Yun-Bo Zhao · Guo-Ping Liu
Yu Kang · Li Yu

Packet-Based Control for Networked Control Systems

A Co-Design Approach

Science Press
Beijing

Springer

Packet-Based Control for Networked Control Systems

Yun-Bo Zhao · Guo-Ping Liu
Yu Kang · Li Yu

Packet-Based Control for Networked Control Systems

A Co-Design Approach

Science Press
Beijing

Springer

Yun-Bo Zhao
Zhejiang University of Technology
Hangzhou
China

Guo-Ping Liu
University of South Wales
Pontypridd
UK

Yu Kang
University of Science and Technology of China
Hefei
China

Li Yu
Zhejiang University of Technology
Hangzhou
China

ISBN 978-981-10-6249-0 ISBN 978-981-10-6250-6 (eBook)
DOI 10.1007/978-981-10-6250-6

Jointly published with Science Press, Beijing, China.
ISBN: 978-7-03-053969-4 Science Press, Beijing, China.

The printed edition is not for sale in China Mainland. Customers from China Mainland please order the print book from Science Press, Beijing, China.

Library of Congress Control Number: 2017951147

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

Networked control systems (NCSs) are control systems whose control links are closed via some form of communication networks. It has become a useful control system model in recent years due to the fast development of the embedded computational devices and the communication technology. These developments have made it possible that a large number of sensors, actuators and controllers can be interconnected over the communication network to interact with the physical environment. This remote and distributed control system structure is the basis of a great many of future applications in information technology, including Internet of Things, cyber-physical systems, smart home.

NCSs can contain a large number of control devices interconnected, and data is exchanged through communication networks which inevitably introduces communication constraints to the control system, e.g. network-induced delay, data packet dropout, data packet disorder, data rate constraint. These communication constraints in NCSs present great challenges for conventional control theory.

The study of NCSs therefore requires multi-field knowledge, and consequently the integration of control, communication and computations, i.e. the "co-design" approach. In this book, we report a class of co-design approach to NCSs—the "packet-based control" approach—which is achieved by taking advantage of the packet-based transmission of the communication network in NCSs, one primary feature distinct from conventional control systems.

For completeness, an introductory chapter is first included which provides a brief tutorial of NCSs, and then the remainder of the book is organized into three parts, covering the design, analysis and extension of the packet-based control approach, respectively.

These studies have shown that the packet-based control approach is both unified and flexible: on the one hand, the approach can stand on its own as a novel class of design and analysis methods different from existing ones; on the other, existing control methods can also be fitted into the packet-based control approach for a better system performance. A unique co-design framework, i.e. packet-based networked control systems, is thus finally constructed.

We hope the reader will find this book useful for their understanding of and research on networked control systems.

Hangzhou, China
April 2017

Yun-Bo Zhao

Acknowledgements

The following financial supports are greatly appreciated: the National Natural Science Foundation of China under Grant 61673350, the Thousand Talents Plan of China and Zhejiang, the Major Projects Foundation of Zhejiang under Grant 2017C03060, and the Monograph and Postgraduate Textbook Foundation of Zhejiang University of Technology under grant 20160103.

Contents

1 A Brief Tutorial of Networked Control Systems 1
1.1 Introduction 1
1.2 Communicational Characteristics of NCSs 3
1.2.1 Network Topology 3
1.2.2 Packet-Based Data Transmission 4
1.2.3 Limited Network Resources. 7
1.3 The Research on Networked Control Systems 9
1.3.1 Control-Centred Research on NCSs. 9
1.3.2 Co-Design for NCSs 10
1.4 Summary 11

Part I Design

2 Packet-Based Control Design for Networked Control Systems 15
2.1 Problem Statement 15
2.2 Packet-Based Control for NCSs 17
2.2.1 Packet-Based Control for NCSs: A Unified Model 17
2.2.2 Design of the Packet-Based Control for NCSs 18
2.3 Stability of Packet-Based NCSs 22
2.3.1 A Switched System Theory Approach. 22
2.3.2 A Delay Dependent Analysis Approach 23
2.4 Controller Design: A GPC-Based Approach. 25
2.5 Numerical and Experimental Examples. 27
2.6 Summary 32

3 Packet-Based Control for Networked Hammerstein Systems 33
3.1 System Description. 33
3.2 Packet-Based Control for Networked Hammerstein Systems . . . 35
3.2.1 Intermediate FCS (FCIS). 35
3.2.2 The Nonlinear Input Process 38

3.2.3 Packet-Based Control for Networked Hammerstein Systems . . . 39
3.3 Stability Analysis of Packet-Based Networked Hammerstein Systems . . . 40
3.3.1 Stability Criterion in Input-Output Description . . . 40
3.3.2 Stability Criterion in State-Space Description . . . 42
3.4 Numerical and Experimental Examples . . . 46
3.5 Summary . . . 49

4 Packet-Based Control for Networked Wiener Systems . . . 51
4.1 System Description . . . 51
4.2 Packet-Based Control for Networked Wiener Systems . . . 52
4.3 Stability Analysis of Packet-Based Networked Wiener Systems . . . 54
4.3.1 Observer Error . . . 54
4.3.2 Closed-Loop Stability . . . 55
4.4 Numerical and Experimental Examples . . . 57
4.5 Summary . . . 58

5 Packet-Based Networked Control Systems in Continuous Time . . . 61
5.1 Packet-Based Control in Continuous Time . . . 61
5.1.1 Packet-Based Control for NCSs in Continuous Time . . . 62
5.1.2 A Novel Model for NCSs . . . 65
5.2 Stability and Stabilization . . . 66
5.3 A Numerical Example . . . 73
5.4 Summary . . . 74

Part II Analysis

6 Stochastic Stabilization of Packet-Based Networked Control Systems . . . 77
6.1 Stochastic Analysis of PBNCSs . . . 77
6.1.1 Stochastic Model of PBNCSs . . . 78
6.1.2 Stochastic Stability and Stabilization . . . 80
6.2 A Numerical Example . . . 84
6.3 Summary . . . 85

7 Stability of Networked Control Systems: A New Time Delay Systems Approach . . . 87
7.1 The Novel Time Delay System Model for PBNCSs . . . 87
7.2 Stability and Stabilization . . . 89
7.3 An Illustrative Example . . . 95
7.4 Summary . . . 97

8 Exploring the Different Delay Effects in Different Channels in Networked Control Systems . . . 99
8.1 Problem Formulation . . . 99
8.2 Categorizing the Control Laws . . . 101
8.2.1 Two General Categories of the Control Laws . . . 101
8.2.2 The Delay-Dependent Control Laws . . . 102
8.3 When and How the Delay Effects in Different Channels Are Different . . . 102
8.3.1 When the Delay Effects Are Different: A Qualitative Analysis . . . 103
8.3.2 How the Delay Effects with (8.4a) and (8.4c) are Different: A Quantitative Analysis . . . 103
8.3.3 A Brief Summary and Discussion . . . 108
8.4 Numerical Examples . . . 108
8.5 Summary . . . 113

Part III Extension

9 Active Compensation for Data Packet Disorder in Networked Control Systems . . . 117
9.1 Data Packet Disorder and Related Work . . . 117
9.1.1 Data Packet Disorder . . . 118
9.1.2 Related Work . . . 119
9.2 Actively Compensating for Data Packet Disorder in NCSs . . . 120
9.3 Modeling and Further Discussion . . . 122
9.3.1 A Unified Model for NCSs . . . 122
9.3.2 Further Discussion: Reduced Communication Constraints . . . 123
9.4 A Numerical Example . . . 124
9.5 Summary . . . 126

10 Error Bounded Sensing for Packet-Based Networked Control Systems . . . 127
10.1 Error Bounded Sensing for PBNCSs . . . 127
10.1.1 Error Bounded Sensing in the Sensor-to-controller Channel . . . 128
10.1.2 Packet-Based Control in the Controller-to-actuator Channel . . . 130
10.1.3 The EBS Strategy for PBNCSs . . . 130
10.2 Stabilization and Further Discussion . . . 132
10.2.1 Stabilization . . . 133
10.2.2 The Effects of the EBS Strategy . . . 135

10.3 Numerical and Experimental Examples 138
10.4 Summary 142

11 Packet-Based Deadband Control for Networked Control Systems 143
11.1 Packet-Based Deadband Control for NCSs 143
11.2 Stability and Stabilization of Packet-Based Deadband Control 146
11.2.1 The Control Laws 146
11.2.2 Stability and Stabilization 148
11.3 Numerical and Experimental Examples 155
11.4 Summary 160

12 Packet-Based Control and Scheduling Co-Design for Networked Control Systems 161
12.1 Problem Statement 161
12.2 Packet-Based Control for Subsystems 162
12.3 Scheduling 164
12.3.1 Static Scheduling 164
12.3.2 Dynamic Feedback Scheduling 167
12.4 Numerical Examples 171
12.5 Summary 174

References 175

Index 183

Acronyms

CARIMA	**C**ontrolled **A**uto **R**egressive **I**ntegrated **M**oving **A**verage
CAS	**C**ontrol **A**ction **S**elector
CTR	**C**ontroller **T**ransmission **R**ule
DCDD	**D**ifferent-**C**hannel-**D**elay-**D**ependent
DCDI	**D**ifferent-**C**hannel-**D**elay-**I**ndependent
DFS	**D**ynamic **F**eedback **S**cheduling
EBS	**E**rror **B**ounded **S**ensing
FCIS	**F**orward **C**ontrol **I**ncrement **S**equence
FCS	**F**orward **C**ontrol **S**equence
FFG	**F**ixed **F**eedback **G**ain
GPC	**G**eneralized **P**redictive **C**ontrol
LGPC	**L**inear **G**eneralized **P**redictive **C**ontrol
LQR	**L**inear **Q**uadratic **O**ptimal
LTI	**L**inear **T**ime-**I**nvariant
MPC	**M**odel **P**redictive **C**ontrol
NCS(s)	**N**etworked **C**ontrol **S**ystem(**s**)
PBNCS(s)	**P**acket-**B**ased **N**etworked **C**ontrol **S**ystem(**s**)
QoP	**Q**uality **o**f **P**erformance
RM	**R**ate **M**onotonic
RHC	**R**eceding **H**orizon **C**ontrol
STR	**S**ensor **T**ransmission **R**ule
SISO	**S**ingle-**I**nput-**S**ingle-**O**utput
SSRTD	**S**tability-guaranteed **S**upremum of **R**ound **T**rip **D**elay
TCP	**T**ransmission **C**ontrol **P**rotocol
TDS(s)	**T**ime **D**elay **S**ystem(**s**)
UDP	**U**ser **D**atagram **P**rotocol
VFG	**V**arying **F**eedback **G**ain

Chapter 1
A Brief Tutorial of Networked Control Systems

A brief tutorial of NCSs is provided in this introductory chapter. The tutorial first introduces the unique characteristics of NCSs and then reviews the research in this field. The former is explored from a perspective that emphasizes the differences between NCSs and conventional control systems, and thus the role played by the computation network in NCSs is extensively examined, including, e.g., the network topology, the packet-based transmission and the limited communication resources, etc. Among all these characteristics, the packet-based transmission will be highlighted as this is the basis of the so-called "packet-based control" for NCSs, the principal theme of the book. The second part of the tutorial covers the state-of-the-art research on NCSs. We particularly make the comparison between conventional studies and the so-called "co-design" approach to NCSs, to which the packet-based control approach belongs.

1.1 Introduction

"Networked control systems" refers to a general class of control systems whose control loop is closed via some form of communication network [1–6]. Interest in such a system configuration can date back to as early as 1980s', when the so-called "Integrated Communication and Control Networks" attracted much attention from the control community [7]. From that time on, other alias such as "Network-based Control Systems" and "Control over (through) Networks" have also been used to describe the similar system configuration as NCSs but are not often used today [8–13].

As indicated by its name, the most distinct feature of NCSs is the use of communication networks in the control loop [14, 15]. Earlier days have witnessed the use of the control-oriented communication networks such as the Control Area Network (CAN), DeviceNet, etc., as the first choice of the communication networks in NCSs, the fast development of the communication technology as well as the increasing needs of large scale systems have now made the Internet or other forms of data networks an

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_1

attracting alternative. The Internet offers us the capability of building a large control system at much lower cost, easier maintenances, with also the more flexible reconfiguration capability. Built on the fundamental theoretical advances in NCSs, we have seen various innovations such as the smart home, smart transportation, remote surgery, Internet of Things, etc. in recent years [16–22].

The advantages brought by NCSs however do not come at no cost. A fundamental basis of conventional control systems is that the data exchanges among the control components are lossless. In NCSs, the data have to be transmitted through the communication network, and the nature of the Internet and other variations of data networks means that perfect data exchanges among the control components is essentially unavailable. The imperfect data translation in NCSs thus introduces the so-called communication constraints to the control system, which include, e.g., the network-induced delay (the delays occurred in transmitting the sensing and control data), the data packet dropout (the data packet may be missing during transmission), the time synchronization issue (different control components may work on different clocks), and so on [23]. These communication constraint can greatly degrade the system performance or even destabilize the system at certain conditions, while simple extensions of conventional control approaches can not be obtained directly in a networked control environment [24–29]. These difficulties thus pose great challenges for the control and communication communities and considerable works have been done to a better understanding and design of such systems at the boundary of control theory and communication technology [30–32].

We provide a brief tutorial on NCSs in this introductory chapter. This consists of two parts. We first give an extensive introduction of the communication networks in NCSs, including its basic characteristics and more importantly its interactions with the control system. Note that we focus on data networks such as the Internet but not the control-oriented networks, simply because of the increasing use and more complicated communication features of the former. We then survey the state-of-the-art research on NCSs, from mainly the control perspective with also an emphasis on the co-design approach which integrates both control and communication.

For simplicity in this tutorial we focus on a simple structure of NCSs. From a general perspective of system structure, NCSs may contain two different structures [33]: the "direct structure" in Fig. 1.1 and the "hierarchical structure" in Fig. 1.2. The latter is different from the former as a local controller is present and the communication network is used to close the loop between the main controller and the local system. This structural distinction may have some theoretical as well as practical values, the latter, however, may be regarded as a hierarchical combination of the direct structured NCS and a conventional local control system and therefore it is not absolutely necessary to investigate the hierarchical structure as a brand new type of NCSs. In fact, most available works on NCSs to date have focused on the direct structure, which is also the main focus of this brief tutorial and the book.

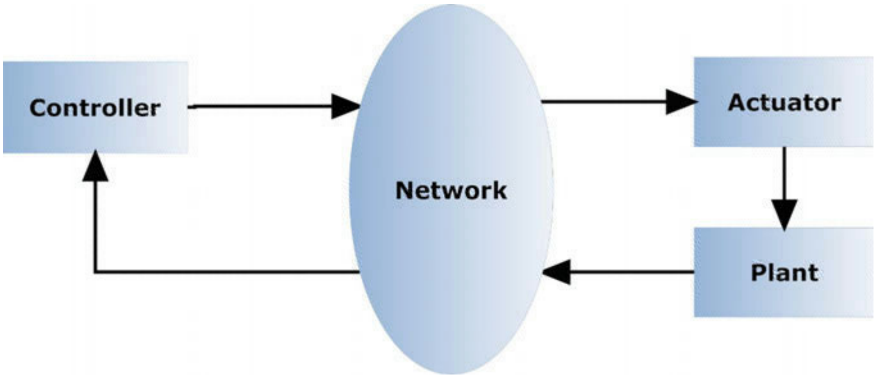


Fig. 1.1 Networked control systems in the direct structure

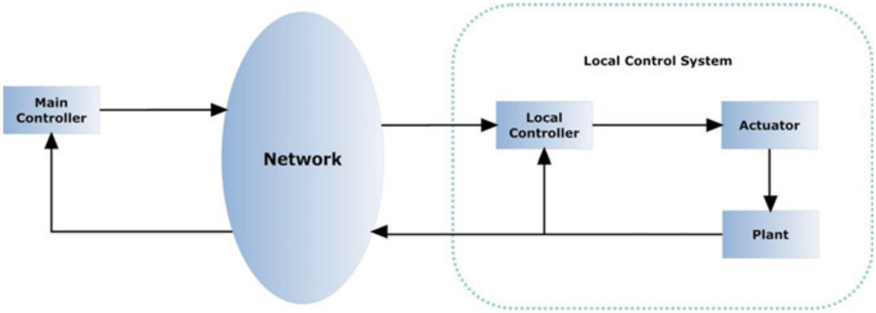


Fig. 1.2 Networked control systems in the hierarchical structure

1.2 Communicational Characteristics of NCSs

This section introduces the basics of NCSs, where the emphasis is on the differences between NCSs and conventional control systems, that is, the distinct and unique characteristics of NCSs that are brought by the inserted communication network.

1.2.1 Network Topology

In the presence of the communication network in the NCSs, the conventional control components including the sensor, the controller and the actuator work as network nodes. From this perspective, two issues need to be addressed, as follows.

1.2.1.1 Time-Synchronization

The control components need to be time synchronized to act properly. This is a fundamental basis of conventional control systems, but is usually missing in NCSs due to the use of the distributed communication networks [34]. Under certain conditions, time-synchronization in NCSs may not be a necessary condition if the network-induced delay in the backward channel is not required for the calculation of the control signals and/or the network-induced delay in the forward channel is not required for the implementation of the control actions. In some other cases, as discussed in [35, 36], time-synchronization together with the use of time stamps in NCSs can offer an advantage over conventional time delay systems since the backward channel delay is known by the controller and the forward channel delay (round trip delay as well) is known by the actuator. This advantage can then be used to derive a better control structure for NCSs as done in [35, 37].

1.2.1.2 Drive Mechanism

The sensor and the actuator can be driven either by time or event. The difference between the two drive mechanisms lies in the trigger method that initiates the control components. For the time-driven mechanism, the control components are trigged to work at regular intervals, while for the event-driven mechanism the control components are only trigged by predefined "events". From a broad perspective time-driven can be regarded as a special case of event-driven, when the trigger events for the latter are chosen as the time. Therefore, it is no wonder why the event-driven mechanisms are more sophisticated and may require ancillary devices for it to work.

The sensor is usually time-driven, while the controller and the actuator can either be time-driven or event-driven. For more information on the drive mechanism for the control components, the reader is referred to [38–41] and the references therein. It is worth mentioning though, with different drive mechanisms different system models for NCSs are obtained and event-driven control components generally lead to a better system performance.

1.2.2 Packet-Based Data Transmission

The data in NCSs is encoded in the data packets and then transmitted through the communication network. A typical data packet is shown in Fig. 1.3. Packet-based transmission is one of the most important characteristics of NCSs that distinguishes it from conventional control systems [42–44]. This characteristic can mean that the perfect data transmission as assumed in conventional control systems is absent in NCSs, posing the most challenging aspect in NCSs. The communication constraints caused by the packet-based transmission in NCSs include the network-induced delay, data packet dropout, data packet disorder, etc., which are detailed in what follows.

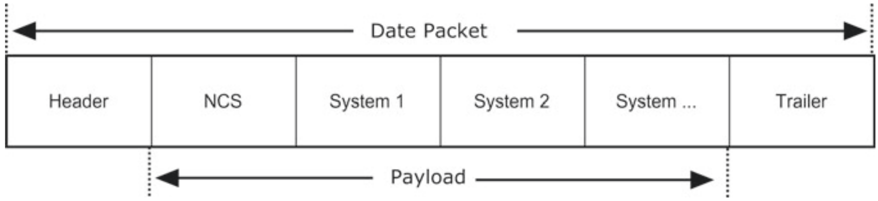


Fig. 1.3 The typical data packet structure where NCS is sharing the data packet with other applications

1.2.2.1 Network-Induced Delay

The transmission time for the data packets introduces network-induced delays to NCSs, which are well known to degrade the performance of the control systems.

There are two types of network-induced delays according to where they occur.

- τ_{sc}: Network-induced delay from the sensor to the controller, i.e., backward channel delay;
- τ_{ca}: Network-induced delay from the controller to the actuator, i.e., forward channel delay.

The two types of network-induced delays may have different characteristics [45]. In most cases, however, these delays are not treated separately and only the round trip delay is of interest [15, 46–48].

According to the types of the communication networks being used in NCSs, the characteristics of the network-induced delay vary, as follows [33, 49, 50].

- Cyclic service networks (e.g., Toking-Ring, Toking-Bus): Bounded delays which can be regarded as constant for most occasions;
- Random access networks (e.g., Ethernet, CAN): Random and unbounded delays;
- Priority order networks (e.g., DeviceNet): Bounded delays for the data packets with higher priority and unbounded delays for those with lower priority.

Network-induced delay is one of the most important characteristics of NCSs which has been widely addressed in the literature to date, see, e.g., [15, 46, 48, 51–60].

1.2.2.2 Data Packet Dropout

Data transmission error in communication networks is inevitable, which in the case of NCSs then produces a situation called "data packet dropout". Data packet dropout can occur either in the backward or forward channel, and it makes either the sensing data or the control signals unavailable to NCSs, thus significantly degrading the performance of NCSs.

In communication networks, two different strategies are applied when a data packet is lost, that is, either to send the packet again or simply discard it. Using the terms from communication networks, these two strategies are called Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) respectively [34]. It is readily seen that with TCP, all the data packets will be received successfully, although it may take a considerably long time for some data packets; while with UDP, some data packets will be lost forever.

As far as NCSs is concerned, UDP is used in most applications due to the real-time requirement and the robustness of control systems. As a result, the effect of data packet dropout in NCSs has to be explicitly considered, as done in, e.g., [61–65].

1.2.2.3 Data Packet Disorder

In most communication networks, different data packets suffer different delays, which then produces a situation where a data packet sent earlier may arrive at the destination later, or vice versa, see Fig. 1.4. This phenomenon is referred to as data packet disorder. The existence of data packet disorder can mean that a newly arrived control signal in NCSs may not be the latest, which never occurs in conventional control systems. The control performance will be inevitably degraded if the control algorithm has not taken explicit consideration of the disordered data. Some preliminary works have been done, usually using an active compensation scheme [66–68].

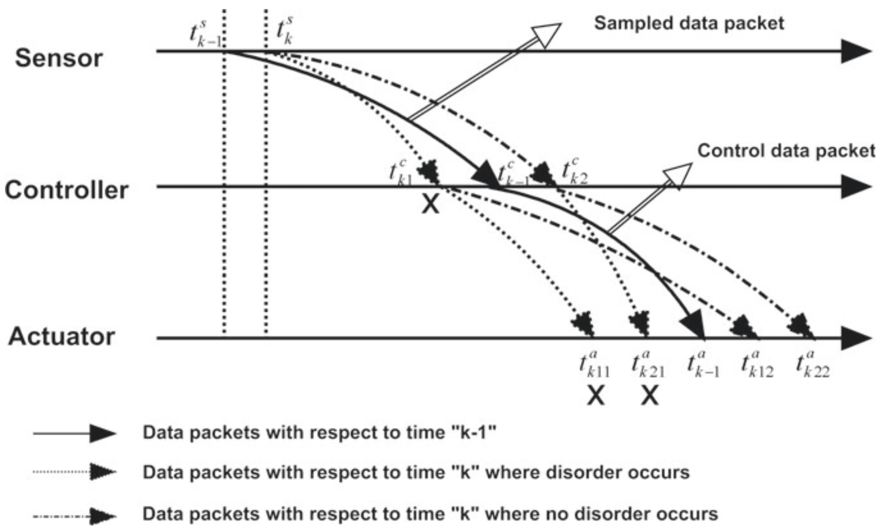


Fig. 1.4 Data packet disorder in NCSs

1.2.2.4 Single and Multi-packet Transmission

When the sensing data and the control signals are sent via data packets of the network, another situation occurs: in a case where, for example, multiple sensors are used and distributed geographically in NCSs and thus they send their sensing data separately to the controller over the network, the controller may have to wait for the arrival of all the sensing data packets before it is able to calculate the control actions, and if only one sensing data packet is lost, all the other sensing data packets have to be discarded due to incompleteness. We call this situation the “multi-packet” transmission of the data in NCSs.

Another situation in NCSs is where the sensing data or the control signals of multiple steps are sent via a single data packet over the network, since the packet size used in NCSs can be very large compared with the data size required to encode a single step of sensing data or control signal. This “single-packet” transmission of the data in NCSs is the fundamental basis of the so-called packet-based control approach [37].

1.2.3 Limited Network Resources

The limitation of the network resources in NCSs is primarily caused by the limited bandwidth of the communication network, which results in the following three situations in NCSs that are distinct from conventional control systems.

1.2.3.1 Sampling Period, Network Loads and System Performance

NCSs is a special class of sampled data systems due to the digital transmission of the data in communication networks. However, in NCSs, the limited bandwidth of the network produces a situation where, a smaller sampling period may not result in a better system performance which is normally true for sampled data systems [4, 25, 69–71].

This situation happens because, with too small a sampling period, too much sensing data will be produced; thus overloading the network and causing congestion, which will result in more data packet dropouts and longer delays, and then degrade the system performance. The relationship between the sampling period, network loads and system performance in NCSs is illustrated in Fig. 1.5. For example, when the sampling period decreases from the value corresponding to point “a” to “b”, the system performance is getting better as in conventional sampled data systems since the network congestion does not appear until point “b”; However, the system performance is likely to deteriorate due to the network congestion when the sampling period is getting even smaller from the value corresponding to point “b” to “c”. Therefore, the relationship shown in Fig. 1.5 implies that there is a trade-off between the period of sampling the plant data and the system performance in NCSs, that is, in

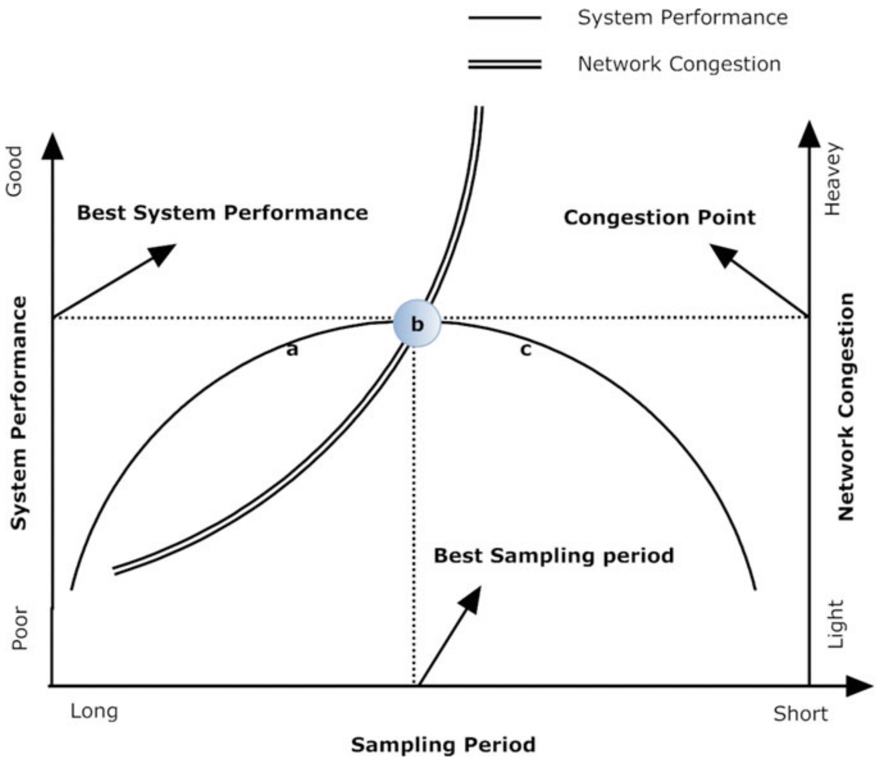


Fig. 1.5 Relationship between the sampling period, network loads and system performance in NCSs

NCSs an optimal sampling period exists which offers the best system performance (point "b" in Fig. 1.5).

1.2.3.2 Quantization

Due to the use of data networks with limited bandwidth, signal quantization is inevitable in NCSs, which has a significant impact on the system performance. Quantization in the meantime is also a potential method to reduce the bandwidth usage which enables it to be an effective tool to avoid the network congestion in NCSs and thus improve the system performance of NCSs. For more information on the quantization effects in NCSs, the reader is referred to [72–76] and the references therein.

1.2.3.3 Network Access Constraint and Scheduling

An NCS may use only part of the payload and share the data packet with other applications. These other applications can be NCSs or not. In such a case, the limited bandwidth of the network makes that all the subsystems can not access the network resource at the same time. A scheduling algorithm is therefore needed to schedule the timeline of when and how long a specific subsystem can occupy the network resource. At the same time, under the satisfactory control performance constraint, the less bandwidth an NCS uses, the better it does to other applications [77–80].

1.3 The Research on Networked Control Systems

In this section, we briefly survey the state-of-the-art research on NCSs. This consists of two parts, categorized according to the methodologies used in these research, i.e. the first category is dominated by the use of the control theory while the second one adopts a co-design strategy by combining control and communication together.

