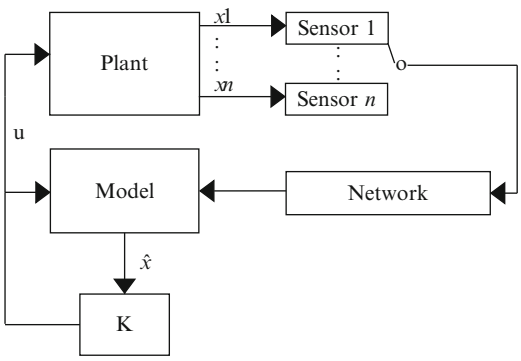

Fig. 13.1 Networked control system with centralized controller and distributed multirate sensor measurements

In what follows we will provide details on how the states of the model are partially updated and how to obtain the response of the corresponding lifted system. The lifted system state equation directly provides necessary and sufficient conditions for the stability of the multirate system.

We consider linear time-invariant systems and models with dynamics described by:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ \hat{x}(k+1) &= \hat{A}\hat{x}(k) + \hat{B}u(k) \end{aligned} \tag{13.1}$$

where $u = K\hat{x}$. In this section we approach the multirate problem by considering an N-partition of the state of the system and, correspondingly, the model (13.1), according to the number of sensors that are used to obtain measurements:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_N \end{bmatrix} \tag{13.2}$$

where $x_i, \hat{x}_i \in \mathbb{R}^{ni}$, $\sum_{i=1}^{N} ni = n$. The case when each scalar element of the state is measured by a different sensor only represents a particular case of (13.2) so, in general, elements of the state may be grouped into different subsets and, consequently, each subset of the system state may have different dimensions. Let the integer s_i represent the update period that sensor i uses to send measurements in order to update the corresponding part of the model state $\hat{x}_i$. Let s represent the minimum common multiple of all s_i.

In order to obtain the response of the multirate system using the model-based control input with partial updates we define all the update instants within a period s by arranging the periods s_i and its multiples up to before s in increasing order as follows:

$$\begin{aligned} &s_1, 2s_1, \ldots, (r_1 - 1)s_1 \\ &s_2, 2s_2, \ldots, (r_2 - 1)s_2 \\ &\vdots \\ &s_N, 2s_N, \ldots, (r_N - 1)s_N \end{aligned} \tag{13.3}$$

where the integer $r_i = \frac{s}{s_i}$ for $i = 1 \ldots N$ represents the relative update rate compared to the update rate given by the period s. Let h_i for $i = 1 \ldots p-1$ represent the update instants in increasing order, where p is the total number of update instants within a period s including the update at time s. Note that at any given instant one or more sets of states $\hat{x}_i$ can be updated. This procedure can be better shown through a simple example.

Consider the state of a system that is partitioned into three subsets x_1, x_2, x_3 with corresponding periods $s_1 = 3, s_2 = 4, s_3 = 6$. Then we proceed to define all update instants within a period $s = 12$, as follows:

$$\begin{aligned} h_1 &= 3 = s_1 \\ h_2 &= 4 = s_2 \\ h_3 &= 6 = 2s_1 = s_3 \\ h_4 &= 8 = 2s_2 \\ h_5 &= 9 = 3s_1 \end{aligned} \tag{13.4}$$

Note that at the time instant h_3 we have two partial updates for this example. Let us define, in general, the partial update matrices:

$$I_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_{ni} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{13.5}$$

that is, the ith partial update matrix $I_i \in \mathbb{R}^{n \times n}$ contains an identity matrix of size ni at the position corresponding to x_i and zeros elsewhere. Define

$$I_{h_i} = I_i + I_j + I_k \ldots \tag{13.6}$$

The matrices I_{h_i} represent all updates that happen at time instant h_i.

Theorem 13.1 *The uncertain system with distributed sensors as shown in Fig. 13.1 with model-based control input and with partial multirate model updates is asymptotically stable for a given selection of update periods s_i if and only if the eigenvalues of*

$$A^s + \sum_{i=1}^{p} A^{h_p - h_i} \Xi_{h_i - h_{i-1}} U_{h_i - 1} \tag{13.7}$$

are within the unit circle of the complex plane, where

$$\begin{aligned} \Xi_{h_i - h_{i-1}} &= \sum_{j=0}^{h_i - h_{i-1} - 1} A^{h_i - h_{i-1} - 1 - j} BK \left(\hat{A} + \hat{B} K\right)^j \\ U_{h_i} &= (I - I_{h_i}) \left(\hat{A} + \hat{B} K\right)^{h_i - h_{i-1}} U_{h_i - 1} + I_{h_i} \left(A^{h_i} + \sum_{q=1}^{i} A^{h_i - h_q} \Xi_{h_q - h_{q-1}} U_{h_q - 1} \right) \end{aligned} \tag{13.8}$$

with $h_0 = 0, \;\; U_{h_0} = I, \;\; h_p = s.$

Proof Let us consider the beginning of a period s. At this time instant, all sensors send measurements and we have that $\hat{x}(ks) = x(ks)$. At the time of the first update after time ks, that is, at time $ks + h_1$ we have:

$$x(ks+h_1) = \left(A^{h_1} + \sum_{j=0}^{h_1-1} A^{h_1-1-j} BK\left(\hat{A} + \hat{B}K\right)^j\right)x(k) = \left(A^{h_1} + \Xi_{h_1}\right)x(ks) \quad (13.9)$$

and the model state after the update has taken place is given by:

$$\hat{x}(ks+h_1) = \left((I - I_{h_1})\left(\hat{A} + \hat{B}K\right)^{h_1} + I_{h_1}\left(A^{h_1} + \Xi_{h_1}\right)\right)x(k) = U_{h_1}x(ks). \quad (13.10)$$

Following a similar analysis we can obtain the response of the system at time $ks + h_2$ as a function of $x(ks + h_1)$ and $\hat{x}(ks + h_1)$:

$$x(ks+h_2) = A^{h_2-h_1}x(ks+h_1) + \sum_{j=0}^{h_2-h_1-1} A^{h_2-h_1-1-j} BK\left(\hat{A} + \hat{B}K\right)^j \hat{x}(ks+h_1). \quad (13.11)$$

Similarly, the response of the model at time $ks + h_2$ as a function of $x(ks + h_1)$ and $\hat{x}(ks + h_1)$ is given by:

$$\hat{x}(ks+h_2) = (I - I_{h_2})\left(\hat{A} + \hat{B}K\right)^{h_2-h_1}\hat{x}(ks+h_1) + I_{h_2}x(ks+h_2) \quad (13.12)$$

but, since both $x(ks + h_1)$ and $\hat{x}(ks + h_1)$ can be expressed in terms of $x(ks)$, we obtain the following:

$$x(ks+h_2) = \left(A^{h_2} + A^{h_2-h_1}\Xi_{h_1} + \Xi_{h_2-h_1}U_{h_1}\right)x(ks) \quad (13.13)$$

and the model response is now described by:

$$\begin{aligned}\hat{x}(ks+h_2) &= \left((I - I_{h_2})\left(\hat{A} + \hat{B}K\right)^{h_2-h_1}U_{h_1} + I_{h_2}\left(A^{h_2} + A^{h_2-h_1}\Xi_{h_1} + \Xi_{h_2-h_1}U_{h_1}\right)\right)x(ks) \\ &= U_{h_2}x(ks).\end{aligned}$$

By following the same analysis for each update instant h_i up to $h_p = s$ we obtain

$$x((k+1)s) = \left(A^s + \sum_{i=1}^{p} A^{h_p-h_i}\Xi_{h_i-h_{i-1}}U_{h_i-1}\right)x(ks). \quad (13.15)$$

Since (13.15) represents a discrete-time LTI system with state matrix given by (13.7), then the networked system is asymptotically stable when the eigenvalues of (13.7) lie inside the unit circle. ♦

Example 13.1 Consider the fifth order unstable system given by:

$$A = \begin{bmatrix} -1.05 & 0.35 & 0.02 & 0.35 & 0.24 \\ 0.35 & 1.1 & 0.1 & 0.035 & 0.035 \\ 0.03 & 0.32 & 0.21 & -0.4 & 0.6 \\ 0.06 & 0.03 & 0.3 & -0.7 & 0.55 \\ 0.035 & 0.03 & 0.6 & 0.3 & 0.2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$$

Its nominal model is given by:

$$\hat{A} = \begin{bmatrix} -1 & 0.35 & 0.02 & 0.35 & 0.24 \\ 0.35 & 1 & 0.05 & 0.035 & 0.035 \\ 0.03 & 0.35 & 0.15 & -0.45 & 0.6 \\ 0.06 & 0.035 & 0.34 & -0.75 & 0.6 \\ 0.035 & 0.035 & 0.5 & 0.3 & 0.4 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$$

If it were possible to send periodic updates of all states, that is, to send all states at the same time instants, then we could use previous results in order to search for the range of stabilizing periods. For instance, we can use the results in Theorem 2.6 in Sect. 2.3 or the equivalent results in Theorem 2.8 in Sect. 2.4.1.

It can be shown that using a single-rate approach we would need to use a period $h < 10$. Figure 13.2 shows the response of the system and the model for the initial conditions shown below and using the update period $h = 9$, the largest stabilizing single-rate period. Note that the entire state of the model is updated at the same time instants. The initial conditions in this simulation are given by:

$$x_0 = [1 \quad 0.5 \quad 0.6 \quad 0.3 \quad -0.5]^T.$$

Now suppose that each state of the system can be measured by a different sensor node using a different update rate. By using the results in this section, it is possible to show the existence of multirate stabilizing update periods. The number of combinations of stabilizing periods is quite large including those that involve lower sampling periods. In many cases it is possible to decrease a few of the individual sampling periods while increasing the rest of them. For instance, if we select the periods $s_1 = 10$, $s_2 = 6$, $s_3 = 15$, $s_4 = 10$, $s_5 = 30$, the eigenvalues of (13.7) are all inside the unit circle, which means that the multirate networked system is asymptotically stable.

Figure 13.3 shows the response of the plant and the model for this particular selection and using the same initial conditions as in Fig. 13.2. It is important to note

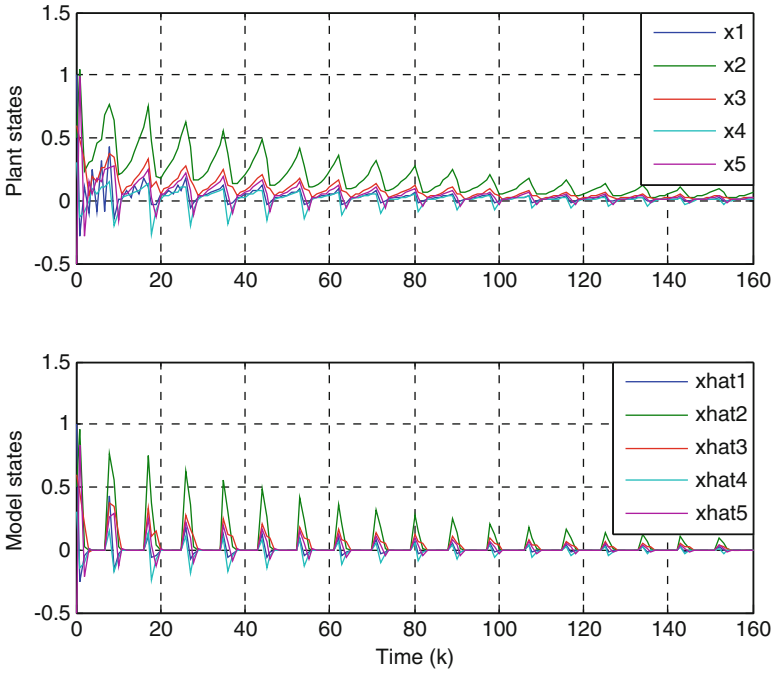

Fig. 13.2 Response of the plant and model for the single-rate system in Example 1 for $h = 9$

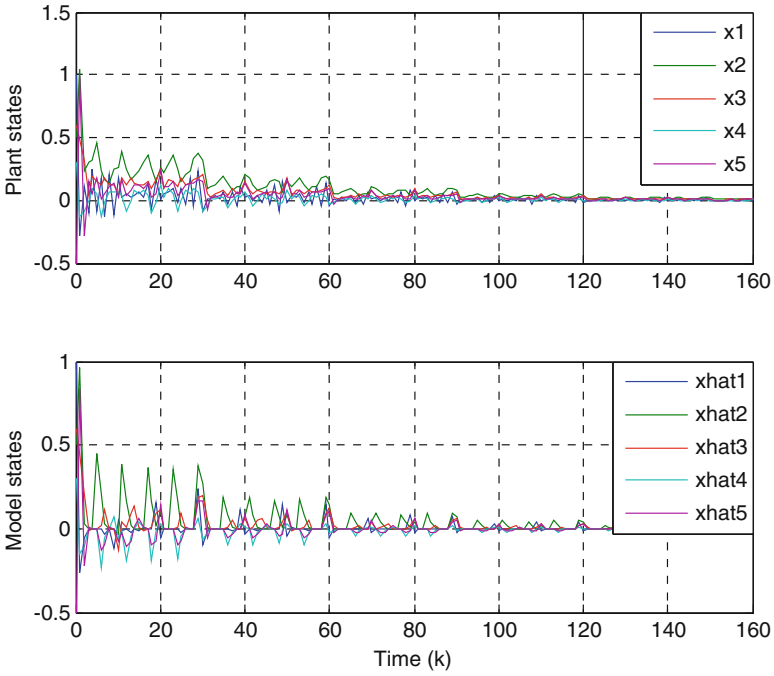

Fig. 13.3 Response of the multirate system for the following choice of update periods: $s_1 = 10,\ s_2 = 6,\ s_3 = 15,\ s_4 = 10,\ s_5 = 30$

that many update period combinations result in a stable system and the one selected is one that provides for significant reduction of network communication compared to the more conservative single-rate implementation in which all sensors would need to use a sampling period $h < 10$.

13.1.2 Updates with Network Constraints

In the previous section we addressed distributed systems and interconnections. We assumed that at certain time instants it is possible for more than one sensor to have access to the network and to transmit a measurement. In this way, flexibility is obtained when choosing different rates for each transmitting node. In many cases this is not a strong assumption, especially if the sampling time of the system is long enough to schedule several nodes to transmit within a sampling interval (here, we can think on T-discretized versions of continuous-time systems, where T is chosen according to specifications related to actuators, sensors, and the communication network as well). However, in other cases, the sampling time may be very small and the network will grant access only to one node at every discrete time k.

In this section we derive similar stabilizing conditions for scheduling transmissions of feedback measurements considering network constraints in the form of restricted access to the communication channel. We consider the architecture shown in Fig. 13.1.

Consider an N-partition of the state of system and model (13.1) according to the number of sensors that are used to obtain measurements. This partition is represented as in (13.2).

Define the augmented state $z(k) = \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix}$. In this problem we consider two time intervals, s and s_1, representing the entire update cycle and the subinterval of empty slots, respectively. The details are as follows: s_1 represents the number of discrete-time instants that other applications may use for communication over the network and are called empty slots since they are not used by the system under analysis. After s_1 time instants the N sensors associated with our system transmit their current measurements sequentially, that is, only one sensor transmits at each one of the following N time slots. Clearly, all sensors use the same update period but with strict scheduling to avoid overlapping. The update period is $s = s_1 + N$.

Theorem 13.2 *The uncertain system with distributed sensors as shown in Fig. 13.1 with model-based control input, with partial model state updates, and with sequential updates is asymptotically stable if and only if the eigenvalues of*

$$V(N)\Gamma^{s_1} \tag{13.16}$$

are within the unit circle of the complex plane, where

$$\Gamma = \begin{bmatrix} A & BK \\ 0 & \hat{A} + \hat{B}K \end{bmatrix}. \tag{13.17}$$

$V(N)$ is of the form:

$$V(N) = \begin{bmatrix} I & 0 \\ I_N & I - I_N \end{bmatrix} \cdots \Gamma \begin{bmatrix} I & 0 \\ I_i & I - I_i \end{bmatrix} \cdots \Gamma \begin{bmatrix} I & 0 \\ I_1 & I - I_1 \end{bmatrix} \tag{13.18}$$

with I_i defined in (13.5).

Proof During the interval $k \in [ks, ks + s_1)$, there are no updates of the state of the model and the augmented system evolves according to:

$$z(k+1) = \Gamma z(k). \tag{13.19}$$

The response of the system at time $ks + s_1$, at the time corresponding to the first update of the current cycle, in terms of $z(ks)$ is given by:

$$z(ks + s_1) = \begin{bmatrix} I & 0 \\ I_1 & I - I_1 \end{bmatrix} \Gamma^{s_1} z(ks). \tag{13.20}$$

The response at the time of the following scheduled update, that is, when the second sensor transmits its measurement update, is given by:

$$z(ks + s_1 + 1) = \begin{bmatrix} I & 0 \\ I_2 & I - I_2 \end{bmatrix} \Gamma \begin{bmatrix} I & 0 \\ I_1 & I - I_1 \end{bmatrix} \Gamma^{s_1} z(ks). \tag{13.21}$$

The response after the entire model state vector was sequentially updated can be represented by:

$$z((k+1)s) = V(N)\Gamma^{s_1} z(ks). \tag{13.22}$$

Since (13.22) represents an LTI system for a given number of nodes, then the networked system is asymptotically stable when the eigenvalues of (13.16) are within the unit circle of the complex plane. ♦

Remark The objective in this section is to find the range of possible update periods s that can be used while maintaining stability. The number of empty slots s_1 can be used by other applications and for transmission of acknowledgement messages, if desired.

Remark In Theorem 13.2, for simplicity, we choose to transmit the measurements pertaining to a given system without empty slots between each other and leaving the

time interval s_1 for other applications to use. This approach can be easily extended to the case in which we can select different update patterns but all states are updated only once during a cycle of duration s. A further extension can combine this approach and the ideas in previous sections in order to schedule several updates of one or more particular sensors during each cycle. However, this represents a lack of flexibility for selection of update rates since the rates have to be multiples of each other in order to prevent the eventual overlap of schedules that would occur otherwise.

Example 13.2 Consider the continuous-time model of a batch reactor.

$$\hat{A} = \begin{bmatrix} 1.3800 & -0.2080 & 6.7150 & -5.6760 \\ -0.5810 & -4.2900 & 0 & 0.6750 \\ 1.0670 & 4.2730 & -6.6540 & 5.8930 \\ 0.0480 & 4.2730 & 1.3430 & -2.1040 \end{bmatrix}, \hat{B} = \begin{bmatrix} 0 & 0 \\ 5.6790 & 0 \\ 1.1360 & -3.1460 \\ 1.1360 & 0 \end{bmatrix}.$$

In this example we consider the case when each state can be measured by a different sensor and we also consider model uncertainties described by $A = \hat{A} + \widetilde{A}$ where the elements in $\widetilde{A}$ take values in the range $\widetilde{a}_{ij} \in [-0.01, 0.01]$. The continuous-time model and plant parameters are discretized using $T = 0.01$ s. The largest value for the update period s is found to be $s = 226$, which corresponds to 2.26 s considering the sampling period T. Figure 13.4 shows the response of the model-based system with sequential updates for two choices of update period that result in a stable networked system. The second option is the limit value and the response is oscillatory and the system is stable with a slow convergence time.

13.2 Two-Channel NCS

We call two- or double-channel MB-NCS an NCS in which not only the path from the sensor to the controller is implemented using a digital network but also the path from the controller to the actuator as well. This configuration offers more flexibility to the designer since there is no need to place the model/controller and actuator/plant in the same node, or directly connect them using a dedicated wire. Depending on the circumstances of the problem, it is many times preferable to use the network to implement this controller to plant connection. The configuration is shown in Fig. 1.1 in Chap. 1. The specific model-based implementation for a system controller pair is shown in Fig. 13.5, where T is the sampling time of the underlying system.

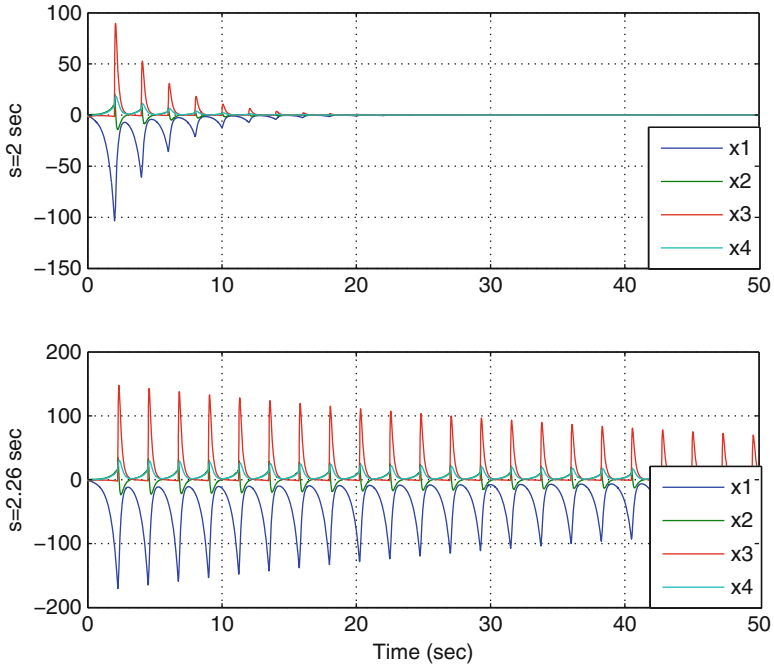

Fig. 13.4 Response of system in Example 13.2 for two different choices of stabilizing update periods: (*top*) $s = 2$ s. (*Bottom*) $s = 2.26$ s

13.2.1 Multirate Approach for Stability

Consider the plant and model dynamics (13.1) implemented as in Fig. 13.5, where the entire state is transmitted from a single sensor node. Here the two switches are closed at different constant rates, giving rise to the constant update intervals n and m. Both n and m are integers and, when multiplied by the underlying sampling period of the plant T, give rise to update periods that are multiples of the underlying sampling period T. As a starting point we wish to find the bounding values of m and n that preserve stability of a discrete-time MB-NCS using instantaneous feedback. Note that in this case, between input updates, the input to the plant is held constant in the actuator and it is equal to the last received value. We do not require any special computation in the actuator it only needs to hold the most current update from the controller.

Assume that $n \geq m$ (low measurement rate, which is typical in many implementations of physical systems). Define $p = n/m$, and p is assumed to be an integer.