1.3.1 Control-Centred Research on NCSs

Since the renewed interest in NCSs, the research on NCSs has been primarily done within the control theory community [33].

From the control theory community, one is concerned with the theoretical analysis of the control performance of NCSs where the network in NCSs is modeled by predetermined parameters to the control system. In this type of research, the communication characteristics of NCSs, e.g., the network-induced delay, can be formulated and incorporated into the system as some parameters, thereby yielding a conventional control system for further analysis and design. This type of research simplifies the modeling and analysis of NCSs, enabling all existing control methods to be readily applied to NCSs. Hence, such a research strategy has been dominating the research field for a significant period [33, 81].

Since the communication characteristics are assumed to be given parameters, the design of NCSs then faces great conservativeness. Most works can only focus on the extension of existing control approaches to NCSs without full use of the communication characteristics of NCSs. This then ignores the possibility of optimizing the system performance by making efficient use of the network characteristics [82–84].

The conventional control approaches and theories that have been applied to NCSs are briefly surveyed, as follows.

- **Time Delay Systems**. As far as the network-induced delay is concerned, it is natural to model NCSs as a special class of time delay systems. This research method covers a vast range of research on NCSs, see, e.g., in [55, 85–88], and the survey in [15, 33].

An interesting issue here is to determine the Maximum Allowable Delay Bound (MADB) of NCSs, which is the upper bound of the transfer interval that ensures the stability or other performance objectives of NCSs. The determination of MADB is important in theory and can also play a guiding role for practical applications. One can refer to the survey paper in [89] for more information on this issue.

- **Stochastic control**. As mentioned above, the communication constraints in NCSs are stochastic in nature, thus enabling the application of conventional stochastic control approaches to NCSs. An early study can be found in [45], where the characteristics of the network-induced delay were explicitly formulated and preliminary stochastic stability criteria of NCSs were obtained; [51] extended the work in [7] to a stochastic optimal control framework and gave the stochastic optimal state feedback and output feedback controllers, respectively; In [56], the sufficient and necessary conditions of the stochastic stability of NCSs were obtained based on the Markov jump system framework. For further information, the reader is referred to the survey in [23].
- **Optimal control**. As a very successful idea both in theory and practical applications, optimal control has also found its position in NCSs. Undoubtedly, conventional optimal control approaches can be used in the networked control environment to design the controller for NCSs, see, e.g., in [60, 62, 90–93]; and as a special class of optimal control approaches, Model Predictive Control (MPC, or Receding Horizon Control (RHC)) seems to be more suitable for the networked control environment and "a major extension required to apply model predictive control in networked environments would be the distributed solution of the underlying optimization problem" [43]. Examples of the application of MPC to NCSs can be seen, e.g., in [36, 94–98].
- **Switched system theory**. Another important tool in the study on NCSs is switched system theory, which is typically used by modelling different network conditions in NCSs as different system modes. This approach can readily deal with network-induced delay as well as data packet dropout in NCSs, and the limitation of the approach is caused mainly by how well we understand the properties of the changes of the network conditions, which is generally difficult. For the research in this area, the reader is referred to [99–103] and the references therein.

1.3.2 Co-Design for NCSs

As has been pointed out earlier, it is the communication network which replaces the direct connections among the control components in conventional control systems that makes NCSs distinct. Therefore, the so-called co-design approach to NCSs, an approach that integrates both control and communication, has been an emerging trend in recent years. The communication constraints are no longer assumed as predetermined parameters but act as designable factors, and by the efficient use of these factors a better performance can be expected [36, 42, 83, 97, 104–108]. We give two examples of the co-design approach to NCSs.

- **Packet-based control approach**. As discussed in Sect. 1.2.2, the packet-based transmission is one of the most distinct characteristics of NCSs. This characteristic can be used to derive a co-design control structure for NCSs, called the packet-based control framework, as done in [36, 37, 109, 110]. The packet-based control approach has its origin in [95, 107], where with the use of generalized predictive control method the packet-based structure of the data transmission was efficiently used to actively compensate for the communication constraints in NCSs.
- **Control and scheduling co-design**. In NCSs, a situation may occur where multiple control components share a network with limited bandwidth. In such a situation, network resource scheduling among the control components is necessary, see Sect. 1.2.3.3. As far as the scheduling algorithms are concerned, [1] proposed a dynamic scheduling algorithm called "Try-Once-Discard" (TOD) which allocates the network resources in a way that the node with the greatest error in the last reported period has access to the network resource. Nesic and Teel [111] proposed a Lyapunov Uniformly Globally Asymptotically Stable (UGAS) protocol based on TOD, which is further improved in [112]. In [113], the authors used the technique of "communication sequence" (see also in [114]) to deal with the network access constraint for such a system configuration and modeled the subsystems as switched systems with two modes "open loop" and "closed loop" which switch according to whether the current subsystem has access to the medium or not. In [115], the authors considered a setup where the channel from controller to actuator is linked directly, and the rate monotonic scheduling algorithm is applied to schedule the transmissions of the sensing data of the subsystems.

1.4 Summary

Despite all the achievements that have been made for networked control systems in the past decades, more efforts are still needed in the future. Most of these ongoing research adopts the co-design methodology, and the collaborations between the control, communication as well as computation communities are desirable.

These collaborations will then reveal the values of networked control systems in broader perspectives, by looking into its close relationship with other systems such as the Internet of things, cyber-physical systems, multi-agent systems, and so forth. All these together then bring us the promising future of the networked, intelligent automation.

Part I
Design

In this part, we discuss the design of packet-based networked control systems in various settings. This starts from a general description of the approach to discrete linear systems in Chap. 2. Then, the approach is extended in two directions, by considering nonlinear dynamics (Hammerstein systems in Chap. 3 and Wiener systems in Chap. 4) and plants in continuous time (Chap. 5), respectively. A unified design framework of the packet-based control approach to networked control system is finally completed.

Chapter 2
Packet-Based Control Design for Networked Control Systems

In this chapter, we exploit the fact that in most communication networks, data is transmitted in a "packet" and within its effective load sending a single bit or several hundred bits of data consumes the same amount of network resources [15]. This makes it possible in NCSs to actively compensate for the communication constraints by sending a sequence of control predictions in one data packet and then selecting the appropriate one corresponding to the current network condition. This packet-based transmission characteristic motivates us to design the packet-based control approach to NCSs. Due to the active compensation process in the packet-based control approach, a better performance can be expected compared to an implementation where no characteristics of the network are specifically considered in the design.

This chapter is organized as follows. The reason why a co-design approach is needed for NCSs is explained in Sect. 2.1. The design of the packet-based control for NCSs is presented in detail in Sect. 2.2, which leads to a novel controller that can compensate for network-induced delay, data packet dropout and data packet disorder simultaneously. The stability criteria for the corresponding closed-loop system are then investigated in Sect. 2.3, using both switched system theory and delay-dependent analysis, respectively. As an example, a GPC based controller under the packet-based control framework is designed in Sect. 2.4, which is more feasible in practice compared with previous results. Numerical and experimental examples to illustrate the effectiveness of the proposed approach are presented in Sects. 2.5 and 2.6 concludes the chapter.

2.1 Problem Statement

The NCS setup considered in this chapter is shown in Fig. 2.1, where $\tau_{sc,k}$ and $\tau_{ca,k}$ are the backward and forward channel delays respectively and the plant is linear in discrete-time which can be represented by

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_2

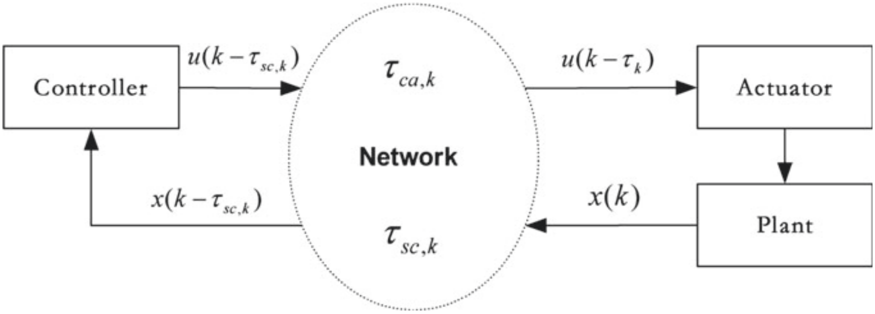


Fig. 2.1 The block diagram of networked control systems in discrete time

$$\mathcal{S}_d : \begin{cases} x(k+1) = Ax(k) + Bu(k) & (2.1a) \\ y(k) = Cx(k) & (2.1b) \end{cases}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{r\times n}$. Though it has not been explicitly shown in Fig. 2.1, the effects of data packet dropout and data packet disorder are also considered in the packet-based control approach.

For such a system setting, conventional state feedback law is generally obtained as follows without the consideration of the communication constraints in NCSs,

$$u(k) = Kx(k) \tag{2.2}$$

where the feedback gain K is time-invariant.

However, when the network-induced delay is considered, the state feedback law can not be simply defined as in (2.2) due to the unavailability of the current state information. The resulting control law using conventional approaches in TDSs would have the following form

$$u(k) = Kx(k - \tau_k) \tag{2.3}$$

where the effect of the delay is not been specially treated in the design. Furthermore, when data packet dropout is also present, it can be seen from Fig. 2.1 that no matter where data packet dropout occurs, a certain control input will be unavailable to the actuator. In conventional TDS theory, there are mainly two ways to deal with this situation, either use the previous control input or adopt zero control [81]. For example, in [116], the last step of the control signal is used as the control strategy in the case of an unsuccessful transmission, as follows,

$$u(k) = \begin{cases} \bar{u}(k) & \text{if transmitted successfully;} \\ u(k-1) & \text{otherwise.} \end{cases} \tag{2.4}$$

where $\bar{u}(k)$ is the newly arrived control signal at time k.

It can be seen that though the conventional control strategies in (2.3) and (2.4) are simple to implement, they are conservative in that they overlook the potential of providing an active prediction for the unavailable control input using available information of the system dynamics and previous system trajectory as done in [36, 107]. Our packet-based control approach is meant to deal with this conservativeness in a unified framework.

2.2 Packet-Based Control for NCSs

2.2.1 Packet-Based Control for NCSs: A Unified Model

As discussed earlier in Sect. 1.2.2, the presence of the network in NCSs brings to the system network-induced delay, data packet dropout, data packet disorder, etc. These communication constraints degrade the system performance significantly whereas the packet-based transmission of the network also offers the potential of transmitting a sequence of control signals simultaneously instead of one at a time as typically done in conventional control systems. This observation is the motivation of the packet-based control approach to NCSs.

The control law based on the packet-based control approach is obtained as follows with explicit compensation for the communication constraint (see Algorithm 2.1 to be given later),

$$u(k) = K(\tau^*_{sc,k}, \tau^*_{ca,k})x(k - \tau^*_{sc,k} - \tau^*_{ca,k}) \tag{2.5}$$

or simply (see Algorithm 2.2 to be given later),

$$u(k) = K(\tau^*_k)x(k - \tau^*_k) \tag{2.6}$$

where $\tau^*_{sc,k}$ and $\tau^*_{ca,k}$ are the network-induced delays of the control action that is actually applied to the plant at time k and $\tau^*_k = \tau^*_{sc,k} + \tau^*_{ca,k}$.

It is seen from the control laws in (2.5) and (2.6) that in the packet-based control approach, different feedback gains apply for different network conditions. This is why we call it a "Varying Feedback Gain" (VFG) scheme for NCSs. As will be presented later, these packet-based control laws can actively deal with the network-induced delay, data packet dropout and data packet disorder simultaneously, and therefore can be regarded as a unified model for NCSs. This control strategy can be compared with the conventional approach as in (2.3) and (2.4) where no active compensation is available.

Remark 2.1 In [56], the authors noticed the unavailability of the forward channel delay $\tau_{ca,k}$ and a controller was designed with the following form

$$u(k) = K(\tau_{sc,k}, \tau_{ca,k-1})x(k - \tau_{sc,k} - \tau_{ca,k}) \tag{2.7}$$

where the forward channel delay of the last step $\tau_{ca,k-1}$ was used instead. However, actually even $\tau_{ca,k-1}$ is generally unavailable for the controller in NCSs since in the case of a arbitrary forward channel delay, $\tau_{ca,k-1}$ can not be known to the controller until the controller receives information of $\tau_{ca,k-1}$ from the actuator. Therefore, it is seen that $\tau_{ca,k-1}$ can not be available for the controller earlier than time $k-1+\tau_{ca,k-1}$ even if an additional delay-free channel exists to send the information of $\tau_{ca,k-1}$ from the actuator to the controller. As a result, the above model in (2.7) is inappropriate in practice unless a special control structure is designed for the networked control environment as done in this chapter.

2.2.2 Design of the Packet-Based Control for NCSs

For the design of the packet-based control approach, the following assumptions are required.

Assumption 2.1 The control components in the considered NCS including the sensor, the controller and the actuator, are time-synchronized and the data packets sent from both the sensor and the controller are time-stamped.

Assumption 2.2 The sum of the maximum forward (backward) channel delay and the maximum number of continuous data packet dropout is upper bounded by $\bar{\tau}_{ca}$ ($\bar{\tau}_{sc}$ accordingly) and

$$\bar{\tau}_{ca} \leq \frac{B_p}{B_c} - 1 \tag{2.8}$$

where B_p is the size of the effective load of the data packet and B_c is the bits required to encode a single step control signal.

Remark 2.2 From Assumption 2.1, the network-induced delay that each data packet experiences is known by the controller and the actuator on its arrival.

Remark 2.3 Assumption 2.2 is required due to the need of packing the forward control signals and compensating for the network-induced delay in the packet-based control approach, which will be detailed later. The constraint in (2.8) is easy to be satisfied, e.g., $B_p = 368$ bit for an Ethernet IEEE 802.3 frame which is often used [34], while an 8-bit data (i.e., $B_c = 8$ bit) can encode $2^8 = 256$ different control actions which is ample for most control implementations; In this case, 45 steps of forward channel delay is allowed by (2.8) which can actually meet the requirements of most practical control systems.

The block diagram of the packet-based control structure is illustrated in Fig. 2.2. It is distinct from a conventional control structure in two aspects: the specially designed packet-based controller and the corresponding Control Action Selector (CAS) at the actuator side.

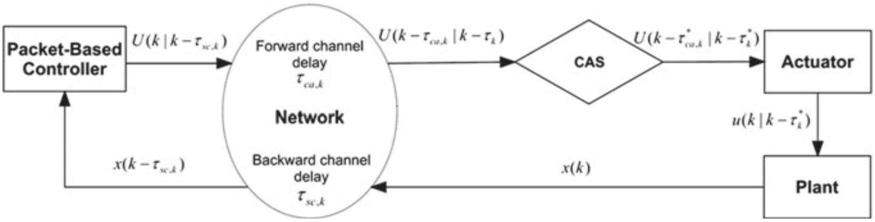


Fig. 2.2 Packet-based networked control systems in discrete time (with time synchronization)

In order to implement the control law in (2.5) and (2.6), we take advantage of the packet-based transmission of the network to design a packet-based controller instead of trying to obtain directly the current forward channel delay as this is actually impossible in practice. As for the control law in (2.5), the packet-based controller determines a sequence of forward control actions (called "Forward Control Sequence" (FCS)) as follows and sends them together in one data packet to the actuator,

$$U_1(k|k-\tau_{sc,k}) = [u(k|k-\tau_{sc,k}) \ \ldots \ u(k+\bar{\tau}_{ca}|k-\tau_{sc,k}]^T \tag{2.9}$$

where $u(k+i|k-\tau_{sc,k}), i = 0, 1, \ldots, \tau_{ca,k}$ are the forward control action predictions based on information up to time $k-\tau_{sc,k}$.

When a data packet arrives at the actuator, the designed CAS compares its time stamp with the one already in CAS and only the one with the latest time stamp is saved. Denote the forward control sequence already in CAS and the one just arrived by $U_1(k_1-\tau_{ca,k_1}|k_1-\tau_{k_1})$ and $U_1(k_2-\tau_{ca,k_2}|k_2-\tau_{k_2})$ respectively, then the chosen sequence is determined by the following comparison rule,

$$U_1(k-\tau^*_{ca,k}|k-\tau^*_k) = \begin{cases} U_1(k_2-\tau_{ca,k_2}|k_2-\tau_{k_2}), & \text{if } k_1-\tau_{k_1} < k_2-\tau_{k_2}; \\ U_1(k_1-\tau_{ca,k_1}|k_1-\tau_{k_1}), & \text{otherwise.} \end{cases} \tag{2.10}$$

The comparison process is introduced due to the fact that different data packets may experience different delays thus producing such a situation where a packet sent earlier may arrive at the actuator later or vice versa, that is, data packet disorder. After the comparison process, only the latest available information is used and the effect of data packet disorder is effectively overcome.

CAS also determines the appropriate control action from the FCS $U_1(k-\tau^*_{ca,k}|k-\tau^*_k)$ at each time instant as follows

$$u(k) = u(k|k-\tau^*_k) \tag{2.11}$$

The timeline of the packet-based control approach is illustrated in Fig. 2.3. It is necessary to point out that the appropriate control action determined by (2.11) is always available provided Assumption 2.2 holds.

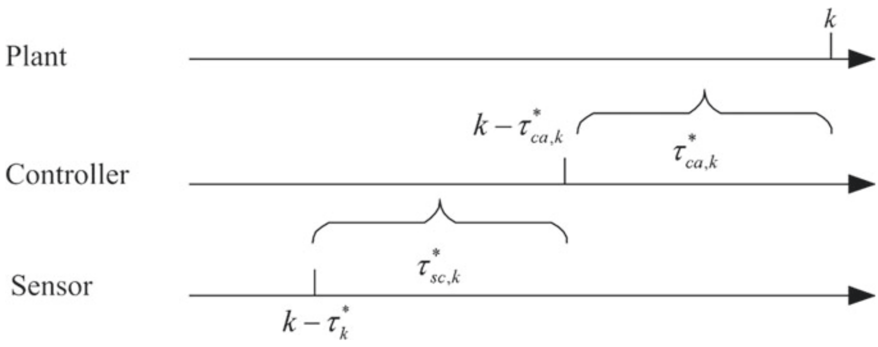


Fig. 2.3 Timeline in the packet-based NCSs

The packet-based control algorithm under Assumptions 2.1 and 2.2 can now be summarized as follows.

Algorithm 2.1 Packet-based control without time synchronization

---

if The packet-based controller receives the delayed state data $x(k-\tau_{sc,k})$ at time k, the controller **then**
  Reads the current backward channel delay $\tau_{sc,k}$
  Calculates the FCS as in (2.9)
  Packs $U_1(k|k-\tau_{sc,k})$ and sends it to the actuator in one data packet with time stamps k and $\tau_{sc,k}$
else
  Let $k=k+1$ and wait for the next time instant
end if
if A data packet arrives at the CAS **then**
  CAS updates its FCS by (2.10)
  The control action in (2.11) is picked out from CAS and applied to the plant
end if

In practice, it is often the case that we do not need to identify separately the forward and backward channel delays since it is normally the round trip delay that affects the system performance. In such a case, the simpler control law in (2.6) instead of that in (2.5) is applied, for which the following assumption is required instead of Assumption 2.2.

Assumption 2.3 The sum of the maximum network-induced delay and the maximum number of continuous data packet dropout in round trip is upper bounded by $\bar{\tau}$ and

$$\bar{\tau} \leq \frac{B_p}{B_c} - 1 \tag{2.12}$$

With the above assumption, the packet-based controller is modified as follows

$$\begin{aligned} &U_2(k-\tau_{sc,k}|k-\tau_{sc,k}) \\ =&[u(k-\tau_{sc,k}|k-\tau_{sc,k}) \ \ldots \ u(k-\tau_{sc,k}+\bar{\tau}|k-\tau_{sc,k}]^T \end{aligned} \tag{2.13}$$

It is noticed that in such a case the backward channel delay $\tau_{sc,k}$ is not required for the controller, since the controller simply produces $(\bar{\tau}+1)$ step forward control actions whenever a data packet containing sensing data arrives. This relaxation implies that the time-synchronization between the controller and the actuator (plant) is not required any more and thus Assumption 2.1 can then be modified as follows.

Assumption 2.4 The data packets sent from the sensor are time-stamped.

The comparison rule in (2.10) and the determination of the actual control action in (2.11) remain unchanged since both of them are based on the round trip delay τ_k.

The packet-based control algorithm with the control law in (2.6) can now be summarized as follows based on Assumptions 2.3 and 2.4.

Algorithm 2.2 Packet-based control with time synchronization

---

if The packet-based controller receives the delayed state data $x(k-\tau_{sc,k})$ at time k, the controller **then**
    Calculates the FCS as in (2.13)
    Packs $U_2(k-\tau_{sc,k}|k-\tau_{sc,k})$ and sends it to the actuator in one data packet
else
    Let $k=k+1$ and wait for the next time instant.
end if
if A data packet arrives at the CAS **then**
    CAS updates its FCS by (2.10)
    The control action in (2.11) is picked out from CAS and applied to the plant
end if

The block diagram of the packet-based control approach in Algorithm 2.2 is illustrated in Fig. 2.4.

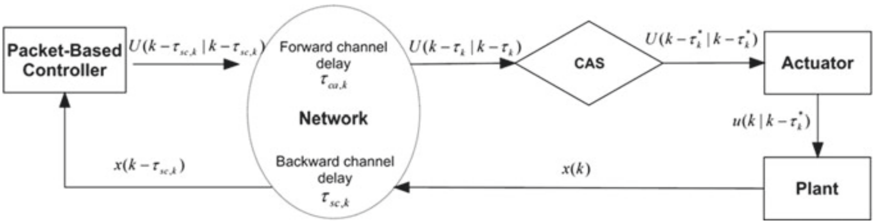


Fig. 2.4 Packet-based networked control systems in discrete time (without time synchronization)

2.3 Stability of Packet-Based NCSs

In this section the stability criteria for the system in (2.1) using the aforementioned packet-based control approach with the control laws in (2.5) and (2.6) are investigated. Two stability analysis approaches, i.e., results from switched system theory and delay-dependent analysis, are applied, by modeling the closed-loop system into different forms. Unless otherwise specified, all the stability related notions are under the Lyapunov framework.

2.3.1 A Switched System Theory Approach

An intuitive observation on the packet-based control approach is that, at every execution time, a specific control action is determined by the CAS according to the current network condition. Thus, regarding this selection process as "switches" among different subsystems, then yields the following analysis from the viewpoint of switched system theory.

Let $X(k) = [x(k)\, x(k-1)\ \dots\ x(k-\bar{\tau})]$. The closed-loop formula for the system in (2.1) using the packet-based controllers in (2.5) and (2.6) can then be represented in augmented forms as

$$X(k+1) = \Xi_{\tau^*_{sc,k},\tau^*_{ca,k}} X(k) \tag{2.14}$$

and

$$X(k+1) = \Xi_{\tau^*_k} X(k) \tag{2.15}$$

respectively, where

$$\Xi_{\tau^*_{sc,k},\tau^*_{ca,k}} = \begin{pmatrix} A & \cdots & BK_{\tau^*_{sc,k},\tau^*_{ca,k}} & \cdots & \cdots \\ I_n & & & & 0 \\ & I_n & & & 0 \\ & & \ddots & & \vdots \\ & & & I_n & 0 \end{pmatrix},$$

$$\Xi_{\tau^*_k} = \begin{pmatrix} A & \cdots & BK_{\tau^*_k} & \cdots & \cdots \\ I_n & & & & 0 \\ & I_n & & & 0 \\ & & \ddots & & \vdots \\ & & & I_n & 0 \end{pmatrix},$$

and I_n is the identity matrix with rank n.

With the closed-loop system model in (2.14), we then obtain the following stability criterion.

Theorem 2.1 *The closed-loop system in* (2.14) *is stable if there exists a positive definite solution* $P = P^T > 0$ *for the following* $(\bar{\tau}_{sc} + 1) \times (\bar{\tau}_{ca} + 1)$ *LMIs*

$$\Xi^T_{\tau^*_{sc,k},\tau^*_{ca,k}} P \Xi_{\tau^*_{sc,k},\tau^*_{ca,k}} - P < 0$$

Proof Let $V(k) = X^T(k)PX(k)$ be a Lyapunov candidate, and then its increment along the system in (2.14) can be obtained as

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) \\ &= X^T(k)(\Xi^T_{\tau^*_{sc,k},\tau^*_{ca,k}} P \Xi_{\tau^*_{sc,k},\tau^*_{ca,k}} - P)X(k) \\ &< 0 \end{aligned}$$

which completes the proof.

The following stability criterion for the closed-loop system model in (2.15) readily follows from Theorem 2.1.

Proposition 2.1 *The closed-loop system in* (2.15) *is stable if there exists a positive definite solution* $P = P^T > 0$ *for the following* $(\bar{\tau} + 1)$ *LMIs*

$$\Xi^T_{\tau^*_k} P \Xi_{\tau^*_k} - P < 0$$

2.3.2 A Delay Dependent Analysis Approach

Unlike the switched system theorem approach, in this subsection the closed-loop stability is investigated using a delay dependent analysis approach as in [117].

Without augmenting the system states as done in the last subsection, the closed-loop formula for the system in (2.1) using the packet-based controller in (2.5) can be obtained as

$$x(k+1) = Ax(k) + BK(\tau^*_{ca,k}, \tau^*_{sc,k})x(k - \tau^*_k), \tag{2.16}$$

It is noticed that in practice there is at least a single step delay in both the forward and backward channels, and therefore we have $\tau^*_k = \tau^*_{sc,k} + \tau^*_{ca,k} \geq 2$. Let $\bar{K} = \max_{2 \leq \tau^*_k \leq \bar{\tau}_{ca} + \bar{\tau}_{sc}} ||K(\tau^*_{ca,k}, \tau^*_{sc,k})||$ where $|| \cdot ||$ denotes the Euclidean norm. Then $BK(\tau^*_{ca,k}, \tau^*_{sc,k})$ can be represented by

$$BK(\tau^*_{ca,k}, \tau^*_{sc,k}) = B_m \cdot K'(\tau^*_{ca,k}, \tau^*_{sc,k}) \tag{2.17}$$

where $B_m = \bar{K}B$ is a constant matrix and $K'(\tau^*_{ca,k}, \tau^*_{sc,k}) = \frac{K(\tau^*_{ca,k}, \tau^*_{sc,k})}{\bar{K}}$.

It is readily to conclude that $||K'(\tau^*_{ca,k}, \tau^*_{sc,k})|| \leq 1, \ \forall\ 1 \leq \tau^*_{ca,k} \leq \bar{\tau}_{ca}, 1 \leq \tau^*_{sc,k} \leq \bar{\tau}_{sc}$.

Theorem 2.2 *If there exists* $P_i = P_i^T > 0, i = 1, 2, 3,\ X = \begin{pmatrix} X_{11} & X_{12} \\ * & X_{22} \end{pmatrix} \geq 0$, $N_i, i = 1, 2$ *with appropriate dimensions and* $\gamma > 0$ *satisfying the following two LMIs,*

$$\begin{pmatrix} X_{11} & X_{12} & N_1 \\ * & X_{22} & N_2 \\ * & * & P_3 \end{pmatrix} \geq 0 \tag{2.18}$$

$$\begin{pmatrix} \Phi_{11} & \Phi_{12} & (A-I)^T H & P_1 B_m \\ * & \Phi_{22} + \gamma I & 0 & 0 \\ * & * & -H & H B_m \\ * & * & * & -\gamma I \end{pmatrix} < 0 \tag{2.19}$$

where

$$\Phi_{11} = (\bar{\tau} - 1) P_2 + P_1 (A - I) + (A - I)^T P_1 + N_1 + N_1^T + \bar{\tau} X_{11},$$

$$\Phi_{12} = N_2^T - N_1 + \bar{\tau} X_{12},$$

$$\Phi_{22} = -P_2 - N_2 - N_2^T + \bar{\tau} X_{22},$$

$$H = P_1 + \bar{\tau} P_3,$$

then the closed-loop system in (2.16) *is stable.*

Proof Let $d_1 = 2, d_2 = \bar{\tau}$, and $\Delta A_d(k) = BK(\tau^*_{ca,k}, \tau^*_{sc,k})$ in Theorem 7.3 in [117], then the above theorem can be obtained using the same techniques as in [117].

Remark 2.4 Following the same procedure, the stability criterion for the system in (2.1) using the packet-based controller in (2.6) can be obtained analogously.

Remark 2.5 It is seen that the aforementioned stability criteria are simple propositions of existing results from switched system theory and delay-dependent analysis respectively. The former emphasizes on the "switch" property of packet-based control while the latter on the "time delay" property of the system. However, none of them is perfect: the switched system theory approach does not consider explicitly the time delay by augmenting the system states, whilst the delay-dependent analysis approach neglects the "switch" property which therefore leads to a stability criterion that is valid for any type of delay changes within an allowed upper bound. Hence, better stability analysis is still needed for the proposed packet-based control approach.

Up to now we have provided the packet-based control structure for NCSs whilst the controller design remains to be open. In fact, under the packet-based control framework, any conventional design approach is eligible to be applied to obtain the VFGs as in (2.5) and (2.6) provided it can result in a satisfactory system performance. In the following section, a Generalized Predictive Control (GPC) based controller is designed as an example.

2.4 Controller Design: A GPC-Based Approach

In GPC, an optimization process is repeated at every control instant to determine a sequence of forward control signals that optimize future open-loop plant behavior based on current system information. Different from conventional GPC implementations where only the first control prediction is actually applied to the plant, in this book the first $\bar{\tau}_{ca}+1$ (or $\bar{\tau}+1$ for the control law in (2.6)) forward control predictions are all used to implement the packet-based control approach proposed in the previous section.

Taking account of the communication constraints in NCSs which delay the sensing data, the objective function for open-loop optimization in GPC is therefore defined as follows,

$$\begin{aligned} J_{k,\tau_{sc,k}} =& X^T(k|k-\tau_{sc,k})QX(k|k-\tau_{sc,k}) \\ &+ U'^T(k|k-\tau_{sc,k})RU'(k|k-\tau_{sc,k}) \end{aligned} \tag{2.20}$$

where $J_{k,\tau_{sc,k}}$ is the objective function at time k, $U'(k|k-\tau_{sc,k}) = [u(k-\tau_{sc,k}|k-\tau_{sc,k})\ldots u(k+N_u-1|k-\tau_{sc,k})]^T$ is the FCS, $X(k|k-\tau_{sc,k}) = [x(k+1|k-\tau_{sc,k})\ldots x(k+N_p|k-\tau_{sc,k})]^T$ is the predictive state trajectory, Q and R are constant weighting matrices and N_p and N_u are the prediction horizon and the control horizon respectively.

The predictive states at time k based on the state at time $k-\tau_{sc,k}$ and the control sequences from $k-\tau_{sc,k}$ can be obtained by iteration as

$$\begin{aligned} x(k+j|k-\tau_{sc,k}) =& A^{j+\tau_{sc,k}}x(k-\tau_{sc,k}) \\ &+ \sum_{l=-\tau_{sc,k}}^{j-1} A^{j-l-1}Bu(k+l|k-\tau_{sc,k}),\ j=1,2,\ldots,N_p \end{aligned}$$

Thus we obtain

$$X(k|k-\tau_{sc,k}) = E_{\tau_{sc,k}} x(k-\tau_{sc,k}) + F_{\tau_{sc,k}} U'(k|k-\tau_{sc,k})$$

where $E_{\tau_{sc,k}} = [(A^{\tau_{sc,k}+1})^T \ \ldots \ (A^{\tau_{sc,k}+N_p})^T]^T$ and $F_{\tau_{sc,k}}$ is a block lower triangular matrix with its non-null blocks defined by $(F_{\tau_{sc,k}})_{ij} = A^{\tau_{sc,k}+i-j}B,\ j-i \le \tau_{sc,k}$.

The optimal control inputs can then be calculated by substituting the above equation to (2.20) and optimizing $J_{k,\tau_{sc,k}}$, which turns out to be state feedback control,

$$u(k+j|k-\tau_{sc,k}) = K_{\tau_{sc,k},j} x(k-\tau_{sc,k}),\ j = 0, 1, 2, \ldots, \bar{\tau}_{ca}$$

where $K_{\tau_{sc,k}} = [K^T_{\tau_{sc,k},0} \ldots K^T_{\tau_{sc,k},\bar{\tau}_{ca}}]^T$, $K_{\tau_{sc,k}}$ can be calculated by

$$K_{\tau_{sc,k}} = -M_{\tau_{sc,k}} (F^T_{\tau_{sc,k}} Q F_{\tau_{sc,k}} + R)^{-1} F^T_{\tau_{sc,k}} Q E_{\tau_{sc,k}}$$

and

$$M_{\tau_{sc,k}} = [0_{m(\bar{\tau}_{ca}+1)\times m\tau_{sc,k}} \ I_{m(\bar{\tau}_{ca}+1)\times m(\bar{\tau}_{ca}+1)} \ 0_{m(\bar{\tau}_{ca}+1)\times m(N_u-\bar{\tau}_{ca})}]$$

The FCS in (2.9) for Algorithm 2.1 can then be constructed by

$$U(k|k-\tau_{sc,k}) = K_{\tau_{sc,k}} x(k-\tau_{sc,k}) \tag{2.21}$$

The FCS in (2.13) for Algorithm 2.2 can also be constructed analogously.

Remark 2.6 (*State observer*) If the state vector x is not available, an observer must be included

$$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + Bu(k) + L(y_m(k) - C\hat{x}(k|k-1)) \tag{2.22}$$

where $\hat{x}(k)$ is the observed state at time k, and $y_m(k)$ is the measured output. If the plant is subject to white noise disturbances affecting the process and the output with known covariance matrices, the observer becomes a Kalman filter and the gain L is calculated solving a Riccati equation.

Remark 2.7 In [107], state feedback $u_k = K\hat{x}_{k|k-\tau_{sc,k}}$ was also used, where K was artificially chosen without consideration of the communication constraints and $\hat{x}_{k|k-\tau_{sc,k}}$ depends on "the state estimation $\hat{x}_{k-\tau_{sc,k}|k-\tau_{sc,k}-1}$, the past control input up to u_{k-1}, and the past output up to $y_{k-\tau_{sc,k}}$ of the system". However, under the compensation scheme in the forward channel in [107], the whole sequence of the optimal forward control signals $U(k|k-\tau^*_{sc,k})$ is sent to the actuator and only one of them is chosen to be applied to the plant. Thus, unless information from the actuator is received we have no idea which control prediction was really used if the data packets in the forward channel were arbitrarily delayed. Hence, the use of the previous control inputs implies an additional communication channel which can send the applied control inputs to the controller efficiently. Without such a channel, the approaches proposed in these publications are only applicable to such a situation where there is no delay or data packet dropout in the forward channel. This requirement is relaxed in this section by redesigning the controller where the objective function includes as part of it the previous control increment sequence from $k - \tau_{sc,k}$ to $k - 1$. As a result, the FCS at time k are only based on data up to time $k - \tau_{sc,k}$, which is always available in practice.

2.5 Numerical and Experimental Examples

In this section, numerical and experimental examples are considered to illustrate the effectiveness of the proposed packet-based control approach to NCSs.

Example 2.1 A second order system in (2.1) is adopted, which is open-loop unstable with the following system matrices,

$$A = \begin{pmatrix} 0.98 & 0.1 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0.04 \\ 0.1 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

In order to illustrate the effectiveness of the proposed packet-based control approach compared with conventional design approach, the Linear Quadratic Optimal (LQR) control method is used to design a state feedback law for this system without consideration of the communication constraints, which yields the time-invariant feedback gain $K_{LQR} = [0.7044\ 1.3611]$. In the simulation, the initial state $x_0 = [-1\ -1]^T$, the upper bounds of the delays and continuous dropout (disorder) are $\bar{\tau} = 3$, $\bar{\tau}_{ca} = 2$, $\bar{\tau}_{sc} = 1$, and the control and prediction horizon in the GPC-based controller proposed in Sect. 2.4 are set as $N_u = 8$, $N_p = 10$ respectively. The delays in both channels are set to vary arbitrarily within their upper bounds.

The simulation results show that it is unstable using this LQR controller (Fig. 2.5) while it is stable using the packet-based control approach (Fig. 2.6) in the presence of communication constraints.

Example 2.2 (*Example 1 in* [107]) The system matrices for the system in (2.1) are as follows,

$$A = \begin{pmatrix} 1.0100 & 0.2710 & -0.4880 \\ 0.4820 & 0.1000 & 0.2400 \\ 0.0020 & 0.3681 & 0.7070 \end{pmatrix}, B = \begin{pmatrix} 5 & 5 \\ 3 & -2 \\ 5 & 4 \end{pmatrix}, C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 3 & 1 \end{pmatrix}.$$

In [107], the above system is illustrated to be stable with the observer in (2.22) where $\bar{\tau}_{sc} = 2$, $\bar{\tau}_{ca} = 1$, and

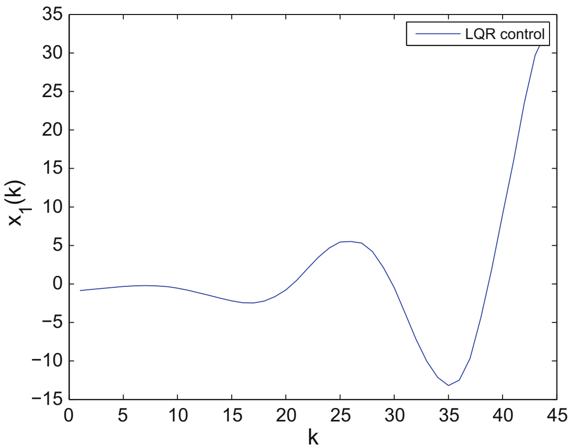


Fig. 2.5 Example 2.1. System is unstable using LQR controller

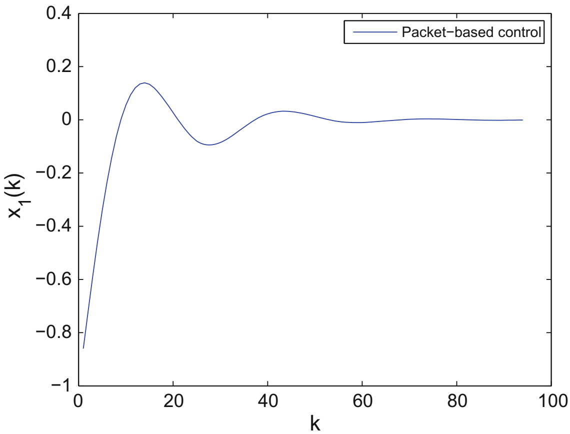


Fig. 2.6 Example 2.1. System is stable using packet-based controller

$$L = \begin{pmatrix} -0.3614 & 0.3326 \\ 0.0332 & 0.0576 \\ 0.2481 & -0.0750 \end{pmatrix}, K = \begin{pmatrix} 0.5858 & -0.1347 & -0.4543 \\ -0.5550 & 0.0461 & 0.4721 \end{pmatrix}.$$

However, using the packet-based control approach, this system is unstable with the same $\bar{\tau}_{sc}$, $\bar{\tau}_{ca}$ and L (see Fig. 2.7). Other parameters: $(N_u = 8, N_p = 10)$. This fact seems to mean the approach in [107] is better than the approach in this chapter, but we need to remember that the approach in [107] takes advantage of more information to design the predictive controller and some of the information used is not easy to obtain in practice (Remark 2.7). On the other hand, the simulation results do illustrate that the VFG scheme in this chapter is superior to the previous Fixed Feedback Gain (FFG) scheme in [107], where the same system is stable using the approach in this chapter when $\bar{\tau}_{sc} = \bar{\tau}_{sc} = 1$ (Fig. 2.8) and yet is unstable using the same state feedback in (2.16) with the fixed K above (Fig. 2.9).

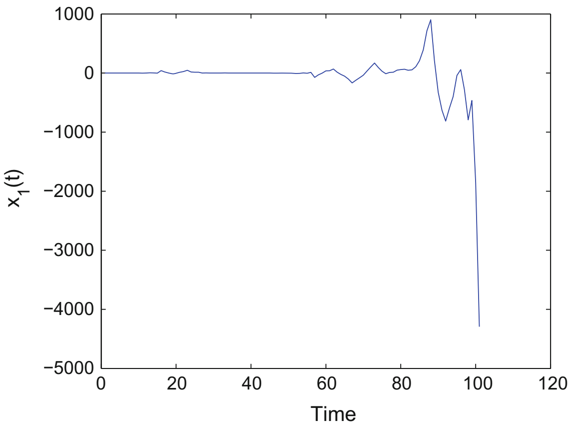


Fig. 2.7 Example 2.2. Packet-based control, unstable, $\bar{\tau}_{sc} = 2$, $\bar{\tau}_{ca} = 1$

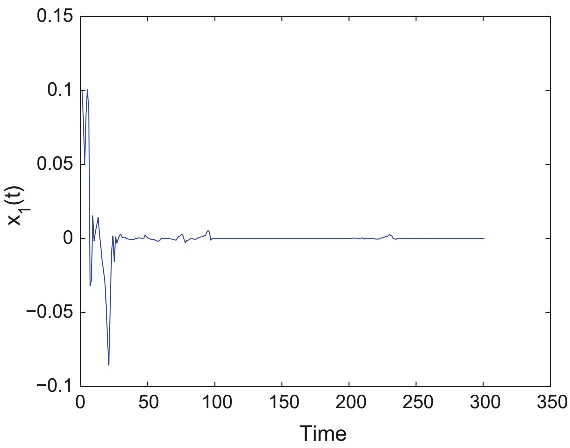


Fig. 2.8 Example 2.2. VFG scheme, stable, $\bar{\tau}_{sc} = 1$, $\bar{\tau}_{ca} = 1$

Example 2.3 The system matrices for the system in (2.1) are set as

$$A = \begin{pmatrix} 0.7 & 0.2 \\ 0.3 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 0.05 \\ 0.2 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

This system can be shown using Theorem 2.2 to be stable under $\bar{\tau}_{sc} = 3$, $\bar{\tau}_{ca} = 2$, $N_u = 8$, $N_p = 10$. The simulation result is illustrated in Fig. 2.10.

Example 2.4 In this example, an Internet-based test rig is used to verify the effectiveness of the packet-based control approach. This test rig consists of a plant (DC servo system, see Fig. 2.11a) which is located in the University of Glamorgan, Pontypridd, UK, and a remote controller which is located in the Institute of Automation, Chinese Academy of Sciences, Beijing, China (see Fig. 2.11b). The plant and the controller are connected via the Internet, whose IP addresses are

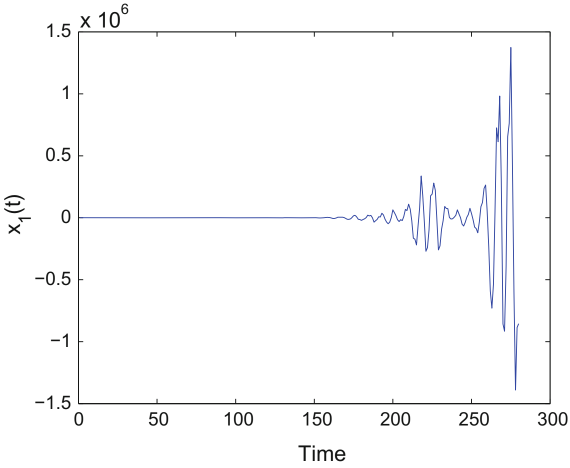


Fig. 2.9 Example 2.2. FFG scheme, unstable, $\bar{\tau}_{sc} = 1$, $\bar{\tau}_{ca} = 1$

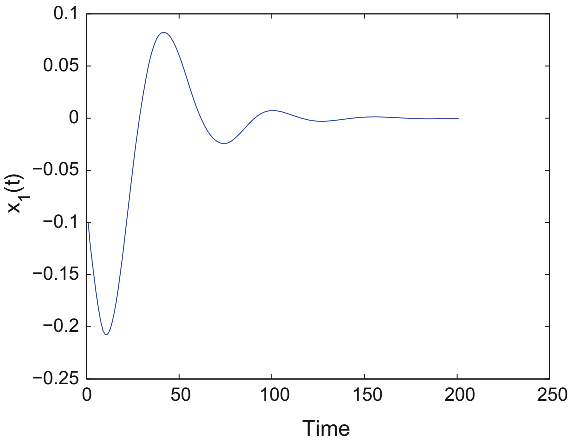


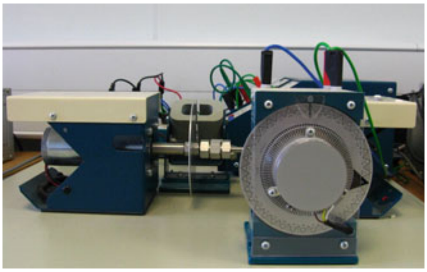
Fig. 2.10 Example 2.3. Packet-based control, stable, $\bar{\tau}_{sc} = 3$, $\bar{\tau}_{ca} = 2$

193.63.131.219 and 159.226.20.109 respectively. A web-based laboratory is also available at www.ncslab.net to implement experiments online. For further information of this test rig, the reader is referred to [38, 118].

The DC servo system is identified by [118] to be a third-order system and in state-space description has the following system matrices,

$$A = \begin{pmatrix} 1.12 & 0.213 & -0.333 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, C = \begin{pmatrix} 0.0541 & 0.0050 & 0.0001 \end{pmatrix}.$$

To enable the use of state feedback in the packet-based control approach, a state observer as in Remark 2.6 is designed with $L = [6\ 6\ 6]^T$. The packet-based controller is calculated by using the GPC-based controller design approach in Sect. 2.4. To this



(a) The DC servo plant in the University of Glamorgan.

(b) The network controller in the Chinese Academy of Sciences.

Fig. 2.11 The Internet-based test rig

end, the upper bounds of the network-induced delays (data packet dropout as well) in both forward and backward channels are assumed to be 4 steps of the sampling period (The sampling period is set as 0.04 s and thus the delay bounds are 0.16 s for both backward and forward channel delays.), since typically the round trip delay in the experiment is not larger than 0.32 s. The packet-based controller can then be obtained as

$$K = [K_0^T \; K_1^T \; K_2^T \; K_3^T \; K_4^T]^T, K_0 = \begin{pmatrix} -1.3217 & 0.1276 & 0.4296 \\ -0.1356 & 0.0306 & 0.0445 \\ 0.2688 & -0.0220 & -0.0816 \\ 0.1255 & -0.0096 & -0.0396 \\ 0.0610 & -0.0061 & -0.0190 \end{pmatrix},$$

$$K_1 = \begin{pmatrix} -0.2193 & 0.0219 & 0.0844 \\ 0.2177 & -0.0032 & -0.0662 \\ 0.1298 & -0.0087 & -0.0381 \\ 0.0621 & -0.0035 & -0.0198 \\ 0.0114 & -0.0014 & -0.0035 \end{pmatrix}, K_2 = \begin{pmatrix} 0.1120 & 0.0005 & -0.0201 \\ 0.1183 & 0.0032 & -0.0348 \\ 0.0726 & -0.0050 & -0.0201 \\ 0.0192 & -0.0007 & -0.0062 \\ 0.0035 & -0.0009 & -0.0010 \end{pmatrix},$$

$$K_3 = \begin{pmatrix} 0.0894 & 0.0021 & -0.0130 \\ 0.0832 & 0.0056 & -0.0239 \\ 0.0398 & -0.0028 & -0.0099 \\ 0.0106 & -0.0001 & -0.0035 \\ 0.0007 & -0.0007 & -0.0002 \end{pmatrix}, K_4 = \begin{pmatrix} 0.0721 & 0.0030 & -0.0076 \\ 0.0515 & 0.0073 & -0.0140 \\ 0.0267 & -0.0021 & -0.0058 \\ 0.0059 & 0.0001 & -0.0021 \\ 0.0005 & -0.0007 & -0.0001 \end{pmatrix},$$

where the subscripts of K_0, K_1, K_2, K_3 and K_4 are with respect to different backward channel delays.

The comparison between the simulation and experimental results is illustrated in Fig. 2.12, which shows that the packet-based control approach is valid in practice.

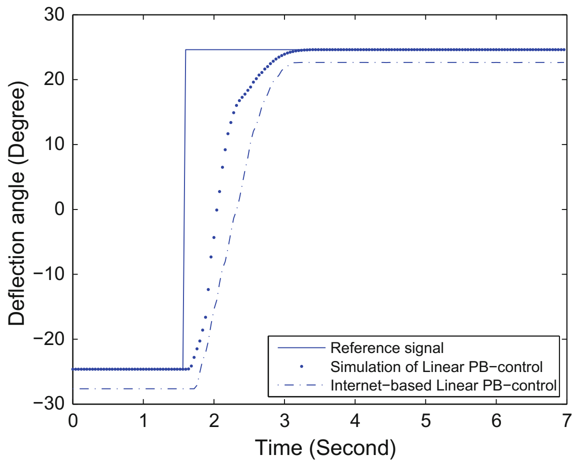


Fig. 2.12 Example 2.4. Comparison between simulation and experimental results of linear packet-based control system

It is seen however that there is some difference between simulation and experimental results. Several possible reasons may contribute to this difference: (1) the identified model for the DC servo system may not be accurate enough; (2) the dead zone of the DC servo plant has not been considered; and (3) the measurement of the network-induced delays is not accurate in practice.

2.6 Summary

Since NCSs is actually the integration of conventional control systems and the communication networks, a natural way to deal with the communication constraints is to put the problem under the co-design framework—design with the integration of control theory and communication technology. Based on the observation of the packet-based transmission in the networked control environment, a packet-based control approach was proposed for NCSs, which can effectively deal with the network-induced delay, data packet dropout and data packet disorder simultaneously. Numerical and experimental examples illustrated the effectiveness of the proposed approach with a GPC-based controller.

Chapter 3
Packet-Based Control for Networked Hammerstein Systems

This chapter extends the application of the packet-based control approach proposed in Chap. 2 to a class of input nonlinear systems described by a Hammerstein model, where a static nonlinear input process is present in the system. A "two-step" approach is adopted to separate the nonlinear input process from the system so that the packet-based control approach can be applied to such systems with minor modifications. Two descriptions of the Hammerstein model, i.e. the input-output description and the state-space description, are considered, where the stability analysis of the former in the case of arbitrary delays proved to be a difficulty due to the constraints of the Popov criterion, while it is solved in the more generalized state-space description by using switched system theory.

This chapter is organized as follows. After presenting the two system descriptions in Sect. 3.1, the design of the packet-based controller for networked Hammerstein systems is then presented in Sect. 3.2, which differs from the standard packet-based controller in Chap. 2 in the compensation for the nonlinear input process. The stability criteria for both descriptions are then obtained in Sect. 3.3 and numerical and experimental examples are presented in Sect. 3.4. Section 3.5 concludes the chapter.

3.1 System Description

The Hammerstein model is a particular category of nonlinear systems which consists of a cascade connection of a static nonlinear input process followed by a dynamic Linear Time-Invariant (LTI) system. This category of nonlinear systems is important in theory, and applies to a number of practical applications, see, e.g., in [119–123]. In this chapter, the Hammerstein model is assumed to be controlled over the network, see Fig. 3.1 for its configuration.

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_3

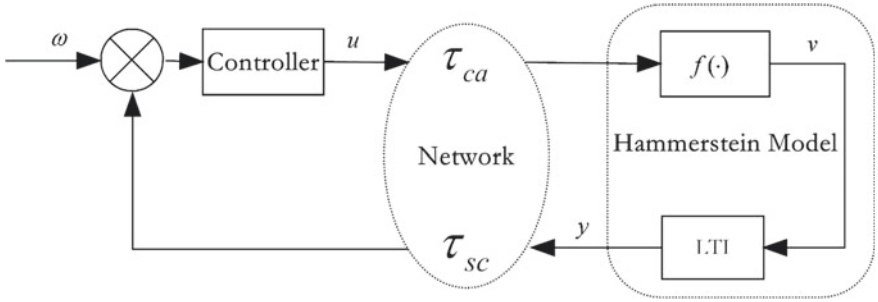


Fig. 3.1 The block diagram of networked Hammerstein systems

Two descriptions of the Hammerstein model are considered in this chapter, shown in (3.1) for the input-output description and in (3.2) for the state-space description, respectively.

In the input-output description, the Hammerstein model is represented by the following formulas with the combination of the Controlled Auto Regressive Integrated Moving Average (CARIMA) model and a static nonlinear function $f(\cdot)$,

$$\mathcal{S}\begin{cases} {}_{I1} : ay(k) = b_{I1}v(k-1) & (3.1a) \\ v(k) = f(u(k)) & (3.1b) \end{cases}$$

where $u, v, y \in \mathbb{R}$ are the input, intermediate input and output respectively, $a = 1 + a_1 z^{-1} + \ldots + a_n z^{-n}$, $b = b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}$ with $a_n \neq 0$, $b_m \neq 0$, and $f(\cdot) : \mathbb{R} \to \mathbb{R}$ is a memoryless static nonlinear function with $f(0) = 0$.

In state-space form, the Hammerstein model with Single-Input-Single-Output (SISO) is represented by

$$\mathcal{S}\begin{cases} {}_{I2} : x(k+1) = Ax(k) + b_{I2}v(k) & (3.2a) \\ y(k) = cx(k) & (3.2b) \\ v(k) = f(u(k)) & (3.2c) \end{cases}$$

where $x \in \mathbb{R}^n$ is the system state and $u, v, y, f(\cdot)$ are defined in (3.1).

In the following section, the packet-based control approach is extended to networked Hammerstein systems with the adoption of a two-step approach to separate the static nonlinear input process from the system. Using this two-step approach, the packet-based control approach proposed in Chap. 2 can be readily implemented with a compensation process for the static input process in the Hammerstein model.

3.2 Packet-Based Control for Networked Hammerstein Systems

It is obvious that the challenge of applying the packet-based control approach to networked Hammerstein systems in $\mathcal{S}_{I1}$ and $\mathcal{S}_{I2}$ is primarily caused by the efficient treatment of the nonlinear input process in the Hammerstein model. Fortunately, it is noticed that the nonlinear input process in the Hammerstein model is memoryless, static. This observation enables the separation of the nonlinear input process from the whole system by using an inverse process, and then design only for the linear part of the Hammerstein model the packet-based controller which has already been done in Chap. 2. This is also why we call it a "two-step" approach to networked Hammerstein systems, see Fig. 3.2 for its typical setup.

Under the two-step approach framework, in this section the intermediate FCS (and Forward Control Increment Sequence (FCIS)) for the linear part of the Hammerstein system are obtained first, and an inverse compensation scheme for the nonlinear input process is then proposed.

3.2.1 Intermediate FCS (FCIS)

In this subsection, we design for the linear part of the Hammerstein system (3.1a) of $\mathcal{S}_{I1}$ and (3.2a), (3.2b) of $\mathcal{S}_{I2}$) the intermediate FCS $V(k|k-\tau_{sc,k})$ (for system $\mathcal{S}_{I1}$) and intermediate FCIS $\Delta V(k|k-\tau_{sc,k})$ (for system $\mathcal{S}_{I2}$), which are defined in (3.3a) and (3.3b) respectively,

$$V(k|k-\tau_{sc,k}) = [v(k|k-\tau_{sc,k}) \cdots v(k+N_{u-1}|k-\tau_{sc,k})]^T \tag{3.3a}$$

$$\Delta V(k|k-\tau_{sc,k}) = [\,\Delta v(k|k-\tau_{sc,k}) \cdots \Delta v(k+N_{u-1}|k-\tau_{sc,k})\,]^T \tag{3.3b}$$

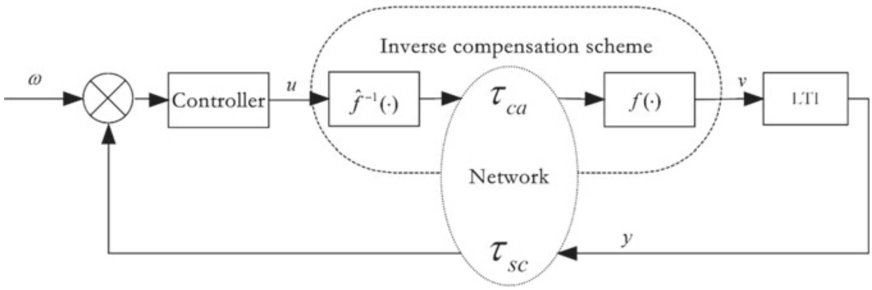


Fig. 3.2 Two-step approach to networked Hammerstein systems

where $\Delta v(k+i|k-\tau_{sc,k}) = v(k+i|k-\tau_{sc,k}) - v(k+i-1|k-\tau_{sc,k}), i = 0, 1, 2, \ldots, N_u - 1$. The FCS (FCIS) is obtained using the GPC approach. The procedure is similar to that in Sect. 2.4 but with minor modifications. One can refer to Sect. 2.4 and [124, 125] for more information on the GPC approach.

Remark 3.1 Note that in (3.3a) and (3.3b) N_u steps of forward control predictions are all used to construct the intermediate FCS (FCIS), which is actually unnecessary in practice, see Sect. 2.4; we do so only for simplicity of presentation. Therefore, in practical implementations, selection of appropriate length of FCS (FCIS) is needed.