Theorem 13.3 *The lifted system corresponding to Fig. 13.5 is asymptotically stable if only if the eigenvalues of*:

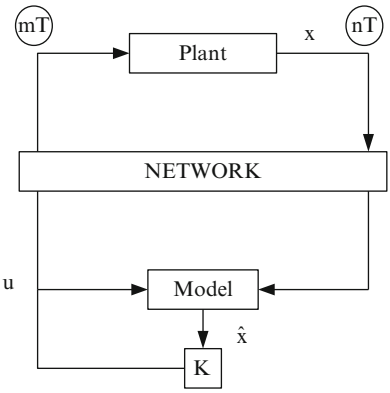

Fig. 13.5 Model-based networked control system with communication network between sensor-controller and controller-actuator

$$A^n + \sum_{i=0}^{p-1} H_i \Gamma^i \tag{13.23}$$

lie inside the unit circle, where

$$H_i = \sum_{j=1}^{m} A^{n-im-j} BK \tag{13.24}$$

$$\Gamma = \left(\hat{A} + \hat{B}K\right)^m. \tag{13.25}$$

Proof Since $p = n/m$ is an integer the period of the equivalent (see Fig. 2.13 in Chap. 2) linear time-varying periodic system is n. Taking a similar approach as in Sect. 13.1 we obtain an LTI system. In order to find the state equations of the lifted system, let us describe the response of the system as a function of the input updates that take place every m clock ticks. Following a similar approach as in Sect. 2.4.1 we obtain:

$$\begin{aligned} x(kn+n) = A^n x(kn) + \left[A^{n-1} + A^{n-2} + \ldots + A^{n-m}\right] Bu(kn) \\ + \left[A^{n-m-1} + A^{n-m-2} + \ldots + A^{n-2m}\right] Bu(kn+m) \\ + \left[A^{n-2m-1} + A^{n-2m-2} + \ldots + A^{n-3m}\right] Bu(kn+2m) + \ldots. \end{aligned} \tag{13.26}$$

The input u is a function of the state of the plant at times kn and a function of the state of the model otherwise. The state of the model between sensor updates can be expressed in terms of the state of the plant at times kn as follows:

$$\begin{aligned} u(kn) &= Kx(kn) \\ u(kn+m) &= K\hat{x}\,(kn+m) = K(\hat{A} + \hat{B}K)^m x(kn) \\ u(kn+2m) &= K\hat{x}\,(kn+2m) = K(\hat{A} + \hat{B}K)^{2m} x(kn) \\ &\vdots \end{aligned} \tag{13.27}$$

Note that the model has access at all times to the input that it generates as it can be deduced from Fig. 13.5. The network connection is between the controller and the plant, and the model is part of the controller. Equation (13.26) becomes:

$$\begin{aligned} x\big((k+1)n\big) &= \big(A^n + H_0\Gamma^0 + H_1\Gamma^1 + H_2\Gamma^2 + ... + H_{p-1}\Gamma^{p-1}\big)x(kn) \\ &= \left(A^n + \sum_{i=0}^{p-1} H_i\Gamma^i\right)x(kn) \end{aligned} \tag{13.28}$$

where H_i and Γ are given in (13.24) and (13.25). Equation (13.28) represents a discrete-time LTI system, therefore stability is achieved when the eigenvalues of (13.23) lie inside the unit circle. ♦

Example 13.3 Consider now the following plant and model implemented as in Fig. 13.5 where the network is also used to connect the controller to the plant. $T = 1$ s.

$$A = \begin{pmatrix} 0.9 & 0.1 \\ 0 & 1.07 \end{pmatrix} \quad B = \begin{pmatrix} 0.01 \\ 0.01 \end{pmatrix}$$

$$\hat{A} = \begin{pmatrix} 0.9117 & 0.1054 \\ 0.0360 & 1.0672 \end{pmatrix} \quad \hat{B} = \begin{pmatrix} 0.0109 \\ 0.0117 \end{pmatrix}$$

$$K = \begin{pmatrix} -2.3294 & -17.6266 \end{pmatrix}.$$

In this case we have two variables, n and m, and, intuitively, we expect a decrease in the necessary value of n as m increases. An appropriate way to proceed here then is as follows: first find the largest value of n for $m = 1$, that is, find the value of h in Theorem 2.8 in Sect. 2.4.1. Then select a value for n less than the value of h that we just found and find the divisors of n. With this information we can plot the eigenvalues of (13.23) as a function of m. For the example in hand, the highest value of h, from Theorem 2.8, is 29, so for illustrative purposes we can fix $n = 24$ (the choice of n is taken considering the existence of a large number of divisors to get $p = n/m$ an integer). For this case the eigenvalues of equation (13.23) are shown in Fig. 13.6. Note that the horizontal axis contains those values of m that result in p being an integer. The response of the plant is shown in Fig. 13.7. The first plot represents the response of the system by choosing parameters n,m according to Fig. 13.6 for which all eigenvalues of (13.23) have magnitude less than one (inside the unit circle). In the second plot the selected parameters result in an unstable system since not all eigenvalues of (13.23) lie inside the unit circle.

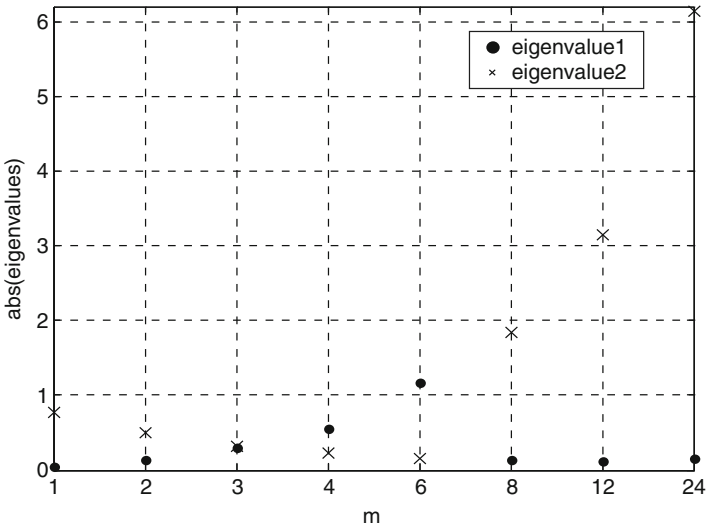

Fig. 13.6 Absolute value of the eigenvalues of equation (13.23) with $n = 24$

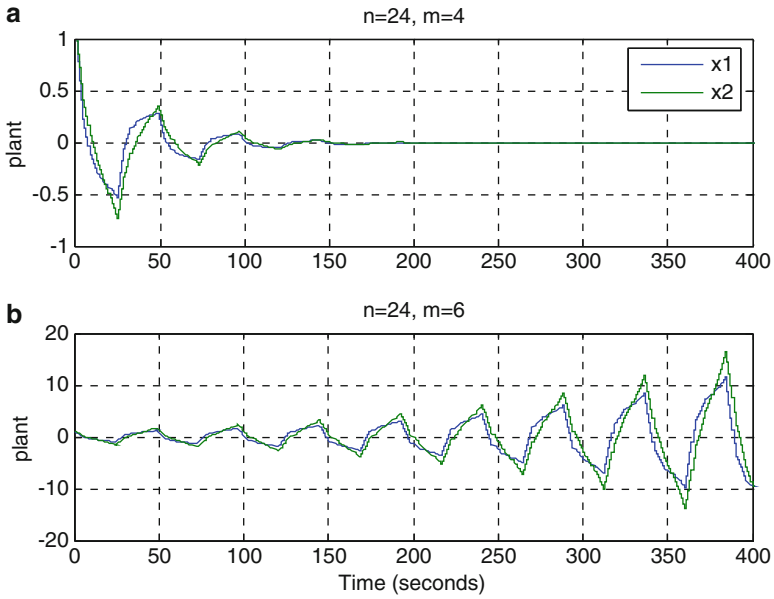

Fig. 13.7 Response of the plant for $n = 24$ and different values of m for which: (**a**) stability is still preserved and (**b**) system becomes unstable

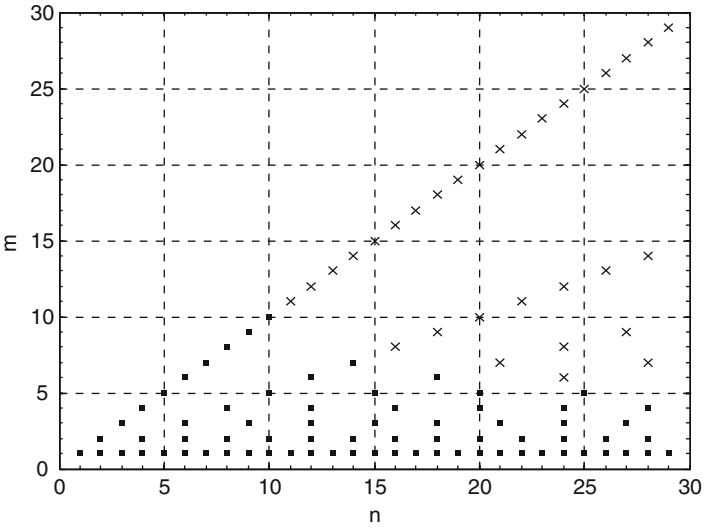

Fig. 13.8 Sets of admissible values of n and m for Example 13.3. (*Filled square*) represent the pairs (n,m) that result in a stable system. (*Multiplication sign*) represent the pairs (n,m) that result in an unstable system, the rest are inadmissible rates

Figure 13.8 shows all admissible pairs (n,m) in the range $n < 30$, i.e., those pairs that result in p having an integer value, it also shows which of those pairs provide stability for the system in Example 13.3.

In the previous analysis it was assumed that the controller update is held constant in the actuator node. Both actuator and sensor node functions are simplified and most of the computations are performed in the controller node. The holding operation is one that, in general, restricts the model-based control properties. By assuming that the actuator is able to receive and synchronize a sequence of input values, we obtain an improved performance in the sense that the controller only needs to receive a measurement update every h time units and compute a sequence of predicted input values and send them all in one larger packet to the actuator.

When the controller node receives a measurement update from the sensor at time $k = t_k$, it predicts the sequence of inputs $u(k)$ for $k = t_k \ldots t_k + h$, that is, the controller computes, instantaneously, the same sequence of control inputs that it would compute for the traditional, one-channel MB-NCS configuration, over the time horizon $[t_k, t_k + h]$ and sends this sequence of input values to the controller. Since the sensor updates are periodic, the model is updated at time $k = t_k + h$ and performs the same computations again but using the new sensor measurements that was just received. The input sequence is sent to the actuator node in a single and larger packet. The actuator synchronizes the sequence by applying each input value at the corresponding time instant. By using this approach the results in Sect. 2.4.1 apply directly to the two-channel MB-NCS configuration where only two updates are necessary every h time units. The update from sensor to controller, then the update from controller to actuator after a sequence of control inputs is calculated.

13.2.2 Two-Channel Networked Systems and Input Tracking

In order to show corresponding results for the input tracking problem when using a two-channel network, we recall the transfer function representation that was used in Sect. 11.3. In that section both the model and the controller were represented using discrete-time input–output models. Additionally, we will also use similar event-based strategies to determine the non-periodic update instants.

With respect to Fig. 13.5, the controller node contains the model $\hat{T}(z)$ and the controller $C(z)$. The controller has access to the reference input signal as well. When the controller node receives a measurement update from the sensor, it predicts the sequence of inputs $u(k)$ and the corresponding sequence of model outputs $\hat{y}(k)$ for $k = t_k \ldots t_k + N$, where t_k is the latest update instant. This input and output prediction is made assuming that the reference signal remains constant during the prediction horizon N. The input sequence is sent to the actuator node in a single and larger packet. The actuator synchronizes the sequence by applying each input value at the corresponding time instant. Similarly, the model output sequence is sent to the sensor node in order to obtain the output error and apply the event policy.

If no error event update has been generated before or at time $t_k + N$, then a time-triggered update takes place and a new output measurement is sent to the controller to compute new sequences in order to repeat the cycle again for $k = t_{k+1} \ldots t_{k+1} + N$ with $t_{k+1} = t_k + N$. Note that the control process can be periodic but not necessarily, since an error event can occur at time $t_k + M$, for $M < N$, which makes $t_{k+1} = t_k + M$.

Finally, the controller also contains the set-point detector and it will send a request for a measurement update to the sensor node if a change in the set-point value occurs. This new event will initialize the prediction cycle no matter if error or time events have not been generated yet.

The results in Sect. 11.3 can be extended to consider this type of implementation and by following some mild assumptions.

Corollary 13.4 *Assume that*:

- *All nodes* (*actuator, controller, and sensor*) *for a given networked system are synchronized.*
- *There exists a time-triggered sensor event equivalent to the duration of the input and model output sequences that are predicted by the controller.*

Then the plant output tracking error is bounded for any bounded reference step input if

(a) *The term* $1 + T(z)C(z)$ *has all its zeros inside the unit circle.*
(b) *The poles of the controller* $C(z)$ *contain the factor* $(z\text{-}1)$.
(c) *The poles of* $\widetilde{T}(z)$ *have magnitude less than one.* ♦

13.3 Notes and References

The contents of Sect. 13.1 were first published in [90]. Results presented in Sect. 13.2 can also be found in [82] and in [89]. The lifting methods used in this chapter are based on common lifting techniques presented in [42]. See also [16, 26, 130] for additional discussion on lifting techniques.

The analysis, development, and controller synthesis for networks of interconnected distributed systems has attracted significant attention within the control systems community. Examples of such systems can be found in a wide variety of applications such as power networks, multi-agent robotic systems, and coordination of autonomous vehicles, large chemical processes comprised several subsystems interacting one with each other, and also in areas that consider economic and/or social systems. In addition, the availability of cheap, fast, embedded sensor, and controller subsystems that are able to communicate via a shared digital network allow for the different subsystems to share their local information with other (possibly all other) subsystems so to achieve a common objective in a more efficient way [34, 64, 233].

Multirate sampling has been used mainly to reduce computation effort in sampled data systems [285]. The work in [285] provides experimental results of a Model Predictive Control (MPC) multirate controller in which a multivariable plant is approximated by three single-output models which are modeled and sampled at different rates. The main purpose in this work is to sample the output representing the slow dynamics of the system at a lower rate in order to reduce the dimension and complexity of the optimization problem at those sampling instants when only the fast dynamics are sampled. Other references concerning the implementation of multirate MPC algorithms are [139, 226].

The multirate implementation generalizes the dual-rate approach frequently used in sampled-data and networked systems [55, 56, 145] where two different rates are used in the control system. These two rates correspond to the actuator (fast rate) and the sensor (slow rate). In the dual-rate approach, it is assumed that the entire output vector is measured and sent through the network at the same time instant.

Multirate systems have also been studied using different approaches. Vadigepalli and Doyle [253] developed a multirate version of the Distributed and Decentralized Estimation and Control (DDEC) for large scale process based on model distribution and internodal communication. The model distribution provides a definition of the states of interest to be estimated locally by each node using measurements that are sampled periodically using different intervals at each node. Communication between nodes is used in order to share information due to the interactions between local subsystems resulting from the model decomposition overlapping states.

Chapter 14
Distributed Control Systems

Many control applications consider the interaction of different subsystems or agents. The increased use of communication networks has made possible the transmission of information among different subsystems in order to apply improved control strategies using information from distant subsystem nodes.

In recent years there has been a strong interest in the analysis, development, and controller synthesis for networks of interconnected systems. The importance and challenges of networks comprised from several to many subsystems or agents have been recognized early by the research community [224]. Examples of such systems can be found in a wide variety of applications such as: power networks, multi-agent robotic systems and coordination of autonomous vehicles, large chemical processes comprised of several subsystems interacting one with each other, and also in areas that consider economic and/or social systems. In addition, the availability of inexpensive, fast, embedded sensor and controller subsystems that are capable to communicate via a shared digital network allows for the different subsystems to share their local information with other (possibly the rest of) subsystems so it can be used to achieve a common objective in a more efficient way [233]. However, digital communication networks have limited bandwidth and not all agents can communicate at a given time instant. It becomes necessary to be able to schedule the broadcast of information by the different nodes in such a way that bandwidth constraints are not violated.

It is in this type of applications where the model-based architectures studied throughout this book provide great advantages in the sense of reducing needed network communication compared to other techniques where models of the systems are not used in the operation of the overall networked system. In the control of distributed coupled systems the improvement in the use of network resources is obtained by implementing models of other subsystems within each local controller.

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8_14,

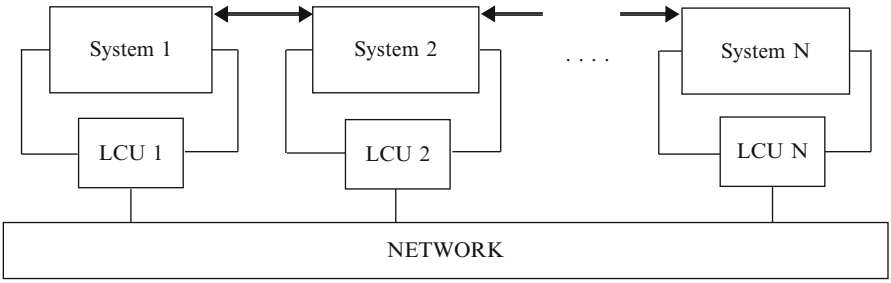

Fig. 14.1 Model-based network interconnected systems

A simple representation of the setup under consideration in this chapter is presented in Fig. 14.1 where the measurements can be transmitted over the network and the arrows represent the physical interconnections or coupling of the subsystems.

The first part of this chapter focuses on distributed systems that transmit information at periodic time intervals. Section 14.1 studies the single-rate implementation where all nodes transmit at the same instants. Section 14.2 investigates multirate implementations where each node uses its own periodic update interval. The second part of the chapter discusses event-triggered control strategies in distributed systems. Section 14.3 offers a centralized triggering condition. Section 14.4 provides a decentralized triggering condition. Notes and references are given in Sect. 14.5.

14.1 Single-Rate Periodic Communication

Throughout this book we have studied different control and communication strategies that have been implemented using the MB-NCS framework. In the present chapter similar strategies are now applied to the stabilization of coupled unstable subsystems.

To simplify the analysis and to clearly explain the model-based implementation we restrict our attention to the case when all subsystems send measurements to update the models in other subsystems using the same sampling period. Next section will generalize this approach to the multirate periodic communication case.

In this section we consider a network of N interconnected agents or subsystems as seen in Fig. 14.1. Each subsystem has a discrete-time state-space representation:

$$
\begin{aligned}
x_1(k+1) &= A_1 x_1(k) + B_1 u_1(k) + \sum_{j=2}^{N} A_{1j} x_j(k) \\
x_2(k+1) &= A_2 x_2(k) + B_2 u_2(k) + \sum_{j=1, j\neq 2}^{N} A_{2j} x_j(k) \\
&\vdots \\
x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + \sum_{j=1, j\neq i}^{N} A_{ij} x_j(k) \\
&\vdots \\
x_N(k+1) &= A_N x_N(k) + B_N u_N(k) + \sum_{j=1}^{N-1} A_{Nj} x_j(k).
\end{aligned}
\tag{14.1}
$$

In this framework each Local Control Unit (LCU) contains copies of the models of *all subsystems* including the model corresponding to its own local dynamics in order to generate estimates of the states of all subsystems in the network. Each LCU does not necessarily need models of all subsystems but only of those agents whose states need to be estimated by that particular local controller. The states that need to be estimated by a particular controller, say LCUi, fall in one of two categories: (a) the states that directly affect the *model dynamics* of system i, and (b) the states that directly or indirectly affect the *model dynamics* of the states in (a), the first category.

The model of each subsystem is represented by:

$$
\hat{x}_i(k+1) = \hat{A}_i \hat{x}_i(k) + \hat{B}_i \hat{u}_i(k) + \sum_{j=1, j\neq i}^{N} \hat{A}_{ij} \hat{x}_j(k) \tag{14.2}
$$

for each $i \in \mathrm{N}$, N denotes the set $\{1, 2, \ldots N\}$ of N integers where $x_i, \hat{x}_i \in \mathbb{R}^{ni}$ represent respectively the real state of the *ith* unit and the state of the corresponding model, $u_i, \hat{u}_i \in \mathbb{R}^{mi}$ represent the local input for subsystem or plant i and the input for model i, respectively. The matrices Ai, Aij, Bi, $\hat{A}i, \hat{A}ij, \hat{B}i$ are of appropriate dimensions. Note that the subsystems could have different dynamics and different dimensions, the dimensions mi and ni could be all different in general. Note also that each LCUi has access to its local state x_i at all times which is used to compute the local subsystem control input:

$$
u_i(k) = K_i x_i(k) + \sum_{j=1, j\neq i}^{N} K_{ij} \hat{x}_j(k). \tag{14.3}
$$

An important point in this model-based distributed approach is to guarantee that all subsystems have identical values of the estimated states, i.e., of the model states (14.2). For instance, if a subset of the LCUs, $\mathrm{N}_1 \subseteq \mathrm{N}$, needs to generate estimates of state x_i, i.e., to generate $\hat{x}_i$, we want to ensure that $\hat{x}_i^j = \hat{x}_i^l$ for all $j, l \in \mathrm{N}_1$, where $\hat{x}_i^j$ is

the estimate of x_i at LCUj. From (14.2) we can see that this task can be accomplished by using the same model parameters at each LCU and also by feeding each model the same input $\hat{u}_i$, and by repeating this procedure for all models in each control unit. The control inputs (14.3) are unsuitable for this task since they depend on the states of the real subsystems. In order to obtain the same estimated states in all necessary control units we define the model control inputs as

$$\hat{u}_i(k) = K_i \hat{x}_i(k) + \sum_{j=1, j\neq i}^{N} K_{ij} \hat{x}_j(k). \tag{14.4}$$

These control inputs are applied to all models in all LCUs whereas (14.3) is applied to each local subsystem. It is clear now that although LCUi computes an estimate $\hat{x}_i$ of x_i, this estimated state is not used to control subsystem i since we have the real state available. At LCUi we use $\hat{x}_i$ as input for the models ensuring that the same model equations with the same model control inputs are implemented at all LCUs.

Define the augmented plant and model-state vectors:

$$\begin{aligned} x &= \left[x1^T \;\; x2^T \;\; \ldots \;\; xn^T\right]^T \\ \hat{x} &= \left[\hat{x}\,1^T \;\; \hat{x}\,2^T \;\; \ldots \;\; \hat{x}\,n^T\right]^T. \end{aligned} \tag{14.5}$$

The equations in (14.5) represent the states of all systems and all corresponding models. The dynamics of the overall system and model can be represented by:

$$x(k+1) = Ax(k) + Bu(k) \tag{14.6}$$

$$\hat{x}\,(k+1) = \hat{A}\,\hat{x}\,(k) + \hat{B}\,\hat{u}\,(k). \tag{14.7}$$

The form of the matrices $A, \hat{A} \in \mathbb{R}^{n\times n}$ and $B, \hat{B} \in \mathbb{R}^{n\times m}$ where $n = \sum_{i=1}^{N} ni$ and $m = \sum_{i=1}^{N} mi$ are as follows:

$$\begin{aligned} A &= \begin{bmatrix} A_1 & A_{12} & \ldots & A_{1n} \\ A_{21} & A_2 & \ldots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{n1} & A_{n2} & \ldots & A_n \end{bmatrix} \quad B = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & B_n \end{bmatrix} \\ \hat{A} &= \begin{bmatrix} \hat{A}_1 & \hat{A}_{12} & \ldots & \hat{A}_{1n} \\ \hat{A}_{21} & \hat{A}_2 & \ldots & \hat{A}_{2n} \\ \vdots & & \ddots & \vdots \\ \hat{A}_{n1} & \hat{A}_{n2} & \ldots & \hat{A}_n \end{bmatrix} \quad \hat{B} = \begin{bmatrix} \hat{B}_1 & 0 & \cdots & 0 \\ 0 & \hat{B}_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & \hat{B}_n \end{bmatrix}. \end{aligned} \tag{14.8}$$

We can describe the dynamics of the overall system as given in the next proposition.

Proposition 14.1 *Assume* $\left(,\hat{A},\hat{B}\right)$ *is stabilizable. The dynamics of the overall system are represented by:*

$$x(k+1) = \left(A + BK_{diag}\right)x + BK_{off}\hat{x} \tag{14.9}$$

where $K_{off} = K - K_{diag}$. $K_{diag} = diag(K_i)$ *is a matrix containing the controller gains* K_i *as main diagonal sub-matrices. The controller* K *is a stabilizing controller for the overall model dynamics,* i.e., $\hat{A} + \hat{B}K$ *is Hurwitz.*

Proof We rewrite (14.6) in the next form

$$x(k+1) = Ax + Bu = Ax + B\left(K_{diag}x(k) + \left(K - K_{diag}\right)\hat{x}(k)\right) \tag{14.10}$$

where u is the augmented vector containing each agent local subsystem control inputs

$$u = \left[u_1{}^T \;\; u_2{}^T \;\; \ldots \;\; u_n{}^T\right]^T. \tag{14.11}$$

Equation (14.10) can be simple rewritten as (14.9). ◆

Theorem 14.2 *System* (14.6) *with control input* $u(k) = K_{diag}x(k) + K_{off}\hat{x}(k)$ *and single-rate periodic updates of the states of the models is asymptotically stable if only if the eigenvalues of*

$$\left(A + BK_{diag}\right)^h + \sum_{j=0}^{h-1}\left(A + BK_{diag}\right)^{h-1-j}BK_{off}\left(\hat{A} + \hat{B}K\right)^j \tag{14.12}$$

lie strictly inside the unit circle, where h *is the sampling period.*

Proof By Proposition 14.1 we have that the state equation containing all subsystems is given by (14.9). Equation (14.9) is of the form (2.64) with state matrix $A + BK_{diag}$, controller gain K_{off}, and single-rate updates. Then Theorem (2.8) in Chap. 2 can be applied directly using these modified matrices. Similarly, the lifted system (Chap. 2)

$$x((k+1)h) = \left(\left(A + BK_{diag}\right)^h + \sum_{j=0}^{h-1}\left(A + BK_{diag}\right)^{h-1-j}BK_{off}\left(\hat{A} + \hat{B}K\right)^j\right)x(kh)$$

is an LTI system and it is asymptotically stable if and only if the eigenvalues of (14.12) lie strictly inside the unit circle. ◆

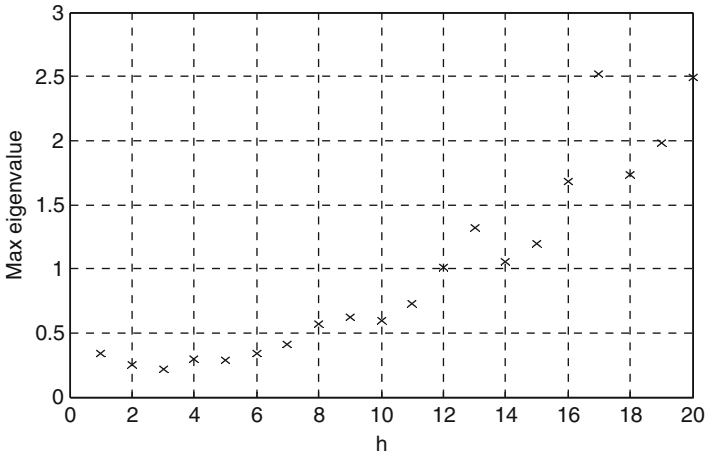

Fig. 14.2 Maximum eigenvalue of (14.12) for different values of h

Example 14.1 We consider a network of $N = 6$ subsystems represented by (14.1) all with different dynamics. The dimensions of the systems vary from 1 to 3 as well. The models for all different parameters represent an uncertainty as follows: 12 % alteration in the A_i matrices, 10 % in A_{ij}, and 6 % in B_i. Some of the systems are unstable and every agent is coupled to all other agents including those with different dimensions by corresponding coupling matrices A_{ij}. The eigenvalues of (14.12) are shown in Fig. 14.2 for different values of h, the single-rate update rate. It can be seen that for $h \leq 11$ the overall networked system is asymptotically stable. Figures 14.3 and 14.4 show the response of all subsystems for the largest stabilizing rate $h = 11$ and for initial conditions:

$$x_1(0) = 1, \quad x_2(0) = 0.5 \quad x_3(0) = \begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix}, \quad x_4(0) = \begin{bmatrix} -0.5 \\ -0.4 \end{bmatrix} \quad x_5(0) = \begin{bmatrix} 1 \\ 0.5 \\ 0.6 \end{bmatrix}, \quad x_6(0) = \begin{bmatrix} 0.3 \\ -0.5 \\ -0.4 \end{bmatrix}$$

14.2 Multirate Periodic Communication

In this section we consider the same problem and approach as in the previous section, but we do not restrict all updates to take place at the same time instants. Instead, we allow each subsystem to use a different sampling period. This approach results in a better usage of network resources. By allowing the sensors to transmit their measurements using different sampling periods we reduce the probability of

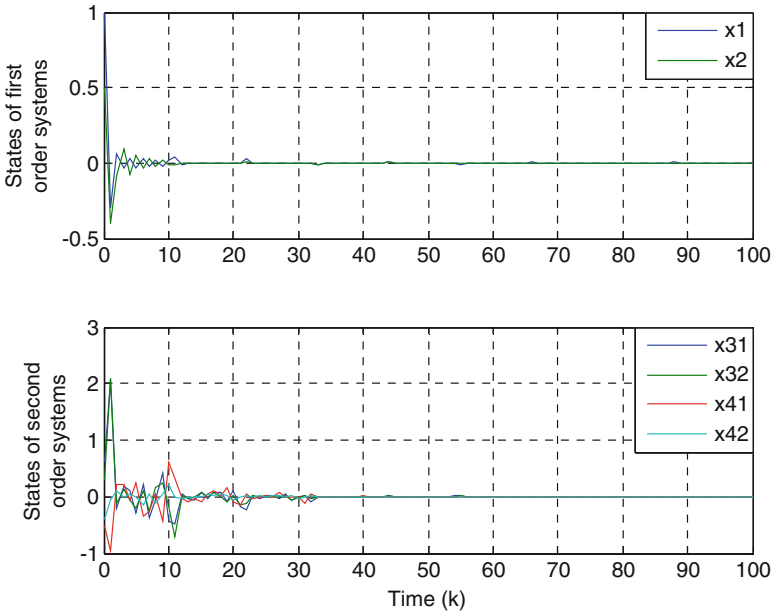

Fig. 14.3 Response of the first order systems (*top*) and the second order systems (*bottom*) for $h = 11$

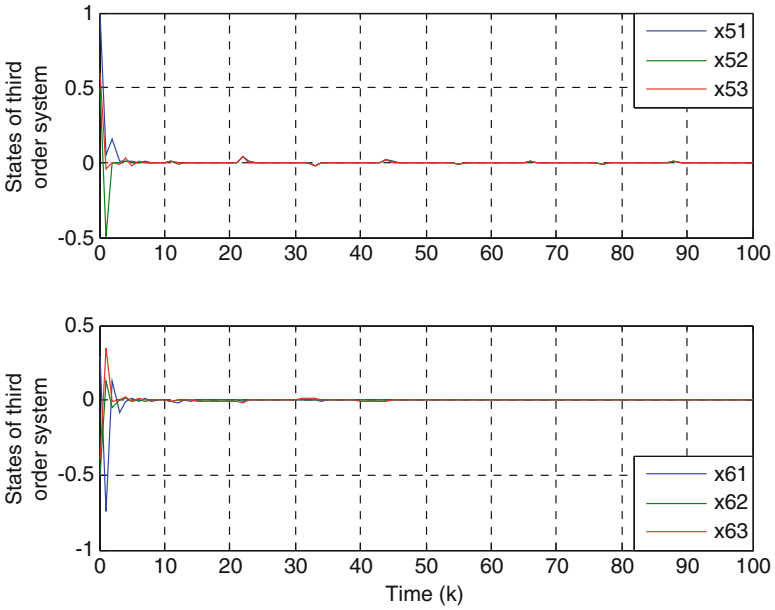

Fig. 14.4 Response of the third order system 1 (*top*) and third order system 2 (*bottom*) for $h = 11$

packet collisions and the size of networked induced delays compared to the case when all of the sensors need to sample and transmit at the same instants. Additionally, we will show that in many cases a further reduction in network communication can be obtained by using different updating rates for each sensor compared to the single-rate case shown in previous section.

Consider a set of N subsystems or agents as represented in Fig. 14.1. Each subsystem can be represented by:

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1, j\neq i}^{N} A_{ij} x_j(k). \tag{14.13}$$

Consider also the models in (14.2),

$$\hat{x}_i(k+1) = \hat{A}_i \hat{x}_i(k) + \hat{B}_i \hat{u}_i(k) + \sum_{j=1, j\neq i}^{N} \hat{A}_{ij} \hat{x}_j(k)$$

for each $i \in$ N, N denotes the set $\{1, 2, \ldots N\}$ of N integers where $x_i, \hat{x}_i \in \mathbb{R}^{ni}$ represent respectively the real state of the *ith* unit and the state of the corresponding model, $u_i, \hat{u}_i \in \mathbb{R}^{mi}$ represent the local input for subsystem i and for model i, respectively. The matrices Ai, Aij, Bi, $\hat{A}i, \hat{A}ij, \hat{B}i$ are of appropriate dimensions. Note that the subsystems could have different dynamics and different dimensions, the dimensions *mi* and *ni* could all be different in general. Note also that each LCU*i* has access to its local state x_i at all times which is used to compute the local subsystem control input. The system control inputs u_i and the model control inputs $\hat{u}_i$ are given by (14.3) and (14.4), respectively. Likewise, the overall networked system can be represented by (14.6) and (14.7), where $A, \hat{A} \in \mathbb{R}^{n\times n}$ and $B, \hat{B} \in \mathbb{R}^{n\times m}$ with $n = \sum_{i=1}^{N} ni$ and $m = \sum_{i=1}^{N} mi$. The system and model matrices are given by (14.8).

Let s_i represent the sampling period that sensor i uses to send measurements in order to update the states of all models corresponding to subsystem i, that is, a measurement x_i to update all models $\hat{x}_i$. Let s represent the minimum common multiple of all s_i.

In order to obtain the response of the overall networked system with multirate samplings and using the model-based control inputs we define all the update instants within a period s by arranging the periods s_i and its multiples up to before s in increasing order as follows:

$$\begin{aligned} &s_1, 2s_1, \ldots, (r_1 - 1)s_1 \\ &s_2, 2s_2, \ldots, (r_2 - 1)s_2 \\ &\vdots \\ &s_N, 2s_N, \ldots, (r_N - 1)s_N \end{aligned}$$

where $r_i = \frac{s}{s_i}$ for $i = 1 \ldots N$ represents the relative sampling rate compared to the sampling rate given by the period s. Let h_i for $i = 1 \ldots p\text{-}1$ represent the update instants in increasing order, where p is the total number of update instants within a period s including the update at time s. Note that at any given instant one or more sets of states $\hat{x}_i$ can be updated. This procedure can be better shown trough a simple example.

Consider a set of three subsystems x_1, x_2, x_3 with corresponding periods $s_1 = 2$, $s_2 = 3$, $s_3 = 4$. Then we proceed to define all update instants within a period $s = 12$, as follows:

$$\begin{aligned} h_1 &= 2 = s_1 \\ h_2 &= 3 = s_2 \\ h_3 &= 4 = 2s_1 = s_3 \\ h_4 &= 6 = 3s_1 = 2s_2 \\ h_5 &= 8 = 4s_1 = 2s_3 \\ h_6 &= 9 = 3s_2 \\ h_7 &= 10 = 5s_1 \end{aligned}$$

Note that at the time instants h_3, h_4, h_5 we have more than one system broadcasting information for this example.

Let us define, in general, the update matrix corresponding to subsystem i:

$$I_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_{ni} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

that is, the ith update matrix $I_i \in \mathbb{R}^{n \times n}$ contains an identity matrix at the position corresponding to x_i and zeros elsewhere.

Define

$$I_{h_i} = I_i + I_j + I_k \ldots$$

The matrices I_{h_i} represent all updates that take place at time instant h_i.

Theorem 14.3 *The overall networked system* (14.6) *with model-based control inputs and with multirate model updates is asymptotically stable for given inter-sample periods s_i if and only if the eigenvalues of*

$$\left(A + BK_{diag}\right)^s + \sum_{i=1}^{p} \left(A + BK_{diag}\right)^{h_p - h_i} \Xi_{h_i - h_{i-1}} U_{h_i - 1} \tag{14.14}$$

lie inside the unit circle, where

$$\Xi_{h_i-h_{i-1}} = \sum_{j=0}^{h_i-h_{i-1}-1} \left(A+BK_{diag}\right)^{h_i-h_{i-1}-1-j} BK_{off}\left(\hat{A}+\hat{B}K\right)^j$$

$$U_{h_i} = (I-I_{h_i})\left(\hat{A}+\hat{B}K\right)^{h_i-h_{i-1}} U_{h_i-1} + I_{h_i}\Big(\left(A+BK_{diag}\right)^{h_i} + \sum_{q=1}^{i}\left(A+BK_{diag}\right)^{h_i-h_q}\Xi_{h_q-h_{q-1}}U_{h_q-1}\Big)$$

with $h_0 = 0,\ \ U_{h_0} = I,\ \ h_p = s.$

Proof Let us consider the beginning of a period s. At this time instant all sensors send measurements and we have that $\hat{x}(ks) = x(ks)$. At the time of the first update after time ks, that is, at time $ks+h_1$ we have:

$$\begin{aligned} x(ks+h_1) &= \Big(\left(A+BK_{diag}\right)^{h_1} + \sum_{j=0}^{h_1-1}\left(A+BK_{diag}\right)^{h_1-1-j} BK_{off}\left(\hat{A}+\hat{B}K\right)^j\Big)x(ks) \\ &= \Big(\left(A+BK_{diag}\right)^{h_1} + \Xi_{h_1}\Big)x\big(ks\big) \end{aligned}$$

and the model state after the update has taken place is given by:

$$\begin{aligned} \hat{x}(ks+h_1) &= \Big(\left(I-I_{h_1}\right)\left(\hat{A}+\hat{B}K\right)^{h_1} + I_{h_1}\Big(\left(A+BK_{diag}\right)^{h_1} + \Xi_{h_1}\Big)\Big)x\big(ks\big) \\ &= U_{h_1}x(k). \end{aligned}$$

Following a similar analysis we can obtain the response of both the system and the model at time $ks+h_2$ as a function of $x(ks+h_1)$ and $\hat{x}(ks+h_1)$:

$$\begin{aligned} x(ks+h_2) &= \left(A+BK_{diag}\right)^{h_2-h_1} x(ks+h_1) \\ &\quad + \sum_{j=0}^{h_2-h_1-1}\left(A+BK_{diag}\right)^{h_2-h_1-1-j} BK_{off}\left(\hat{A}+\hat{B}K\right)^j \hat{x}(ks+h_1) \\ \hat{x}(ks+h_2) &= (I-I_{h_2})\left(\hat{A}+\hat{B}K\right)^{h_2-h_1}\hat{x}(ks+h_1) + I_{h_2}x(ks+h_2) \end{aligned}$$

but, since both $x(ks+h_1)$ and $\hat{x}(ks+h_1)$ can be expressed in terms of $x(ks)$, we get:

$$\begin{aligned} x(ks+h_2) &= \Big(\left(A+BK_{diag}\right)^{h_2} + \left(A+BK_{diag}\right)^{h_2-h_1}\Xi_{h_1} + \Xi_{h_2-h_1}U_{h_1}\Big)x(ks) \\ \hat{x}(ks+h_2) &= \Big(\left(I-I_{h_2}\right)\left(\hat{A}+\hat{B}K\right)^{h_2-h_1}U_{h_1} \\ &\quad + I_{h_2}\Big(\left(A+BK_{diag}\right)^{h_2} + \left(A+BK_{diag}\right)^{h_2-h_1}\Xi_{h_1} + \Xi_{h_2-h_1}U_{h_1}\Big)\Big)x\big(ks\big) \\ &= U_{h_2}x(ks). \end{aligned}$$

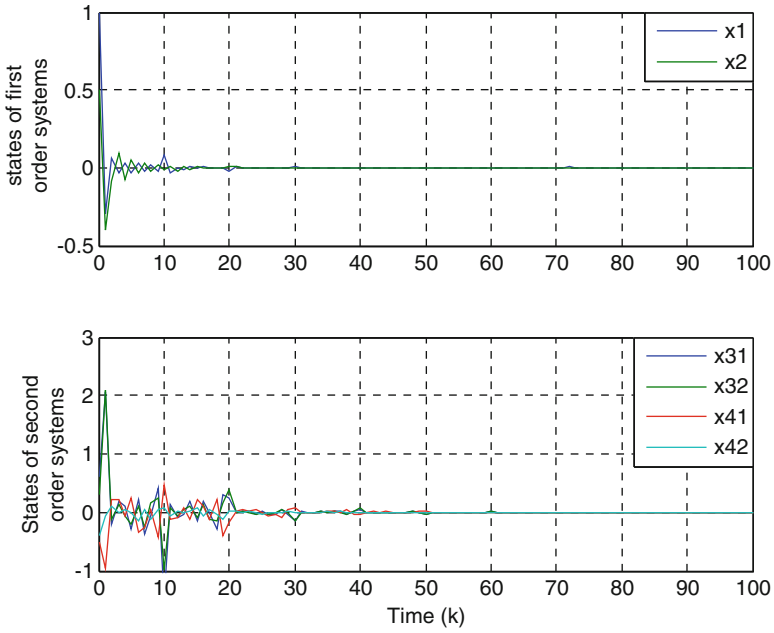

Fig. 14.5 Response of the first order systems (*top*) and the second order systems (*bottom*) for $s_1 = 60,\ s_2 = 15,\ s_3 = 10,\ s_4 = 10$

By following the same analysis for each updating instant h_i up to $h_p = s$ we obtain the following

$$x((k+1)s) = \left(\left(A + BK_{diag}\right)^s + \sum_{i=1}^{p} \left(A + BK_{diag}\right)^{h_p - h_i} \Xi_{h_i - h_{i-1}} U_{h_i - 1} \right) x(ks) \tag{14.15}$$

Since (14.15) represents an LTI system then the overall networked system is asymptotically stable when the eigenvalues of (14.14) lie inside the unit circle. ◆

Example 14.2 We use the same set of systems and models as in Example 14.1 with the same initial conditions. By allowing the agents to broadcast their states using different rates we are able to asymptotically stabilize the overall system and to further reduce the communication between agents. There exist many combinations of periods s_i that result in a stable system. The next selection that significantly increases the sampling periods and reduces network traffic compared to $h = 11$ in Example 14.1 was used in the simulation shown in Figs. 14.5 and 14.6,

$$s_1 = 60,\ \ s_2 = 15,\ \ s_3 = 10,\ \ s_4 = 10,\ \ s_5 = 30,\ \ s_6 = 12.$$

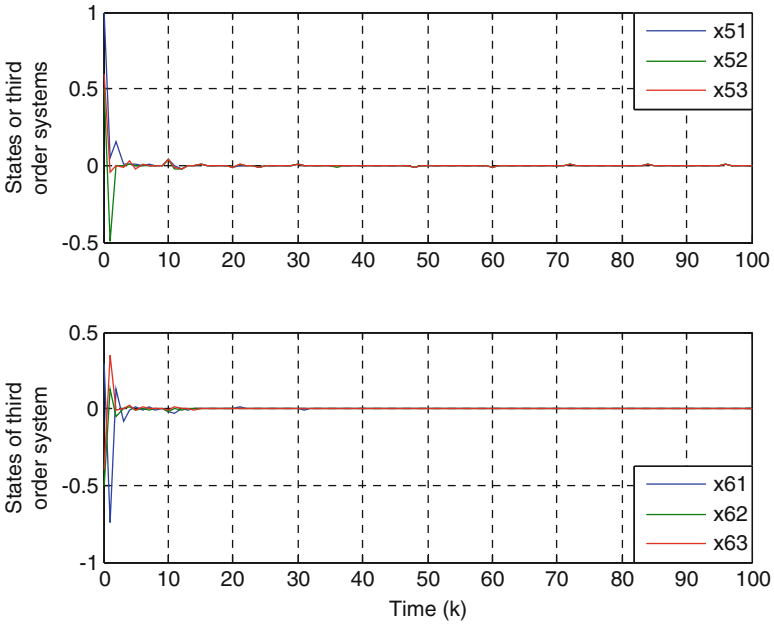

Fig. 14.6 Response of the third order system 1 (*top*) and third order system 2 (*bottom*) for $s_5 = 30,\ s_6 = 12$

This selection increases the update time intervals of subsystems 1,2,5, and 6 by 49, 6, 19, and 3 sampling instants of the plant, respectively, while reducing the update time intervals of subsystems 3 and 4 by only one sampling interval of the plant.