3.2.1.1 FCS for the Input-Output Description

Without consideration of the nonlinear input process of the Hammerstein system in (3.1b), the Linear Generalized Predictive Control (LGPC) problem for (3.1a) is solved for the following objective function:

$$J^{I1}_{k,\tau_{sc,k}} = ||Y(k|k-\tau_{sc,k}) - \varpi||^2_Q + ||\Delta V'(k|k-\tau_{sc,k})||^2_R \tag{3.4}$$

where $\varpi = [\omega\ \omega\ \cdots\ \omega]^T_{N_p+\tau_{sc,k}}$, ω is the set-point, $Q_{N_p+\tau_{sc,k}}$, $R_{N_u+\tau_{sc,k}}$ are diagonal weight matrixes, $Y(k|k-\tau_{sc,k}) = [y(k-\tau_{sc,k}+1|k-\tau_{sc,k})\ y(k-\tau_{sc,k}+2|k-\tau_{sc,k})\ \cdots\ y(k+N_p|k-\tau_{sc,k})]^T$, $y(k+i|k-\tau_{sc,k}), i = -\tau_{sc,k}+1, \cdots, N_p$ are the predicted outputs, $\Delta V'(k|k-\tau_{sc,k}) = [\Delta v(k-\tau_{sc,k}|k-\tau_{sc,k})\ \Delta v(k-\tau_{sc,k}+1|k-\tau_{sc,k})\ \cdots\ \Delta v(k+N_u-1|k-\tau_{sc,k})]^T$, N_p is the predictive horizon and $||\psi||^2_\Theta$ means $\psi^T\Theta\psi$.

Introduce the following Diophantine equations for $j = -\tau_{sc,k}+1, \cdots, N_p$,

$$1 = E_j a\Delta + z^{-j-\tau_{sc,k}} F_j$$

and

$$E_j b_{I1} = z^{-(j+\tau_{sc,k})} E^0_j + G_j, \text{ when } m > 0$$

where $E_j = 1 + e_{j,1}z^{-1} + \ldots + e_{j,j+\tau_{sc,k}-1}z^{-(j+\tau_{sc,k}-1)}$, $F_j = f_{j,0} + f_{j,1}z^{-1} + \cdots + e_{j,n}z^{-n}$, $E^0_j = e^0_{j,0} + e^0_{j,1}z^{-1} + \ldots + e^0_{j,m-1}z^{-(m-1)}$, and $G_j = g_{j,0} + g_{j,1}z^{-1} + \cdots + g_{j,j-1}z^{-(j+\tau_{sc,k}-1)}$.

Define $E = [E^0_{-\tau_{sc,k}+1}\ E^0_{-\tau_{sc,k}+2}\ \cdots\ E^0_{N_p}]^T$, if $m > 0$; $0_{(N_p+\tau_{sc,k})\times 1}$, otherwise; $G \in R^{(N_p+\tau_{sc,k})\times(N_u+\tau_{sc,k})}(z^{-1})$ with all the entries 0 but $G(j,j) = G_j$ if $m > 0$; $G(j,j) = E_j b_{I1}$, otherwise, for $j = -\tau_{sc,k}+1, -\tau_{sc,k}+2, \cdots, N_u$, and $F = [F_{-\tau_{sc,k}+1}\ F_{-\tau_{sc,k}+2}\ \cdots\ F_{N_p}]^T$, $T = (G^TQG+R)^{-1}G^TQ$, $Y_0(k|k-$

$\tau_{sc,k}) = E\Delta v(k-\tau_{sc,k}-1) + Fy(k-\tau_{sc,k})$, $M = [1\ 1\ \cdots\ 1]^T_{(N_u+\tau_{sc,k})\times 1}$,

$$P = [0_{N_u\times\tau_{sc,k}} I_{N_u\times N_u}],\ S = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{N_u+\tau_{sc,k}}.$$

The FCS from k to $k+N_u-1$ using objective function in (3.4) based on the information up to time $k-\tau_{sc,k}$ is then obtained as

$$V(k|k-\tau_{sc,k}) = P(Mv(k-\tau_{sc,k}-1) + ST(\varpi - Y_0(k|k-\tau_{sc,k})) \tag{3.5}$$

3.2.1.2 FCIS for the State-Space Description

For the state-space description, the following objective function is adopted,

$$J^{I2}_{k,\tau_{sc,k}} = \sum_{j=N_1}^{N_2} q_j(\hat{y}(k+j|k-\tau_{sc,k}) - \omega(k+j))^2 + \sum_{j=1}^{N_u} r_j(\Delta v(k+j-1))^2 \tag{3.6}$$

Let $\bar{x}(k) = [x^T(k)\ v(k-1)]^T$, and then system $\mathcal{S}_{I2}$ can be transformed to $\mathcal{S}'_{I2}$ as follows,

$$\mathcal{S}'_{I2}: \begin{cases} \bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{b}\Delta v(k) & (3.7a) \\ y(k) = \bar{c}\bar{x}(k) & (3.7b) \end{cases}$$

where $\bar{A} = \begin{pmatrix} A & b_{I2} \\ 0 & 1 \end{pmatrix}$, $\bar{b} = \begin{pmatrix} b_{I2} \\ 1 \end{pmatrix}$, $\bar{c} = \begin{pmatrix} c & 0 \end{pmatrix}$.

The forward output predictions at time k based on the information of the state on time $k-\tau_{sc,k}$ and control signals from time $k-\tau_{sc,k}-1$ is

$$\hat{y}(k+j|k-\tau_{sc,k}) = \bar{c}\bar{A}^{j+\tau_{sc,k}}\bar{x}(k-\tau_{sc,k}) + \sum_{l=-\tau_{sc,k}}^{j-1} \bar{c}\bar{A}^{j-l-1}\bar{b}\Delta v(k+l|k-\tau_{sc,k})$$

Let $\hat{Y}(k|k-\tau_{sc,k}) = [\hat{y}(k+N_1|k-\tau_{sc,k})\ \cdots\ \hat{y}(k+N_2|k-\tau_{sc,k})]^T$, $\Delta V'(k|k-\tau_{sc,k}) = [\Delta v(k-\tau_{sc,k}|k-\tau_{sc,k})\ \cdots\ \Delta v(k+N_u-1|k-\tau_{sc,k})]^T$. Then we obtain

$$\hat{Y}(k|k-\tau_{sc,k}) = E_{\tau_{sc,k}}\bar{x}(k-\tau_{sc,k}) + F_{\tau_{sc,k}}\Delta V'(k|k-\tau_{sc,k})$$

where $E_{\tau_{sc,k}} = [(\bar{c}\bar{A}^{N_1+\tau_{sc,k}})^T\ (\bar{c}\bar{A}^{N_1+\tau_{sc,k}+1})^T\ \cdots (\bar{c}\bar{A}^{N_2+\tau_{sc,k}})^T]^T$ and $F_{\tau_{sc,k}}$ is a $(N_2 - N_1 + 1) \times (N_u + \tau_{sc,k})$ matrix with the non-null entries defined by $(F_{\tau_{sc,k}})_{ij} =$

$\bar{c}\bar{A}^{N_1+\tau_{sc,k}+i-j-1}\bar{b}$, $j-i \le N_1+\tau_{sc,k}-1$. Note here that $E_{\tau_{sc,k}}$ and $F_{\tau_{sc,k}}$ vary with different $\tau_{sc,k}$s.

Let $\varpi_k = [\omega(k+N_1)\ \cdots\ \omega(k+N_2)]^T$, and then the optimal predictive control increments from k to $k+N_u-1$ can be calculated by following standard techniques in GPC approach,

$$\Delta V(k|k-\tau_{sc,k}) = M_{\tau_{sc,k}}(\varpi_k - E_{\tau_{sc,k}}\bar{x}(k-\tau_{sc,k})) \tag{3.8}$$

where $M_{\tau_{sc,k}} = H_{\tau_{sc,k}}(F^T_{\tau_{sc,k}} Q F_{\tau_{sc,k}} + R)^{-1} F^T_{\tau_{sc,k}} Q$, Q, R are diagonal matrices with $Q_{i,i} = q_i$, $R_{i,i} = r_i$ respectively and $H_{\tau_{sc,k}} = [0_{N_u\times\tau_{sc,k}} I_{N_u\times N_u}]$, $I_{N_u\times N_u}$ is the identity matrix with rank N_u.

Remark 3.2 Normally, the minimum prediction horizon can be set as 1. Rewrite the maximum prediction horizon N_2 as N_p. The following constraint between N_u and N_p needs to be always held in order to implement the LGPC method successfully,

$$N_u \le N_p \tag{3.9}$$

3.2.2 The Nonlinear Input Process

With the designed intermediate FCIS in (3.8), the nonlinear input process in $\mathcal{S}_{I2}$ is first considered as follows.

Assume the nonlinear function $f(\cdot)$ in (3.2c) is invertible and denote its inverse by $\hat{f}^{-1}(\cdot)$. Then we obtain

$$\Delta u(k|k-\tau_{sc,k}) = \hat{f}^{-1}(\Delta v(k|k-\tau_{sc,k})) \tag{3.10}$$

Thus, at every time instant k, the intermediate control increments $\Delta v(k|k-\tau_{sc,k})$, $k = 1, 2, \cdots, N_u$ can be obtained from (3.8), and then the real control increments $\Delta u(k|k-\tau_{sc,k})$, $k = 1, 2, \cdots, N_u$ can be calculated from (3.10) thus enabling the control law to be defined for system $\mathcal{S}'_{I2}$.

If $\Delta u(k|k-\tau_{sc,k})$ can be calculated accurately using (3.10), thus enabling the function $\hat{f}^{-1}(\cdot)$ to be exactly known, then the system with compensation for the nonlinear input process is equivalent to LGPC and the system is stable if and only if the linear part of system $\mathcal{S}_{I2}$ with LGPC is stable. However, in practice, it is usually impossible to calculate $\Delta u(k|k-\tau_{sc,k})$ that accurately, i.e., $\hat{f}^{-1}(f(\cdot)) \not\equiv 1(\cdot)$. This inaccuracy introduces to the LGPC a nonlinear disturbance, which makes the stability analysis difficult.

For simplicity of notation, let $\hat{\bar{f}}^{-1}(\cdot): \mathbb{R}^{N_u} \to \mathbb{R}^{N_u}$ with $\hat{\bar{f}}^{-1}(\Delta V(k|k-\tau_{sc,k})) = [\hat{f}^{-1}(\Delta v(k|k-\tau_{sc,k}))\ \cdots\ \hat{f}^{-1}(\Delta v(k+N_u-1|k-\tau_{sc,k}))]^T$. Then from the discussion above, the real FCIS for system $\mathcal{S}_{I2}$ can be obtained as

$$\Delta U(k|k-\tau_{sc,k}) = \hat{\hat{f}}^{-1}(\Delta V(k|k-\tau_{sc,k}) \tag{3.11}$$

where $\Delta U(k|k-\tau_{sc,k}) = [\Delta u(k|k-\tau_{sc,k}) \ \cdots \ \Delta u(k+N_u-1|k-\tau_{sc,k})]^T$.

Remark 3.3 Note that the control increment instead of the control signal itself is used in the compensation for the nonlinear input process in (3.10). Though the use of control increments complicates the problem in that the past control increments are also needed to determine the current control increment, it is inevitable since the objective function to be optimized in (3.6) takes the form of control increments. In order to implement the predictive controller in this chapter, the past control increments are sent to the controller as well as the state information, which is different from both CCSs and standard packet-based control approach. Note that for a system without a nonlinear input process in (3.2c), it makes no difference whether the intermediate control increment or the intermediate control signal itself is used to calculate the real control signal, while for system $\mathcal{S}_{I2}$, generally, these two methods give different control input at time k, i.e., $f(\Delta v(k)) \neq f(v(k)) - f(v(k-1))$.

Remark 3.4 For system $\mathcal{S}_{I1}$, the real FCS can be obtained analogously as follows using the similar inverse compensation scheme as aforementioned

$$U(k|k-\tau_{sc,k}) = \hat{\hat{f}}^{-1}(V(k|k-\tau_{sc,k}) \tag{3.12}$$

where $U(k|k-\tau_{sc,k}) = [u(k|k-\tau_{sc,k}) \ \cdots \ u(k+N_u-1|k-\tau_{sc,k})]^T$.

3.2.3 *Packet-Based Control for Networked Hammerstein Systems*

With the aforementioned discussion, the packet-based controllers for networked Hammerstein systems have been successfully obtained for both descriptions, which enables the packet-based control structure proposed in Sect. 2.2 to be implemented. Since the packet-based control structure for networked Hammerstein systems is exactly the same as the linear system case in Sect. 2.2, we therefore will not address the design details but only present the packet-based control algorithm as follows, where we take the state-space description as an example; the reader is referred to Sect. 2.2 for more details of the packet-based control structure.

The block diagram of the packet-based control structure for networked Hammerstein systems in state-space description is shown in Fig. 3.3.

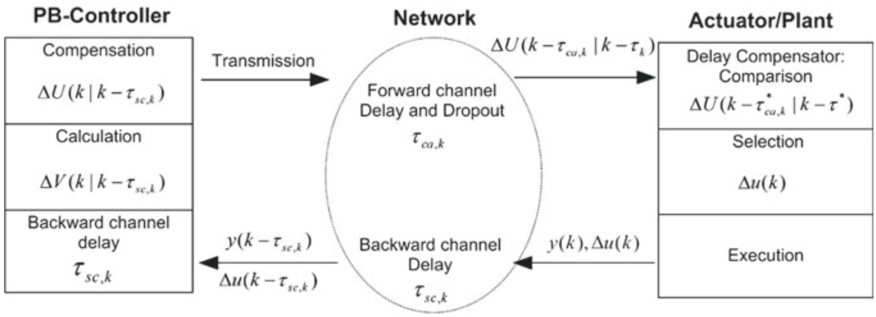


Fig. 3.3 Packet-based control for networked Hammerstein systems

---

Algorithm 3.1 Packet-based control for networked Hammerstein systems

---

Calculation. The packet-based controller calculates the intermediate FCIS $\Delta V(k|k-\tau_{sc,k})$ using (3.8) and then obtains the real FCIS $\Delta U(k|k-\tau_{sc,k})$ by compensating for the nonlinear input process using (3.11)
Forward-transmission. $\Delta U(k|k-\tau_{sc,k})$ is packed and sent to the actuator simultaneously with time stamps k and $\tau_{sc,k}$
Comparison. CAS updates its FCIS according to the time stamps once a data packet arrives
Execution. An appropriate control increment signal is picked out from CAS and applied to the plant
Backward-transmission. The information of the applied control increment with the sensing state is sent to the controller

---

3.3 Stability Analysis of Packet-Based Networked Hammerstein Systems

In this section, the stability conditions of networked Hammerstein systems using the packet-based control approach are investigated, for both descriptions in (3.1) and (3.2). For the input-output description, the Popov criterion is adopted from which a stability criterion is derived only for a constant network-induced delay whereas for the state-space description, switched system theory is applied which yields a stability criterion that is valid for arbitrary network-induced delays.

3.3.1 Stability Criterion in Input-Output Description

From the design of the CAS in Sect. 2.2, the control action adopted by the actuator at time k is readily obtained as

$$u(k) = d^T_{\tau^*_{ca,k}} U(k - \tau^*_{ca,k}|k - \tau^*_k) \tag{3.13}$$

where $d_{\tau^*_{ca,k}}$ is a $N_u \times 1$ column vector with all entries 0 but the $\tau^*_{ca,k}$th being 1, and τ^*_k is the round trip delay with respect to $\tau^*_{ca,k}$, i.e. $\tau^*_k = \tau^*_{ca,k} + \tau^*_{sc,k}$.

Combining (3.1), (3.5), (3.12) , (3.13), the packet-based control approach applied to the networked Hammerstein system in (3.1) can then be fully described by the following system $\mathcal{S}^*_{I1}$ (ω is set to 0 without loss of generality),

$$\mathcal{S}^*_{I1}: \begin{cases} ay(k) = b_{I1}v(k-1) & (3.14a) \\ v(k) = f(u(k)) & (3.14b) \\ u(k) = d^T_{\tau^*_{ca,k}} \hat{\hat{f}}^{-1}(V(k-\tau^*_{ca,k}|k-\tau^*_k)) & (3.14c) \\ V(k-\tau^*_{ca,k}|k-\tau^*_k) = L_\tau(z^{-1})y(k-\tau^*_k) & (3.14d) \end{cases}$$

where $L_\tau(z^{-1}) = (z^{-\tau^*_k-1}PMd^T_{\tau^*_{ca,k}} - z^{-\tau^*_k-1}PCDE\Delta d^T_{\tau^*_{ca,k}} - I)^{-1}PSTF$ and (3.14d) is obtained by noticing

$$\begin{aligned} V(k-\tau^*_{ca,k}|k-\tau^*_k) = &\ PMv(k-\tau^*_k-1) \\ &- PSTE\Delta v(k-\tau^*_k-1) - PSTFy(k-\tau^*_k), \end{aligned}$$

$$v(k-\tau^*_k-1) = z^{-\tau^*_k-1}d^T_{\tau^*_{ca,k}}V(k-\tau^*_{ca,k}|k-\tau^*_k),$$

and substituting the latter to the former.

In order to derive the stability criterion for system $\mathcal{S}^*_{I1}$, the following Popov criterion is required.

Lemma 3.1 (Popov criterion, see [120]) *Suppose that $H(z^{-1})$ in Fig. 3.4 is stable and $0 \le \Phi(\theta) \le \mathcal{K}\theta$. Then the closed-loop system is stable if $1/\mathcal{K} + Re(H(z^{-1}) > 0, \forall |z| = 1$.*

In the case of constant delays, we have that $\tau^* = \tau^*_k, \tau^*_{ca} = \tau^*_{ca,k}, \tau^*_{sc} = \tau^*_{sc,k}, \forall k$, are all constant. Apply Lemma 3.1 to system $\mathcal{S}^*_{I1}$ and denote the characteristic polynomial of a transfer function $H(z^{-1})$ by $\delta(H(z^{-1}))$, we then obtain the following theorem.

Theorem 3.1 *Suppose the roots of $\delta(A_\tau(z^{-1})) = 0$ are all located in the unit circle. Then the system in (3.14) is stable if there exists a positive constant $\mathcal{K}$ such that*

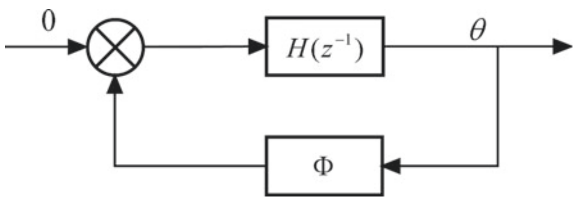


Fig. 3.4 Popov criterion

1. *the input nonlinearity of the plant satisfies*

$$0 \leq v \leq \mathcal{K}\bar{v}, \tag{3.15a}$$

2. *the network-induced delay satisfies*

$$\frac{1}{\mathcal{K}} + Re\{A_\tau(z^{-1})\} > 0, \forall |z| = 1, \tag{3.15b}$$

where $A_\tau(z^{-1}) = \frac{z^{-\tau^*-1} d^T_{\tau^*_{ca}} L_\tau(z^{-1}) b_{I1}}{a}$, *and* $\bar{v}(k) = A_\tau(z^{-1} v(k))$ *is the theoretical input value to the CARIMA model.*

Proof Without loss of generality assume $\omega = 0$. Notice that for any column vector P with an appropriate dimension, $f(d^T_{\tau^*_{ca}} \hat{\hat{f}}^{-1}(P)) = f \cdot \hat{f}^{-1}(d^T_{\tau^*_{ca}} P)$ from the definition of $\hat{\hat{f}}^{-1}(\cdot)$. Then from (3.14) we obtain

$$\begin{aligned} v(k) &= f(u(k)) \\ &= f(d^T_{\tau^*_{ca}} \hat{\hat{f}}^{-1}(V(k - \tau^*_{ca} | k - \tau^*)) \\ &= f \cdot \hat{f}^{-1}(d^T_{\tau^*_{ca}} L_\tau(z^{-1}) y(k - \tau^*)) \\ &= f \cdot \hat{f}^{-1}(A_\tau(z^{-1}) v(k)) \\ &= f \cdot \hat{f}^{-1}(\bar{v}(k)) \end{aligned}$$

This is equivalent to the block diagram shown in Fig. 3.5. Thus the theorem can be easily obtained by applying Lemma 1 to Fig. 3.5.

3.3.2 Stability Criterion in State-Space Description

Similar to (3.13), from the design of the CAS in Sect. 2.2, the incremental control action adopted by the actuator at time k is readily obtained as

$$\Delta u(k) = \Delta u(k|k - \tau^*_k) = d^T_{\tau^*_{ca,k}} \Delta U(k - \tau^*_{ca,k} | k - \tau^*_k) \tag{3.16}$$

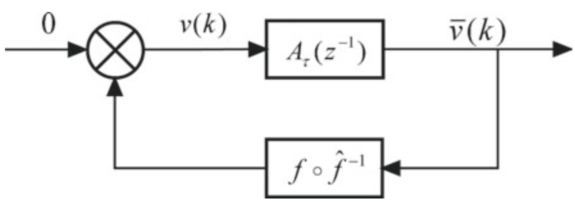


Fig. 3.5 The simplified block diagram of packet-based control for networked Hammerstein systems in (3.1)

where $d_{\tau^*_{ca,k}}$, τ^*_k, $\tau^*_{ca,k}$ and $\tau^*_k = \tau^*_{ca,k} + \tau^*_{sc,k}$ are defined in (3.13).

From (3.13), (3.11), (3.16) and noticing for any vector V with an appropriate dimension, $d^T_{\tau^*_{ca,k}} \hat{\bar{f}}^{-1}(V) = \hat{f}^{-1}(d^T_{\tau^*_{ca,k}} V)$, we then obtain (assume the set point $\omega = 0$ w.l.o.g.)

$$\begin{aligned} \Delta u(k) &= d^T_{\tau^*_{ca,k}} \Delta U(k - \tau^*_{ca,k} | k - \tau^*_k) \\ &= d^T_{\tau^*_{ca,k}} \hat{\bar{f}}^{-1}(\Delta V(k - \tau^*_{ca,k} | k - \tau^*_k) \\ &= \hat{f}^{-1}(d^T_{\tau^*_{ca,k}} \Delta V(k - \tau^*_{ca,k} | k - \tau^*_k) \\ &= \hat{f}^{-1}(-K^*_{\tau,k} \bar{x}(k - \tau^*_k)) \end{aligned} \tag{3.17}$$

where $K^*_{\tau,k} = d^T_{\tau^*_{ca,k}} M_{\tau_{sc,k}} E_{\tau_{sc,k}}$.[1] The real incremental control action for the linear system in (3.2a) and (3.2b) at time k can then be obtained as

$$\Delta v(k) = f(\Delta u(k)) = f \circ \hat{f}^{-1}(-K^*_{\tau,k} \bar{x}(k - \tau^*_k)) \tag{3.18}$$

where $f \circ \hat{f}^{-1}(\cdot) = f(\hat{f}^{-1}(\cdot))$ is the composite function of $f(\cdot)$ and $\hat{f}^{-1}(\cdot)$.

Let $X(k) = [\bar{x}^T(k - \bar{\tau}) \ \cdots \ \bar{x}^T(k)]^T$, $w(k) = \Delta v(k)$. The closed-loop formula for the system in (3.7) with the controller in (3.18) can then be represented by

$$\mathcal{S}^*_{I2} : \begin{cases} X(k+1) = \widetilde{A} X(k) + \widetilde{b} w(k) & \text{(3.19a)} \\ w(k) = f \circ \hat{f}^{-1}(-K^*_{\bar{\tau},k} X(k)) & \text{(3.19b)} \end{cases}$$

where $\widetilde{b} = [0_{n+1,1} \ \cdots \ 0_{n+1,1} \bar{b}^T_{n+1,1}]^T$, $K^*_{\bar{\tau},k}$ is a $1 \times (\bar{\tau} + 1)$ block matrix with block size of $1 \times (n+1)$ and all its blocks 0 except the $(\bar{\tau} + 1 - \tau^*_k)$th being $K^*_{\tau,k}$ (the set of all the possible $K^*_{\bar{\tau},k}$ will be denoted by $\mathbb{K}$), and $\widetilde{A} = \begin{pmatrix} 0_{n+1} & I_{n+1} & & \\ & 0_{n+1} & I_{n+1} & \mathbf{0} \\ & & \ddots & \ddots \\ \mathbf{0} & & & 0_{n+1} \ I_{n+1} \\ & & & \bar{A} \end{pmatrix}$.

As has been pointed out in Sect. 3.2.2, the compensation for the nonlinear input process using (3.10) is generally not accurate, and this inaccuracy introduces to the linear part of the Hammerstein system in (3.2a) and (3.2b) a nonlinear disturbance, which appears in the form of $f \circ \hat{f}^{-1}(\cdot)$. Though generally $f \circ \hat{f}^{-1}(\cdot) \not\equiv 1(\cdot)$, it is reasonable to assume that the calculation error meets some accuracy requirement

[1] Note that the value of $K^*_{\tau,k}$ varies with the delays in both channels, and thus it has $(\bar{\tau}_{ca} + 1)(\bar{\tau}_{sc} + 1)$ different values in total.

to a certain extent, which results in a sector constraint for the term $f \circ \hat{f}^{-1}(\cdot)$, as described in Assumption 3.1 as follows.[2]

Assumption 3.1 The nonlinearity due to the calculation inaccuracy is supposed to satisfy the following sector constraint: $\exists 0 < \underline{\varepsilon} \leq \bar{\varepsilon} < \infty$, s.t.

$$\underline{\varepsilon}\alpha \leq f \circ \hat{f}^{-1}(\alpha) \leq \bar{\varepsilon}\alpha, \forall \alpha \in \mathbb{R} \tag{3.20a}$$

or denoted by

$$f \circ \hat{f}^{-1}(\cdot) \in [\underline{\varepsilon}, \bar{\varepsilon}] \tag{3.20b}$$

Notice here generally $0 < \underline{\varepsilon} \leq 1 \leq \bar{\varepsilon} < \infty$.

Assumption 3.1 implies that for any specific $\alpha \in \mathbb{R}$, there exists a real number $\varepsilon_\alpha, \underline{\varepsilon} \leq \varepsilon_\alpha \leq \bar{\varepsilon}$ such that $f \circ \hat{f}^{-1}(\alpha) = \varepsilon_\alpha \alpha$. With this observation, (3.19b) can then be rewritten as

$$w(k) = f \circ \hat{f}^{-1}(-K^*_{\bar{\tau},k} X(k)) = -\varepsilon_k K^*_{\bar{\tau},k} X(k) \tag{3.21}$$

where $\varepsilon_k \in [\underline{\varepsilon}, \bar{\varepsilon}]$ represents the compensation for the specific nonlinearity for the term $K^*_{\bar{\tau},k} X(k)$ at time k.

With (3.19a) and (3.21), the closed-loop system in $\mathcal{S}^*_{I2}$ can then be written as

$$\begin{aligned} X(k+1) &= \widetilde{A} X(k) + \widetilde{b} w(k) \\ &= (\widetilde{A} - \varepsilon_k \widetilde{b} K^*_{\bar{\tau},k}) X(k) \\ &= \Lambda(\varepsilon_k, K^*_{\bar{\tau},k}) X(k) \end{aligned} \tag{3.22}$$

where the closed loop matrix $\Lambda(\varepsilon_k, K^*_{\bar{\tau},k}) = \widetilde{A} - \varepsilon_k \widetilde{b} K^*_{\bar{\tau},k}$ has the following form

$$\Lambda(\varepsilon_k, K^*_{\bar{\tau},k}) = \begin{pmatrix} 0_{n+1} & I_{n+1} & & & \\ & 0_{n+1} & I_{n+1} & & \mathbf{0} \\ & & \ddots & \ddots & \\ & \mathbf{0} & & 0_{n+1} & I_{n+1} \\ \cdots & -\varepsilon_k \bar{b} K^*_{\tau,k} & \cdots & & \bar{A} \end{pmatrix}.$$

where the position and value of the term $-\varepsilon_k \bar{b} K^*_{\tau,k}$ depend on specific delays in both channels at time k, i.e., $(\Lambda(\varepsilon_k, K^*_{\bar{\tau},k}))_{\bar{\tau}+1,j} = -\varepsilon_k \bar{b} K^*_{\tau,k}$, $j = \tau^*_k = 1, 2, \cdots, \bar{\tau}$, and $(\Lambda(\varepsilon_k, K^*_{\bar{\tau},k}))_{\bar{\tau}+1,\bar{\tau}+1} = \bar{A} - \varepsilon_k \bar{b} K^*_{\tau,k}$, if $\tau^*_k = \bar{\tau} + 1$.