14.3 Centralized Model-Based Event-Triggered Control

A different way to schedule model updates using the model-based approach was shown in Chap. 6. The Model-Based Event-Triggered (MB-ET) approach, in contrast to the periodic implementation, adjusts the broadcasting intervals according to the current state error. In this section we extend the MB-ET strategy to consider multiple coupled subsystems. Each LCU is in charge of evaluating its own local state error in order to decide the time instants in which the real state needs to be broadcasted to the other agents so all LCUs that implement the model corresponding to that particular control unit that has sent information update the corresponding model state.

In this section we consider a network of N interconnected continuous-time subsystems or agents. Each subsystem can be described by a state-space representation as follows:

$$\dot{x}_i = A_i x_i + B_i u_i + \sum_{j=1, j\neq i}^{N} A_{ij} x_j \tag{14.16}$$

for each $i \in \mathrm{N}$, where N is the set $\{1, 2, \ldots N\}$ of N integers; $x_i \in \mathbb{R}^{ni}$ represents the state of the *ith* subsystem, $u_i \in \mathbb{R}^{mi}$ represents the local input for subsystem i. $A_i \in \mathbb{R}^{ni \times ni}, A_{ij} \in \mathbb{R}^{ni \times nj}, B_i \in \mathbb{R}^{ni \times mi}$ represent respectively the state, coupling, and input matrices for the *ith* subsystem.

In this framework each LCU contains copies of the models of all subsystems including the model corresponding to its own local dynamics in order to generate estimates of the states of all subsystems in the network. Each LCU does not necessarily need models of all subsystems but only of those agents whose states need to be estimated by that particular local controller as it was explained earlier in this chapter.

The model of each subsystem is

$$\dot{\hat{x}}_i = \hat{A}_i \hat{x}_i + \hat{B}_i \hat{u}_i + \sum_{j=1, j\neq i}^{N} \hat{A}_{ij} \hat{x}_j \tag{14.17}$$

where $\hat{x}_i \in \mathbb{R}^{ni}$ represents the state of the *ith* model, $\hat{u}_i \in \mathbb{R}^{mi}$ represents the local input for model i. The matrices $\hat{A}_i \in \mathbb{R}^{ni \times ni}, \hat{A}_{ij} \in \mathbb{R}^{ni \times nj}, \hat{B}_i \in \mathbb{R}^{ni \times mi}$ represent the nominal parameters of the dynamics of the *ith* subsystem. Note that the subsystems could have different dynamics and different dimensions, the dimensions *mi* and *ni* could be all different in general. Note also that each LCUi has access to its local state x_i at all times which is used to compute the local subsystem control input:

$$u_i = K_i x_i + \sum_{j=1, j\neq i}^{N} K_{ij} \hat{x}_j \tag{14.18}$$

and the local model-state error which is defined as:

$$e_i = \hat{x}_i - x_i \tag{14.19}$$

where K_i and K_{ij} are the stabilizing control gains to be designed.

By measuring its local error, each LCUi is able to decide the appropriate times at which it should broadcast the current measured state x_i to all other units so all LCU's can update the state of their local models $\hat{x}_i$ corresponding to x_i. At the same time the LCU that broadcasted its state needs to update its own local model corresponding to x_i and the error e_i is set to 0. We assume that the communication delay is negligible and the initial conditions of the plant are nonzero but finite.

Note that when the LCUs update the state of the model corresponding to x_i then the error e_i is set to 0 and therefore it is less than the positive threshold that is used to determine the update instants.

Similar to previous sections in this chapter, we want to guarantee that the models in all necessary LCUs of any given state x_i produce the same estimated state. For the MB-ET approach used in this section enforcing this step also guarantees that the local error (14.19) is the same for all nodes as well.

Using this framework, we can see from (14.18) that the input u_i for the agent i is not an appropriate input for the corresponding model. The input for the local subsystem is a function of the real state which is not always available to the other agents. In order to make sure that every agent in the network computes the same estimate of the states of all agents we need to use the same parameters for the model equations (14.17) and we also need to implement control inputs for the models that can be executed at every LCU. We define the model inputs

$$\hat{u}_i = K_i \hat{x}_i + \sum_{j=1, j\neq i}^{N} K_{ij} \hat{x}_j. \tag{14.20}$$

These control inputs are applied to all models in all LCUs whereas (14.18) is applied to each local subsystem. It is clear now that although LCUi computes an estimate $\hat{x}_i$ of x_i, this estimated state is not used to control subsystem i since we have the real state available. At LCUi we use $\hat{x}_i$ as input for the models ensuring that the same model equations with the same model control inputs are implemented at all LCUs.

The first approach to compute the stabilizing controllers and thresholds is presented in this section and it is based on the dynamics of the overall system and model. The time instants at which each agent needs to send its information to the network can be computed locally by each LCU.

Let us introduce the augmented vectors:

$$\begin{aligned} x &= \begin{bmatrix} x_1{}^T & x_2{}^T & \dots & x_n{}^T \end{bmatrix}^T \\ \hat{x} &= \begin{bmatrix} \hat{x}_1{}^T & \hat{x}_2{}^T & \dots & \hat{x}_n{}^T \end{bmatrix}^T \\ e &= \begin{bmatrix} e_1{}^T & e_2{}^T & \dots & e_n{}^T \end{bmatrix}^T. \end{aligned} \tag{14.21}$$

The dynamics of the overall system and model can be represented by:

$$\dot{x} = Ax + Bu \tag{14.22}$$

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}\hat{u}. \tag{14.23}$$

The form of the matrices $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$, where $n = \sum_{i=1}^{N} ni$ and $m = \sum_{i=1}^{N} mi$, are as follows:

$$A = \begin{bmatrix} A_1 & A_{12} & \dots & A_{1n} \\ A_{21} & A_2 & \dots & A_{2n} \\ \vdots & & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_n \end{bmatrix}$$
$$B = \begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & B_n \end{bmatrix} \tag{14.24}$$

and similarly for $\hat{A}$ and $\hat{B}$. We describe the dynamics of the overall system in the next proposition.

Proposition 14.4 *Assume* $\left(,\hat{A},\hat{B}\right)$ *is stabilizable. The dynamics of the overall system can be represented by:*

$$\dot{x} = (A + BK)x + BK_{off}e \tag{14.25}$$

where $K_{off} = K - K_{diag}$. $K_{diag} = diag(K_i)$ *is a matrix containing the controller gains* K_i *as main diagonal sub-matrices. The controller* K *is a stabilizing controller for the overall model dynamics,* i.e., *the matrix* $\hat{A} + \hat{B}K$ *is Hurwitz.*

Proof We rewrite (14.22) in the next form

$$\dot{x} = Ax + Bu = Ax + B\left(K_{diag}x + \left(K - K_{diag}\right)\hat{x}\right)$$

where u is the augmented vector containing each agent local subsystem control inputs

$$u = \left[u1^T \;\; u2^T \;\; \dots \;\; un^T\right]^T. \tag{14.26}$$

From (14.21) we have that $e = \hat{x} - x$ and we can write

$$\begin{aligned} \dot{x} &= Ax + BK(x + e) - BK_{diag}e \\ &= (A + BK)x + BK_{off}e. \end{aligned}$$

◆

In order to asymptotically stabilize all agents from their finite initial conditions we implement a similar strategy as in Chap. 6. The main idea is to reduce the threshold value as we approach the equilibrium point of the system. This can be achieved by comparing the norm of the error to a function of the norm of the state. The norm used in the next results is the Euclidean norm.

Consider again the plant and model described by (14.22) and (14.23) and by using the control input $u = K_{diag}x + K_{off}\hat{x}$ we obtain expression (14.25) for

the plant with K rendering $\hat{A} + \hat{B}K$ Hurwitz, i.e., the closed-loop model is globally asymptotically stable. Then we can always find a P which is symmetric positive definite and is the solution of the closed-loop model Lyapunov equation:

$$\left(\hat{A} + \hat{B}K\right)^T P + P\left(\hat{A} + \hat{B}K\right) = -Q \tag{14.27}$$

where Q is a symmetric positive definite matrix.

Let us first analyze the case when $\hat{B} = B$ for simplicity, and define the uncertainty $\widetilde{A} = A - \hat{A}$. Also assume that the next bound on the uncertainty $\left|\widetilde{A}^T P + P\widetilde{A}\right| \leq \Delta < q$ holds where $q = \underline{\sigma}(Q)$, the smallest singular value of Q in the Lyapunov equation (14.27) and Δ is a bound on the norm of the uncertainty. This bound can be seen as a measure of how close A and $\hat{A}$ should be.

The next theorem provides conditions on the error and its threshold value so the networked system is asymptotic stable. The error threshold is defined as a function of the norm of the state and Δ. Additionally, the error events can be computed locally, that is, once the thresholds have been designed each agent can decide when to broadcast its current measurement to the rest of the agents based only on the measurements of its own state and its own error.

Theorem 14.5 *System* (14.22) *with* $u = K_{diag}x + K_{off}\hat{x}$ *and feedback based on error events generated when the relation:*

$$|e_i| > \alpha |x_i| \tag{14.28}$$

is first satisfied, is globally asymptotically stable, where $\alpha = \sigma(q - \Delta)/b$, $b = 2\left|P\hat{B}K_{off}\right|$ *and* $0 < \sigma < 1$.

Proof In order to prove this theorem we will set a bound on the derivative of $V = x^T P x$ along the trajectories of the system (14.25) which is equal to (14.22) when the input $u = K_{diag}x + K_{off}\hat{x}$ has already been substituted and expressed in terms of the state error, then we can easily show that this bound can be appropriately tuned by the choice of the threshold on the error.

$$\begin{aligned}
\dot{V} &= x^T\left[(A + BK)^T P + P(A + BK)\right]x + e^T K_{off}{}^T B^T P x + x^T P B K_{off} e \\
&= x^T\left[\left(\hat{A} + \widetilde{A} + \hat{B}K\right)^T P + P\left(\hat{A} + \widetilde{A} + \hat{B}K\right)\right]x + 2x^T P\hat{B}K_{off}e \\
&= -x^T Q x + x^T\left(\widetilde{A}^T P + P\widetilde{A}\right)x + 2x^T P\hat{B}K_{off}e.
\end{aligned}$$

We have just expressed $\dot{V}$ in terms of the model parameters and the uncertainty of the state matrix A. We now proceed to evaluate the contributions of each, the model, the uncertainty, and the error.

$$\begin{aligned}\dot{V} &\leq -q|x|^2 + \left|\widetilde{A}^T P + P\widetilde{A}\right||x|^2 + 2\left|P\hat{B}\,K_{off}\right||e||x| \\ &\leq (-q+\Delta)|x|^2 + b|e||x|. \end{aligned} \tag{14.29}$$

Now, by sending updates according to (14.28), which sets the error equal to 0 at every update time, we can see first that

$$|e_i|^2 \leq \alpha^2 |x_i|^2$$

and

$$|e|^2 = \sum_{i=1}^{N} |e_i|^2 \leq \sum_{i=1}^{N} \alpha^2 |x_i|^2 = \alpha^2 |x|^2.$$

Since $\alpha > 0$ and the norms are always nonnegative then we have that

$$|e| \leq \alpha |x|. \tag{14.30}$$

We can use (14.30) in (14.29) to obtain

$$\dot{V} \leq (\sigma - 1)(q - \Delta)|x|^2. \tag{14.31}$$

Then V is guaranteed to decrease for any σ such $0 < \sigma < 1$ and updating the elements of the state of the models in all LCUs according to the events in (14.28). ◆

The extension to consider the case of $\hat{A} \neq A$ and $\hat{B} \neq B$ is straightforward by assuming that the next bounds on the uncertainty matrices hold:

$$\left|\left(\widetilde{A} + \widetilde{B}K\right)^T P + P\left(\widetilde{A} + \widetilde{B}K\right)\right| \leq \Delta < q \tag{14.32}$$

$$\left|\widetilde{B}\right| \leq \beta \tag{14.33}$$

where $\widetilde{B} = B - \hat{B}$. In order to obtain (14.31) the local errors are set to satisfy (triggering an update otherwise):

$$|e_i| \leq \frac{\sigma(q-\Delta)}{\overline{b}} |x_i| \tag{14.34}$$

where $\overline{b} = b + 2\beta|K||P|$.

By following the approach described above each LCU is capable of determining the time instants at which it should send its current measurement to the network. An important drawback in Theorem 14.5 is that the controller is designed based on the

model dynamics of the overall system and the threshold is calculated as a function of the bounds on the uncertainty of the augmented system as well. In the next section we offer a complete decentralized solution, that is, not only the LCUs update their states based on local information but the local controllers and the local thresholds, which are not necessarily the same for every agent as in Theorem 14.5, can also be designed based on local model dynamics and uncertainty bounds.

14.4 Decentralized Model-Based Event-Triggered Control

Consider the network of coupled subsystems represented by (14.16) with models (14.17). We assume that every pair $(\hat{A}_i, \hat{B}_i)$ is stabilizable. We design controllers K_i that render the matrices $\hat{A}_i + \hat{B}_i K_i$ Hurwitz. Then for every agent i, $i \in \mathrm{N}$, there exists a symmetric and positive definite P_i which is the solution of the closed-loop local decoupled model

$$\left(\hat{A}_i + \hat{B}_i K_i\right)^T P_i + P_i\left(\hat{A}_i + \hat{B}_i K_i\right) = -Q_i \tag{14.35}$$

where Q_i is a symmetric and positive definite matrix.

Theorem 14.6 *Let* (14.18) *be the control input for each agent in the networked system* (14.16). *Assume that the following bounds are satisfied*

$$\left|\left(\widetilde{A}_i + \widetilde{B}_i K_i\right)^T P_i + P_i\left(\widetilde{A}_i + \widetilde{B}_i K_i\right)\right| \leq \Delta_i < q_i \tag{14.36}$$

$$\sum_{j=1, j\neq i}^{N} \left|P_j \widetilde{A}_{ji}\right|^2 \leq W_i \leq \frac{f_i^2}{8(N-1)} \tag{14.37}$$

where $q_i = \underline{\sigma}(Q_i)$, $f_i = q_i - \Delta_i$ *and* $\widetilde{A}_{ji} = A_{ji} - B_j K_{ji}$. *Then the networked system* (14.16) *is globally asymptotically stable when the local events are triggered by*

$$|e_i|^2 > \frac{\chi_i}{\beta_i}|x_i|^2 \tag{14.38}$$

where $\chi_i = f_i - 2(N-1)\delta_i - \dfrac{W_i}{\delta_i}$, $\beta_i = \displaystyle\sum_{j=1, j\neq i}^{N} \frac{\left|P_j B_j K_{ji}\right|^2}{\delta_i}$, *and* δi *is such that*

$$\begin{cases} \delta_{i1} < \delta_i < \delta_{i2} & \text{if} \quad \delta_{i1} > 0 \\ 0 < \delta_i < \delta_{i2} & \text{if} \quad \delta_{i1} \leq 0 \end{cases} \tag{14.39}$$

with

$$\delta_{i1} = \frac{f_i}{N-1}\left(\frac{1}{4} - \sqrt{\frac{1}{16} - \frac{(N-1)W_i}{2{f_i}^2}}\right) \tag{14.40}$$

$$\delta_{i2} = \frac{f_i}{N-1}\left(\frac{1}{4} + \sqrt{\frac{1}{16} - \frac{(N-1)W_i}{2{f_i}^2}}\right). \tag{14.41}$$

Proof We consider the candidate Lyapunov function

$$V(x) = \sum_{i=1}^{N} V_i(x_i) \tag{14.42}$$

and use the following proposition.

Proposition 14.7 *The derivative of* $V_i = {x_i}^T P_i x_i$ *along the trajectories of subsystem* i *in* (14.16) *with control input* (14.18) *satisfies the inequality.*

$$\begin{aligned}\dot{V}_i \leq & -\left(f_i - 2(N-1)\delta_i\right)|x_i|^2 \\ & + \sum_{j=1, j\neq i}^{N} \frac{\left|P_i \widetilde{A}_{ij}\right|^2}{\delta_i}|x_j|^2 + \sum_{j=1, j\neq i}^{N} \frac{\left|P_i B_i K_{ij}\right|^2}{\delta_i}|e_j|^2.\end{aligned} \tag{14.43}$$

Proof of Proposition 14.7 Consider the local Lyapunov function $V_i = {x_i}^T P_i x_i$ and compute its derivative along the trajectories of subsystem i in (14.16) using the local control input (14.18) to obtain

$$\begin{aligned}\dot{V}_i = & {x_i}^T\left(\left(\hat{A}_i + \hat{B}_i K_i\right)^T P_i + P_i\left(\hat{A}_i + \hat{B}_i K_i\right)\right)x_i + {x_i}^T\left(\left(\widetilde{A}_i + \widetilde{B}_i K_i\right)^T P_i + P_i\left(\widetilde{A}_i + \widetilde{B}_i K_i\right)\right)x_i \\ & + {x_i}^T P_i\left(\sum_{j=1 j\neq i}^{N} \widetilde{A}_{ij} x_j + \sum_{j=1 j\neq i}^{N} B_i K_{ij} x_j\right) + \left(\sum_{j=1 j\neq i}^{N} \widetilde{A}_{ij} x_j + \sum_{j=1 j\neq i}^{N} B_i K_{ij} x_j\right)^T P_i x_i\end{aligned} \tag{14.44}$$

and consider the next inequality involving the vectors $\mu \in \mathbb{R}^n, \nu \in \mathbb{R}^m$

$$|\delta\mu - \Pi\nu|^2 \geq 0 \tag{14.45}$$

where $\Pi \in \mathbb{R}^{n\times m}$ and δ is any positive real constant. Equation (14.45) can be expanded to yield

$$\mu^{\mathrm{T}}\Pi\nu \leq \frac{|\Pi\nu|^2}{2\delta} + \frac{\delta|\mu|^2}{2} \tag{14.46}$$

Applying (14.44) to (14.41) we obtain

$$\dot{V}_i \leq -q_i|x_i|^2 + \Delta_i|x_i|^2 + \sum_{j=1,j\neq i}^{N}\left(\frac{\left|P_i\widetilde{A}_{ij}\right|^2}{\delta_i}|x_j|^2 + \delta_i|x_i|^2\right) + \sum_{j=1,j\neq i}^{N}\left(\frac{|P_iB_iK_{ij}|^2}{\delta_i}|e_j|^2 + \delta_i|x_i|^2\right)$$

Finally, we write the terms involving $|x_i|^2$ together and we obtain (14.43). ◆

Now, taking the derivative of the Lyapunov function V along the trajectories of the state $x = [x_1{}^T\ x_2{}^T\ \ldots\ x_n{}^T]^T$ results in the next expression

$$\begin{aligned}\dot{V}(x) &= \sum_{i=1}^{N}\dot{V}_i(x_i)\\ &\leq \sum_{i=1}^{N} -(f_i - 2(N-1)\delta_i)|x_i|^2 + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{\left|P_i\widetilde{A}_{ij}\right|^2}{\delta_i}|x_j|^2 + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{|P_iB_iK_{ij}|^2}{\delta_i}|e_j|^2.\end{aligned}$$

We consider the case where all subsystems can receive measurement updates from the rest of the agents in the network, although this is not a necessary condition for the validity of the results in this theorem. It suffices for each agent i, $i \in \mathrm{N}$ to establish a bidirectional communication to those agents for which exchange of information is needed, that is, those agents that need to estimate the state x_i in any of their models and the agents for which agent i needs to estimate their state to use in any of their models. Then we can use the symmetry property of this type of interconnection to obtain

$$\begin{aligned}\dot{V}(x) &\leq \sum_{i=1}^{N} -(f_i - 2(N-1)\delta_i)|x_i|^2 + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{\left|P_j\widetilde{A}_{ji}\right|^2}{\delta_i}|x_i|^2 + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\frac{|P_jB_jK_{ji}|^2}{\delta_i}|e_i|^2\\ &\leq -\left(\sum_{i=1}^{N}\chi_i|x_i|^2 - \sum_{i=1}^{N}\beta_i|e_i|^2\right).\end{aligned} \tag{14.47}$$

It is clear that coefficients β_i are positive and, in order for condition (14.36) to be a valid threshold we need the χ_i coefficients to be positive as well, which requires to solve the following inequalities for the real parameter δ_i

$$\begin{aligned}\chi_i &= f_i - 2(N-1)\delta_i - \frac{W_i}{\delta_i} > 0\\ \delta_i &> 0\end{aligned} \tag{14.48}$$

It can be verified that the solution for the above inequalities is given by (14.39) with $\delta_{i1,i2}$ as in (14.40) and (14.41), moreover the solution is a real number by the assumption on the bounds W_i. Since we showed that $\chi_i, \beta_i > 0$ then the Lyapunov function is guaranteed to decrease by updating the models in all LCUs corresponding to the state x_i according to the threshold (14.38).

The parameters Δ_i represent given bounds on the norm of the uncertainty for every agent and they can be seen as a measure of how close the model and system dynamics are. The bound W_i represents a measure of how close we are able to cancel the effects of other subsystems on system i using the control gains that are designed based on the nominal models.

Example 14.3 Consider a network of $N = 10$ unstable subsystems represented as in (14.16) all with different dynamics. The dimensions of the systems vary from 1 to 3 as well. The models for all different parameters represent an uncertainty as follows: 12 % alteration in the A_i matrices, 10 % in A_{ij}, and 6 % in B_i. Every agent is coupled to all other agents including those with different dimensions by corresponding coupling matrices A_{ij}.

The unknown dynamics of the subsystems are given by:

$$A_1 = 0.4, \quad B_1 = 1$$

$$A_2 = 0.5, \quad B_2 = 1$$

$$A_3 = 0.2, \quad B_3 = 1$$

$$A_4 = 1, \quad B_4 = 1$$

$$A_5 = \begin{bmatrix} 0.21 & -0.4 \\ 0.3 & -0.7 \end{bmatrix}, \quad B_5 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} 0.3 & 2 \\ 0.6 & -1.8 \end{bmatrix}, \quad B_6 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} 0.05 & 0.003 \\ 0.0023 & -0.7 \end{bmatrix}, \quad B_7 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A_8 = \begin{bmatrix} 0.11 & -0.41 & 0 \\ 0 & 0.3 & -0.7 \\ 0 & -0.3 & -0.5 \end{bmatrix}, \quad B_8 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_9 = \begin{bmatrix} 0.3 & 2 & -0.07 \\ 0 & 0.09 & -1 \\ 0.1 & 0 & 0.2 \end{bmatrix}, \quad B_9 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_{10} = \begin{bmatrix} 0.05 & 0.003 & 0.5 \\ 0 & 0.0023 & -0.7 \\ -0.3 & 0 & -0.5 \end{bmatrix}, \quad B_{10} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The nominal model parameters are given by:

$$\hat{A}_1 = 0.352, \quad \hat{B}_1 = 0.95$$
$$\hat{A}_2 = 0.44, \quad \hat{B}_2 = 0.95$$
$$\hat{A}_3 = 0.176, \quad \hat{B}_3 = 0.95$$
$$\hat{A}_4 = 0.88, \quad \hat{B}_4 = 0.95$$

$$\hat{A}_5 = \begin{bmatrix} 0.1848 & -0.352 \\ 0.264 & -0.616 \end{bmatrix}, \quad \hat{B}_5 = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.95 \end{bmatrix}$$

$$\hat{A}_6 = \begin{bmatrix} 0.264 & 1.76 \\ 0.528 & -1.584 \end{bmatrix}, \quad \hat{B}_6 = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.95 \end{bmatrix}$$

$$\hat{A}_7 = \begin{bmatrix} 0.044 & 0.0026 \\ 0.002 & -0.616 \end{bmatrix}, \quad \hat{B}_7 = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.95 \end{bmatrix}$$

$$\hat{A}_8 = \begin{bmatrix} 0.0968 & -0.3608 & 0 \\ 0 & 0.264 & -0.616 \\ 0 & -0.264 & -0.44 \end{bmatrix}, \quad \hat{B}_8 = \begin{bmatrix} 0.95 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix}$$

$$\hat{A}_9 = \begin{bmatrix} 0.264 & 1.76 & -0.0616 \\ 0 & 0.0792 & -0.88 \\ 0.88 & 0 & 0.176 \end{bmatrix}, \quad \hat{B}_9 = \begin{bmatrix} 0.95 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix}$$

$$\hat{A}_{10} = \begin{bmatrix} 0.044 & 0.0026 & 0.44 \\ 0 & 0.002 & -0.616 \\ -0.264 & 0 & -0.44 \end{bmatrix}, \quad \hat{B}_{10} = \begin{bmatrix} 0.95 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 0.95 \end{bmatrix}$$

Every agent is coupled to all other agents including those with different dimensions by corresponding coupling matrices *Aij*. The unknown coupling matrices are given by:

$$A_{ij} = c_1 \quad for \;\; i = 1,2,3,4; \;\; j = 1,2,3,4; \;\; i \neq j.$$
$$A_{ij} = [\, c_1 \quad c_1 \,] \quad for \;\; i = 1,2,3,4; \;\; j = 5,6,7.$$
$$A_{ij} = [\, c_2 \quad c_2 \quad c_2 \,] \quad for \;\; i = 1,2,3,4; \;\; j = 8,9,10.$$
$$A_{ij} = [\, c_1 \quad c_1 \,]^T \quad for \;\; i = 5,6,7; \;\; j = 1,2,3,4.$$
$$A_{ij} = \begin{bmatrix} c_2 & 0 \\ 0 & c_2 \end{bmatrix} \quad for \;\; i = 5,6,7; \;\; j = 5,6,7; \;\; i \neq j.$$
$$A_{ij} = \begin{bmatrix} c_3 & 0 & c_3 \\ 0 & c_3 & 0 \end{bmatrix} \quad for \;\; i = 5,6,7; \;\; j = 8,9,10.$$
$$A_{ij} = [\, c_2 \quad c_2 \quad c_2 \,]^T \quad for \;\; i = 8,9,10; \;\; j = 1,2,3,4.$$

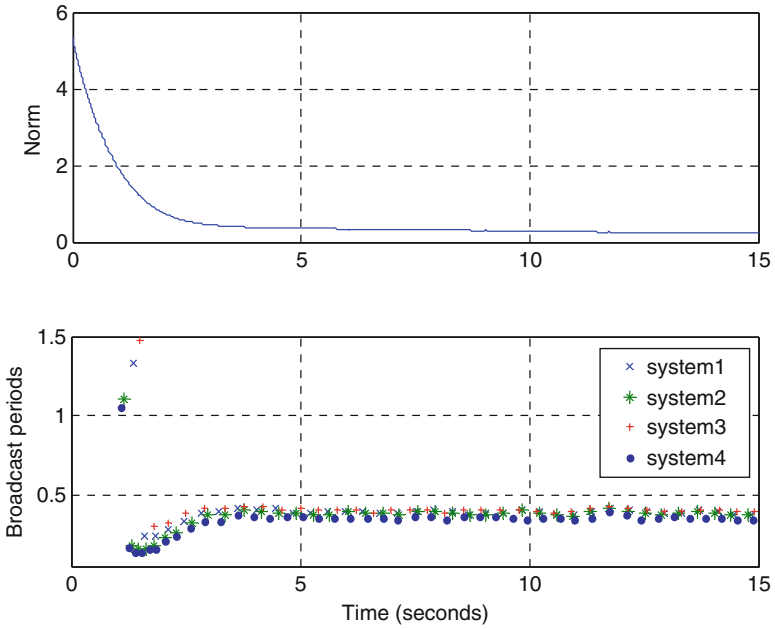

Fig. 14.7 The norm of the state of the overall system is shown in the *top*. The broadcasting period for subsystems 1–4 is shown at the *bottom*

$$A_{ij} = \begin{bmatrix} c_3 & 0 & c_3 \\ 0 & c_3 & 0 \end{bmatrix}^T \quad for \;\; i = 8, 9, 10; \;\; j = 5, 6, 7.$$

$$A_{ij} = \begin{bmatrix} c_3 & 0 & 0 \\ 0 & c_3 & 0 \\ 0 & 0 & c_3 \end{bmatrix} \quad for \;\; i = 8, 9, 10; \;\; j = 8, 9, 10; \;\; i \neq j.$$

for $c_1 = 0.5$, $c_2 = 0.4$, $c_3 = 0.1$. The nominal coupling matrices $\hat{A}_{ij}$ are of the same form but with $\hat{c}_1 = 0.45$, $\hat{c}_2 = 0.36$, $\hat{c}_3 = 0.09$.

The local controllers and thresholds are designed following the decentralized approach in this section, where only the model parameters and the bounds on the uncertainties are used. The results of simulations are shown in Figs. 14.7 and 14.8. In the top portion of Fig. 14.7 it can be seen the norm of the augmented state, that is, the response of all states of all subsystems. Figure 14.8 and the bottom portion of Fig. 14.7 show the broadcasting periods for every agent in the networked system. Figure 14.7 (bottom) represent the broadcasting periods for the four first order systems, Fig. 14.8 (top) for the 3 s order systems, and Fig. 14.8 (bottom) for the three third order systems.

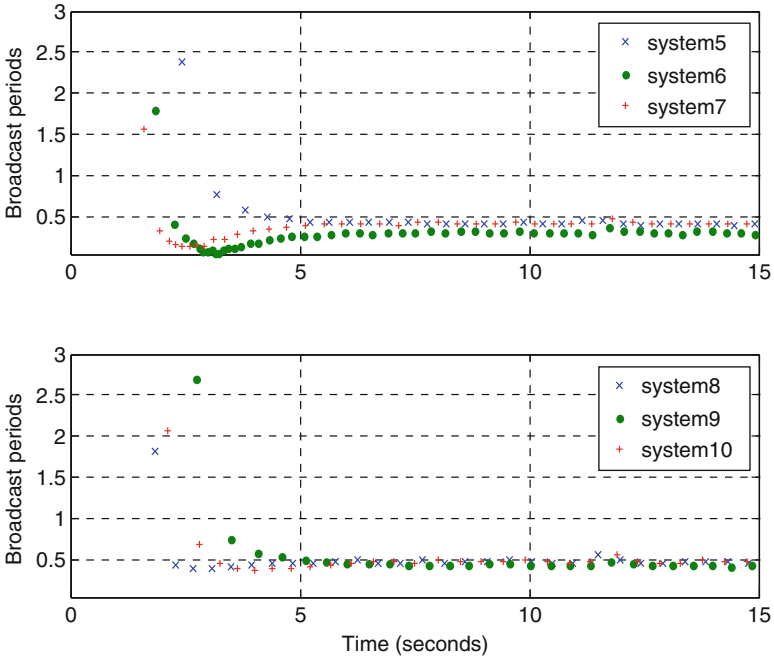

Fig. 14.8 Broadcasting period for subsystems 5–7 (*top*) and 8–10 (*bottom*)

Example 14.4 In this example we consider a collection of three coupled carts. The physical coupling corresponds to the springs used to connect the carts to each other, and assume that at the equilibrium of the system, all springs are not stretched. The dynamics of each cart and its corresponding model can be described by (14.16) and (14.17), respectively with

$$A_i = \begin{bmatrix} 0 & 1 \\ -c_i k & 0 \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad A_{ij} = \begin{bmatrix} 0 & 0 \\ d_{ij} & 0 \end{bmatrix},$$
$$\hat{A}_i = \begin{bmatrix} 0 & 1 \\ -c_i \hat{k} & 0 \end{bmatrix}, \quad \hat{B}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \hat{A}_{ij} = \begin{bmatrix} 0 & 0 \\ \hat{d}_{ij} & 0 \end{bmatrix},$$

where $c_1 = c_3 = 1$, $c_2 = 2$, $k = 5$, $d_{12} = d_{32} = d_{21} = d_{23} = 1$, and $d_{13} = d_{31} = 0$. The model parameters are $\hat{k} = 4.95$, $\hat{d}_{13} = \hat{d}_{31} = 0$, and the remaining $\hat{d}_{ij} = 1.01$. Results of simulations of this example are shown in Fig. 14.9. In this simulation, the carts start with an initial condition that represents stretched springs and the purpose is to control the carts motion to reach an equilibrium where the springs are not stretched.

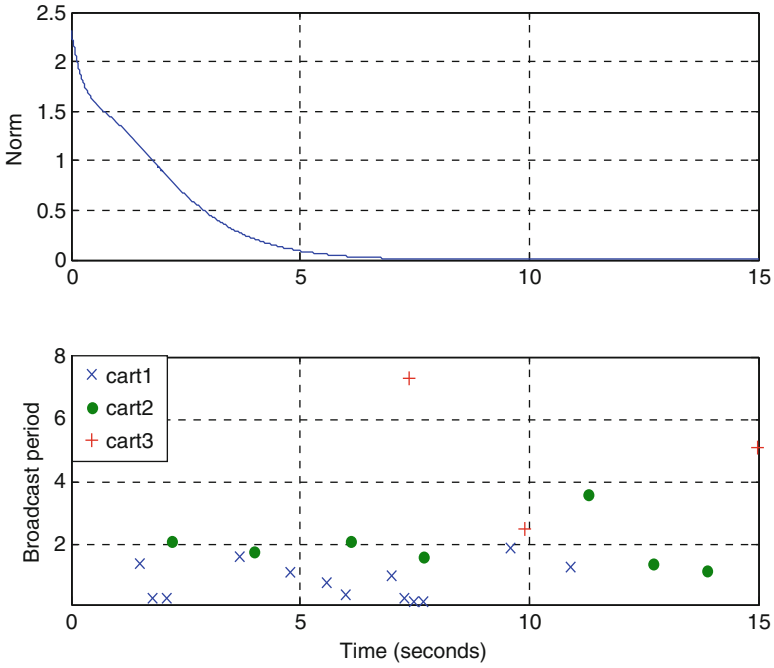

Fig. 14.9 The norm of the state of the overall system in example 14.4 is shown in the *top*. The broadcasting period (in seconds) for each cart is shown at the *bottom*

14.5 Notes and References

In this chapter the MB-NCS approach that was introduced in Chap. 2 has been extended to consider networks of coupled systems using periodic communication and using multirate sampling for discrete-time systems and using event-triggered control for continuous-time systems.

The work discussed in Sects. 14.1 and 14.2 appeared first in [90]. The work discussed in Sects. 14.3 and 14.4 was first published in [85].

The event-triggered control has been extended to consider networked interconnected systems in [50, 53, 54, 95, 102–104, 177, 178, 228, 265]. A common characteristic in the previous work on event-triggered control is the use of a Zero-Order-Hold (ZOH) model in the controller node and the assumption that the models being used are the same as the plants they represent, i.e., no model uncertainties are considered. Similar model-based approaches for stabilization of coupled systems and for cooperation of multi-agent systems have been presented in [52, 96, 240–243].

In Chap. 6 the Model-Based Event-Triggered (MB-ET) control framework that considered model uncertainties was presented. Here, we extended that approach to consider multiple interconnected subsystems. This strategy represents considerable

savings on network bandwidth compared to similar work in the literature that use model-based implementations.

Similar model-based implementations to the one presented in this chapter that consider systems with distributed sensors can be found in [66, 240–243]. In [240] the authors study the stabilization of coupled continuous-time systems employing the MB-NCS framework described in Chap. 2 of this book and using a single-rate approach which forces all agents to send measurement updates through the network all at the same time instants. That approach was extended in [66, 242, 243] to the case when a schedule for the updates is pre-assigned but all subsystems still use the same update rate, i.e., a single-rate approach in which the agents send updates at different time instants.

In [241] the same problem is considered using event-triggered control techniques but the proposed solution requires the opposite updating strategy as the one we present in Sect. 14.4; all agents need to communicate or send their information to agent i when the error e_i grows large, that is, agent i needs to send a request for updates to all other agents, then all agents need to respond and send their current measurements to agent i all of them at the same time instant, which may produce packet collisions and loss of information. In addition, since the update request is based on the local error e_i, we could be requesting all other agents to send their information to agent i when it is not necessary, i.e., their local errors are small and the growth of e_i may be due to large local parameter uncertainty or due to only one or very few errors from other agents.

The strategy proposed here avoids the unnecessary increase in communication by simply making agent i to broadcast its state according to its local error. If all agents including agent i have the same estimate $\hat{x}_i$ of x_i then when the error e_i is large by an appropriate measure we know it is necessary for all agents to receive the real state and update the state of the model $\hat{x}_i$.

The decentralized method described in this section extends the results provided by [261]. In [261] only a ZOH model is used, that is, the control input for each agent remains constant between updates. With respect to previous work in event-triggered control, the implementation of this strategy using MB-NCS accounts for the unavoidable existence of model uncertainties in the stability analysis.

Appendix A

This appendix contains a brief review of basic tools, concepts, and methods that have been used in this book. Essential notions concerning linear algebra, control theory, linear systems, and Lyapunov analysis are described in the following sections.

Linear Algebra

Vector norms and matrix norms. Let $\mathbf{V}$ be a vector space over $\mathbb{R}$ (the field of real numbers) or $\mathbb{C}$ (the field of complex numbers). A real valued scalar function $\|\cdot\|$ defined on $\mathbf{V}$ is said to be a vector norm if it satisfies the following properties:

- $\|x\| \geq 0$.
- $\|x\| = 0$ if and only if $x = 0$.
- $\|\alpha x\| = |\alpha|\|x\|$, for any scalar α. $|\alpha|$ denotes the absolute value of α.
- $\|x + y\| \leq \|x\| + \|y\|$, also called triangle inequality property.

for any $x, y \in \mathbf{V}$.

The norm generalizes the usual concept of length in lower dimensional Euclidean spaces. The p-norm of vector x is defined as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{A.1}$$

for $1 \leq p \leq \infty$. Particular cases of interest include $p = 1, 2, \infty$ and the corresponding norms are defined as follows:

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8,

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \tag{A.2}$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} \tag{A.3}$$

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|. \tag{A.4}$$

For a matrix $A \in \mathbb{C}^{m \times n}$ we can define the *matrix norm* induced by a vector p-norm in the following way

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}. \tag{A.5}$$

From a system theoretic point of view the matrix induced norms can be seen as input/output amplification gains. A particular case of this type of norm is given by the induced 2-norm which can be computed as follows

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^* A)} \tag{A.6}$$

where $\lambda_{\max}(A^*A)$ represents the largest eigenvalue of the symmetric matrix A^*A, where A^* denotes the conjugate transpose matrix of A.

Eigenvalues and eigenvectors. The eigenvalues λ_i, $i = 1,\ldots,n$ of a square matrix $A \in \mathbb{C}^{n \times n}$ are the n roots of its characteristic polynomial

$$p(\lambda) = \det(\lambda I - A). \tag{A.7}$$

A *right eigenvector* of matrix A is a non-zero vector $x \in \mathbb{C}^n$ that satisfies

$$Ax = \lambda x. \tag{A.8}$$

Note that if x is an eigenvector of A, then any αx, $\alpha \neq 0$ scalar, is also an eigenvector of A. Similarly, a *left eigenvector* of matrix A is a non-zero vector $y \in \mathbb{C}^n$ that satisfies the following

$$y^* A = \lambda y^* \tag{A.9}$$

where the notation y^* represents the conjugate transpose of the vector y.

A matrix $A \in \mathbb{R}^{n \times n}$ that satisfies $[a_{ij}] = [a_{ji}]$, where $[a_{ij}]$ represents the entry of matrix A in the ith row and jth column is called a real *symmetric matrix*.

The eigenvalues of a symmetric matrix are all real and they can be ordered in the following way:

$$\lambda_1(A) \leq \lambda_2(A) \leq \ldots \leq \lambda_n(A). \tag{A.10}$$

We define $\lambda_{\min}(A) = \lambda_1(A)$, the smallest eigenvalue of the real symmetric matrix A, and $\lambda_{\max}(A) = \lambda_n(A)$, the largest eigenvalue of the real symmetric matrix A.

A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called a *positive definite* (*semi-definite*) matrix, represented as $A > 0$ ($A \geq 0$), if $z^T A z$ is positive (non-negative) for any non-zero vector $z \in \mathbb{R}^n$.

The eigenvalues of a positive definite matrix A are all real and positive. Every positive definite matrix is invertible and its inverse is also a positive definite matrix. For $A, B \in \mathbb{R}^{n \times n}$, the notation $A > B$ ($A \geq B$) means that the matrix $A - B$ is positive definite (semi-definite).

Consider the dynamical system

$$\dot{x} = Ax \tag{A.11}$$

and consider the change of variables $x = Tx$', where T is a constant, square, and invertible matrix. By substituting x in (A.11) we have

$$T\dot{x}' = ATx'. \tag{A.12}$$

By multiplying on the left by T^{-1} we obtain the following

$$\dot{x}' = T^{-1}ATx' = A'x' \tag{A.13}$$

where $A' = T^{-1}AT$. The characteristic polynomials of A and A' are the same; this can be shown as follows

$$\begin{aligned} \det(sI - A') &= \det\left(sI - T^{-1}AT\right) \\ &= \det\left(sT^{-1}T - T^{-1}AT\right) \\ &= \det\left(T^{-1}(sI - A)T\right) \\ &= \det T^{-1} \det(sI - A) \det T \\ &= \det(sI - A). \end{aligned} \tag{A.14}$$

A and A' are called *similar matrices* and T is called a *similarity transformation*.

Now, assume that the matrix A has distinct eigenvalues $\lambda_1, \ldots, \lambda_n$. Then A has n linearly independent eigenvectors $\{v_1, \ldots, v_n\}$. Let $A_D = P^{-1}AP$ where $P = [v_1, \ldots, v_n]$, are the right eigenvectors of matrix A. The n diagonal entries in A_D are the eigenvalues of A

$$A_D = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}. \tag{A.15}$$

A and A_D are matrix representations of the same abstract linear transformation; A and A_D are similar matrices and the similarity transformation is

$$P = [v_1, \ldots, v_n]. \tag{A.16}$$

When a matrix A has repeated eigenvalues, then it is not always possible to diagonalize it. The Jordan canonical form is a useful representation of matrix A when diagonalization is not possible. This transformation can be obtained by means of generalized eigenvectors.

A vector v is called a *generalized eigenvector* of rank k of matrix A, associated with an eigenvalue λ if and only if

$$(A - \lambda I)^k v = 0 \quad \text{and} \quad (A - \lambda I)^{k-1} v \neq 0. \tag{A.17}$$

Note that for $k = 1$ the definition of a generalized eigenvector is the definition of an eigenvector of A (A.8).

Define a chain of generalized eigenvectors $\{v_1, \ldots, v_k\}$ as follows:

$$\begin{aligned} v_k &= v \\ v_{k-1} &= (A - \lambda I)v = \big(A - \lambda I\big)v_k \\ v_{k-2} &= (A - \lambda I)^2 v = \big(A - \lambda I\big)v_{k-1} \\ &\vdots \\ v_1 &= (A - \lambda I)^{k-1} v = \big(A - \lambda I\big)v_2. \end{aligned} \tag{A.18}$$

For generalized eigenvectors we have the following results

- The generalized eigenvectors $\{v_1, \ldots, v_k\}$ defined in (A.18) are linearly independent.
- The generalized eigenvectors of A associated with different eigenvalues are linearly independent.
- If u and v are generalized eigenvectors of rank k and l, respectively, associated with the same eigenvalue λ, and if u_i and v_j are defined by

$$\begin{aligned} u_i &= (A - \lambda I)^{k-i} u \quad i = 1, \ldots, k, \\ v_j &= (A - \lambda I)^{l-j} v \quad j = 1, \ldots, l, \end{aligned} \tag{A.19}$$

and if u_i and v_j are linearly independent, then the generalized eigenvectors $\{u_1, \ldots, u_k\}$ and $\{v_1, \ldots, v_l\}$ are linearly independent.