[2]Note that though it is reasonable to place a sector constraint as in Assumption 3.1 to $f \circ \hat{f}^{-1}(\cdot)$, it is somewhat conservative since the calculation of some strongly nonlinear function may not be that accurate and thus does not satisfy Assumption 3.1.

Theorem 3.2 *The closed-loop system $\mathcal{S}_{I2}^*$ is stable if Assumption (3.14) holds and there exists a positive definite solution $P = P^T > 0$ for the following $2(\bar{\tau}_{ca}+1)(\bar{\tau}_{sc}+1)$ LMIs*

$$\Lambda^T(\underline{\varepsilon}, K_{\bar{\tau},k}^*) P \Lambda(\underline{\varepsilon}, K_{\bar{\tau},k}^*) - P \leq 0 \tag{3.23a}$$

$$\Lambda^T(\bar{\varepsilon}, K_{\bar{\tau},k}^*) P \Lambda(\bar{\varepsilon}, K_{\bar{\tau},k}^*) - P \leq 0 \tag{3.23b}$$

where $K_{\bar{\tau},k}^* \in \mathbb{K}$.

Proof Let $V(k) = X^T(k)PX(k)$ be a Lyapunov function candidate for system $\mathcal{S}_{I2}^*$. The increment of $V(k)$ along the trajectory of system $\mathcal{S}_{I2}^*$ can be obtained using (3.22) as

$$\begin{aligned}\Delta V(k) &= X^T(k)(\Lambda(\varepsilon_k, K_{\bar{\tau},k}^*)^T P \Lambda(\varepsilon_k, K_{\bar{\tau},k}^*) - P)X(k) \\ &= X^T(k)(\widetilde{A}^T P \widetilde{A} - P - \varepsilon_k \widetilde{A}^T P \widetilde{b} K_{\bar{\tau},k}^* - \varepsilon_k K_{\bar{\tau},k}^{*T} \widetilde{b}^T P \widetilde{A} \\ &\quad + \varepsilon_k^2 K_{\bar{\tau},k}^{*T} \widetilde{b}^T P \widetilde{b} K_{\bar{\tau},k}^*)X(k) \\ &\triangleq X^T(k)\mathscr{A}(\varepsilon_k, K_{\bar{\tau},k}^*)X(k)\end{aligned} \tag{3.24}$$

where $\varepsilon_k \in [\underline{\varepsilon}, \bar{\varepsilon}]$, $K_{\bar{\tau},k}^* \in \mathbb{K}$.

Notice that for any $\varepsilon_k \in [\underline{\varepsilon}, \bar{\varepsilon}]$, there exists $0 \leq \lambda_k \leq 1$ s.t. $\varepsilon_k = \lambda_k \underline{\varepsilon} + (1-\lambda_k)\bar{\varepsilon}$, and thus we obtain by substituting this into (3.24) that

$$\begin{aligned}&\mathscr{A}(\varepsilon_k, K_{\bar{\tau},k}^*) \\ &= \lambda_k \mathscr{A}(\underline{\varepsilon}, K_{\bar{\tau},k}^*) + (1-\lambda_k)\mathscr{A}(\bar{\varepsilon}, K_{\bar{\tau},k}^*) - \lambda_k(1-\lambda_k)(\underline{\varepsilon} - \bar{\varepsilon})^2 K_{\bar{\tau},k}^{*T} \widetilde{b}^T P \widetilde{b} K_{\bar{\tau},k}^*\end{aligned}$$

From (3.23) and (3.24) it is seen that $\mathscr{A}(\underline{\varepsilon}, K_{\bar{\tau},k}^*)$ and $\mathscr{A}(\bar{\varepsilon}, K_{\bar{\tau},k}^*)$ are semi-negative definite for all $K_{\bar{\tau},k}^* \in \mathbb{K}$. Notice that P is symmetric positive definite, and then $K_{\bar{\tau},k}^{*T} \widetilde{b}^T P \widetilde{b} K_{\bar{\tau},k}^*$ is semi-positive definite as a symmetric matrix, thus enabling $\mathscr{A}(\varepsilon_k, K_{\bar{\tau},k}^*)$ to be semi-negative definite for any $\varepsilon_k \in [\underline{\varepsilon}, \bar{\varepsilon}]$ and $K_{\bar{\tau},k}^* \in \mathbb{K}$, which completes the proof.

Remark 3.5 It is necessary to point out that according to Theorem 3.2, what is required for the stability of the system is to satisfactorily meet the sector constraint in Assumption 3.1, no matter how the inverse function $\hat{f}^{-1}(\cdot)$ is calculated. It implies that the function $f(\cdot)$ does not need to be theoretically invertible as long as its inverse can be obtained by a numerical method and satisfies the sector constraint. The reader is referred to [127] and the references therein for more information of the calculation of $\hat{f}^{-1}(\cdot)$.

The following two special cases are also considered for system $\mathcal{S}_{I2}$.

Case 3.1 The network-induced delays in both channels are constant (noted by τ_{sc}^0 and τ_{ca}^0 respectively).

Case 3.2 The calculation of the inverse of the nonlinear function is accurate.

The following corollary readily follows from Theorem 3.2.

Corollary 3.1 *The closed loop system $\mathcal{S}_{I2}^{*}$ is stable if any one of the following three conditions holds.*

1. *Assumption 3.1 and Case 3.2 hold and there exists a positive definite solution $P = P^T > 0$ for the following two LMIs*

$$\Lambda^T(\underline{\varepsilon}, K^{*}_{\bar{\tau},k}) P \Lambda(\underline{\varepsilon}, K^{*}_{\bar{\tau},k}) - P \leq 0 \tag{3.25a}$$

$$\Lambda^T(\bar{\varepsilon}, K^{*}_{\bar{\tau},k}) P \Lambda(\bar{\varepsilon}, K^{*}_{\bar{\tau},k}) - P \leq 0 \tag{3.25b}$$

   where $\tau_{sc,k} \equiv \tau_{sc}^0$, $\tau_{ca,k} \equiv \tau_{ca}^0$ and $K^{*}_{\bar{\tau},k}$ is therefore fixed.
2. *Case 3.2 holds and there exists a positive definite solution $P = P^T > 0$ for the following $(\bar{\tau}_{ca} + 1)(\bar{\tau}_{sc} + 1)$ LMIs*

$$\Lambda^T(1, K^{*}_{\bar{\tau},k}) P \Lambda(1, K^{*}_{\bar{\tau},k}) - P \leq 0$$

   where $K^{*}_{\bar{\tau},k} \in \mathbb{K}$.
3. *Both of Cases 3.1 and 3.2 hold and there exists a positive definite solution $P = P^T > 0$ for the following LMI*

$$\Lambda^T(1, K^{*}_{\bar{\tau},k}) P \Lambda(1, K^{*}_{\bar{\tau},k}) - P \leq 0$$

   where $\tau_{sc,k} \equiv \tau_{sc}^0$, $\tau_{ca,k} \equiv \tau_{ca}^0$ and $K^{*}_{\bar{\tau},k}$ is therefore fixed.

3.4 Numerical and Experimental Examples

In this section, numerical and experimental examples are considered to illustrate the effectiveness of the proposed packet-based control approach to networked Hammerstein systems.

Example 3.1 This numerical example is used to illustrate the effectiveness of the packet-based control approach to system $\mathcal{S}_{I1}$ in input-output description.

The linear part in (3.1a) is adopted as

$$y(k) - 0.8y(k-1) = 2v(k-1) + 3v(k-2)$$

The nonlinear input process in (3.1b) is chosen as $v = f(u) = u^2$ and the practical inverse of $f(\cdot)$ is $\hat{f}^{-1} = \sqrt{v} \times \epsilon$, where ϵ is a random number with a uniform distribution in [0 1]. ϵ is introduced to represent the uncertainty in a practical implementation. From (3.15a) in Theorem 3.1 it is seen that $\mathcal{K} = 1$. The predictive horizon and control horizon are chosen as $N_p = N_u = 12$ in the simulation.

It is seen that the system is stable only for the first two cases according to Theorem 3.1 since for too large a time delay the system will not satisfy (3.15b) in Theorem 3.1. The simulation results of three cases: (i) $(\tau_{ca}, \tau_{sc}) = (0, 0)$; (ii) $(\tau_{ca}, \tau_{sc}) = (2, 3)$; and (iii) $(\tau_{ca}, \tau_{sc}) = (3, 7)$ are shown in Figs. 3.6 and 3.7 and illustrate the validity of the theoretical analysis.

Example 3.2 This numerical example is used to illustrate the effectiveness of the packet-based control approach to system $\mathcal{S}_{I2}$ in state-space description.

The linear part of system $\mathcal{S}_{I2}$ is defined as follows which is open-loop unstable,

$$A = \begin{pmatrix} 0.98 & 0.1 \\ 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 0.04 \\ 0.1 \end{pmatrix}, c = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

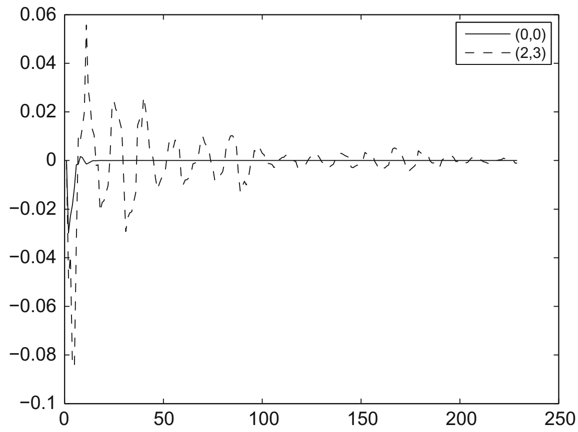


Fig. 3.6 Example 3.1. (i) $(\tau_{ca}, \tau_{sc}) = (0, 0)$; (ii) $(\tau_{ca}, \tau_{sc}) = (2, 3)$;

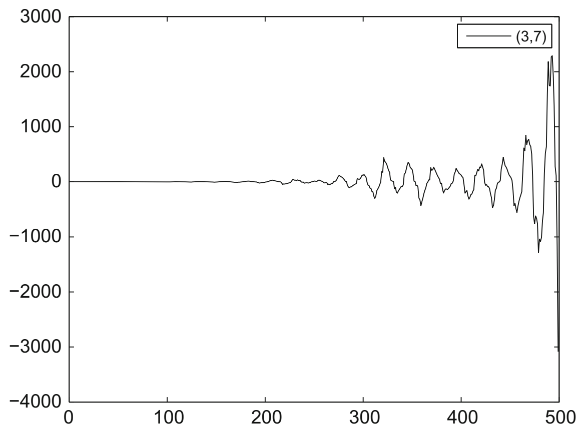


Fig. 3.7 Example 3.1. (iii) $(\tau_{ca}, \tau_{sc}) = (3, 7)$

Note the fact that with an inverse process to compensate for the static nonlinear input process in system $\mathcal{S}_{I2}$, from (3.19b) we know that the system performance depends only on the accuracy of the compensation process, i.e., the size of the sector constraint $[\underline{\varepsilon}, \bar{\varepsilon}]$ for $f \circ \hat{f}^{-1}(\cdot)$ (see Assumption 3.1). In this simulation, we set $[\underline{\varepsilon}, \bar{\varepsilon}] = [0.5, 1.5]$ which means there is approximately 50% error in the compensation for the input nonlinearity while the input nonlinear function $f(\cdot)$ can be of any form provided this compensation accuracy is satisfied. All the other parameters are set the same as above. Such a system with those parameters can be proved to be stable using Theorem 3.2.

The compensation for the nonlinear input process is shown in Fig. 3.9 (under the arbitrary delays in the forward channel as shown in Fig. 3.8), from which it is seen that this compensation strategy is effective for networked Hammerstein systems, where the parameters are set as $N_u = 8$, $N_p = 10$, $\bar{\tau} = 3$, $\bar{\tau}_{ca} = 2$, $\bar{\tau}_{sc} = 1$ and $x_0 = [-1\ -1]^T$.

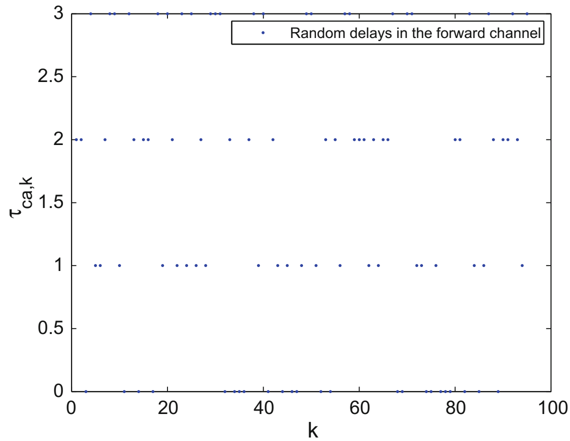


Fig. 3.8 Example 3.2. Arbitrary delays in the forward channel

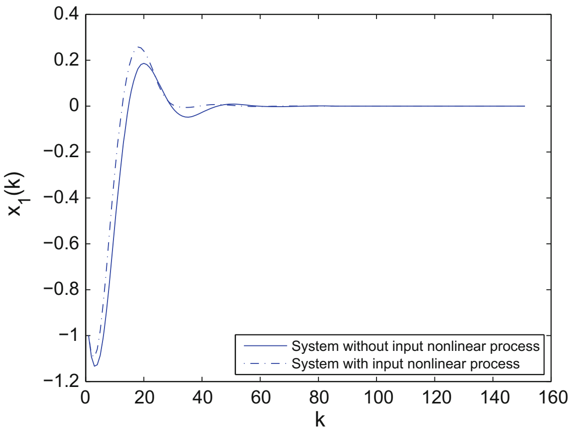


Fig. 3.9 Example 3.2. The effectiveness of packet-based control for networked Hammerstein systems

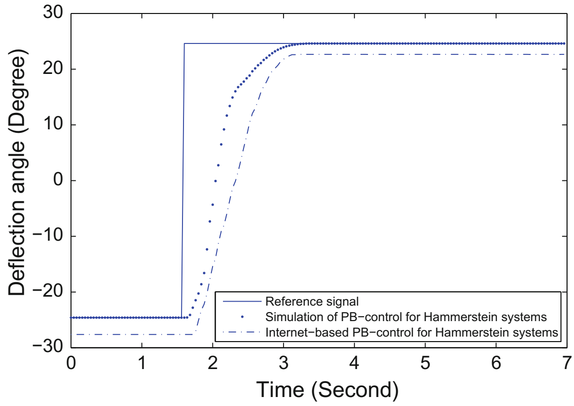


Fig. 3.10 Example 3.3. Comparison between simulation and experimental results of packet-based control for Hammerstein systems

Example 3.3 In this example, we use the same experiment setup as in Example 2.4 only that a compensation scheme for an input nonlinear process is present with $[\underline{\varepsilon}, \bar{\varepsilon}] = [0.8, 1.2]$. Since the linear part of the system remains the same, the same packet-based controller is designed here as in Example 2.4. The comparison between the simulation and experimental results is illustrated in Fig. 3.10, where it is seen that the compensation scheme is effective in practice.

3.5 Summary

In this chapter, the packet-based control approach proposed in Chap. 2 was extended to networked Hammerstein systems. In order to deal with the nonlinear input process in the Hammerstein system, a two-step approach was applied to separate the nonlinear input process from the whole system, which proved to be effective for both descriptions of the Hammerstein system, i.e., the input-output description and the state-space description. For input-output description, a stability criterion was obtained using Popov criterion, which is valid for a constant delay, while for the state-space description, stability conditions were obtained for arbitrary delays by using switched system theory. Numerical and experimental examples illustrated the effectiveness of the proposed approaches.

Chapter 4
Packet-Based Control for Networked Wiener Systems

Following the extension of the packet-based control approach to networked Hammerstein systems in Chap. 3, another extension readily follows which is the category of output nonlinear systems described by the Wiener model, where a static nonlinear output process is present in the system. For this type of nonlinear systems, the two-step approach proposed in Chap. 3 can still be applied to separate the nonlinear process from the system, thus enabling the packet-based control approach to be implemented in this case. Different from the input nonlinearity case, a specially designed observer is proposed for the implementation of the two-step approach to networked Wiener systems, and as a result, the stability criterion of the corresponding closed-loop system depends not only on the communication conditions but the error of the observer.

This chapter is organized as follows. Section 4.2 presents the design details of the packet-based control approach to networked Wiener systems; Sect. 4.3 analyzes the stability of the closed-loop system; Sect. 4.4 presents numerical and experimental examples to illustrate the effectiveness of the proposed approach and Sect. 4.5 concludes the chapter.

4.1 System Description

This chapter considers a class of SISO Wiener system $\mathcal{S}_o$ [128–131], described as follows,

$$\mathcal{S}_o : \begin{cases} x(k+1) = Ax(k) + bu(k) & (4.1a) \\ y(k) = cx(k) & (4.1b) \\ z(k) = f(y(k)) & (4.1c) \end{cases}$$

where $x \in \mathbb{R}^n$, $u, y, z \in \mathbb{R}$, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^{n\times 1}$, $c \in \mathbb{R}^{1\times n}$, $f(\cdot)$ is a memoryless static nonlinear function and $u(k)$ is to be determined (see Sect. 4.2). In this chapter the Wiener system is assumed to be controlled over the network, see Fig. 4.1 for its configuration.

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_4

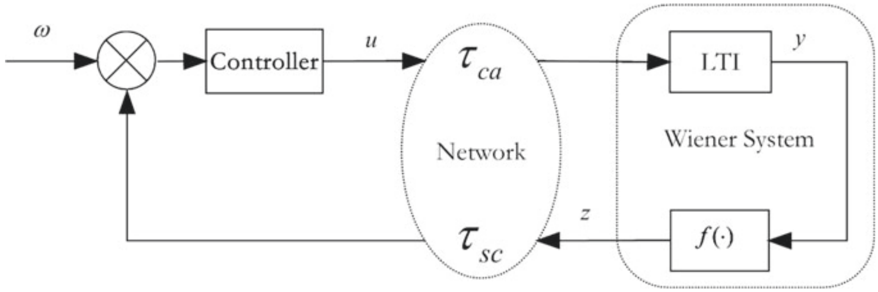


Fig. 4.1 The block diagram of networked Wiener Systems

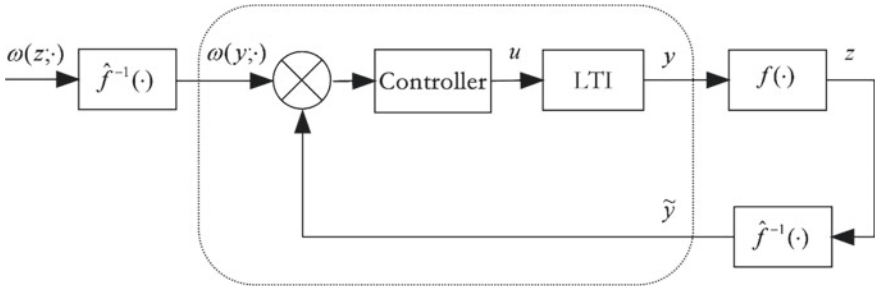


Fig. 4.2 Two-step approach to networked Wiener systems

As in Chap. 3, the memoryless static nonlinear function $f(\cdot)$ in this chapter is assumed to be invertible with its inverse denoted by $\hat{f}^{-1}(\cdot)$. Notice that $\hat{f}^{-1}(\cdot)$ can not be obtained accurately in practice which means $\varphi(\cdot) \triangleq \hat{f}^{-1}(f(\cdot)) \not\equiv 1(\cdot)$. The approximate intermediate output $\tilde{y}(k)$ (Fig. 4.2) can thus be obtained as follows,

$$\tilde{y}(k) = \hat{f}^{-1}(z(k)) = \varphi(y(k)) \tag{4.2}$$

With this inverse process, the packet-based controller for networked Wiener systems in (4.1) can then be obtained using a LGPC method and a specially designed state observer as follows.

4.2 Packet-Based Control for Networked Wiener Systems

Let the objective function for system $\mathcal{S}_o$ be defined by

$$\begin{aligned} J^o_{k,\tau_{sc,k}} = &\sum_{j=N_1}^{N_2} q_j(\hat{y}(k+j|k-\tau_{sc,k}) - \omega(y;k+j))^2 \\ &+ \sum_{j=1}^{N_u} r_j(\Delta u(k+j-1))^2 \end{aligned} \tag{4.3}$$

where N_1 and N_2 are the minimum and maximum prediction horizons, N_u is the control horizon, $q_j, N_1 \leq j \leq N_2$ and $r_j, 1 \leq j \leq N_u$ are weighting factors, $\Delta u(k) = u(k) - u(k-1)$ is the control increment, $\hat{y}(k+j|k-\tau_{sc,k})$, $j = N_1, ..., N_2$ are the forward predictions of the system outputs, which are obtained on data up to time $k - \tau_{sc,k}$; $\omega(y; k + j)$ is the set point with respect to y and can be obtained approximately by inverting corresponding set point $\omega(z; k + j)$ with respect to z, i.e.,

$$\omega(y; k + j) = \hat{f}^{-1}(\omega(z; k + j)), j = N_1, ..., N_2 \tag{4.4}$$

Letting $\bar{x}(k) = [x^T(k)\ u(k-1)]^T$, then the linear part of system $\mathcal{S}_o$ (i.e. (4.1a) and (4.1b)) can be rewritten as follows,

$$\mathcal{S}_o' : \begin{cases} \bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{b}\Delta u(k) & (4.5a) \\ y(k) = \bar{c}\bar{x}(k) & (4.5b) \end{cases}$$

where $\bar{A} = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$, $\bar{b} = \begin{pmatrix} b \\ 1 \end{pmatrix}$, $\bar{c} = \begin{pmatrix} c & 0 \end{pmatrix}$.

Following the same procedure as in Sect. 2.4, the optimal FCIS from k to $k + N_u - 1$ can then be obtained as

$$\Delta U(k|k - \tau_{sc,k}) = M_{\tau_{sc,k}}(\varpi_k(y; \cdot) - E_{\tau_{sc,k}}\bar{x}(k - \tau_{sc,k}))$$

where $\Delta U(k|k-\tau_{sc,k}) = [\Delta u(k|k-\tau_{sc,k}) \cdots \Delta u(k+N_u-1|k-\tau_{sc,k})]^T$, $E_{\tau_{sc,k}} = [(\bar{c}\bar{A}^{N_1+\tau_{sc,k}})^T \cdots (\bar{c}\bar{A}^{N_2+\tau_{sc,k}})^T]^T$, $F_{\tau_{sc,k}}$ is a $(N_2 - N_1 + 1) \times (N_u + \tau_{sc,k})$ matrix with the non-null entries defined by $(F_{\tau_{sc,k}})_{ij} = \bar{c}\bar{A}^{N_1+\tau_{sc,k}+i-j-1}\bar{b}$, $j - i \leq N_1 + \tau_{sc,k} - 1$, $\varpi_k(y; \cdot) = [\omega(y; k + N_1) \cdots \omega(y; k + N_2)]^T$, $M_{\tau_{sc,k}} = H_{\tau_{sc,k}}(F_{\tau_{sc,k}}^T Q F_{\tau_{sc,k}} + R)^{-1} F_{\tau_{sc,k}}^T Q$, Q, R are diagonal matrices with $Q_{i,i} = q_i$, $R_{i,i} = r_i$ respectively, $H_{\tau_{sc,k}} = [0_{N_u \times \tau_{sc,k}}\ I_{N_u \times N_u}]$, and $I_{N_u \times N_u}$ is the identity matrix with rank N_u.

Since the system states are normally unavailable for the controller, the following observed system is then constructed,

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + bu(k) & (4.6a) \\ \hat{y}(k) = \varphi(c\hat{x}(k)) & (4.6b) \end{cases}$$

to observe the system states,

$$\hat{x}(k+1) = A\hat{x}(k) + bu(k) + L(\tilde{y}(k) - \hat{y}(k)) \tag{4.7}$$

where $\hat{x}(k)$ is the observed state at time k.

Letting $\hat{\bar{x}}(k) = [\hat{x}^T(k)\ u^T(k-1)]^T$, the real FCIS can then be obtained as follows when the state observer in (4.6) is present,

$$\Delta U(k|k - \tau_{sc,k}) = M_{\tau_{sc,k}}(\varpi_k(y; \cdot) - E_{\tau_{sc,k}}\hat{\bar{x}}(k - \tau_{sc,k})) \tag{4.8}$$

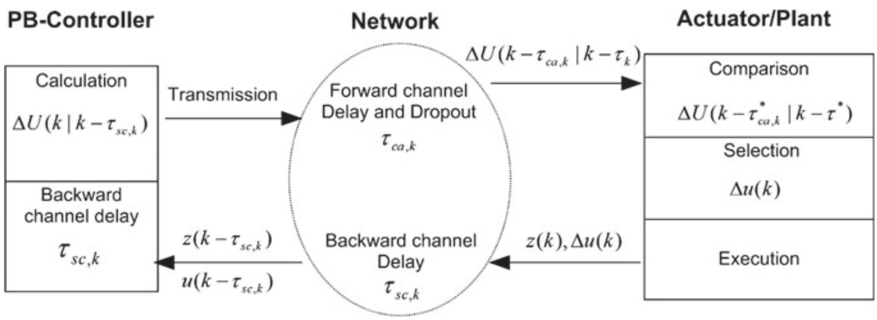


Fig. 4.3 Packet-based control for networked Wiener systems

With the FCIS obtained in (4.8), the packet-based control approach can then be implemented to networked Wiener systems (4.1). Since the whole packet-based control structure here is exactly the same as in Sect. 2.2, we therefore only illustrate its block diagram in Fig. 4.3 without further discussion; the reader is referred to Sect. 2.2 for more information on the design of the packet-based control approach.

4.3 Stability Analysis of Packet-Based Networked Wiener Systems

In this section, we first prove the proposed state observer in (4.6) is stable under certain conditions. This fact enables us to construct the stability criterion for the closed-loop system.

4.3.1 Observer Error

Let the observer error $e(k) = x(k) - \hat{x}(k)$. From (4.1a), (4.6a) we obtain

$$\begin{aligned} e(k+1) = &\ x(k+1) - \hat{x}(k+1) \\ = &\ Ae(k) - L(\tilde{y}(k) - \hat{y}(k)) \end{aligned} \tag{4.9}$$

Assume $\varphi(\cdot) \in C^1$, then by mean value theorem,

$$\begin{aligned} \tilde{y}(k) - \hat{y}(k) = &\ \varphi(cx(k)) - \varphi(c\hat{x}(k)) \\ = &\ c\varphi'(\xi_k))e(k) \end{aligned} \tag{4.10}$$

where $\xi_k \in [\min\{cx(k), c\hat{x}(k)\}\ \max\{cx(k), c\hat{x}(k)\}]$.

Combining equations (4.9) and (4.10) yields

$$e(k+1) = (A - Lc\varphi'(\xi_k))e(k) \tag{4.11}$$

Notice that though $\varphi(\cdot) \not\equiv 1(\cdot)$, it is reasonable to assume that the compensation for the nonlinear function $f(\cdot)$ is smooth, which means there exists $\varepsilon > 0$ s.t. $|\varphi'(\alpha) - 1| \leq \varepsilon, \forall \alpha \in \mathbb{R}$. Thus the dynamics of the observer error can be obtained as

$$\begin{aligned} e(k+1) =& (A - Lc - \zeta_k Lc)e(k) \\ =& A_{\zeta_k} e(k) \end{aligned} \tag{4.12}$$

where $A_{\zeta_k} = A - Lc - \zeta_k Lc$, $|\zeta_k| \leq \varepsilon$.

Theorem 4.1 ***(Observer Error)*** *The observer error converges to 0 if there exists a positive definite solution $P_e = P_e^T > 0$ for the following two LMIs*

$$\begin{aligned} A_\varepsilon^T P_e A_\varepsilon - P_e < 0 \\ A_{-\varepsilon}^T P_e A_{-\varepsilon} - P_e < 0 \end{aligned} \tag{4.13}$$

where $A_\varepsilon = A - Lc - \varepsilon Lc$ and $A_{-\varepsilon} = A - Lc + \varepsilon Lc$.

Proof Let $V(k) = e^T(k)P_e e(k)$ be a Lyapunov function candidate for the system in (4.12). Notice the fact that for any ζ_k, there exists $0 \leq \lambda_k \leq 1$ such that $\zeta_k = \lambda_k \varepsilon + (1 - \lambda_k)(-\varepsilon)$. Thus by simple calculation, the incremental $V(k)$ for the system in (4.12) can be obtained as

$$\begin{aligned} \Delta V(k+1) =& e^T(k)\Gamma_{\zeta_k} e(k) \\ =& e^T(k)(\lambda_k \Gamma_\varepsilon + (1 - \lambda_k)\Gamma_{-\varepsilon} - 4\lambda_k(1 - \lambda_k)(Lc)^T P_e Lc)e(k) \end{aligned}$$

where $\Gamma_{\zeta_k} = A_{\zeta_k}^T P_e A_{\zeta_k} - P_e$.