The Jordan canonical form of a matrix A can be obtained by using a new basis of the vector space constructed using the above results on the generalized eigenvectors. A Jordan matrix J can be written as follows:

$$J = \begin{bmatrix} J_0 & & & 0 \\ & J_1 & & \\ & & \ddots & \\ 0 & & & J_s \end{bmatrix} \tag{A.20}$$

where J_0 is a diagonal matrix with diagonal elements $\{\lambda_1, \ldots, \lambda_k\}$ (not necessarily distinct), and each J_p is an $n_p \times n_p$ matrix of the form

$$J_p = \begin{bmatrix} \lambda_{k+p} & 1 & 0 & \ldots & 0 \\ & \lambda_{k+p} & 1 & & \vdots \\ \vdots & \vdots & \ddots & & 1 \\ 0 & 0 & & \ldots & \lambda_{k+p} \end{bmatrix} \tag{A.21}$$

for $p = 1,\ldots,s$, where λ_{k+p} need not be different than λ_{k+q} for $p \neq q$ and $k + n_1 + \ldots + n_s = n$. The blocks $J_0, J_1, \ldots, J_s$ are called the Jordan blocks and the numbers λ_i, $i = 1, \ldots, k+s$ are the eigenvalues of A. If λ_i is a simple non-repeated eigenvalue of A, then it appears in the block J_0.

Note that a matrix can be similar to a diagonal matrix without having distinct eigenvalues; in fact, a $n \times n$ matrix is similar to a diagonal matrix if and only if there are n linearly independent eigenvectors. Also, it can be shown that any real symmetric matrix is similar to a diagonal matrix.

Linear Systems and Control Theory

Control theory addresses the behavior of dynamical systems and the tools and methods to influence this behavior in order to attain a desired response. Dynamical systems can be classified as different types (e.g., lumped or distributed) and described using different representations (e.g., ordinary or partial differential equations; integral equations). Consider the state space representation of a continuous-time linear finite dimensional time-invariant (LTI) system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{A.22}$$

$$y(t) = Cx(t) + Du(t) \tag{A.23}$$

where $x \in \mathbb{R}^n$ is the state of the system (n is the dimension), $u \in \mathbb{R}^m$ represents the input to the system, and $y \in \mathbb{R}^p$ is the output of the system. The expression in (A.22) is called the *state equation* and (A.23) is called the *output equation*. The input $u(t)$ is the available signal that can modify the behavior of the system and it is referred to as the *control input*. When the control input is a function of the state of the system, i.e., $u(t) = f(x(t))$, then a *closed-loop control* is applied to the system, otherwise, an *open-loop control* is used. The solution to (A.22) given the initial conditions $x(t_0)$ is

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \tag{A.24}$$

and the output response is given by

$$y(t) = Ce^{A(t-t_0)}x(t_0) + C\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau + Du(t). \tag{A.25}$$

Usually in LTI systems t_0 is taken to be zero ($t_0 = 0$).

Definition A.1 Let $A \in \mathbb{C}^{n\times n}$ be a constant matrix. The exponential matrix e^{At} is defined as follows:

$$e^{At} = I + \sum_{k=1}^{\infty} \frac{t^k}{k!}A^k \tag{A.26}$$

for any $-\infty < t < \infty$.

Similarly, a discrete-time linear time-invariant system can be described using the following state space representation

$$x(k+1) = Ax(k) + Bu(k) \tag{A.27}$$

$$y(k) = Cx(k) + Du(k). \tag{A.28}$$

The solutions to the discrete-time *state equation* (A.27) and *output equation* (A.28) are given, respectively, by the following expressions:

$$x(k) = A^{(k-k_0)}x(k_0) + \sum_{j=k_0}^{k-1} A^{(k-(j+1))}Bu(j) \tag{A.29}$$

$$y(k) = CA^{(k-k_0)}x(k_0) + \sum_{j=k_0}^{k-1} CA^{(k-(j+1))}Bu(j) + Du(k) \tag{A.30}$$

for $k > k_0$, where $x(k_0)$ is a given initial condition.

A state $\chi \in \mathbb{R}^n$ is called reachable if there exists an open- or closed-loop control that transfers the state of the system $x(t)$ from the zero state to the state χ in some finite time T. A state $\xi \in \mathbb{R}^n$ is called controllable if there exists an open- or closed-loop control that transfers the state from ξ to the zero state in some finite time T.

A time-invariant system (A.22) or (A.27) is said to be reachable when all states are reachable. This is true if and only if its controllability matrix given by

$$[A|B] = \left[B, AB, \ldots, A^{n-1}B\right] \in \mathbb{R}^{n\times mn} \tag{A.31}$$

Table A.1 Laplace transforms

$f(t), (t \geq 0)$	$\hat{f}(s) = L[f(t)]$
$\delta(t)$	1
$u(t)$	$1/s$
$t^k/k!$	$1/s^{k+1}$
e^{-at}	$1/(s+a)$
$t^k e^{-at}$	$k!/(s+a)^{k+1}$
$e^{-at}\sin bt$	$b/[(s+a)^2+b^2]$
$e^{-at}\cos bt$	$(s+a)/[(s+a)^2+b^2]$

Table A.2 Laplace transform properties

Time differentiation	$d^k f(t)/dt^k$	$s^k\hat{f}(s) - \left[s^{k-1}f(0) + \ldots + f^{(k-1)}(0)\right]$
Frequency shift	$e^{-at}f(t)$	$\hat{f}(s+a)$
Time shift	$f(t-a)u(t-a), a>0$	$e^{-as}\hat{f}(s)$
Scaling	$f(t/\alpha), \alpha>0$	$\alpha\hat{f}(\alpha s)$
Convolution	$\int_0^t f(\tau)g(t-\tau)d\tau$	$\hat{f}(s)\hat{g}(s)$
Initial value	$\lim_{t\to 0^+} f(t)$	$\lim_{s\to\infty} s\hat{f}(s)^{\dagger}$
Final value	$\lim_{t\to\infty} f(t)$	$\lim_{s\to 0} s\hat{f}(s)^{\diamond}$

† If the limit exists

$^{\diamond}$ If $s\hat{f}(s)$ has no singularities on the imaginary axis or in the right half s plane

has full row rank n. Note that reachability implies controllability but the other way around is not true in general in discrete-time systems. In continuous-time systems reachability implies and is implied by controllability always.

A common tool for analysis and design in control systems is the Laplace transform. The one-sided Laplace transform is a linear operator of a function $f(t)$ with a real argument t, $t \geq 0$, that transforms $f(t)$ to a function $\hat{f}(s)$ with complex argument s. The Laplace transform, denoted as $L[f(t)]$, is given by the integral

$$\hat{f}(s) = \int_0^\infty f(t)e^{-st}dt. \tag{A.32}$$

If $f(t) = [f_1(t), \ldots, f_n(t)]^T$, where $f_i(t) : [0, \infty) \to \mathbb{R}$, $i = 1, \ldots, n$, and if each $f_i(t)$ is Laplace transformable, then we define the Laplace transform of the vector f component-wise: i.e., $\hat{f}(s) = \left[\hat{f}_1(s), \ldots, \hat{f}_n(s)\right]^T$, where $\hat{f}_i(s) = L[f_i(t)]$. Table A.1 shows the Laplace transform of common time signals. In Table A.2 important properties of the Laplace transform are summarized. Note that for all functions in the table $t \geq 0$; for $t < 0$ the functions are zero

A transformation commonly used for discrete-time systems is the Z-transform. The one-sided Z-transform of a discrete-time signal $f(n)$, where $n \geq 0$ is an integer, is defined as

$$\hat{f}(z) = Z[f(n)] = \sum\nolimits_{n=0}^{\infty} f[n]z^{-n} \tag{A.33}$$

where z is a complex number in general.

Stability and Lyapunov Analysis

Consider the dynamical system described by the nonlinear first order ordinary differential equation

$$\dot{x} = f(x), \quad x(t_0) = x_0 \tag{A.34}$$

where $x \in \mathbb{R}^n$ and $x(t_0)$ is the initial value, or initial condition, at time t_0, and f: $D \to \mathbb{R}^n$ is a locally Lipschitz map from a domain $D \subset \mathbb{R}^n$ into $\mathbb{R}^n$

Definition A.2 A function $f: \mathbb{R}^d \to \mathbb{R}^m$ is called Lipschitz continuous if there exists a constant $L_x > 0$ such that

$$\|f(x_1) - f(x_2)\| \leq L_x \|x_1 - x_2\|. \tag{A.35}$$

If (A.35) is only valid on a subset D of $\mathbb{R}^n$, then the function $f(x)$ is called locally Lipschitz.

Assume that (A.34) possesses a unique solution $\phi(t, t_0, x_0)$ for all $t \geq t_0$.

Definition A.3 A point $x_e \in \mathbb{R}^n$ is called an equilibrium point of (A.34), or simply an equilibrium of (A.34), if

$$f(x_e) = 0. \tag{A.36}$$

We will assume that the equilibrium of interest is located at the origin of $\mathbb{R}^n$, i.e., $f(0) = 0$.

Several definitions of stability (called Lyapunov stability) can be associated with an equilibrium of the homogeneous system (A.34).

Definition A.4 The equilibrium $x = 0$ of (A.34) is said to be stable if for every $\varepsilon > 0$, there exists a $\delta(\varepsilon) > 0$ such that

$$\|x_0\| < \delta(\varepsilon) \Rightarrow \|\phi(t, t_0, x_0)\| < \varepsilon \tag{A.37}$$

for all $t \geq t_0$. It is said to be unstable if it is not stable, and it is said to be asymptotically stable if it is stable and

$$\|x_0\| < \delta(\varepsilon) \Rightarrow \lim_{t \to \infty} \|\phi(t, t_0, x_0)\| = 0. \tag{A.38}$$

Definition A.5 The equilibrium $x = 0$ of (A.34) is said to be globally asymptotically stable if it is asymptotically stable when the system (A.34) is arbitrarily initialized.

Stability of a given dynamical system can be established by means of candidate Lyapunov functions.

Let $V:\mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable and positive definite function, i.e., $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$. If $V(x) \to \infty$ when $\|x\| \to \infty$, then V is called radially unbounded.

If system (A.34) admits a function V described in the previous paragraph and we have $\dot{V}(x) \leq 0$ for all x along the trajectories of (A.34), then the *system is stable*.

Similarly, if $\dot{V}(x) < 0$, then the *system is asymptotically stable*. Furthermore, if the Lyapunov function V is radially unbounded, then the system is *globally asymptotically stable*.

A notion of stability frequently used in this book concerning systems with exogenous inputs is input-to-state stability (ISS). Consider the system with input u described by

$$\dot{x} = f(t, x, u) \tag{A.39}$$

where $f:[0,\infty) \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x and u. The input $u(t)$ is piecewise continuous and it is a bounded function of t for all $t \geq 0$.

In order to provide a definition for ISS, let us first define the following types of functions:

Definition A.6 A continuous function $\alpha:[0,a) \to [0,\infty)$ is said to belong to class $\mathtt{K}$ if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class $\mathtt{K}_\infty$ if $a = \infty$ and $\alpha(r) \to \infty$ as $r \to \infty$.

Definition A.7 A continuous function $\beta:[0,a) \times [0,\infty) \to [0,\infty)$ is said to belong to class $\mathtt{KL}$ if for each fixed s, the mapping $\beta(r,s)$ belongs to class $\mathtt{K}$ with respect to r and, for each fixed r, the mapping $\beta(r,s)$ is decreasing with respect to s and $\beta(r,s) \to 0$ as $s \to \infty$.

We now provide a definition for ISS using the comparison functions defined in the previous two items.

Definition A.8 The system (A.39) is said to be input-to-state stable (ISS) if there exists a class $\mathtt{KL}$ function β and a class $\mathtt{K}$ function γ such that for any initial state $x(t_0)$ and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq t_0$ and satisfies

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) + \gamma\big(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\|\big). \tag{A.40}$$

Lyapunov functions can also be used in this case in order to certify ISS of a given nonlinear system with exogenous input.

Theorem A.9 *Let $V:[0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that*

$$\alpha_1(\|x\|) \leq V(t,x) \leq \alpha_2(\|x\|) \tag{A.41}$$

$$\frac{\partial V}{\partial t}+\frac{\partial V}{\partial x}f(t,x,u)\leq -W_3(x)\quad \textit{for all}\ \ \|x\|\geq \rho(\|u\|)>0 \tag{A.42}$$

for all $(t,x,u)\in[0,\infty)\times\mathbb{R}^n\times\mathbb{R}^m$, *where* α_1,α_2 *are class* K_∞ *functions*, ρ *is a class* K *function, and* $W_3(x)$ *is a continuous positive definite function on* $\mathbb{R}^n$. *Then, the system* (A.39) *is input-to-state stable with* $\gamma=\alpha_1^{-1}\circ\alpha_2\circ\rho$.

For the continuous-time linear time-invariant system described by (A.11) we have that $x=0$ is always an equilibrium of (A.11). Let $x(t)=\phi(t,0,x_0)$ represent the solution of (A.11) where $t_0=0$ without loss of generality.

Theorem A.10 *The following statements are equivalent*:

(a) *The equilibrium* $x=0$ *of* (A.11) *is asymptotically stable.*
(b) *The equilibrium* $x=0$ *of* (A.11) *is globally asymptotically stable.*
(c) $\lim_{t\to\infty}\|x(t)\|=0$.

Theorem A.11 *The equilibrium* $x=0$ *of* (A.11) *is stable if and only if all eigenvalues of A have non-positive real parts, and every eigenvalue with zero real part has an associated Jordan block of order one. The equilibrium* $x=0$ *of* (A.11) *is asymptotically stable if and only if all eigenvalues of A have negative real parts.*

Stability of linear systems can also be analyzed using Lyapunov functions. A typical choice of Lyapunov function for linear systems is a quadratic function of the form:

$$V(x)=x^TPx \tag{A.43}$$

where $P>0$. The derivative of (A.41) along the trajectories of the linear system (A.11) is given by:

$$\begin{aligned}\dot{V}(x)&=\dot{x}^TPx+x^TP\dot{x}\\&=x^T\left(A^TP+PA\right)x.\end{aligned} \tag{A.44}$$

Recall that if $\dot{V}(x)<0$ then system (A.11) is globally asymptotically stable. Therefore we wish to find a $P>0$ such that the $A^TP+PA<0$. The following theorem formalizes this discussion.

Theorem A.12 *The equilibrium* $x=0$ *of* (A.11) *is asymptotically stable if and only if for any given positive definite symmetric matrix Q there exists a positive definite symmetric matrix P that satisfies*

$$A^TP+PA=-Q. \tag{A.45}$$

Equation (A.45) *is called the Lyapunov equation.*

Discrete-time systems. Lyapunov stability concepts can also be used to characterize and study discrete-time systems. Consider the following nonlinear discrete-time system with dynamics given by:

$$x(k+1) = f(x(k)), \quad x(k_0) = x_0. \tag{A.46}$$

For the time-invariant system (A.46), assume without loss of generality that $k_0 = 0$ and denote its solution by $\phi(k, x_0)$.

Definition A.13 A point $x_e \in \mathbb{R}^n$ is called an equilibrium point of (A.46), or simply an equilibrium of (A.46), if

$$f(x_e) = x_e. \tag{A.47}$$

Definition A.14 The equilibrium $x = 0$ of (A.46) is said to be stable if for every $\varepsilon > 0$, there exists a $\delta(\varepsilon) > 0$ such that

$$\|x_0\| < \delta(\varepsilon) \Rightarrow \|\phi(k, x_0)\| < \varepsilon \tag{A.48}$$

for all $k \geq 0$. It is said to be unstable if it is not stable, and it is said to be asymptotically stable if it is stable and

$$\|x_0\| < \delta(\varepsilon) \Rightarrow \lim_{k\to\infty} \|\phi(k, x_0)\| = 0. \tag{A.49}$$

Similarly, if the initial conditions x_0 can be arbitrarily chosen, then the system is said to be globally asymptotically stable.

The Lyapunov function characterization in the discrete-time case is similar to its continuous-time counterpart. Instead of derivatives we consider the differences. Let $V(x) > 0$ for all $x \neq 0$. If the following is satisfied

$$V(x(k+1)) - V(x(k)) < 0 \tag{A.50}$$

for all $x \neq 0$, along the trajectories of the system, then the system is asymptotically stable, furthermore, if $V(x)$ is radially unbounded then the system is globally asymptotically stable.

In the case of linear time-invariant discrete-time systems described by

$$x(k+1) = Ax(k) \tag{A.51}$$

we have that $x = 0$ is always an equilibrium of (A.51). Let $x(k) = \phi(k, x_0)$ represent the solution of (A.51) where $k_0 = 0$ without loss of generality.

Theorem A.15 *The following statements are equivalent*:

(a) *The equilibrium* $x = 0$ *of* (A.51) *is asymptotically stable.*
(b) *The equilibrium* $x = 0$ *of* (A.51) *is globally asymptotically stable.*
(c) $\lim_{k\to\infty} \|A^k\| = 0$.

Theorem A.16 *The equilibrium* $x = 0$ *of* (A.51) *is asymptotically stable if and only if all eigenvalues of A are within the unit circle of the complex plane, i.e.,* $|\lambda_i(A)| < 1$, $i = 1, \ldots, n$. *In this case we say that the matrix A is Schur stable.*

The equilibrium $x = 0$ of (A.51) is stable if and only if $|\lambda_i(A)| \leq 1,\ i = 1, \ldots, n$ and each eigenvalue with $|\lambda_i(A)| = 1$ has associated Jordan block of order one.

Similar to the continuous-time case, a typical choice of Lyapunov function for linear systems is a quadratic function of the form (A.43).

The time difference of (A.43) along the trajectories of the linear system (A.51) is given by:

$$\begin{aligned} V\big(x(k+1)\big) - V\big(x(k)\big) &= x(k+1)^T P x(k+1) - x(k)^T P x(k) \\ &= x(k)^T \big(A^T P A - P\big) x(k) \end{aligned} \tag{A.52}$$

which leads to the discrete-time Lyapunov equation shown next.

Theorem A.17 *The equilibrium $x = 0$ of (A.51) is asymptotically stable if and only if for any given positive definite symmetric matrix Q there exists a positive definite symmetric matrix P that satisfies*

$$A^T P A - P = -Q. \tag{A.53}$$

Equation (A.53) is called the discrete-time Lyapunov equation.

Notes and References

The contents of this section are based on several books on linear systems, control engineering, and nonlinear systems [5, 61, 76, 77, 131].

Bibliography

1. A. Anta, P. Tabuada, Self-triggered stabilization of homogeneous control systems, in *Proceedings of the American Control Conference*, 2008, pp. 4129–4134
2. A. Anta, P. Tabuada, To sample or not to sample: Self-triggered control for nonlinear systems. IEEE T. Automat. Contr. **55**(9), 2030–2042 (2010)
3. P. J. Antsaklis and J. Baillieul, Special issue on technology of networked control systems, *Proceeding of the IEEE*, vol. 95, no. 1, 2007
4. P. J. Antsaklis, J. C. Kantor, Sensitivity analysis for model based control configurations, in *Proceedings of the American Control Conference*, 1984
5. P.J. Antsaklis, A. Michel, *Linear Systems*, 1st edn. (McGraw Hill, New York, 1997)
6. D. Antunes, J. P. Hespanha, C. Silvestre, Stochastic networked control systems with dynamic protocols, in *Proceedings of the American Control Conference*, 2011, pp. 1686–1691
7. D. Antunes, J. Hespanha, C. Silvestre, Stability of networked control systems with asynchronous renewal links: An impulsive systems approach. Automatica **49**(2), 402–413 (2013)
8. J. Araujo, Design and implementation of resource-aware wireless networked control systems, Licentiate Thesis, KTH, 2011
9. J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, K. H. Johansson, Self-triggered control over wireless sensor and actuator networks, in *International Conference on Distributed Computing in Sensor Systems and Workshops*, 2011, pp. 1–9
10. K.J. Astrom, Event based control, in *Analysis and Design of Nonlinear Control Systems*, ed. by A. Astolfi, L. Marconi (Springer, Berlin, 2008), pp. 127–147
11. K. J. Astrom, B. M. Bernhardson, Comparison or Riemann and Lebesgue sampling for first order stochastic systems, in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 2011–2016
12. K.J. Astrom, P. Eykhoff, System identification: A survey. Automatica **7**(2), 123–162 (1971)
13. B. Azimi-Sadjadi, Stability of networked control systems in the presence of packet losses, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 676–681
14. J. Baillieul, Feedback coding for information-based control: operating near the data-rate limit, in *Proceedings of the 41st IEEE Conference on Decision and Control,* 2003, pp. 3229–3236
15. B. Bamieh, Intersample and finite wordlength effects in sampled-data problems, in *Proceedings of the 35th IEEE Conference on Decision and Control*, 1996, pp. 3890–3895
16. B. Bamieh, J.B. Pearson, B.A. Francis, A. Tannenbaum, A lifting technique for linear periodic systems with applications to sampled-data control. Syst. Control. Lett. **17**, 79–88 (1991)
17. L. Bao, M. Skoglund, and K. H. Johansson, Encoder-decoder design for event-triggered feedback control over bandlimited channels, in *Proceedings of the American Control Conference*, 2006, pp. 4183–4188

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8,

18. D. Bauer, M. Deistler, W. Scherrer, User choices in subspace algorithms, in *Proceedings of the 37th IEEE Conference on Decision and Control*, 1998, pp. 731–736
19. P. Bauer, M. Schitiu, C. Lorand, K. Premaratne, Total delay compensation in LAN control systems and implications for scheduling, in *Proceedings of the American Control Conference*, 2001, pp. 4300–4305
20. O. Beldiman, L. Bushnell, G. C. Walsh, H.O. Wang, Y. Hong, Perturbations in networked control systems, in *Proceedings of ASME-IMECE*, 2001
21. O. Beldiman, G. C. Walsh, L. Bushnell, Predictors for networked control systems, in *Proceedings of the American Control Conference*, 2000, pp. 2347–2351
22. O. Beldiman, G.C. Walsh, L. Bushnell, Asymptotic behavior of nonlinear networked control systems. IEEE T. Automat. Contr. **46**(7), 1093–1097 (2001)
23. A. Bemporad, M. Morari, Robust model predictive control: a survey, *in Robustness in Identification and Control,* ed. by A. Garulli, A. Tesi, A. Vicino. Lecture Notes in Control and Information Sciences, vol. 245 (Springer-Verlag, 1999) pp. 207–226
24. D. Bernardini, A. Bemporad, Energy-aware robust model predictive control based on wireless sensor feedback, in *Proceedings of 47th IEEE Conference on Decision and Control,* 2008, pp. 3342–3346
25. S.P. Bhattacharya, A.C. Del Nero Gomes, J.W. Howze, The structure of robust disturbance rejection control. IEEE T. Automat. Contr. **28**(9), 874–881 (1983)
26. S. Bittanti, P. Colaneri, Invariant representations of discrete-time periodic systems. Automatica **36**(12), 1777–1793 (2000)
27. R. Bishop, S.J. Paynter, J.W. Sunkel, Adaptive control of space station with control moment gyros. IEEE Contr. Syst. Mag. **12**, 23–28 (1992)
28. M.S. Branicky, V. Liberatore, S.M. Phillips, Networked control system co-simulation for co-design, in *Proceedings of the American Control Conference*, 2003, pp. 3341–3346
29. M. S. Branicky, S. M. Phillips, W. Zhang, Stability of networked control systems: explicit analysis of delays, *in Proceedings of the American Control Conference*, 2000, pp. 2352–2357
30. M. S. Branicky, S.M. Phillips, W. Zhang, Scheduling and feedback co-design for networked control systems, in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 1211–1216
31. R. Brockett, Minimum attention control, in *Proceedings of the 36th Conference on Decision and Control*, 1997, pp. 2628–2632
32. R.G. Brown, P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd edn. (Wiley, New York, 1992)
33. L. Bushnell, O. Beldiman, G. Walsh, An equivalence between a control network and a switched hybrid system, *International Workshop on Hybrid Systems: Computation and Control*, 1998, pp. 64–79
34. E. Camponogara, D. Jia, B.H. Krogh, S. Talukdar, Distributed model predictive control. IEEE Contr. Syst. Mag. **22**, 44–52 (2002)
35. Y. Cao, W. Ren, Sample-data formation control under dynamic directed interaction, in *Proceedings of the American Control Conference*, 2009, pp. 5186–5191
36. Y. Cao, W. Ren, Multi-vehicle coordination for double integrator dynamics under fixed undirected/directed interaction in a sampled data setting. Int. J. Robust. Nonlin. **20**, 987–1000 (2010)
37. D. Carnevale, A.R. Teel, D. Nesic, A Lyapunov proof of an improved maximum allowable transfer interval for networked control systems. IEEE T. Automat. Contr. **52**(5), 892–897 (2007)
38. C. S. Carver, M. F. Scheier, *On the self-regulation of behavior.* (Cambridge University Press, 1998)
39. B. Castillo, S. Di Genaro, S. Monaco, D. Normand-Cyrot, On regulation under sampling. IEEE T. Automat. Contr. **42**(6), 864–868 (1997)
40. A. Cervin, T. Henningson, Scheduling of event-triggered controllers on a shared network, in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, pp. 3601–3606

41. A. Chaillet, A. Bicchi, Delay compensation in packet-switching networked controlled systems, in *Proceedings of the 47th IEEE Conference on Decision and control*, 2008, pp. 3620–3625
42. T. Chen, B. Francis, *Optimal sampled data control systems*, (Springer, 1995)
43. H. Chu, S. Fei, D. Yue, C. Peng, J. Sun, H∞ quantized control for nonlinear networked control systems. Fuzzy Set. Syst. **174**(1), 99–113 (2011)
44. M. B. G. Cloosterman, Control over communication networks: modeling, analysis, and synthesis, Ph.D. dissertation, Tech.Univ. Eindhoven, Eindhoven, The Netherlands, 2008
45. M.B.G. Cloosterman, N. van de Wouw, W.P.M.H. Heemels, H. Nijmeijer, Stability of networked control systems with uncertain time-varying delays. IEEE T. Automat. Contr. **54**(7), 1575–1580 (2009)
46. M.B.G. Cloosterman, L. Hetel, N. van de Wouw, W.P.M.H. Heemels, J. Daafouz, H. Nijmeijer, Controller synthesis for networked systems. Automatica **46**, 1584–1594 (2010)
47. G. Conte, A. M. Perdon, G. Vitaioli, A model based control scheme with sampled information, in *Proceedings of the 17th Mediterranean Conference on Control and Automation,* 2009, pp. 1354–1360
48. G. Conte, A. M. Perdon, G. Vitaioli, Stabilization of nonlinear plants with sampled information, in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 4746–4752
49. O.L.V. Costa, M.D. Fragoso, Stability results for discrete-time linear systems with Markovian jumping parameters. J. Math. Anal. Appl. **179**, 154–178 (1993)
50. C. De Persis, R. Sailer, F. Wirth, On a small-gain approach to distributed event-triggered control, in *18th IFAC World Congress*, 2011, pp. 2401–2406
51. A. Dembo, O. Zeitouni, *Large Deviation Techniques and Applications* (Springer, New York, 1998)
52. O. Demir, J. Lunze, Cooperative control of multiagent systems with event-based communication, in *Proceedings of the American Control Conference*, 2012, pp. 4504–4509
53. D. V. Dimarogonas, K. H. Johansson, Event-triggered control of multi-agent systems, *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009
54. D.V. Dimarogonas, E. Frazzoli, K.H. Johansson, Distributed event-triggered control for multi-agent systems. IEEE T. Automat. Contr. **57**(5), 1291–1297 (2012)
55. F. Ding, T. Chen, Hierarchical identification of lifted state-space models for general dual-rate systems. IEEE T. Circuits. Syst. **52**, 1179–1187 (2005)
56. F. Ding, T. Chen, Combined parameter and output estimation of dual-rate systems using an auxiliary model. Automatica **40**, 1739–1748 (2004)
57. M.C.F. Donkers, W.P.M.H. Heemels, D. Bernardini, A. Bemporad, V. Shneer, Stability analysis of stochastic networked control systems. Automatica **48**(5), 917–925 (2012)
58. M. C. F. Donkers, W. P. M. H. Heemels, Output-based event-triggered control with guaranteed Linfty-gain and improved event-triggering, in *Proceedings of the 49th IEEE Conference on Decision and Control,* 2010, pp. 3246–3251
59. M.C.F. Donkers, W.P.M.H. Heemels, Output-based event-triggered control with guaranteed L∞-gain and improved and decentralized event-triggering. IEEE T. Automat. Contr. **57**(6), 1362–1376 (2012)
60. M.C.F. Donkers, W.P.M.H. Heemels, N. van de Wouw, L. Hetel, Stability analysis of networked control systems using a switched linear systems approach. IEEE T. Automat. Contr. **56**(9), 2101–2115 (2011)
61. R.C. Dorf, R.H. Bishop, *Modern Control Systems*, 11th edn. (Prentice Hall, Upper Saddle River, 2008)
62. J. Douglas, M. Athans, Robust linear quadratic designs with real parameter uncertainty. IEEE T. Automat. Contr. **39**(1), 107–111 (1994)
63. G. Dullerud, S. Lall, Asynchronous hybrid systems with jumps: Analysis and synthesis methods. Syst. Control. Lett. **37**(2), 61–69 (1999)
64. W.B. Dunbar, A distributed receding horizon control algorithm for dynamically coupled nonlinear systems. IEEE T. Automat. Contr. **52**, 1249–1263 (2007)

65. N. Elia, S.K. Mitter, Stabilization of linear systems with limited information. IEEE T. Automat. Contr. **46**(9), 1384–1400 (2001)
66. N.H. El-Farra, Y. Sun, Quasi-decentralized control of process systems using wireless sensor networks with scheduled sensor transmissions, in *Proceedings of the American Control Conference*, 2009, pp. 3390–3396
67. T. Estrada, Model-based Networked Control Systems with intermittent feedback, Ph. D. Dissertation, University of Notre Dame, Notre Dame, IN, 2009
68. T. Estrada, H. Lin, P. J. Antsaklis, Model based control with intermittent feedback, in *Proceedings of the 14th Mediterranean Conference on Control and Automation*, 2006, pp. 1–6
69. T. Estrada, P. J. Antsaklis, Stability of discrete-time plants using model-based control with intermittent feedback, in *Proceedings of the 16th Mediterranean Conference on Control and Automation,* 2008, pp. 1130–1136
70. T. Estrada, P. J. Antsaklis, Stability of model-based networked control systems with intermittent feedback, in *Proceedings of the 17th IFAC World Congress*, 2008
71. T. Estrada, P. Antsaklis, Performance of model-based networked control systems with discrete time plants, in *Proceedings of the 17th Mediterranean Conference on Control and Automation*, 2009, pp. 628–633
72. F. Fagnani, S. Zampieri, Stabilizing quantized feedback with minimal information flow: the Scalar Case, *15th International Symposium on Mathematical Theory of Networks and Systems*, 2002
73. Y. Fang, A new general sufficient condition for almost sure stability of jump linear systems. IEEE T. Automat. Contr. **42**(3), 378–382 (1997)
74. W. Favoreel, B. DeMoor, P. Van Overschee, Subspace state space system identification for industrial processes. J. Process. Contr. **10**, 149–155 (2000)
75. B.A. Francis, W.M. Wonham, The internal model principle of control theory. Automatica **12**, 457–465 (1976)
76. G.F. Franklin, J.D. Powell, M. Workman, *Digital Control of Dynamic Systems*, 3rd edn. (Addison Wesley Longman, Inc., Menlo Park, 1998)
77. B. Friedland, *Control system design: an introduction to state-space methods*, Dover ed., 2005
78. E. Fridman, A. Seuret, J.P. Richard, Robust sampled-data stabilization of linear systems: An input delay approach. Automatica **40**(8), 1441–1446 (2004)
79. M. Fu, Robust stabilization of linear uncertain systems via quantized feedback, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 199–203
80. K. Furuta, M. Iwase, S. Hatakeyama, Internal model and saturating actuation in human operation from human adaptive mechatronics. IEEE T. Ind. Electron. **52**(5), 1236–1245 (2005)
81. C.E. Garcia, D.M. Prett, M. Morari, Model predictive control: Theory and practice—a survey. Automatica **25**(3), 335–348 (1989)
82. E. Garcia, P. J. Antsaklis, Model-based control using a lifting approach, in *Proceedings of the 18th Mediterranean Conference on Control and Automation*, 2010, pp. 105–110.
83. E. Garcia and P. J. Antsaklis, Adaptive stabilization of model-based networked control systems, in *Proceedings of the American Control Conference*, 2011, pp. 1094–1099
84. E. Garcia, P. J. Antsaklis, Model-based event-triggered control with time-varying network delays, in *Proceedings of the 50th IEEE Conference on Decision and Control - European Control Conference,* 2011, pp. 1650–1655
85. E. Garcia, P. J. Antsaklis, Decentralized model-based event-triggered control of networked systems, in *Proceedings of the American Control Conference*, 2012, pp. 6485–6490
86. E. Garcia, P. J. Antsaklis, Model-based control of continuous-time and discrete-time systems with large network induced delays, in *Proceedings of the 20th Mediterranean Conference on Control and Automation,* 2012, pp. 1129–1134
87. E. Garcia, P.J. Antsaklis, Model-based event-triggered control for systems with quantization and time-varying network delays. IEEE T. Automat. Contr. **58**(2), 422–434 (2013)

88. E. Garcia, P.J. Antsaklis, Parameter estimation and adaptive stabilization in time-triggered and event-triggered model-based control of uncertain systems. Int. J. Control. **85**(9), 1327–1342 (2012)
89. E. Garcia, P. J. Antsaklis, Output feedback model-based control of uncertain discrete-time systems with network induced delays, in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 6647–6652
90. E. Garcia, P.J. Antsaklis, Model-based control of networked distributed systems with multirate state feedback updates. Int. J. Control. **86**(9), 1503–1517 (2013)
91. E. Garcia, P. J. Antsaklis, Model-based control of continuous-time systems with limited intermittent feedback, in *Proceedings of the 21st Mediterranean Conference on Control and Automation,* 2013, pp. 452–457
92. E. Garcia, P.J. Antsaklis, Output feedback networked control with persistent disturbance attenuation. Syst. Control. Lett. **62**, 943–948 (2013)
93. E. Garcia, G. Vitaioli, P. J. Antsaklis, Model-based tracking control over networks, in *Proceedings of the 2011 I.E. International Conference on Control Applications*, 2011, pp. 1226–1231
94. E. Garcia, P. J. Antsaklis, Optimal model-based control with limited communication, in *19th IFAC World Congress*, 2014
95. E. Garcia, Y. Cao, H. Yu, P.J. Antsaklis, D.W. Casbeer, Decentralised event-triggered cooperative control with limited communication. Int. J. Control. **86**(9), 1479–1488 (2013)
96. E. Garcia, Y. Cao, D. W. Casbeer, Cooperative control with general linear dynamics and limited communication: centralized and decentralized event-triggered control strategies, in *American Control Conference*, 2014
97. H. Gao, T. Chen, Network-based H-inf output tracking control. IEEE T. Automat. Contr. **53** (3), 655–667 (2008)
98. P. J. Gawthrop, L. Wang Intermittent model predictive control, *Proceedings of the Institution of Mechanical Engineers, Pt. I, Journal of Systems and Control Engineering*, 2007
99. P.J. Gawthrop, L. Wang, Event-driven intermittent control. Int. J. Control. **82**(12), 2235–2248 (2009)
100. D. Georgiev, D. M. Tilbury Packet-based control in *Proceedings of the American Control Conference,* 2004, pp. 329–336
101. G.C. Goodwin, D.E. Quevedo, E.I. Silva, Architectures and coder design for networked control systems. Automatica **44**, 248–257 (2008)
102. M. Guinaldo, D. Lehmann, J. Sánchez, S. Dormido, K.H. Johansson, Distributed event-triggered control with network delays and packet losses, in *Proceedings of the 51st IEEE Conference on Decision and Control,* 2012, pp. 1–6
103. M. Guinaldo, D. V. Dimarogonas, K. H. Johansson, J. Sanchez, and S. Dormido, Distributed event-based control for interconnected linear systems, in *Proceedings of the* 50th *IEEE Conference on Decision and Control and European Control Conference* (CDC-ECC), 2011, pp. 2553–2558
104. M. Guinaldo, D.V. Dimarogonas, K.H. Johansson, J. Sánchez, S. Dormido, Distributed event-based control strategies for interconnected linear systems. IEE P-Contr. Theor. Ap. **7**(6), 877–886 (2013)
105. W.P.M.H. Heemels, M.C.F. Donkers, Model-based periodic event-triggered control for linear systems. Automatica **49**(3), 698–711 (2013)
106. W.P.M.H. Heemels, M.C.F. Donkers, A.R. Teel, Periodic event-triggered control for linear systems. IEEE T. Automat. Contr. **58**(4), 847–861 (2013)
107. W. P. M. H. Heemels, M. C. F. Donkers, A. R. Teel, Periodic event-triggered control based on state feedback, in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 2571–2576
108. W.P.M.H. Heemels, J.H. Sandee, P.P.J. van den Bosch, Analysis of event-driven controllers for linear systems. Int. J. Control. **81**(4), 571–590 (2008)

109. W.P.M.H. Heemels, A.R. Teel, N. van de Wouw, D. Nesic, Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. IEEE T. Automat. Contr. **55**(8), 1781–1796 (2010)
110. W. P. M. H. Heemels, N. van de Wouw, R. H. Gielen, M. C. F. Donkers, L. Hetel, S. Olaru, M. Lazar, J. Daafouz, S. Niculescu, Comparison of overapproximation methods for stability analysis of networked control systems, in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, 2010, pp. 181–190
111. J. P. Hespanha, A. Ortega, L. Vasudevan, Towards the control of linear systems with minimum bit-rate, in *Proceedings of the Int. Symposium on the Mathematical Theory of Networks and Sys.,* 2002
112. J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in Networked Control Systems. P. IEEE **95**(1), 138–162 (2007)
113. L. Hetel, J. Daafouz, C. Iung, Analysis and control of LTI and switched systems in digital loops via an event-based modeling. Int. J. Control. **81**(7), 1125–1138 (2008)
114. D.J. Hill, P.J. Moylan, The stability of nonlinear dissipative systems. IEEE T. Automat. Contr. **21**(5), 708–711 (1976)
115. D.J. Hill, P.J. Moylan, Stability results of nonlinear feedback systems. Automatica **13**(4), 377–382 (1977)
116. P. F. Hokayem, C. T. Abdallah, Inherent issues in networked control systems: a survey, in *Proceedings of the American Control Conference*, 2004, pp. 329–336
117. D. Hristu-Varsakelis, Feedback control systems as users of a shared network: communication sequences that guarantee stability, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 3631–3636
118. L.S. Hu, J. Lam, Y.Y. Cao, H.H. Shao, A linear matrix inequality (LMI) approach to robust H 2 sampled-data control for linear uncertain systems. IEEE Trans. Syst. Man Cybern. B Cybern. **33**(1), 149–155 (2003)
119. M. Huang, S. Dey, Stability of Kalman filtering with markovian packet losses. Automatica **43** (4), 598–607 (2007)
120. O. C. Imer, T. Basar, Optimal estimation with limited measurements, in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference,* 2005, pp. 1029–1034
121. O. C. Imer, T. Basar, Optimal control with limited controls, in *Proceedings of the American Control Conference,* 2006, pp. 298–303
122. P. Ioannou, B. Fidan, *Adaptive Control Tutorial* (Society for Industrial and Applied Mathematics, Philadelphia, 2006)
123. H. Ishii, B. Francis, Stabilization with control networks. Automatica **38**(10), 1745–1751 (2002)
124. H. Ishii, B. Francis, Stabilizing a linear system by switching control and output feedback with dwell time, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 191–196
125. Y. Ishido, K. Takaba, D.E. Quevedo, Stability analysis of networked control systems subject to packet-dropouts and finite-level quantization. Syst. Control. Lett. **60**(5), 325–332 (2011)
126. M. Iwase, S. Hatakeyama, K. Furuta. Analysis of intermittent control systems by human operations, *IECON 2006, 32nd Annual Conference on IEEE Industrial Electronics*, 2006, pp. 4516–1521
127. M. Ji, M. Egerstedt, Distributed coordination control of multiagent systems while preserving connectedness. IEEE T. Robotic. **23**(4), 693–703 (2007)
128. Z.P. Jiang, Y. Wang, Input-to-state stability for discrete-time nonlinear systems. Automatica **37**, 857–869 (2001)
129. C.R. Johnson, *Lectures on Adaptive Parameter Estimation* (Prentice Hall, Englewood Cliffs, 1988)
130. C. A. Kahane, L. B. Mirkin, J. Z. Palmor, Discrete-time lifting via implicit descriptor systems, *in Proceedings of the 5th European Control Conference*, 1999
131. H.K. Khalil, *Nonlinear Systems*, 3rd edn. (Prentice-Hall, Upper Saddle River, 2002)

132. D.S. Kim, Y.S. Lee, W.H. Kwon, H.S. Park, Maximum allowable delay bounds of networked control systems. Control. Eng. Pract. **11**(11), 1301–1313 (2003)
133. M. Kim, Controlling chemical turbulence by global delayed feedback: Pattern formation in catalytic co oxidation on pt(110). Science **292**(5520), 1357–1360 (2001)
134. K. Koay, G. Bugmann, Compensating intermittent delayed visual feedback inrobot navigation, *Proceedings of the IEEE Conference on Decision and Control Including The Symposium on Adaptive Processes*, 2004
135. T. Kobayashi, D.L. Simon, Application of a bank of Kalman filters for aircraft engine fault diagnosis, *ASME Turbo Expo,* Atlanta, paper GT2003-38550, 2003
136. F. Kozin, A survey of stability of stochastic systems. Automatica **5**, 95–112 (1968)
137. P.R. Kumar, P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control* (Prentice Hall, Englewood Cliffs, 1986)
138. G. Kreisselmeier, Adaptive observers with exponential rate of convergence. IEEE T. Automat. Contr. **AC-22**(1), 2–8 (1977)
139. J.H. Lee, M.S. Gelormino, M. Morari, Model predictive control of multirate sample-data systems: a state approach. Int. J. Control. **55**, 153–191 (1992)
140. D. Lehmann, J. Lunze, Event-based control using quantized state information, in *Proceedings of IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2010
141. D. Lehmann, J. Lunze, Event-based control with communication delays, in *Proceedings of IFAC World Congress*, 2011, pp. 3262–3267
142. D. Lehmann, J. Lunze, Event-based output feedback control, in *Proceedings of the 19th Mediterranean Conference on Control and Automation,* 2011, pp. 982–987
143. E. Lenneberg, *Biological Foundations of Language*. Wiley, 4th ed., 1967
144. N. Leonard, P. Krishnaprasad, Averaging for attitude control and motion planning, *in Proceedings of the 32nd IEEE Conference on Decision and Control*, 1993, pp. 3098–3104
145. D. Li, S.L. Shah, T. Chen, Analysis of dual-rate inferential control systems. Automatica **38**, 1053–1059 (2002)
146. L. Li, M. Lemmon, Event-triggered output feedback control of finite horizon discrete-time multi-dimensional linear processes, in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 3221–3226
147. F.L. Lian, J.R. Moyne, D.M. Tilbury, Performance evaluation of control networks: Ethernet, controlnet, and devicenet. IEEE Contr. Syst. Mag. **21**(1), 66–83 (2001)
148. F. L. Lian, J. R. Moyne, D. M. Tilbury, Time delay modeling and sample time selection for networked control systems, in *Proceedings of the ASME Dynamic Systems and Control Division,* 2001
149. F.L. Lian, J.M. Moyne, D.M. Tilbury, Network design consideration for distributed control systems. IEEE T. Contr. Syst. T. **10**(2), 297–307 (2002)
150. D. Liberzon, R. Brockett, Quantized feedback stabilization of linear systems. IEEE T. Automat. Contr. **45**(7), 1279–1289 (2000)
151. D. Liberzon, Hybrid feedback stabilization of systems with quantized signals. Automatica **39**, 1543–1554 (2003)
152. D. Liberzon, On stabilization of linear systems with limited information. IEEE T. Automat. Contr. **48**(2), 304–307 (2003)
153. D. Liberzon, Stabilizing a nonlinear system with limited information feedback, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 182–186
154. F. Lin, A. W. Olbrot, An LQR approach to robust control of linear systems with uncertain parameters, in *Proceedings of the 35th IEEE Conference on Decision and Control,* 1996, pp. 4158–4163
155. H. Lin, G. Zhai, P.J. Antsaklis, Robust stability and disturbance attenuation of a class of networked control systems, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 1182–1187