Noticing $\lambda_k(1 - \lambda_k) \geq 0$ and $(Lc)^T P_e Lc$ is semi positive definite, then yields that $\Delta V(k)$ is decreasing which completes the proof.

4.3.2 Closed-Loop Stability

From (4.2) and the design of the CAS, the incremental control action adopted by the actuator at time k is readily obtained as

$$\begin{aligned} \Delta u(k) &= d_{\tau_{ca,k}^*}^T \Delta U(k - \tau_{ca,k}^* | k - \tau_k^*) \\ &= -d_{\tau_{ca,k}^*}^T M_{\tau_k^*} E_{\tau_k^*} \hat{\hat{x}}(k - \tau_k^*) \\ &= -\Sigma_{\tau_k} \hat{\hat{x}}(k - \tau_k^*) \end{aligned} \tag{4.14}$$

where $d_{\tau^*_{ca,k}}$ is a $N_u \times 1$ matrix with all entries 0 except the $(\tau^*_{ca,k}+1)$th being 1, τ^*_k, $\tau^*_{ca,k}$ and $\tau^*_{sc,k}$ are defined in (2.6), $\Sigma_{\tau_k} = d^T_{\tau^*_{ca,k}} M_{\tau^*_k} E_{\tau^*_k}$ and the set point is assumed to be 0 without loss of generality.

Let $\bar{e}(k) = \bar{x}(k) - \hat{\bar{x}}(k) = [e(k)\ 0]^T$. Then

$$\bar{e}(k+1) = \bar{A}_{\xi_k}\bar{e}(k)$$

where $\bar{A}_{\xi_k} = \begin{pmatrix} A - Lc\varphi'(\xi_k) & 0 \\ 0 & 0 \end{pmatrix}$.

Let $Z(k) = [\bar{x}^T(k-\bar{\tau}) \ \cdots \ \bar{x}^T(k)\bar{e}(k-\bar{\tau}) \ \cdots \ \bar{e}(k)]^T$. The closed-loop system can then be obtained as

$$Z(k+1) = \Lambda_{\xi_k,\tau_k} Z(k) \tag{4.15}$$

where $\Lambda_{\xi_k,\tau_k} = \begin{pmatrix} \Lambda^{11}_{\tau_k} & \Lambda^{12}_{\tau_k} \\ 0 & \Lambda^{22}_{\xi_k} \end{pmatrix}$, $\Lambda^{11}_{\tau_k} = \begin{pmatrix} 0_{n+1} & I_{n+1} & & & \\ & & I_{n+1} & & \\ & & & \ddots & \\ & & & & I_{n+1} \\ \cdots & -\Sigma_{\tau_k} & \cdots & & \bar{A} \end{pmatrix}$,

$\Lambda^{22}_{\xi_k} = \begin{pmatrix} 0_{n+1} & I_{n+1} & & & \\ & 0_{n+1} & I_{n+1} & & \\ & & \ddots & \ddots & \\ & & & \ddots & I_{n+1} \\ & & & & \bar{A}_{\xi_k} \end{pmatrix}$, and $\Lambda^{12}_{\tau_k}$ is a block matrix with all its entries (blocks) 0 except $(\Lambda^{12}_{\tau_k})_{(\bar{\tau}-1)\times(\bar{\tau}-\tau^*_k+1)} = -\Sigma_{\tau_k}$.

Theorem 4.2 *[Closed-loop stability] The closed-loop system in (4.15) is stable if (4.1a) holds and there exists a positive definite solution $P_c = P_c^T > 0$ for the following $(\bar{\tau}_{ca}+1)(\bar{\tau}_{sc}+1)$ LMIs*

$$(\Lambda^{11}_{\tau_k})^T P_c \Lambda^{11}_{\tau_k} - P_c \leq 0 \tag{4.16}$$

Proof By noticing the block-triangular structure of the system matrix Λ_{ξ_k,τ_k} for the closed-loop system, it is seen that the state observer in (4.6) can be designed separately without affecting the stability of the system and the closed-loop system is stable if we can guarantee the stability of the state observer (Theorem 4.1) and the following system,

$$X(k+1) = \Lambda^{11}_{\tau_k} X(k) \tag{4.17}$$

where $X(k) = [\bar{x}^T(k-\bar{\tau}) \ \cdots \bar{x}^T(k)]$.

Let $V(k) = X^T(k)P_cX(k)$ be a Lyapunov function candidate for the system in (4.17). The incremental $V(k)$ along the trajectory of the system in (4.17) is then obtained as

$$\Delta V(k) = X^T(k)((\Lambda_{\tau_k}^{11})^T P_c \Lambda_{\tau_k}^{11} - P_c)X(k)$$

which completes the proof using (4.16).

Remark 4.1 It is worth mentioning that the two conditions (4.13) and (4.16) that guarantee the stability of the closed-loop system are with respect to the compensation accuracy for the nonlinearity and the effect of the network constraints respectively.

4.4 Numerical and Experimental Examples

Example 4.1 The linear system in Example 3.2 with a static nonlinear output process and random delays in both channels and data packet dropout in the forward channel, is adopted, with other parameters of the simulation chosen as $\bar{\tau} = 8$, $\bar{\tau}_{ca} = 4$, $\bar{\tau}_{sc} + \bar{\chi} = 4$, $N_u = 8$, $N_p = 10$, $\varepsilon = 0.5$ and the initial state $x(0) = x_0 = [-0.1\ 0.2]^T$. The delays in both channels are set to vary randomly within their upper bounds. Such a system using the packet-based control approach can be proved to be stable under Theorem 4.2.

Two cases which illustrate the validity of the compensation for the communication constraints and the compensation for the output nonlinearity, are shown in Fig. 4.4 and Fig. 4.5 respectively. In both cases, all the other parameters remain the same and only the evolution of the first state of the system is illustrated. The simulation results show that the system is stable with the compensation scheme while unstable without it, which illustrate the validity of the proposed approach in this chapter.

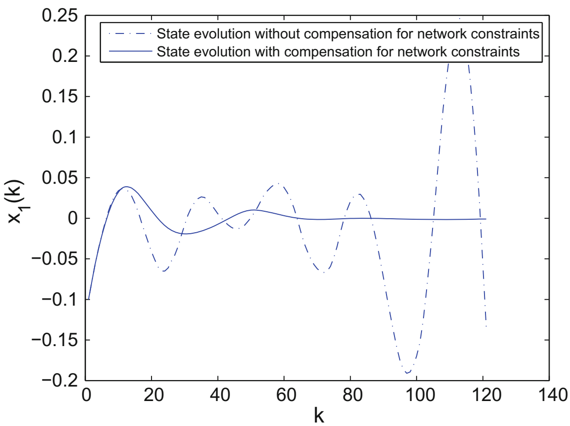


Fig. 4.4 Example 4.1. A comparison between with/without compensation for network constraints

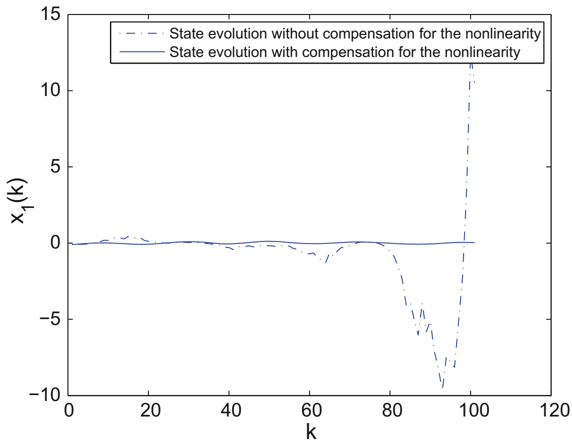


Fig. 4.5 Example 4.1. A comparison between with/without compensation for output nonlinearity

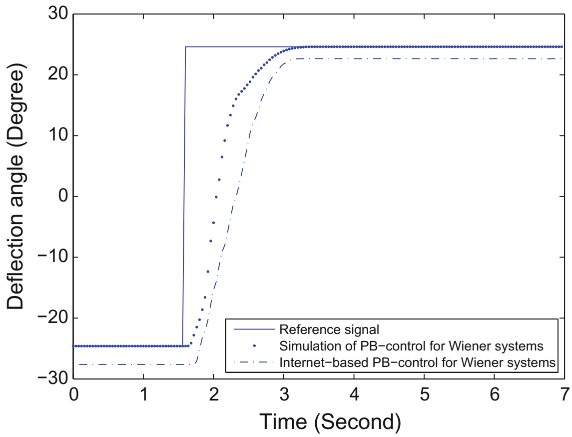


Fig. 4.6 Example 4.2. Comparison between simulation and experimental results of packet-based control for Wiener systems

Example 4.2 In this example, we use the same experiment setup as in Example 2.4 only that a compensation scheme for an output nonlinear process is present with $[\underline{\varepsilon}, \bar{\varepsilon}] = [0.8, 1.2]$. Since the linear part of the system remains the same, the same packet-based controller is designed here as in Example 2.4. The comparison between the simulation and experimental results is illustrated in Fig. 4.6, where it is seen that the compensation scheme is effective in practice.

4.5 Summary

In this chapter, the packet-based control approach was extended to networked Wiener systems. The idea of the two-step approach proposed for networked Hammerstein systems in Chap. 3 was still adopted, which together with a specially designed state observer enabled the packet-based control approach to be implemented in this case. Closed-loop stability was obtained by using the separate principle and switched system theory, the validity of which was illustrated by numerical and experimental examples.

Chapter 5
Packet-Based Networked Control Systems in Continuous Time

In all the previous chapters (Chaps. 2, 3, and 4) the packet-based control approach is considered for plants in discrete time and discrete network-induced delay. The packet-based control approach is extended to the continuous time case in this chapter, with the use of a discretization technique for the continuous network-induced delay. The derived approach leads to a novel model for NCSs in continuous time. This model, as in the discrete time case, offers the designer the freedom of designing different controllers with respect to specific network conditions, which is distinct from previously reported results and results in a better performance. By applying switched system theory, the stability criterion for the derived model is obtained, which is then used to obtain an LMI-based stabilized controller for the continuous-time PBNCSs.

This chapter is organized as follows. The design details of the packet-based control approach to NCSs in continuous time is first presented in Sect. 5.1, which leads to a novel model for NCSs. This model is then further analyzed in Sect. 5.2 to obtain the stability criterion and a stabilized controller by using the results from switched system theory. A numerical example is given in Sect. 5.3 to illustrate the effectiveness of the proposed approach and Sect. 5.4 concludes the chapter.

5.1 Packet-Based Control in Continuous Time

The following linear plant in continuous time is considered in this chapter, which is assumed to be controlled over the network as shown in Fig. 5.1,

$$\mathcal{S}_c : \dot{x}(t) = Ax(t) + Bu(t) \tag{5.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$.

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_5

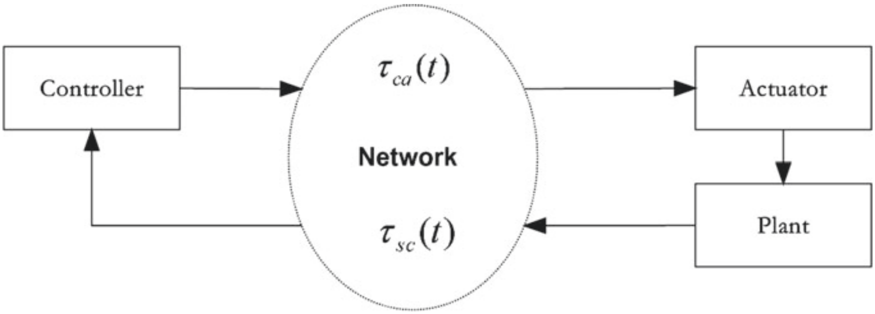


Fig. 5.1 The block diagram of networked control systems in continuous time

In this section the packet-based control approach proposed in Chap. 2 in the discrete time fashion is extended to the continuous time case, with a discretization technique to merge the gap between discrete and continuous time. The reader is referred to Sect. 2.2 for more information on the packet-based control approach in the discrete time case.

5.1.1 Packet-Based Control for NCSs in Continuous Time

A fundamental basis of the implementation of the packet-based control for NCSs in discrete time is the construction of the FCS in (2.9) and (2.13), which can be readily obtained in discrete time. However, a time delay system in continuous time is of infinite dimension, thus making it difficult to implement readily the packet-based control approach in the continuous time case because of the difficulty in determining the FCS as in (2.9) and (2.13). To deal with this difficulty, the continuous network-induced delay is discretized as follows.

Let $\bar{\tau}_d = dh + \bar{\tau}$ and $\vartheta = \frac{\bar{\tau}_d}{N}$, where h is the sampling period. A different FCS structure compared to (2.9) and (2.13) can then be constructed as follows, which uses N discrete levels to approach the real network-induced delay,

$$\begin{aligned} & U(t-\tau_{sc}(t)|t-\tau_{sc}(t)) \\ = \ & [u(t-\tau_{sc}(t)|t-\tau_{sc}(t)) \ \ldots \ u(t-\tau_{sc}(t)+(N-1)\vartheta|t-\tau_{sc}(t))] \end{aligned} \tag{5.2}$$

where $\tau_{sc}(t)$ is the continuous backward channel delay of the data packet received by the controller at time t.

The FCS in (5.2) can now be transmitted in one data packet by the network, provided the data size required for encoding a single step of the control signal is the same as the discrete time case. This is generally true since, both single step control signals are the specific values at one time instant, and therefore the data sizes of encoding both signals depend only on the range of the signals and the corresponding quantization levels, which can be assumed to be the same in both cases.

In order to implement the packet-based control approach, a similar CAS is also designed at the actuator side. The designed CAS consists of a simple comparison logic and a memory which can store only a single forward control sequence. When a FCS arrives at the actuator, it will first be compared using the comparison logic of the CAS with the one already in the memory of the CAS and only the latest is stored and applied to the plant. This comparison process is also introduced to overcome the effect of data packet disorder as done in Sect. 2.2.

For clarity, a FCS is called an "effective" one if it is actually stored after the comparison process. Note that the kth effective FCS is $U(t_k^* - \tau_k^* | t_k^* - \tau_k^*)$, where t_k^* is the time when this sequence is received by the actuator and τ_k^* the corresponding round trip delay. The control law during the time period $[t_{i_k}^*, t_{i_k+1}^*)$ can then be defined by

$$u(t) = u(t_k^* - \tau_k^* + i_k\vartheta | t_k^* - \tau_k^*), t \in [t_{i_k}^*, t_{i_k+1}^*) \tag{5.3}$$

where $[t_{i_k}^*, t_{i_k+1}^*) = [t_k^* - \tau_k^* + i_k\vartheta, t_k^* - \tau_k^* + (i_k + 1)\vartheta)$ with $i_k \in \mathbb{N}$ satisfying $t_k^* \leq t_k^* - \tau_k^* + i_k\vartheta < t_{k+1}^*$, and $u(t_k^* - \tau_k^* + i_k\vartheta | t_k^* - \tau_k^*)$ is selected from $U(t_k^* - \tau_k^* | t_k^* - \tau_k^*)$ which in this chapter is of the form of state feedback as follows,

$$u(t_k^* - \tau_k^* + i_k\vartheta | t_k^* - \tau_k^*) = K(i_k)x(t_k^* - \tau_k^*) \tag{5.4}$$

Note here that the value of the feedback gain $K(i_k)$ is dependent on the current range of the network-induced delay which, for $t \in [t_{i_k}^*, t_{i_k+1}^*)$, is

$$\tau_k^*(t) = t - (t_k^* - \tau_k^*) \in [i_k\vartheta, (i_k + 1)\vartheta) \tag{5.5}$$

Remark 5.1 Note that $t_k^* - \tau_k^*$ is the time when the sensing data packet is sent from the sensor from which the kth effective forward control sequence is calculated, and the sum of the continuous data packet dropout and network-induced delay is upper bounded by $\bar{\tau}_d$, see Fig. 5.2. Therefore the time when the $(k + 1)$th effective FCS arrives at the actuator is not later than $t_k^* - \tau_k^* + \bar{\tau}_d$, that is,

$$t_{k+1}^* \leq t_k^* - \tau_k^* + \bar{\tau}_d, \forall k \geq 1$$

The definition of i_k yields

$$\tau_k^* \leq i_k\vartheta < t_{k+1}^* - t_k^* + \tau_k^* \leq \bar{\tau}_d, \forall k \geq 1$$

Thus

$$\lceil \frac{\tau_k^*}{\vartheta} \rceil \leq i_k < \frac{\bar{\tau}_d}{\vartheta} = N, \forall k \geq 1$$

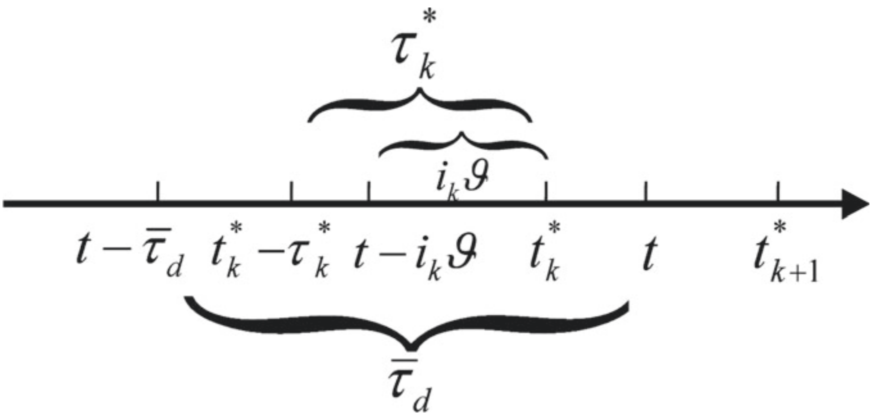


Fig. 5.2 Timeline of packet-based networked control systems

where $\lceil \frac{\tau_k^*}{\vartheta} \rceil = \min\{\varsigma | \varsigma \in \mathbb{N}, \varsigma \geq \frac{\tau_k^*}{\vartheta}\}$. Noticing the structure of $U(t_k^* - \tau_k^* | t_k^* - \tau_k^*)$ it is seen that the control action in (5.3) is always available from $U(t_k^* - \tau_k^* | t_k^* - \tau_k^*)$.

Remark 5.2 It is necessary to point out that there exists a situation where, for some $k \geq 1$ and i_k, the following relationship holds,

$$t_k^* \leq t_{i_k}^* < t_{k+1}^* < t_{i_k+1}^*$$

By the control law in (5.3), in this situation the $(k+1)$th effective FCS is not applied to the plant immediately but waits until $t_{i_k+1}^*$, and during the time period $[t_{k+1}^*, t_{i_k+1}^*)$ the kth effective FCS is still in action.

It is seen that this strategy artificially increases the delay (less than ϑ) however it provides the advantage that it produces a constant switch interval between two subsequent switches of control actions. A constant switch interval undoubtedly simplifies the modeling and analysis, and what is more important, it avoids a situation where, the switch interval is too short which may affect the stability of the system according to switched system theory [132].

Based on the above analysis, the algorithm of the packet-based control for NCSs in continuous time under Assumptions 2.3 and 2.4 can now be summarized as follows.

Algorithm 5.1 Packet-based control in continuous time

if The data packet containing the state information $x(t - \tau_{sc}(t))$ is received, the controller **then**
  Calculates the FCS as in (5.2)
  Packs $U(t - \tau_{sc}(t) | t - \tau_{sc}(t))$ into one data packet and sends it to the actuator
end if
CAS updates its FCS once a data packet arrives
The effective FCS is applied to the plant by the control law in (5.3)

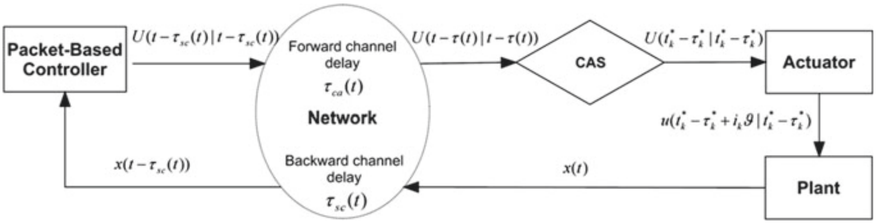


Fig. 5.3 Packet-based networked control systems in continuous time

The block diagram of the continuous packet-based network control system is shown in Fig. 5.3.

5.1.2 A Novel Model for NCSs

Under Algorithm 5.1, and assuming $u(t) = 0, t \in [t_0^*, \ t_1^*], t_0^* = t_1^* - \bar{\tau}_d$, a novel model for NCSs can now be obtained as

$$\mathcal{S}_c^* : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \ t \in [t_{i_k}^*, t_{i_k+1}^*), \ k \geq 1 & (5.6a) \\ u(t) = K(i_k)x(t_k^* - \tau_k^*), \ k \geq 1 & (5.6b) \end{cases}$$

with initial state evolving as $x(t) = x(t_0^*)e^{A(t-t_0^*)} \triangleq \phi(t), t \in [t_0^*, \ t_1^*]$, where i_k and $K(i_k)$ are defined in (5.3) and (5.4), respectively.

It is noticed that the derived model for NCSs in (5.6) is distinct from the previous models as in, e.g., [133] in that the network-induced delay is considered more precisely and the effects of the data packet dropout and disorder are also included in the same model. As shown in (5.6b), the discretization of the network-induced delay and the implementation of the packet-based control approach offer us the advantage of designing different control actions as in (5.3) for different network conditions. It is obvious that this advantage results in at least the same system performance as previous approaches (by designing the same control action for all network conditions), whereas a better performance is expected since more freedom is given to designers.

Remark 5.3 If Assumption 2.1 holds, then the network-induced delay in the backward channel can be known to the controller by using time stamps as done in Sect. 2.2. Thus a different FCS compared with (5.2) can be used, which is defined by

$$U'(t|t - \tau_{sc}(t)) = [u(t|t - \tau_{sc}(t)) \ \ldots \ u(t + (N-1)\vartheta|t - \tau_{sc}(t))]$$

That is, the control signals from time $t - \tau_{sc}(t)$ to $t - 1$ which are obviously useless, are discarded from FCS. As a result, the network-induced delay (data packet dropout as well) in the backward channel will not affect the delay range that the packet-based control approach can handle. One can see that, both cases, with or without the time synchronization, have very similar models (simply replace the round trip delay related parameters in the aforementioned model to the forward channel delay related ones in the presence of the time synchronization). Therefore without loss of generality we will focus only on the system model in (5.6) in the following stability and stabilization analysis.

5.2 Stability and Stabilization

In this section, switched system theory is applied to system $\mathcal{S}_c^*$ to derive a stability criterion. To this end, we first evaluate the growth of the following Lyapunov function candidate $V_{i_k}(x(t)), t \in [t_{i_k}^*, t_{i_k+1}^*), k \geq 1$ defined by

$$\begin{aligned} V_{i_k}(x(t)) =& x^T(t)P_{i_k}x(t) \\ &+ \int_{t-i_k\vartheta}^{t} x^T(s)e^{\alpha(s-t)}R_{i_k}x(s)ds \\ &+ \int_{-i_k\vartheta}^{0}\int_{t+\theta}^{t} \dot{x}^T(s)e^{\alpha(s-t)}Q_{i_k}^1\dot{x}(s)dsd\theta \\ &+ \int_{-(i_k+1)\vartheta}^{-i_k\vartheta}\int_{t+\theta}^{t} \dot{x}^T(s)e^{\alpha(s-t)}Q_{i_k}^2\dot{x}(s)dsd\theta \\ &+ \int_{-(i_k+1)\vartheta}^{0}\int_{t+\theta}^{t} \dot{x}^T(s)e^{\alpha(s-t)}Q_{i_k}^3\dot{x}(s)dsd\theta. \end{aligned}$$

Note that $i_k = 0$ is a special case where $R_0 = 0$, $Q_0^1 = 0$, $Q_0^2 = 0$. For simplicity this case will not be specially addressed in the following analysis.

Lemma 5.1 *For a given constant $\alpha > 0$ and given feedback gain matrices $K(i_k)$, if the following LMI-based problems are feasible,*

$$\mathscr{P}_{i_k}(\alpha): \begin{cases} \exists P_{i_k} > 0, R_{i_k} > 0, Q_{i_k}^i > 0, N_{i_k}^i, i = 1, 2, 3, \\ s.t. \\ \Xi_{i_k}(\alpha) < 0. \end{cases} \tag{5.7}$$

where

$$\Xi_{i_k}(\alpha) = \begin{pmatrix} \Xi'_{i_k}(\alpha) & \Xi_{i_k}^5 \\ * & \Xi_{i_k}^6 \end{pmatrix}, \tag{5.8}$$

$$\Xi'_{i_k}(\alpha) = \begin{pmatrix} \Xi_{i_k}^1(\alpha) + \Xi_{i_k}^2 + (\Xi_{i_k}^2)^T & \Xi_{i_k}^3 \\ * & \Xi_{i_k}^4(\alpha) \end{pmatrix},$$

$$\Xi_{i_k}^1(\alpha) = \begin{pmatrix} \Xi_{i_k}^{11}(\alpha) & 0 & \Xi_{i_k}^{13} \\ * & \Xi_{i_k}^{22}(\alpha) & 0 \\ * & * & 0 \end{pmatrix},$$

$$\Xi_{i_k}^{11}(\alpha) = P_{i_k} A + A^T P_{i_k} + \alpha P_{i_k} + R_{i_k},$$

$$\Xi_{i_k}^{13} = P_{i_k} BK(i_k),$$

$$\Xi_{i_k}^{22}(\alpha) = -e^{-\alpha i_k \vartheta} R_{i_k},$$

$$\Xi_{i_k}^2 = [N_{i_k}^1 + N_{i_k}^3 \quad -N_{i_k}^1 + N_{i_k}^2 \quad -N_{i_k}^2 - N_{i_k}^3],$$

$$\Xi_{i_k}^3 = [N_{i_k}^1 \quad N_{i_k}^2 \quad N_{i_k}^3],$$

$$\Xi_{i_k}^4(\alpha) = -diag\{(i_k\vartheta)^{-1} e^{-\alpha i_k \vartheta} Q_{i_k}^1, \vartheta^{-1} e^{-\alpha(i_k+1)\vartheta} Q_{i_k}^2, ((i_k+1)\vartheta)^{-1} e^{-\alpha(i_k+1)\vartheta} Q_{i_k}^3\},$$

$$\Xi_{i_k}^5 = [\bar{Q}_{i_k} A \quad 0 \quad \bar{Q}_{i_k} BK(i_k) \quad 0 \quad 0 \quad 0]^T,$$

$$\Xi_{i_k}^6 = -\bar{Q}_{i_k},$$

$$\bar{Q}_{i_k} = i_k \vartheta Q_{i_k}^1 + \vartheta Q_{i_k}^2 + (i_k+1)\vartheta Q_{i_k}^3,$$

then along the trajectory of the system in (5.6), the following inequality holds

$$V_{i_k}(x(t)) \le e^{-\alpha(t-t_{i_k}^*)} V_{i_k}(x(t_{i_k}^*)), t \in [t_{i_k}^*, t_{i_k+1}^*), k \ge 1 \tag{5.9}$$

Proof Note that for any $N_{i_k}^i$, $i = 1, 2, 3$, with appropriate dimensions we have

$$\Gamma_{i_k}^1 = \xi^T(t) N_{i_k}^1 [x(t) - x(t - i_k\vartheta) - \int_{t-i_k\vartheta}^{t} \dot{x}(s) ds] = 0 \tag{5.10a}$$

$$\Gamma_{i_k}^2 = \xi^T(t)N_{i_k}^2[x(t-i_k\vartheta) - x(t_k^* - \tau_k^*) - \int_{t_k^*-\tau_k^*}^{t-i_k\vartheta} \dot{x}(s)ds] = 0 \tag{5.10b}$$

$$\Gamma_{i_k}^3 = \xi^T(t)N_{i_k}^3[x(t) - x(t_k^* - \tau_k^*) - \int_{t_k^*-\tau_k^*}^{t} \dot{x}(s)ds] = 0 \tag{5.10c}$$

where $\xi(t) = [x^T(t), x^T(t-i_k\vartheta), x^T(t_k^* - \tau_k^*)]^T$.

Using (5.10) and noticing $t-(i_k+1)\vartheta < t_k^* - \tau_k^* \le t - i_k\vartheta$ for $t \in [t_{i_k}^*, t_{i_k+1}^*)$ we then obtain

$$\begin{aligned}
&\dot{V}_{i_k}(x(t)) + \alpha V_{i_k}(x(t)) \\
= \; & 2x^T(t)P_{i_k}\dot{x}(t) + x^T(t)(\alpha P_{i_k} + R_{i_k})x(t) - x^T(t-i_k\vartheta)e^{-\alpha i_k\vartheta}R_{i_k}x(t-i_k\vartheta) \\
& + \dot{x}^T(t)(i_k\vartheta Q_{i_k}^1 + \vartheta Q_{i_k}^2 + (i_k+1)\vartheta Q_{i_k}^3)\dot{x}(t) - \int_{t-i_k\vartheta}^{t} \dot{x}^T(s)e^{\alpha(s-t)}Q_{i_k}^1\dot{x}(s)ds \\
& - \int_{t-(i_k+1)\vartheta}^{t-i_k\vartheta} \dot{x}^T(s)e^{\alpha(s-t)}Q_{i_k}^2\dot{x}(s)ds - \int_{t-(i_k+1)\vartheta}^{t} \dot{x}^T(s)e^{\alpha(s-t)}Q_{i_k}^3\dot{x}(s)ds \\
\le \; & 2x^T(t)P_{i_k}\dot{x}(t) + x^T(t)(\alpha P_{i_k} + R_{i_k})x(t) - x^T(t-i_k\vartheta)e^{-\alpha i_k\vartheta}R_{i_k}x(t-i_k\vartheta) \\
& + \dot{x}^T(t)(i_k\vartheta Q_{i_k}^1 + \vartheta Q_{i_k}^2 + (i_k+1)\vartheta Q_{i_k}^3)\dot{x}(t) - \int_{t-i_k\vartheta}^{t} \dot{x}^T(s)e^{-\alpha i_k\vartheta}Q_{i_k}^1\dot{x}(s)ds \\
& - \int_{t_k^*-\tau_k^*}^{t-i_k\vartheta} \dot{x}^T(s)e^{-\alpha(i_k+1)\vartheta}Q_{i_k}^2\dot{x}(s)ds - \int_{t_k^*-\tau_k^*}^{t} \dot{x}^T(s)e^{-\alpha(i_k+1)\vartheta}Q_{i_k}^3\dot{x}(s)ds \\
& + 2\Gamma_{i_k}^1 + 2\Gamma_{i_k}^2 + 2\Gamma_{i_k}^3 \\
= \; & \xi^T(t)(\Xi_{i_k}^1 + \Xi_{i_k}^2 + (\Xi_{i_k}^2)^T + \Xi_{i_k}^7 + \Xi_{i_k}^8)\xi(t) - \sum_{i=9}^{11} \Xi_{i_k}^i
\end{aligned}$$

where

$$\Xi_{i_k}^7 = \begin{pmatrix} A^T\bar{Q}_{i_k}A & 0 & A^T\bar{Q}_{i_k}BK(i_k) \\ * & 0 & 0 \\ * & * & (BK(i_k))^T\bar{Q}_{i_k}BK(i_k) \end{pmatrix},$$

$$\begin{aligned}
\Xi_{i_k}^8 = \; & i_k\vartheta N_{i_k}^1 e^{\alpha i_k\vartheta}(Q_{i_k}^1)^{-1}N_{i_k}^1 + \vartheta N_{i_k}^2 e^{\alpha(i_k+1)\vartheta}(Q_{i_k}^2)^{-1}N_{i_k}^2 \\
& + (i_k+1)\vartheta N_{i_k}^3 e^{\alpha(i_k+1)\vartheta}(Q_{i_k}^3)^{-1}N_{i_k}^3,
\end{aligned}$$

$$\Xi_{i_k}^9 = \int_{t-i_k\vartheta}^{t} (\Xi_{i_k}^{91})^T e^{\alpha i_k\vartheta}(Q_{i_k}^1)^{-1}\Xi_{i_k}^{91},$$

$$\Xi_{i_k}^{91} = (N_{i_k}^1)^T \xi(t) + e^{-\alpha i_k \vartheta} Q_{i_k}^1 \dot{x}(s),$$

$$\Xi_{i_k}^{10} = \int_{t_k^*-\tau_k^*}^{t-i_k\vartheta} (\Xi_{i_k}^{101})^T e^{\alpha(i_k+1)\vartheta} (Q_{i_k}^2)^{-1} \Xi_{i_k}^{101},$$

$$\Xi_{i_k}^{101} = (N_{i_k}^2)^T \xi(t) + e^{-\alpha(i_k+1)\vartheta} Q_{i_k}^2 \dot{x}(s),$$

$$\Xi_{i_k}^{11} = \int_{t_k^*-\tau_k^*}^{t} (\Xi_{i_k}^{111})^T e^{\alpha(i_k+1)\vartheta} (Q_{i_k}^3)^{-1} \Xi_{i_k}^{111},$$

$$\Xi_{i_k}^{111} = (N_{i_k}^3)^T \xi(t) + e^{-\alpha(i_k+1)\vartheta} Q_{i_k}^3 \dot{x}(s).$$

Notice that $Q_{i_k}^i > 0, i = 1, 2, 3$ implies $\Xi_{i_k}^4 < 0$, $\Xi_{i_k}^6 < 0$ and $\Xi_{i_k}^i \geq 0, i = 9, 10, 11$. Then by Schur complements, $\Xi_{i_k}(\alpha) < 0$ guarantees

$$\begin{pmatrix} \Xi_{i_k}^1 + \Xi_{i_k}^2 + (\Xi_{i_k}^2)^T + \Xi_{i_k}^7 & \Xi_{i_k}^3 \\ * & \Xi_{i_k}^4 \end{pmatrix} < 0,$$

which furthermore guarantees $\Xi_{i_k}^1 + \Xi_{i_k}^2 + (\Xi_{i_k}^2)^T + \Xi_{i_k}^7 + \Xi_{i_k}^8 < 0$. Thus we obtain

$$\dot{V}_{i_k}(x(t)) + \alpha V_{i_k}(x(t)) \leq 0, t \in [t_{i_k}^*, t_{i_k+1}^*), \forall k \geq 1$$

Integrating this inequality then completes the proof.

Using Lemma 5.1, we then obtain the following stability criterion for the system in (5.6), based on the average dwell time analysis [134].

Theorem 5.1 *Suppose for the system in (5.6) the following inequality holds*

$$\vartheta > \vartheta^* \tag{5.11}$$

where $\vartheta^* = \inf_{\alpha \in \Omega} \{\frac{\ln \mu_\alpha}{\alpha}\}$ *with* $\mu_\alpha = \inf\{\mu | \mu \geq 1, P_{i_k} \leq \mu P_{j_k}, R_{i_k} \leq \mu R_{j_k}, Q_{i_k}^i \leq \mu Q_{j_k}^i, i = 1, 2, 3, \forall i_k, j_k \in \mathbb{M}\}$, $\Omega = \{\alpha | \alpha > 0, \mathscr{P}_{i_k}(\alpha)$ *feasible*, $\forall i_k \in \mathbb{M}\}$ *and* $\mathbb{M} = \{0, 1, 2, \ldots, N-1\}$. *Then the system in (5.6) is exponentially stable.*

Proof For any given $\alpha \in \Omega$, define for the system in (5.6) the following piecewise Lyapunov functional

$$V(x(t)) = V_{i_k}(x(t)), t \in [t_{i_k}^*, t_{i_k+1}^*), k \geq 1$$

From Lemma 5.1 the following inequality holds for $t \in [t^*_{i_k}, t^*_{i_k+1}), k \geq 1$,

$$V(x(t)) = V_{i_k}(x(t)) \leq e^{-\alpha(t-t^*_{i_k})} V_{i_k}(x(t^*_{i_k})) = e^{-\alpha(t-t^*_{i_k})} V(x(t^*_{i_k}))$$

The definition of μ_α implies that

$$V_{i_k}(x(t^*_{i_k})) \leq \mu_\alpha V_{i_k-1}(x(t^{*-}_{i_k})), \forall k \geq 1, \text{if } i_k \geq \lceil \frac{\tau^*_k}{\vartheta} \rceil + 1,$$

or

$$V_{i_k}(x(t^*_{i_k})) \leq \mu_\alpha V_{\bar{i}_{k-1}}(x(t^{*-}_{i_k})), \forall k \geq 2, \text{if } i_k = \lceil \frac{\tau^*_k}{\vartheta} \rceil,$$

where $\bar{i}_{k-1} = \max\{i_{k-1} | i_{k-1} \text{ satisfying (5.3)}\}$.

Thus by iteration we obtain

$$V(x(t)) \leq e^{-\alpha(t-t^*_1)} \mu_\alpha^{I_k - 1} V_1(x(t^*_1)), t \in [t^*_{\bar{i}_k}, t^*_{\bar{i}_k+1})$$

where $V_1(x(t^*_1))$ is defined over $[t^*_1, t^*_1 + \vartheta)$ and $I_k = \sum_k \sum_{i_k} 1$. From Remark 5.2, it is readily seen that $I_k - 1 = \lfloor \frac{(t-t^*_1)}{\vartheta} \rfloor$, and thus

$$V(x(t)) \leq e^{-(\alpha - \frac{\ln \mu_\alpha}{\vartheta})(t-t^*_1)} V_1(x(t^*_1)), t \in [t^*_{\bar{i}_k}, t^*_{\bar{i}_k+1})$$

The definition of ϑ^* implies that $\forall \varepsilon > 0, \exists \alpha_\varepsilon \in \Omega$ and correspondingly μ_{α_ε} such that

$$\vartheta^* + \varepsilon > \frac{\ln \mu_{\alpha_\varepsilon}}{\alpha_\varepsilon}$$

Choosing a sufficiently small $\varepsilon = \varepsilon_0$ such that $\vartheta > \vartheta^* + \varepsilon_0$ yields

$$\vartheta > \vartheta^* + \varepsilon_0 > \frac{\ln \mu_{\alpha_{\varepsilon_0}}}{\alpha_{\varepsilon_0}}$$

which implies

$$\alpha_{\varepsilon_0} - \frac{\ln \mu_{\alpha_{\varepsilon_0}}}{\vartheta} > 0$$

Correspondingly, we obtain

$$V(x(t)) \leq e^{-(\alpha_{\varepsilon_0} - \frac{\ln \mu_{\alpha_{\varepsilon_0}}}{\vartheta})(t-t^*_1)} V_1(x(t^*_1)), t \in [t^*_{\bar{i}_k}, t^*_{\bar{i}_k+1})$$

which completes the proof following the same procedure as in, e.g., [134].

The following proposition solves the synthesis problem of the packet-based control approach to NCSs based on Theorem 5.1.

Proposition 5.1 *Suppose for the system in (5.6) the following inequality holds,*

$$\vartheta > \vartheta^{*'} \tag{5.12}$$

where $\vartheta^{*'} = \inf_{\beta \in \Omega'} \{\frac{\ln \mu'_\beta}{\beta}\}$ *with* $\mu'_\beta = \inf\{\mu' | \mu' \geq 1, L_{i_k} \leq \mu' L_{j_k}, Z_{i_k} \leq \mu' Z_{j_k}, Y^i_{i_k} \leq \mu' Y^i_{j_k}, i = 1, 2, 3, \forall i_k, j_k \in \mathbb{M}\}$*,* $\Omega' = \{\beta | \beta > 0, \mathscr{L}_{i_k}(\beta)$ *feasible*, $\forall i_k \in \mathbb{M}\}$*, and*

$$\mathscr{L}_{i_k}(\beta) : \begin{cases} \exists L_{i_k} > 0, Z_{i_k} > 0, Y^i_{i_k} > 0, M^i_{i_k}, i = 1, 2, 3, V_{i_k}, \\ s.t. \\ \Pi_{i_k}(\beta) < 0. \end{cases}$$

where

$$\Pi_{i_k}(\beta) = \begin{pmatrix} \Pi'_{i_k}(\beta) & \Pi^5_{i_k} \\ * & \Pi^6_{i_k} \end{pmatrix}, \tag{5.13}$$

$$\Pi'_{i_k}(\beta) = \begin{pmatrix} \Pi^1_{i_k}(\beta) + \Pi^2_{i_k} + (\Pi^2_{i_k})^T & \Pi^3_{i_k} \\ * & \Pi^4_{i_k}(\beta) \end{pmatrix},$$

$$\Pi^1_{i_k}(\beta) = \begin{pmatrix} \Pi^{11}_{i_k}(\beta) & 0 & \Pi^{13}_{i_k} \\ * & \Pi^{22}_{i_k}(\beta) & 0 \\ * & * & 0 \end{pmatrix},$$

$$\Pi^{11}_{i_k}(\beta) = AL_{i_k} + L_{i_k}A^T + \beta L_{i_k} + Z_{i_k},$$

$$\Pi^{13}_{i_k} = BV_{i_k},$$

$$\Pi^{22}_{i_k}(\beta) = -e^{-\beta i_k \vartheta} Z_{i_k},$$

$$\Pi^2_{i_k} = [M^1_{i_k} + M^3_{i_k} \quad -M^1_{i_k} + M^2_{i_k} \quad -M^2_{i_k} - M^3_{i_k}],$$

$$\Pi^3_{i_k} = [M^1_{i_k} \;\; M^2_{i_k} \;\; M^3_{i_k}],$$

$$\Pi_{i_k}^4(\beta) = -diag\{(i_k\vartheta)^{-1}e^{-\beta i_k\vartheta}Y_{i_k}^1, \vartheta^{-1}e^{-\beta(i_k+1)\vartheta}Y_{i_k}^2, ((i_k+1)\vartheta)^{-1}e^{-\beta(i_k+1)\vartheta}Y_{i_k}^3\},$$

$$\Pi_{i_k}^5 = [AL_{i_k} \;\; 0 \;\; BV_{i_k} \;\; 0 \;\; 0 \;\; 0]^T,$$

$$\Pi_{i_k}^6 = -L_{i_k}(\bar{Y}_{i_k})^{-1}L_{i_k},$$

$$\bar{Y}_{i_k} = i_k\vartheta Y_{i_k}^1 + \vartheta Y_{i_k}^2 + (i_k+1)\vartheta Y_{i_k}^3.$$

Then, the system in (5.6) is exponentially stabilizable by the control law $K(i_k) = V_{i_k}L_{i_k}^{-1}, i_k \in \mathbb{M}$.

Proof Pre- and post-multiply $diag\{P_{i_k}^{-1}, P_{i_k}^{-1}, P_{i_k}^{-1}, P_{i_k}^{-1}, P_{i_k}^{-1}, P_{i_k}^{-1}, \bar{Q}_{i_k}^{-1}\}$ to (5.8) and let $L_{i_k} = P_{i_k}^{-1}$, $\bar{Y}_{i_k} = L_{i_k}\bar{Q}_{i_k}L_{i_k}$, $Z_{i_k} = L_{i_k}R_{i_k}L_{i_k}$, $M_i^{i_k} = L_{i_k}N_{i_k}^iL_{i_k}$, $Y_i^{i_k} = L_{i_k}Q_{i_k}^iL_{i_k}$, $i = 1, 2, 3$, and $V_{i_k} = K(i_k)L_{i_k}$. Then we complete the proof by using Theorem 5.1.

It is noticed that the feasibility problem of $\mathscr{L}_{i_k}(\beta)$ is no longer LMI conditions because of the term $\Pi_{i_k}^6$. There are several techniques available to deal with this difficulty, among which the cone complementarity technique is one of the most commonly used [135]. In the following theorem, this technique is used to derive a suboptimal solution for $\mathscr{L}_{i_k}(\beta)$ by transforming the feasibility problem of $\mathscr{L}_{i_k}(\beta)$ to a nonlinear minimization problem involving LMI conditions.

Theorem 5.2 *Suppose (5.12) holds for the system in (5.6), where the feasibility problem of* $\mathscr{L}_{i_k}(\beta)$ *is redefined to the following nonlinear minimization problem involving LMI conditions,*

$$\mathscr{L}'_{i_k}(\beta) : \begin{cases} \text{Minimize } Tr(S_{i_k}T_{i_k} + L_{i_k}J_{i_k} + \bar{Y}_{i_k}U_{i_k}) \\ \text{Subject to } L_{i_k} > 0, Z_{i_k} > 0, Y_{i_k}^i > 0, i = 1, 2, 3, \\ \Psi_{i_k}^1 \le 0, \Psi_{i_k}^2 \ge 0, \Psi_{i_k}^3 \ge 0, \Psi_{i_k}^4 \ge 0, \Psi_{i_k}^5 \ge 0. \end{cases}$$

where $\Psi_{i_k}^1 = \begin{pmatrix} \Pi'_{i_k}(\beta) & \Pi_{i_k}^5 \\ * & -S_{i_k} \end{pmatrix}$, $\Psi_{i_k}^2 = \begin{pmatrix} T_{i_k} & J_{i_k} \\ * & U_{i_k} \end{pmatrix}$, $\Psi_{i_k}^3 = \begin{pmatrix} S_{i_k} & I \\ * & T_{i_k} \end{pmatrix}$, $\Psi_{i_k}^4 = \begin{pmatrix} L_{i_k} & I \\ * & J_{i_k} \end{pmatrix}$, $\Psi_{i_k}^5 = \begin{pmatrix} \bar{Y}_{i_k} & I \\ * & U_{i_k} \end{pmatrix}$.

If the solution of $\mathscr{L}'_{i_k}(\beta) = 6n, \forall i_k \in \mathbb{M}$*, then the system in (5.6) is exponentially stabilizable by the control law defined in Proposition 5.1.*

Proof Applying the cone complementarity technique proposed in [135] to (5.13) in Proposition 5.1 then we complete the proof.

5.3 A Numerical Example

Example 5.1 Consider the system in (5.1) with the following system matrices borrowed from [64],

$$A = \begin{pmatrix} -1 & 0 & -0.5 \\ 1 & -0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

When the plant is sampled with $h = 0.1s$, it yields the following discretized system

$$x(k+1) = \begin{pmatrix} 0.9048 & 0 & -0.04881 \\ 0.09278 & 0.9512 & -0.002419 \\ 0 & 0 & 1.051 \end{pmatrix} x(k) + \begin{pmatrix} -0.00246 \\ -8.13 \times e^{-5} \\ 0.1025 \end{pmatrix} u(k).$$

Let $d = 3$, $\bar{\tau} = 0.6s$, and thus $\bar{\tau}_d = 0.9s$. Assume $N = 3$ which means one data packet of the network can contain three steps of control signals. Applying Theorem 5.2 we then obtain the following packet-based controllers with respect to different network conditions,

$$K(0) = \begin{pmatrix} 0.0200 & 0.0004 & -1.3267 \end{pmatrix},$$

$$K(1) = \begin{pmatrix} -0.0004 & -0.0001 & -1.0088 \end{pmatrix},$$

$$K(2) = \begin{pmatrix} -0.0002 & -0.0001 & -1.0098 \end{pmatrix}.$$

Notice here that the continuous network-induced delay (data packet dropout as well) is discretized into three levels by $\vartheta = 0.3s$, corresponding to the above three different packet-based controllers. That is, for different delays, different controllers apply.

The system response and the network-induced delay (data packet dropout as well) is shown in Fig. 5.4 with the initial state $x(t) = [-5\ 0\ 5]^T, t \in [-0.9\ 0)$, which illustrates the effectiveness of the packet-based control approach for NCSs in the presence of network-induced delay, data packet dropout and data packet disorder simultaneously. This can be compared with the example in [64] where only data packet dropout and an unit step delay is considered.

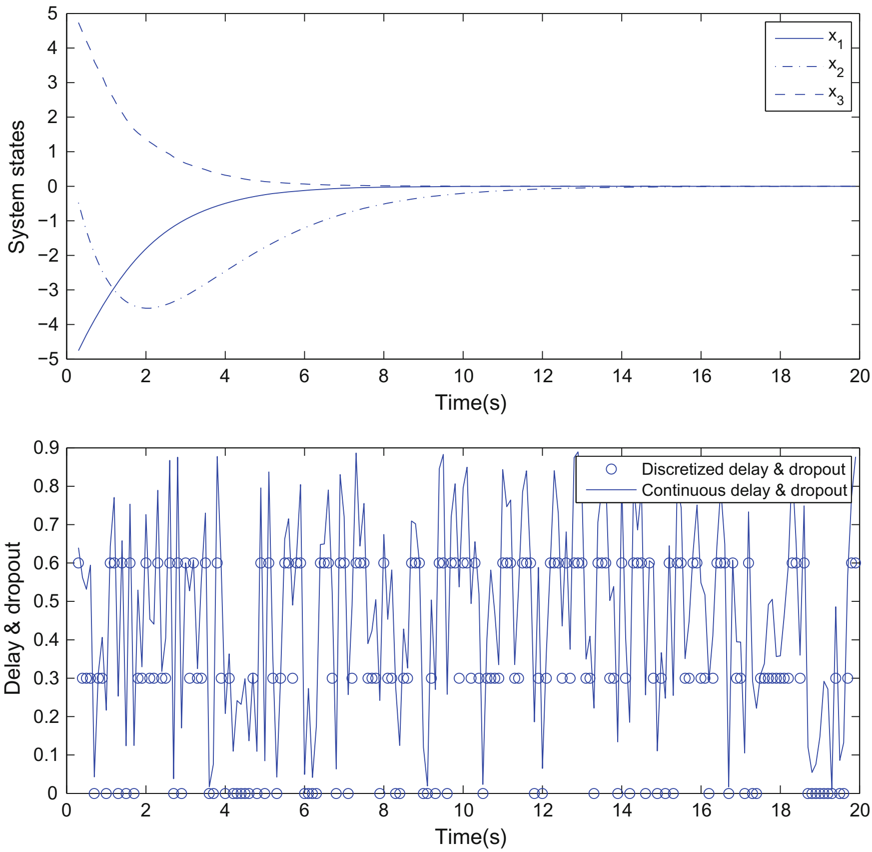


Fig. 5.4 Example 5.1. State response and communication constraints. The discretized delay and dropout is obtained by $i_k\vartheta$

5.4 Summary

By applying the discretization technique to the continuous network-induced delay, the packet-based control approach was extended to the continuous time case in this chapter, from which a novel model for NCSs was derived. The proposed approach and the derived model can deal with network-induced delay, data packet dropout and data packet disorder simultaneously as in the discrete time case in Chap. 2 and offer the designer the freedom of designing different controllers for different network conditions. The stability criterion was obtained using switched system theory and the stabilization problem was also solved, the effectiveness of which was illustrated by a numerical example.

Part II
Analysis

Built on the design framework of the packet-based control approach, this part further discusses the theoretical analysis of the corresponding packet-based networked control systems. This includes the stochastic modelling and analysis of packet-based networked control systems in Chap. 6, a new stability analysis approach inspired by the packet-based control approach in Chap. 7, and a comprehensive analysis of the different delay effects in different channels in networked control systems in Chap. 8. These analyses complete the theoretical foundation of the packet-based control framework for networked control systems.

Chapter 6
Stochastic Stabilization of Packet-Based Networked Control Systems

In the previous chapters the communication constraints including network-induced delay, data packet dropout and data packet disorder, are all assumed to be deterministic which, however, are actually stochastic in nature. This observation motivates the study in this chapter on the stochastic stabilization of PBNCSs under the Markov jump system framework, where the network-induced delay (data packet dropout as well) in round trip is modeled as a homogeneous ergodic Markov chain. Under this framework, the sufficient and necessary conditions for stochastic stability and stabilization of PBNCSs are obtained, which can be compared with the deterministic analysis in Chap. 2 where only sufficient conditions to guarantee the closed-loop stability are obtained and no stabilization analysis is given.

This chapter is organized as follows. The stochastic analysis of PBNCSs is presented in Sect. 6.1, covering the stochastic model of PBNCSs and the corresponding stochastic stability and stabilization analysis. A numerical example is then given in Sect. 6.2 to illustrate the validity of the theoretical analysis and Sect. 6.3 concludes the chapter.

6.1 Stochastic Analysis of PBNCSs

Note that all the analysis in this chapter is based on the packet-based control approach designed in Sect. 2.2; the reader is referred to Sect. 2.2 for more information on the design details of the packet-based control approach and this chapter only focuses on the corresponding stochastic analysis. It is noticed that the control law in (2.5) equals that in (2.6) if $K(\tau_k^*) = K(\tau_{sc,k}^*, \tau_{ca,k}^*)$ which is generally true in practice. Thus for simplicity only the closed-loop system with the control law in (2.6) (i.e., Algorithm 2.2) is analyzed in this chapter. The augmented closed-loop system of system $\mathcal{S}_d$ in (2.1) with the control law in (2.6) was shown in (2.15) as follows,

$$X(k+1) = \Xi(\tau_k^*)X(k)$$

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_6

where $X(k) = [x^T(k)\ x^T(k-1)\ \cdots\ x^T(k-\bar{\tau})]^T$,

$$\Xi(\tau_k^*) = \begin{pmatrix} A & \cdots & BK(\tau_k^*) & \cdots & \cdots \\ I_n & & & & 0 \\ & I_n & & & 0 \\ & & \ddots & & \vdots \\ & & & I_n & 0 \end{pmatrix},$$

and I_n is the identity matrix with rank n.

In this section, the stochastic model of the closed-loop system in (2.15) is first obtained, which is then further analyzed to derive the stochastic stability and stabilization conditions as follows.

6.1.1 Stochastic Model of PBNCSs

In NCSs, it is reasonable to model the round trip delay $\{\tau_k; k = 0, 1, \ldots\}$ as a homogeneous ergodic Markov chain [56]. Here in order to take explicit account of data packet dropout, Markov chain $\{\tau_k; k = 0, 1, \ldots\}$ is assumed to take values from $\mathcal{M} = \{0, 1, 2, \ldots, \bar{\tau}, \infty\}$ where $\tau_k = 0$ means no delay in round trip while $\tau_k = \infty$ implies a data packet dropout in either the backward or the forward channel. Let the transition probability matrix of $\{\tau_k; k = 0, 1, \ldots\}$ be denoted by $\Lambda = [\lambda_{ij}]$ where

$$\lambda_{ij} = P\{\tau_{k+1} = j | \tau_k = i\}, i, j \in \mathcal{M}$$

$P\{\tau_{k+1} = j | \tau_k = i\}$ is the probability of τ_k jumping from state i to j, $\lambda_{ij} \geq 0$ and

$$\sum_{j \in \mathcal{M}} \lambda_{ij} = 1, \forall i, j \in \mathcal{M}$$

The initial distribution of $\{\tau_k; k = 0, 1, \ldots\}$ is defined by

$$P\{\tau_0 = i\} = p_i, i \in \mathcal{M}$$

According to the comparison rule in (2.10), the round trip delay of the control actions that are actually applied to the plant can be determined by the following formula,

$$\tau_{k+1}^* = \begin{cases} \tau_k^* + 1, & \text{if } \tau_{k+1} > \tau_k^*; \\ \tau_k^* - r, & \text{if } \tau_k^* - r = \tau_{k+1} \leq \tau_k^*. \end{cases} \tag{6.1}$$

The following lemma shows that the delay τ_k^* in (6.1) that is derived from the comparison rule in (2.10) is a Markov chain.

Lemma 6.1 $\{\tau_k^*; k = 0, 1, \ldots\}$ *is a non-homogeneous Markov chain with state space* $\mathcal{M}^* = \{0, 1, 2, \ldots, \bar{\tau}\}$ *whose transition probability matrix* $\Lambda^*(k) = [\lambda_{ij}^*(k)]$ *is defined by*

$$\lambda_{ij}^*(k) = \begin{cases} \dfrac{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}(k)\lambda_{l_1 j}}{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}(k)}, \ j \leq i; \\ \dfrac{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \sum\limits_{l_2 \in \mathcal{M}, l_2 > i} \pi_{l_1}(k)\lambda_{l_1 l_2}}{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}(k)}, \ j = i + 1; \\ 0, \ otherwise. \end{cases} \tag{6.2}$$

where $\pi_j(k) = \sum\limits_{i \in \mathcal{M}} p_i \lambda_{ij}^{(k)}$ *and* $\lambda_{ij}^{(k)}$ *is the k-step transition probability of* τ_k *from state* i *to* j.

Proof The comparison rule in (6.1) implies that the probability event $\{\tau_k^* = i\} \in \sigma(\tau_k, \tau_{k-1}, \ldots, \tau_1, \tau_0)$. Thus it is readily concluded that τ_k^* is also a Markov chain since τ_k as a Markov chain evolves independently. It is obvious that τ_k^* can not be ∞ and thus its state space is $\mathcal{M}^* = \{0, 1, 2, \ldots, \bar{\tau}\}$. Furthermore, since $\{\tau_k^* = i\} = \{\tau_{k-1}^* = i - 1, \tau_k > i - 1\} \cup \{\tau_{k-1}^* \geq i, \tau_k = i\}$ we have

1. If $j \leq i$, then

$$\begin{aligned} P\{\tau_{k+1}^* = j | \tau_k^* = i\} &= P\{\tau_{k+1} = j | \tau_k^* = i\} \\ &= P\{\tau_{k+1} = j | \tau_k \geq i\} \\ &= \frac{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}(k)\lambda_{l_1 j}}{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}(k)} \end{aligned}$$

2. If $j = i + 1$, then

$$\begin{aligned} P\{\tau_{k+1}^* = j | \tau_k^* = i\} &= P\{\tau_{k+1} > i | \tau_k^* = i\} \\ &= P\{\tau_{k+1} > i | \tau_k \geq i\} \\ &= \sum_{l_2 \in \mathcal{M}, l_2 > i} P\{\tau_{k+1} = l_2 | \tau_k \geq i\} \\ &= \frac{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \sum\limits_{l_2 \in \mathcal{M}, l_2 > i} \pi_{l_1}(k)\lambda_{l_1 l_2}}{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}(k)} \end{aligned}$$

which completes the proof.