156. Q. Ling, M.D. Lemmon, Robust performance of soft real-time networked control systems with data dropouts, in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 1225–1230
157. Q. Ling, M.D. Lemmon, Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain, in *Proceedings of the American Control Conference*, 2003, pp. 4845–4850
158. Q. Ling, M.D. Lemmon, Optimal dropout compensation in networked control systems, in *Proceedings of the IEEE Conference on Decision and Control*, 2003, pp. 670–675
159. Q. Ling, M.D. Lemmon, Stability of quantized control systems under dynamic bit assignment, in *Proceedings of the 2004 American Control Conference*, 2004, pp. 4915–4920
160. X. Liu, A model based approach to nonlinear networked control systems, PhD thesis, University of Alberta, 2009
161. X. Liu, A. Goldsmith, Kalman filtering with partial observation losses, in *Proceedings of the IEEE Conference on Decision and Control*, 2004, pp. 4180–4186
162. L. Ljung, Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. IEEE T. Automat. Contr. **AC-24**(1), 36–50 (1979)
163. L. Ljung, T. Soderstrom, *Theory and Practice of Recursive Identification* (The MIT Press, Cambridge, 1983)
164. J. Lu, L. J. Brown, Stability analysis of a proportional with intermittent integral control system, in *Proceedings of the American Control Conference*, 2010, pp. 3257–3262
165. J. Lunze, D. Lehmann, A state-feedback approach to event-based control. Automatica **46**(1), 211–215 (2010)
166. S.E. Lyshevski, State space model identification of deterministic nonlinear systems: nonlinear mapping technology and application of the Lyapunov theory. Automatica **34**(5), 659–664 (1998)
167. W. MacKunis, J. W. Curtis, P. E. K. Berg-Yuen, Optimal estimation of multidimensional data with limited measurements, in *Proceedings of the American Control Conference,* 2011, pp. 4257–4262
168. D.T. Magill, Optimal adaptive estimation of sampled stochastic processes. IEEE T. Automat. Contr. **AC-10**(4), 434–439 (1965)
169. Z. Mao, B. Jiang, P. Shi, H∞ fault detection filter design for networked control systems modelled by discrete Markovian jump systems, *IET Control Theory & Applications*, vol. 5, 2007, pp. 1336–1343
170. M. Mariton, *Jump Linear Systems in Automatic Control* (Marcel Dekker, New York, 1990)
171. M.J. Mataric, Reinforcement learning in the multi-robot domain. Auton. Robot. **4**, 77–83 (1997)
172. T. Matiakis, S. Hirche, M. Buss, Independent-of-delay stability of nonlinear networked control systems by scattering transformation, *in Proceedings of the American Control Conference*, 2006, pp. 2801–2806
173. A. S. Matveev, A.V. Savkin, Optimal state estimation in networked systems with asynchronous communication channels and switched sensors, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 825–830
174. A. S. Matveev, A.V. Savkin, Optimal control of networked systems via asynchronous communication channels with irregular delays, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 2323–2332
175. M. Mazo, A. Anta, P. Tabuada, An ISS self-triggered implementation of linear controllers. Automatica **46**(8), 1310–1314 (2010)
176. M. Mazo, P. Tabuada, On event-triggered and self-triggered control over sensor/actuator networks, in *Proceedings of the 47th IEEE Conference on Decision and Control,* 2008, pp. 435–440
177. M. Mazo, P. Tabuada, Decentralized event-triggered control over wireless sensor/actuator networks. IEEE T. Automat. Contr. **56**(10), 2456–2461 (2011)

178. M. Mazo, M. Cao. Decentralized event-triggered control with asynchronous updates. in *Proceedings of the 50th IEEE Conference on Decision and Control,* 2011, pp. 2547–2552
179. M.J. McCourt, E. Garcia, P.J. Antsaklis, Model-based event-triggered control of nonlinear dissipative systems, in *American Control Conference*, 2014
180. L. Mirkin, Some remarks on the use of time-varying delay to model sample-and-hold circuits. IEEE T. Automat. Contr. **52**(6), 1109–1112 (2007)
181. L. Mirkin, Z. Palmor, A new representation of the parameters of lifted systems. IEEE T. Automat. Contr. **34**(4), 833–840 (1999)
182. M. Miskowicz, Send-On-Delta concept: an event-based data reporting strategy. Sensors **6**, 49–63 (2006)
183. P. Misra, LQR design with prescribed damping and degree of stability, in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, 1996, pp. 68–70
184. A. Molin and S. Hirche, On LQG joint optimal scheduling and control under communication constraints, in *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 5832-5838.
185. L.A. Montestruque, Model-based networked control systems, Ph. D. Dissertation, University of Notre Dame, Notre Dame, IN, 2004
186. L.A. Montestruque, P.J. Antsaklis, Model-based networked control systems: necessary and sufficient conditions for stability, in *Proceedings of the 10th Mediterranean Conference on Control and Automation*, 2002
187. L.A. Montestruque, P.J. Antsaklis, State and output feedback control in model-based networked control systems, in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, pp. 1620–1625
188. L.A. Montestruque, P.J. Antsaklis, On the model-based control of networked systems. Automatica **39**(10), 1837–1843 (2003)
189. L.A. Montestruque, P.J. Antsaklis, Stability of model-based control of networked systems with time varying transmission times. IEEE T. Automat. Contr. **49**(9), 1562–1572 (2004)
190. L.A. Montestruque, P.J. Antsaklis, Performance evaluation for model-based networked control systems. *Networked Embedded Sensing and Control*. (Springer, Berlin, 2006), pp. 231–249
191. L.A. Montestruque, P.J. Antsaklis, Static and dynamic quantization in model-based networked control systems. Int. J. Control. **80**(1), 87–101 (2007)
192. M. Morari, J.H. Lee, Model predictive control: past, present and future. Comput. Chem. Eng. **23**(4–5), 667–682 (1999)
193. M. Morari, E. Zafiriou, *Robust Process Control* (Prentice Hall, Englewood Cliffs, 1989)
194. L. Moreau, Stability of continuous-time distributed consensus algorithms, in *IEEE Conference on Decision and Control*, 2004, pp. 3998–4003
195. J.M. Moyne, D.M. Tilbury, The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. P. IEEE **95**(1), 29–47 (2007)
196. S. Mu, T. Chu, L. Wang, An improved model-based control scheme for networked systems, *IEEE International Conference on Systems, Man and Cybernetics*, 2004
197. D. Munoz de la Pena, P. Christofides, Model based control of nonlinear systems subject to sensor data losses: a chemical process case study, in *Proceedings of the 46th IEEE Conference on Decision and Control,* 2007, pp. 3333–3338
198. P. Naghshtabrizi, J.P. Hespanha, A.R. Teel, Exponential stability of impulsive systems with application to uncertain sampled-data systems. Syst. Control. Lett. **57**(5), 378–385 (2008)
199. G. Nair, R. Evans, Communication-limited stabilization of linear systems in *Proceedings of the 39th IEEE Conference on Decision and Control,* 2000, pp.1005–1010
200. G. Nair, R. Evans, Stabilization with data-rate-limited feedback: tightest attainable bounds. Syst. Control. Lett. **41**, 49–56 (2000)
201. G. Nair, S. Dey, R. Evans, Infimum data rates for stabilizing Markov jump linear systems, in *Proceedings of the 42nd IEEE Conference on Decision and Control,* 2003, pp. 1176–1181

202. D. Nesic, D. Liberzon, A unified framework for design and analysis of networked and quantized control systems. IEEE T. Automat. Contr. **54**, 732–747 (2009)
203. D. Nesic, A. Teel, Input to state stability of networked control systems. Automatica **40**(12), 2121–2128 (2004)
204. L.W. Nelson, E. Stear, The simultaneous on-line estimation of parameters and states in linear systems. IEEE T. Automat. Contr. **AC-21**(1), 94–98 (1976)
205. C. Nowzari, J. Cortes, Self-triggered coordination of robotic networks for optimal deployment. Automatica **48**(6), 1077–1087 (2012)
206. D. Oldroyd, L. Goldblatt, Effect of uniform versus intermittent product withdrawal from distillation columns, *Industrial and Engineering Chemistry*, **12**, (1946)
207. L. Orihuela, F. Rubio, F. Gomez-Estern, Model-based networked control systems under parametric uncertainties, *IEEE Int. Conference on control Applications*, 2009
208. P. G. Otanez, J.R. Moyne, D.M. Tilbury Using deadbands to reduce communication in Networked Control Systems in *Proceedings of the American Control Conference*, 2002, pp. 3015–3020
209. I.R. Petersen, A.V. Savkin, Multi-rate stabilization of multivariable discrete-time linear systems via a limited capacity communication channel, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 304–309
210. I.G. Polushin, P.X. Liu, C.H. Lung, On the model-based approach to nonlinear networked control systems. Automatica **44**(9), 2409–2414 (2008)
211. D.E. Quevedo, K.H. Johansson, A. Ahlén, I. Jurado, Adaptive controller placement for wireless sensor–actuator networks with erasure channels. Automatica **49**(11), 3458–3466 (2013)
212. D.E. Quevedo, E.I. Silva, G.C. Goodwin, Packetized predictive control over erasure channels, in *Proceedings of the American Control Conference*, 2007, pp. 1003–1008
213. F. Rasool, D. Huang, S.K. Nguang, Robust H∞ output feedback control of networked control systems with multiple quantizers. J. Franklin Inst. **349**(3), 1153–1173 (2012)
214. B. Recht, R. DAndrea, Exploiting symmetry for the distributed control of spatially interconnected systems, in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 598–603
215. H. Rehbinder, M. Sanfridson, Scheduling of a limited communication channel for optimal control, in *Proceedings of the Conference on Decision and Control*, 2000, pp. 1011–1016
216. W. Ren, R.W. Beard, E.M. Atkins, Information consensus in multivehicle cooperative control. IEEE Contr. Syst. Mag. **27**(2), 71–82 (2007)
217. J. Richalet, A. Rault, J.L. Testud, J. Papon, Model predictive heuristic control: applications to industrial processes. Automatica **14**, 413–428 (1978)
218. T. L. Richard A. Schmidt, *Motor Control and Learning: A Behavioral Emphasis*. 4th ed., (Human Kinetics, 2005)
219. E. Ronco, T. Arsan, P. J. Gawthrop, Open-loop intermittent feedback control: practical continuous-time GPC, *IEE Proceedings-Control Theory and Applications*, 1999
220. E. Ronco, D. J. Hill, Open-loop intermittent feedback optimal predictive control: a human movement control model, NIPS99, 1999
221. R.S.R. Ronsse, P. Lefevre, Sensorless stabilization of bounce juggling. IEEE T. Robotic. **2** (1), 147–159 (2006)
222. A. Sala, Computer control under time-varying sampling period: An LMI gridding approach. Automatica **41**(12), 2077–2082 (2005)
223. B.H. Salzberg, A.J. Wheeler, L.T. Devar, B.L. Hopkins, The effect of intermittent feedback and intermittent contingent access to play on printing of kindergarten children. J. Appl. Behav. Anal. **4**(3), 163–171 (1971)
224. N.R. Sandell, P. Varaiya, M. Athans, M.G. Safonov, Survey of decentralized control methods for large scale systems. IEEE T. Automat. Contr. **AC-23**(2), 108–128 (1978)
225. A.V. Savkin, T.M. Cheng, Detectability and output feedback stabilizability of nonlinear networked control systems. IEEE T. Automat. Contr. **52**(4), 730–735 (2007)

226. R. Scattolini, N. Schiavoni, A multirate model-based predictive controller. IEEE T. Automat. Contr. **40**, 1093–1097 (1995)
227. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, S. Sastry, Foundations of control and estimation over lossy networks. P. IEEE **95**(1), 163–187 (2007)
228. S. Seyboth, D.V. Dimarogonas, K.H. Johansson, Event-based broadcasting for multi-agent average consensus. Automatica **49**(1), 245–252 (2013)
229. Y. Shi, B. Yu, Output feedback stabilization of networked control systems with random delays modeled by Markov chains. IEEE T. Automat. Contr. **54**(7), 1668–1674 (2009)
230. H. Shousong, Z. Qixin, Stochastic optimal control and analysis of stability of networked control systems with long delay. Automatica **39**(11), 1877–1884 (2003)
231. D. Simon, *Optimal State Estimation* (Wiley, Hoboken, 2006)
232. B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, S. Sastry, Kalman filtering with intermittent observations. IEEE T. Automat. Contr. **49**(9), 1453–1464 (2004)
233. B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, S. Sastry, Distributed control applications within sensor networks. P. IEEE **91**(8), 1235–1246 (2003)
234. R. Sipahi, S.I. Niculescu, C.T. Abdallah, W. Michiels, K. Gu, Stability and stabilization of systems with delays: limitations and opportunities. IEEE Contr. Syst. Mag. **31**(1), 38–65 (2011)
235. B. Skinner, *About behaviorism.* Vintage, 1st ed., 1974
236. Y. Song, J.W. Grizzle, The extended Kalman filter as a local asymptotic observer for discrete-time nonlinear systems. J. Math. Syst. Estimat. Contr. **5**(1), 59–78 (1995)
237. G. Stewart, *Introduction to Matrix Computations*, 1st edn. (Academic Press, New York, 1973)
238. C. Stocker, D. Vey, J. Lunze, Decentralized event-based control: Stability analysis and experimental evaluation. Nonlinear Analysis: Hybrid Systems **10**, 141–155 (2013)
239. A. Stubbs, G.E. Dullerud, Networked control of distributed systems, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 203–204
240. Y. Sun, N.H. El-Farra, Quasi-decentralized model-based networked control of process systems. Comput. Chem. Eng. **32**, 2016–2029 (2008)
241. Y. Sun, N. H. El-Farra, Quasi-decentralized networked process control using an adaptive communication policy, in *Proceedings of the American Control Conference,* 2010
242. Y. Sun, N.H. El-Farra, Integrating control and scheduling of distributed energy resources over networks. Chem. Eng. Sci. **67**, 80–91 (2012)
243. Y. Sun, N.H. El-Farra, Resource aware quasi-decentralized control of networked process systems over wireless sensor networks. Chem. Eng. Sci. **69**, 93–106 (2012)
244. V. Suplin, E. Fridman, U. Shaked, Sampled-data H∞ control and filtering: Nonuniform uncertain sampling. Automatica **43**(6), 1072–1083 (2007)
245. M. Tabbara, D. Nesic, A.R. Teel, Stability of wireless and wireline networked control systems. IEEE T. Automat. Contr. **52**(9), 1615–1630 (2007)
246. P. Tabuada, X. Wang, Preliminary results on state-triggered scheduling of stabilizing control tasks, in *Proceedings of the 45th IEEE Conference on Decision and Control,* 2006, pp. 282–287
247. P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks. IEEE T. Automat. Contr. **52**(9), 1680–1685 (2007)
248. A. H. Tahoun, H. Fang, Adaptive stabilization of networked control systems with delays and state-dependent disturbances, in *Proceedings of 17th Mediterranean Conference on Control and Automation*, 2009, pp. 622–627
249. N.F. Thornhill, B. Huang, S.L. Shah, Controller performance assessment in set point tracking and regulatory control. Int. J. Adapt. Control. **17**, 709–727 (2003)
250. P. Tallapragada, N. Chopra, Event-triggered dynamic output feedback control for LTI systems in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 6597–6602
251. D. Tolic, R. Fierro, Stability of feedback linearization under intermittent information: a target-pursuit case, in *Proceedings of the American Control Conference,* 2011, pp. 3184–3190

252. K. Tsumura, H. Ishii, H. Hoshina, Tradeoffs between quantization and packet loss in networked control of linear systems. Automatica **45**(12), 2963–2970 (2009)
253. R. Vadigepalli, F.J. Doyle, A distributed state estimation and control algorithm for plantwide processes. IEEE T. Contr. Syst. T. **11**, 119–127 (2003)
254. R. van der Merwe, E. A. Wan, The square root unscented Kalman filter for state and parameter estimation, *International Conference on Acoustics, Speech and Signal Processing*, 2001
255. P. Varutti, B. Kern, T. Foulwasser, R. Findeisen, Event based model predictive control for networked control systems, *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference,* 2009, pp. 567–572
256. M. Velasco, P. Marti J.M. Fuertes The self triggered task model for real-time control systems, *24th IEEE Real Time Systems Symposium*, 2003
257. G. C. Walsh, O. Beldiman, L. Bushnell, Asymptotic behavior of networked control systems, in *Proceedings of the International Conference on Control Applications*, 1999
258. G. C. Walsh, O. Beldiman, L. Bushnell, Error encoding algorithms for networked control systems, in *Proceedings of the IEEE Conference on Decision and Control*, 1999, pp. 4933–4939
259. G.C. Walsh, H. Ye, Scheduling of networked control systems. IEEE Contr. Syst. Mag. **21**(1), 57–65 (2001)
260. G.C. Walsh, H. Ye, L. Bushnell, Stability analysis of networked control systems. IEEE T. Contr. Syst. T. **10**(3), 438–446 (2002)
261. X. Wang, M. D. Lemmon, Event triggered broadcasting across distributed networked control systems, in *Proceedings of the American Control Conference,* 2008, pp. 3139–3144
262. X. Wang and M. D. Lemmon, Decentralized event-triggered broadcasts over networked control systems, in *Hybrid Systems: computation and control*, (Springer, Berlin, 2008), pp. 674–677
263. X. Wang, M.D. Lemmon, Self-triggered feedback control systems with finite-gain stability. IEEE T. Automat. Contr. **54**(3), 452–467 (2009)
264. X. Wang, M. D. Lemmon, Event design in event-triggered feedback control systems, in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, pp. 2105–2110
265. X. Wang, M.D. Lemmon, Event-triggering in distributed networked control systems. IEEE T. Automat. Contr. **56**, 586–601 (2011)
266. Z. Wang, B. Shen, H. Shu, G. Wei, Quantized H-infinity control for nonlinear stochastic time-delay systems with missing measurements. IEEE T. Automat. Contr. **57**(6), 1431–1444 (2012)
267. J. Warland, A. Dimakis, *Random Processes in Systems—Lecture Notes* (Department of Electrical Engineering and Computer Sciences, Berkeley, 2006)
268. J.C. Willems, Dissipative dynamical systems, Part I: General theory. Arch. Ration. Mech. An. **45**(5), 321–351 (1972)
269. J.C. Willems, Dissipative dynamical systems Part II: Linear systems with quadratic supply rates. Arch. Ration. Mech. An. **45**(5), 352–393 (1972)
270. S. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*, 2nd edn. (Prentice Hall, Upper Saddle River, 1994)
271. N.V. Wouw, P. Naghshtabrizi, M. Cloosterman, J.P. Hespanha, Tracking control for sampled-data systems with uncertain time-varying sampling intervals and delays. Int. J. Robust. Nonlin. **20**(4), 387–411 (2010)
272. L. Xiao, M. Johansson, H. Hindi, S. Boyd, A. Goldsmith, Joint optimization of communication rates and linear systems, in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 2321–2326
273. J. Xiong, J. Lam, Stabilization of linear systems over networks with bounded packet loss. Automatica **43**(1), 80–87 (2007)
274. Y. Xu, J. Hespanha, Optimal communication logics in networked control systems, in *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004, pp. 3527–3532
275. R. Yang, P. Shi, G.P. Liu, H. Gao, Network-based feedback control for systems with mixed delays based on quantization and dropout compensation. Automatica **47**(12), 2805–2809 (2011)

276. F. Yang, Z. Wang, Y.S. Hung, M. Gani, H< sub>∞</sub> control for networked systems with random communication delays. IEEE T. Automat. Contr. **51**(3), 511–518 (2006)
277. H. Ye, A.N. Michel, L. Hou, Stability analysis of systems with impulse effects. IEEE T. Automat. Contr. **43**(12), 1719–1723 (1998)
278. J.K. Yook, D.M. Tilbury, N.R. Soparkar, Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. IEEE T. Contr. Syst. T. **10**(4), 503–518 (2002)
279. H. Yu, Adaptive model-based control of networked systems, private communication
280. H. Yu, P. J. Antsaklis, Event-triggered output feedback control for networked control systems using passivity: time-varying network induced delays, in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 205–210
281. H. Yu, P. J. Antsaklis, Quantized output synchronization of networked passive systems with event-driven communication, in *Proceedings of the American Control Conference*, 2012, pp. 5706–5711
282. H. Yu, E. Garcia, P. J. Antsaklis, Model-based scheduling for networked control systems, in *Proceedings of the American Control Conference*, 2013, pp. 2350–2355
283. H. Yu, Z. Wang, Y. Zheng, On the model-based networked control for singularly perturbed systems. J. Contr. Theor. Appl. **6**(2), 153–162 (2008)
284. M. Yu, L. Wang, T. Chu, G. Xie, Stabilization of networked control systems with data packet dropout and network delays via switching system approach, in *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004, pp. 3539–3544
285. D.W. Yu, D.L. Yu, Multirate model predictive control of a chemical reactor based on three neural models. Biochem. Eng. J. **37**, 86–97 (2007)
286. D. Yue, Q.L. Han, C. Peng, State feedback controller design of networked control systems. IEEE T. Circuits. Syst. II **51**(11), 640–644 (2004)
287. S. Yuksel, H. Hindi, L. Crawford, Optimal tracking with feedback-feedforward control separation over a network, *in Proceedings of the American Control Conference*, 2006
288. G. Zhang, X. Chen, T. Chen, A model predictive control approach to networked systems, *in Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 3339–3344
289. L. Zhang, D. Hristu-Varsakelis, LQG control under limited communication, in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference,* 2005, pp. 185–190
290. J. Zhang, K.H. Johansson, J. Lygeros, S. Sastry, Zeno hybrid systems. Int. J. Robust. Nonlin. **11**(5), 435–451 (2001)
291. W. Zhang, M.S. Branicky, S.M. Phillips, Stability of networked control systems. IEEE Contr. Syst. Mag. **21**(1), 84–99 (2001)
292. W.A. Zhang, L. Yu, Output feedback stabilization of networked control systems with packet dropouts. IEEE T. Automat. Contr. **52**(9), 1705–1710 (2007)
293. Y.-B. Zhao, J. Kim, G.-P. Liu, Error bounded sensing for packet-based networked control systems. IEEE T. Ind. Electron. **58**(5), 1980–1989 (2011)
294. P.V. Zhivoglyadov, R.H. Middleton, Networked control design for linear systems. Automatica **39**, 743–750 (2003)
295. F. Zhu, P. J. Antsaklis, Switched systems using intermittent feedback control: modeling and stability analysis, ISIS Technical Report, ISIS-2013-010, 2013

Index

E. Garcia et al., *Model-Based Control of Networked Systems*, Systems & Control: Foundations & Applications, DOI 10.1007/978-3-319-07803-8,