With Lemma 6.1, the closed-loop system in (2.15) can now be regarded as a Markov jump system where the system matrix $\Xi(\tau_k^*)$ evolves with the Markov chain $\{\tau_k^*; k = 0, 1, \ldots\}$ whose transition probability matrix is defined in (6.2).

Remark 6.1 The data packet dropout is explicitly considered by including the state $\tau_k = \infty$ into the state space Λ; The data packet disorder is also considered by (6.1): In our stochastic model the network-induced delay, data packet dropout and data packet disorder are all considered simultaneously.

The following well-known result for homogeneous ergodic Markov chains is required for the stochastic stability analysis in this chapter.

Lemma 6.2 *([136]) For the homogeneous ergodic Markov chain* $\{\tau_k; k = 0, 1, \ldots\}$ *with any initial distribution, there exists a limit probability distribution* $\pi = \{\pi_i; \pi_i > 0, i \in \mathcal{M}\}$ *such that for each* $j \in \mathcal{M}$,

$$\sum_{i \in \mathcal{M}} \lambda_{ij}\pi_i = \pi_j, \quad \sum_{i \in \mathcal{M}} \pi_i = 1$$

and

$$|\pi_i(k) - \pi_i| \leq \eta\xi^k \tag{6.3}$$

for some $\eta \geq 0$ *and* $0 < \xi < 1$.

Proposition 6.1 *For* N_1 *that is large enough and some nonzero* η^* *the following inequality holds*

$$|\lambda_{ij}^*(k) - \lambda_{ij}^*| \leq \eta^*\xi^k, \; k > N_1$$

where $\Lambda^* = [\lambda_{ij}^*]$ *with*

$$\lambda_{ij}^* = \begin{cases} \dfrac{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}\lambda_{l_1 j}}{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}}, & \text{if } j \leq i; \\ \dfrac{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \sum\limits_{l_2 \in \mathcal{M}, l_2 > i} \pi_{l_1}\lambda_{l_1 l_2}}{\sum\limits_{l_1 \in \mathcal{M}, l_1 \geq i} \pi_{l_1}}, & \text{if } j = i + 1; \\ 0, & \text{otherwise}. \end{cases} \tag{6.4}$$

Proof It can be readily obtained from (6.2), (6.3) and (6.4).

6.1.2 Stochastic Stability and Stabilization

The following definition of stochastic stability is used in this chapter.

Definition 6.1 *(Stochastic stability, see [56].)* The closed-loop system in (2.15) is said to be stochastically stable if for every finite $X_0 = X(0)$ and initial state $\tau_0^* = \tau^*(0) \in \mathcal{M}$, there exists a finite $W > 0$ such that the following inequality holds,

$$E\{\sum_{k=0}^{\infty}||X(k)||^2|X_0,\tau_0^*\} < X_0^T W X_0$$

where $E\{X\}$ is the expectation of the random variable X.

Theorem 6.1 (Stochastic stability) *The closed-loop system in (2.15) is stochastically stable if and only if there exists $P(i) > 0, i \in \mathcal{M}^*$ such that the following $(\bar{\tau}+1)$ LMIs hold*

$$L(i) = \sum_{j\in\mathcal{M}^*}\lambda_{ij}^*\Xi^T(j)P(j)\Xi(j) - P(i) < 0, \forall i \in \mathcal{M}^* \tag{6.5}$$

Proof Sufficiency. For the closed-loop system in (2.15), consider the following quadratic function given by

$$V(X(k),k) = X^T(k)P(\tau_k^*)X(k)$$

We have

$$\begin{aligned}
&E\{\Delta V(X(k),k)\}\\
=&E\{X^T(k+1)P(\tau_{k+1}^*)X(k+1)|X(k),\tau_k^*=i\} - X^T(k)P(i)X(k)\\
=&\sum_{j\in\mathcal{M}^*}\lambda_{ij}^*(k+1)X^T(k)\Xi^T(j)P(j)\Xi(j)X(k) - X^T(k)P(i)X(k)\\
=&X^T(k)[\sum_{j\in\mathcal{M}^*}\lambda_{ij}^*(k+1)\Xi^T(j)P(j)\Xi(j) - P(i)]X(k)
\end{aligned}$$

From (6.5) we obtain

$$\begin{aligned}
&X^T(k)[\sum_{j\in\mathcal{M}^*}\lambda_{ij}^*\Xi^T(j)P(j)\Xi(j) - P(i)]X(k)\\
\le& -\lambda_{\min}(-L(i))X^T(k)X(k)\\
\le& -\beta||X(k)||^2
\end{aligned}$$

where $\beta = \inf\{\lambda_{\min}(-L(i)); i \in \mathcal{M}^*\} > 0$. Thus for $k > N_1$,

$$\begin{aligned}
&E\{\Delta V(X(k),k)\}\\
=&X^T(k)[\sum_{j\in\mathcal{M}^*}\lambda_{ij}^*(k+1)\Xi^T(j)P(j)\Xi(j) - P(i)]X(k)\\
\le&X^T(k)[\sum_{j\in\mathcal{M}^*}\lambda_{ij}^*\Xi^T(j)P(j)\Xi(j) - P(i)]X(k)\\
&+X^T(k)\sum_{j\in\mathcal{M}^*}|\lambda_{ij}^*(k+1)-\lambda_{ij}^*|\Xi^T(j)P(j)\Xi(j)X(k)
\end{aligned}$$

$$\leq -\beta||X(k)||^2 + \eta^*\xi^{k+1}X^T(k)\sum_{j\in\mathcal{M}^*}\Xi^T(j)P(j)\Xi(j)X(k)$$
$$\leq(\alpha\eta^*\xi^{k+1} - \beta)||X(k)||^2$$

where $\alpha = \sup\{\lambda_{\max}(\Xi^T(j)P(j)\Xi(j));\ j \in \mathcal{M}^*\} > 0$. Let $N_2 = \inf\{M;\ M \in \mathbb{N}^+,\ M > \max\{N_1, \log_\xi \frac{\beta}{\alpha\eta^*} - 1\}\}$. Then we have for $k \geq N_2$

$$E\{\Delta V(X(k), k)\} \leq -\beta^*||X(k)||^2$$

where $\beta^* = \beta - \alpha\eta^*\xi^{N_2+1} > 0$. Summing from N_2 to $N > N_2$ we obtain

$$E\{\sum_{k=N_2}^{N}||X(k)||^2\}$$
$$\leq\frac{1}{\beta^*}(E\{V(X(N_2), N_2)\} - E\{V(X(N+1), N+1)\})$$
$$\leq\frac{1}{\beta^*}E\{V(X(N_2), N_2)\}$$

which implies that

$$E\{\sum_{k=0}^{\infty}||X(k)||^2\} \leq \frac{1}{\beta^*}E\{V(X(N_2), N_2)\} + E\{\sum_{k=0}^{N_2-1}||X(k)||^2\}$$

This proves the stochastic stability of the closed-loop system in (2.15) by Definition 6.1.

Necessity. Suppose the closed-loop system in (2.15) is stochastically stable, that is,

$$E\{\sum_{k=0}^{\infty}||X(k)||^2|X_0, \tau_0^*\} < X_0^T W X_0 \tag{6.6}$$

Define

$$X^T(n)\bar{P}(N-n, \tau_n^*)X(n) = E\{\sum_{k=n}^{N}X^T(k)Q(\tau_k^*)X(k)|X_n, \tau_n^*\}$$

with $Q(\tau_k^*) > 0$. It is noticed that $X^T(n)\bar{P}(N-n, \tau_n^*)X(n)$ is upper bounded from (6.6) and non-decreasing as N increases since $Q(\tau_k^*) > 0$. Therefore its limit exists which is denoted by

$$X^T(n)P(i)X(n) = \lim_{N\to\infty}X^T(n)\bar{P}(N-n, \tau_n^* = i)X(n) \tag{6.7}$$

Since (6.7) is valid for any $X(n)$, we obtain

$$P(i) = \lim_{N\to\infty} \bar{P}(N-n, \tau_n^* = i) > 0$$

Now consider

$$\begin{aligned}
&E\{X^T(n)\bar{P}(N-n, \tau_n^*)X(n) \\
&\quad - X^T(n+1)\bar{P}(N-n-1, \tau_{n+1}^*)X(n+1)|X_n, \tau_n^* = i\} \\
=&X^T(n)[\bar{P}(N-n, i) - \sum_{j\in\mathcal{M}^*} \lambda_{ij}^*(n+1)\Xi^T(j)\bar{P}(N-n-1, j)\Xi(j)]X(n) \\
=&X^T(n)Q(i)X(n)
\end{aligned} \tag{6.8}$$

Since (6.8) is valid for any $X(n)$, we obtain

$$\bar{P}(N-n, i) - \sum_{j\in\mathcal{M}^*} \lambda_{ij}^*(n+1)\Xi^T(j)\bar{P}(N-n-1, j)\Xi(j)) = Q(i) > 0$$

Let $N \to \infty$,

$$P(i) - \sum_{j\in\mathcal{M}^*} \lambda_{ij}^*(n+1)\Xi^T(j)P(j)\Xi(j) > 0, \forall n$$

Let $n \to \infty$,

$$P(i) - \sum_{j\in\mathcal{M}^*} \lambda_{ij}^*\Xi^T(j)P(i)\Xi(j) > 0$$

which completes the proof.

The stochastic stabilization result in Corollary 6.1 readily follows using the Schur complement.

Corollary 6.1 (Stochastic stabilization) *System $\mathcal{S}_d$ is stochastically stabilizable using the packet-based control approach with the control law in (2.6) if and only if there exist $P(i) > 0, Z(i) > 0, K(i), i \in \mathcal{M}^*$ such that the following $(\bar{\tau}+1)$ LMIs hold*

$$\begin{pmatrix} P(i) & R(i) \\ R^T(i) & Q \end{pmatrix} > 0, \ i \in \mathcal{M}^*$$

with the equation constraints

$$P(i)Z(i) = I, \forall i \in \mathcal{M}^* \tag{6.9}$$

where $R(i) = [(\lambda_{i0}^*)^{\frac{1}{2}} \Xi^T(0) \ldots (\lambda_{i\bar{\tau}}^*)^{\frac{1}{2}} \Xi^T(\bar{\tau})]$, $Q = diag\{Z(0) \ldots Z(\bar{\tau})\}$ *and* $\Xi(i)$ *(consequently* $K(i)$*) is defined in (2.15).*

The LMIs in Corollary 6.1 with the matrix inverse constraints in (6.9) can be solved using the Cone Complementarity Linearization (CCL) algorithm [137].

6.2 A Numerical Example

In this section, a numerical example is considered to illustrate the validity of Theorem 6.1 and Corollary 6.1.

Example 6.1 Consider the example in [56] where the system matrices are as follows,

$$A = \begin{pmatrix} 1.0000 & 0.1000 & -0.0166 & -0.0005 \\ 0 & 1.0000 & -0.3374 & -0.0166 \\ 0 & 0 & 1.0996 & 0.1033 \\ 0 & 0 & 2.0247 & 1.0996 \end{pmatrix}, B = \begin{pmatrix} 0.0045 \\ 0.0896 \\ -0.0068 \\ -0.1377 \end{pmatrix}.$$

This system is open-loop unstable with the eigenvalues at 1, 1, 1.5569 and 0.6423. In the simulation, the random round trip delay is bounded by 4, i.e., $\tau_k \in \mathcal{M} = \{0, 1, 2, 3, 4, \infty\}$, with the transition probability matrix as follows,

$$\Lambda = \begin{pmatrix} 0.1 & 0.2 & 0.2 & 0.3 & 0.2 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.24 & 0.06 & 0.48 & 0.12 & 0.1 & 0 \\ 0.15 & 0.25 & 0.3 & 0.15 & 0.1 & 0.05 \\ 0.3 & 0.3 & 0.2 & 0.1 & 0.1 & 0 \\ 0.3 & 0.3 & 0.15 & 0.15 & 0.1 & 0 \end{pmatrix}.$$

The limit distribution of the above ergodic Markov chain can be simply obtained as in Lemma 6.2,

$$\pi = \begin{pmatrix} 0.1982 & 0.1814 & 0.3000 & 0.1738 & 0.1198 & 0.0268 \end{pmatrix}.$$

Λ^* in Proposition 6.1 can then be calculated by (6.4) as

$$\Lambda^* = \begin{pmatrix} 0.1982 & 0.8018 & 0 & 0 & 0 \\ 0.2224 & 0.1767 & 0.6008 & 0 & 0 \\ 0.2290 & 0.1699 & 0.3612 & 0.2398 & 0 \\ 0.2186 & 0.2729 & 0.2501 & 0.1313 & 0.1271 \\ 0.3000 & 0.3000 & 0.1909 & 0.1091 & 0.1000 \end{pmatrix}.$$

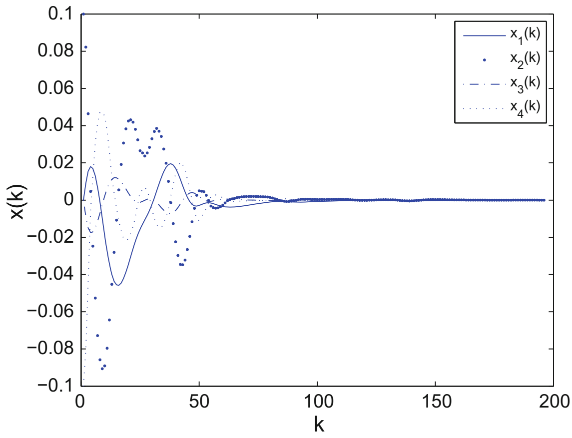


Fig. 6.1 Example 6.1. States evolution of the packet-based control approach to NCSs

From Corollary 6.1, the packet-based controller is obtained as follows,

$$K(0) = \left(0.5292\ 0.6489\ 22.4115\ 2.8205\right),$$
$$K(1) = \left(0.3792\ 0.8912\ 20.2425\ 5.3681\right),$$
$$K(2) = \left(0.0499\ 0.4266\ 15.6574\ 5.7322\right),$$
$$K(3) = \left(-0.4400\ -0.3003\ 9.2976\ 5.0540\right),$$
$$K(4) = \left(-0.8400\ -1.3422\ 2.7723\ 2.9173\right).$$

The state trajectories of the closed-loop system under the packet-based controller are shown in Fig. 6.1 with the initial states $x(-3) = x(-2) = x(-1) = x(0) = [0\ 0.1\ 0\ -0.1]^T$, which illustrates the stochastic stability of the closed-loop system.

6.3 Summary

It is observed that the communication constraints in NCSs including network-induced delay, data packet dropout and data packet disorder, are stochastic in nature. Based on this observation, a stochastic analysis was presented for the packet-based control approach proposed in Chap. 2. Both stochastic stability and stabilization conditions were obtained, which was then validated by a numerical example.

Chapter 7
Stability of Networked Control Systems: A New Time Delay Systems Approach

A large number of research works on Networked Control Systems (NCSs) are from the time delay system perspective, however, it is noticed that the description of the network-induced delay is too general to represent the practical reality. By recognizing this fact, a novel time delay system model for NCSs is thus obtained by depicting the network-induced delay more specifically, inspired by the packet-based control approach. Based on this model, stability (robust stability) and stabilization results are obtained using delay-dependent analysis approach, which are less conservative compared with conventional models due to the specific description of the network-induced delay in the new model. A numerical example illustrates the effectiveness of the proposed approach.

This chapter is organized as follows. In Sect. 7.1, we first present the novel model for NCSs, based on which the stability and stabilization results are then obtained in Sect. 7.2. A numerical example is considered to illustrate the effectiveness of the proposed approach in Sects. 7.3 and 7.4 concludes the chapter.

7.1 The Novel Time Delay System Model for PBNCSs

Consider the NCSs setup illustrated in Fig. 2.1, where $\tau_{sc,k}$ and $\tau_{ca,k}$ are the network-induced delays in the sensor-to-controller and the controller-to-actuator channels, respectively, and the plant is represented by the following discrete-time linear model with the full state information measurable,

$$x(k+1) = Ax(k) + Bu(k) \tag{7.1}$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the control input, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are constant system matrices.

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_7

We first discuss the control law based on the packet-based control approach for completeness; a simpler discussion can be found in Sect. 2.2.1. For simplicity, in this chapter the network-induced delays in the sensor-to-controller and the controller-to-actuator channels are not considered separately but only the round trip delay is of interest, which is denoted by τ_k at time k, i.e., $\tau_k = \tau_{sc,k} + \tau_{ca,k}$. Using conventional modeling approaches, the control law for the system in (7.1) is typically obtained as

$$u(k) = Kx(k - \tau_k) \tag{7.2}$$

where the feedback gain K is fixed for all the network conditions. In view of the time-varying network conditions, a more reasonable control law is of the following form,

$$u(k) = K(\tau_k)x(k - \tau_k) \tag{7.3}$$

where the feedback gain $K(\tau_k)$ is designed with respect to different network conditions and thus gives the system designer more freedom to compensate for the communication constraints.

In the system models in both (7.2) and (7.3), the round trip delay τ_k is typically assumed to be time-varying and upper bounded. This assumption is generally true in practice as well as necessary in theory. However, this assumption can readily result in a situation where for some specific time k',

$$k' + 1 - \tau_{k'+1} < k' - \tau_{k'}$$

The above inequality means that, the control action at time $k' + 1$ is based on the outdated state information at time $k' + 1 - \tau_{k'+1}$ instead of the more updated information at time $k' - \tau_{k'}$ which is already available for the actuator. This control strategy is obviously unacceptable in practice.

By recognizing this defect in conventional models for NCSs, we thus have the following reasonable assumption for the network-induced delay in NCSs, denoted by τ_k^* to distinguish from τ_k in (7.2) and (7.3),

$$\tau_{k+1}^* \leq \tau_k^* + 1, \forall k \tag{7.4}$$

where τ_k^* is time-varying and upper bounded, i.e., $\tau_k^* \in \Omega \triangleq \{2, \ldots, \bar{\tau}\}$. τ_k^* is not less than 2 due to the fact that there is at least one step delay in both the sensor-to-controller and the controller-to-actuator channels respectively. Notice that the condition in (7.4) is not naturally held for conventional control approaches to NCSs but can be readily realized in practice by the packet-based control approach.

The control law in (7.3) can now be rewritten as

$$u(k) = K(\tau_k^*)x(k - \tau_k^*) \tag{7.5}$$

where τ_k^* satisfies (7.4).

Based on the control law in (7.5), the closed-loop system for the NCS in (7.1) with the assumption in (7.4) can be represented by

$$x(k+1) = Ax(k) + BK(\tau_k^*)x(k - \tau_k^*) \tag{7.6}$$

where τ_k^* satisfies (7.4) and the feedback gains $K(\tau_k^*)$ are to be designed. This model is different from conventional models available in the literature in mainly two aspects: (1) the assumption for the network-induced delay in (7.4); (2) the delay-dependent feedback gains $K(\tau_k^*)$. Based on this model, stability and stabilization analysis is then conducted in the following section which results in less conservative conditions compared with those with conventional models. Another case in the presence of the following time-varying uncertainties will also be considered within this framework,

$$x(k+1) = (A + \Delta A(k))x(k) + (B + \Delta B(k))K(\tau_k^*)x(k - \tau_k^*) \tag{7.7}$$

where the time-varying parameter uncertainties are norm-bounded, i.e.,

$$[\Delta A(k)\ \Delta B(k)] = DE(k)[F_A\ F_B] \tag{7.8}$$

with D, F_A and F_B being known constant matrices and

$$E^T(k)E(k) \leq I$$

7.2 Stability and Stabilization

In this section, the stability of the nominal system in (7.6) is first considered. The result obtained is then extended to the case with time-varying parameter uncertainties in (7.7). Furthermore, a stabilized controller design method is also obtained in terms of LMIs.

The following stability theorem for the closed-loop system in (7.6) can be obtained based on delay-dependent analysis.

Theorem 7.1 *Given $\lambda \geq 1$. The closed-loop system in (7.6) is stable if there exist $P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \geq 0$, T_i^1, T_i^2 with appropriate dimensions such that*

1. $\forall i \in \Omega$,

$$\Phi_i = \begin{pmatrix} \Phi_i^{11} & \Phi_i^{12} & (A-I)^T H_i \\ * & \Phi_i^{22} & (BK(i))^T H_i \\ * & * & -H_i \end{pmatrix} < 0 \tag{7.9}$$

$$\Psi_i = \begin{pmatrix} S_i^{11} & S_i^{12} & T_i^1 \\ * & S_i^{22} & T_i^2 \\ * & * & \frac{1}{\lambda} R_i \end{pmatrix} \geq 0 \tag{7.10}$$

2. $\forall i, j \in \Omega$

$$P_i \leq \lambda P_j, Q_i \leq \lambda Q_j, R_i \leq \lambda R_j \tag{7.11}$$

where

$$\Phi_i^{11} = (\lambda - 1)P_i + Q_i + 2\lambda P_i(A - I) + T_i^1 + (T_i^1)^T + iS_i^{11},$$

$$\Phi_i^{12} = \lambda P_i BK(i) - T_i^1 + (T_i^2)^T + iS_i^{12},$$

$$\Phi_i^{22} = -T_i^2 - (T_i^2)^T + iS_i^{22},$$

$$H_i = \lambda P_i + \bar{\tau} R_i.$$

Proof Let

$$z(l) = x(l + 1) - x(l)$$

Then

$$x(k) - x(k - \tau_k^*) - \sum_{l=k-\tau_k^*}^{k-1} z(l) = 0$$

Define the following Lyapunov functional where we suppose at time k, $\tau_k^* = i \in \Omega$,

$$V_i(k) = V_i^1(k) + V_i^2(k) + V_i^3(k)$$

with

$$V_i^1(k) = x^T(k)P_i x(k)$$

$$V_i^2(k) = \sum_{l=k-\tau_k^*}^{k-1} x^T(l)Q_{\tau_l^*}x(l)$$

$$V_i^3(k) = \sum_{m=-\bar{\tau}+1}^{0} \sum_{l=k+m-1}^{k-1} z^T(l) R_{\tau_l^*} z(l)$$

Define $\Delta V_i(k) = V_{\tau_{k+1}^*}(k+1) - V_i(k)$. Then along the trajectory of the system in (7.6), we have

$$\begin{aligned}\Delta V_i^1(k) =& x^T(k+1) P_{\tau_{k+1}^*} x(k+1) - x^T(k) P_i x(k) \\ \leq& (\lambda - 1) x^T(k) P_i x(k) + 2\lambda x^T(k) P_i z(k) + \lambda z^T(k) P_i z(k)\end{aligned} \tag{7.12}$$

due to (7.11),

$$\Delta V_i^2(k) = \left(\sum_{l=k-\tau_{k+1}^*+1}^{k-1} - \sum_{l=k-\tau_k^*}^{k-1} \right) x^T(l) Q_{\tau_l^*} x(l) + x^T(k) Q_i x(k) \leq x^T(k) Q_i x(k) \tag{7.13}$$

due to (7.4), and

$$\begin{aligned}\Delta V_i^3(k) =& \sum_{m=-\bar{\tau}+1}^{0} \left(\sum_{l=k+m}^{k} - \sum_{l=k+m-1}^{k-1} \right) z^T(l) R_{\tau_l^*} z(l) \\ =& \bar{\tau} z^T(k) R_i z(k) - \sum_{l=k-\bar{\tau}}^{k-1} z^T(l) R_{\tau_l^*} z(l) \\ \leq& \bar{\tau} z^T(k) R_i z(k) - \sum_{l=k-\tau_k^*}^{k-1} z^T(l) R_{\tau_l^*} z(l)\end{aligned} \tag{7.14}$$

Notice that

$$z(k) = (A - I)x(k) + BK(i)x(k - \tau_k^*) \tag{7.15}$$

and

$$R_i \geq \frac{1}{\lambda} R_j,\ Q_i \geq \frac{1}{\lambda} Q_j, \forall i, j \in \Omega \tag{7.16}$$

In addition, we have for any T_i^1, T_i^2 with appropriate dimensions,

$$2[x^T(k)T_i^1 + x^T(k - \tau_k^*)T_i^2] \times [x(k) - x(k - \tau_k^*) - \sum_{l=k-\tau_k^*}^{k-1} z(l)] = 0 \tag{7.17}$$

and for any S_i with appropriate dimensions,

$$i\zeta_1^T(k)S_i\zeta_1(k) - \sum_{l=k-\tau_k^*}^{k-1} \zeta_1^T(k)S_i\zeta_1(k) = 0 \tag{7.18}$$

where $\zeta_1(k) = [x^T(k)\ x^T(k-\tau_k^*)]^T$.

From (7.12)–(7.18) we then obtain

$$\Delta V_i(k) \le \zeta_1^T(k)\Xi_i\zeta_1(k) - \frac{1}{\lambda}\sum_{l=k-\tau_k^*}^{k-1} \zeta_2^T(k,l)\Psi_i\zeta_2(k,l) \tag{7.19}$$

where $\Xi_i = \begin{pmatrix} \Phi_i^{11} + \Upsilon_i^{11} & \Phi_i^{12} + \Upsilon_i^{12} \\ * & \Phi_i^{22} + \Upsilon_i^{22} \end{pmatrix}$, $\Upsilon_i^{11} = (A-I)^T H_i (A-I)$, $\Upsilon_i^{12} = (A-I)^T H_i BK(i)$, $\Upsilon_i^{22} = (BK(i))^T H_i BK(i)$, $H_i = \lambda P_i + \bar{\tau} R_i$, and $\zeta_2(k,l) = [\zeta_1^T(k), z^T(l)]^T$. It is noticed that the system is stable if $\Xi_i < 0$ and $\Psi_i \ge 0$. Furthermore, noticing that by Schur complement that $\Xi_i < 0$ is equivalent to $\Phi_i < 0$, we then complete the proof.

Remark 7.1 In [117], a typical discrete-time system with time-varying state delay was considered, where the Lyapunov functional was constructed with an additional item being (using the notations in this chapter)

$$V_i^4 = \sum_{m=-\bar{\tau}+2}^{3} \sum_{l=k+1-m}^{k-1} x^T(l)Q_{\tau_l^*}x(l)$$

This item was included mainly to cancel out the first item of the difference of $\Delta V_i^2(k)$ in (7.13), since the value of $(\sum_{l=k-\tau_{k+1}^*+1}^{k-1} - \sum_{l=k-\tau_k^*}^{k-1})x^T(l)Q_{\tau_l^*}x(l)$ can not be estimated without the assumption in (7.4) and thus can not be dropped directly as done in this chapter. In [133], the Lyapunov functional used in [117] was further improved by adding another new item to eliminate the negative effect brought by the introduction of V_i^4. However, without using the item V_i^4 in our Lyapunov functional, a less complex result is obtained in this chapter which is also less conservative since no such inequalities are used in the proof. On the other hand, in a recent article [138], a similar delay system was considered in the switched system context, which derived a very similar model to that used in this chapter. The aforementioned additional item in the Lyapunov functional V_i^4 was still used, and for the reduction of the coupling under the switched system context, common Q and R were used in the Lyapunov functional which obviously led to conservativeness compared with the result in this chapter.

Based on Theorem 7.1, a robust stability theorem can then be obtained for the closed-loop system with time-varying uncertainties in (7.7).

Theorem 7.2 *Given $\lambda \ge 1$ and the feedback gains $K(i), i \in \Omega$. The closed-loop system with time-varying uncertainties in* (7.7) *is robust stable if there exist*

$P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \geq 0$, T_i^1, T_i^2 *with appropriate dimensions and a scalar* $\gamma > 0$ *such that*

1. $\forall i \in \Omega$,

$$\begin{pmatrix} \Phi_i^{11} + \gamma F_A^T F_A & \Phi_i^{12} + \gamma F_A F_B K(i) & (A-I)H_i & P_i D \\ * & \Phi_i^{22} + \gamma (F_B K(i))^T F_B K(i) & (BK(i))^T H_i & 0 \\ * & * & -H_i & H_i D \\ * & * & * & -\gamma I \end{pmatrix} < 0 \tag{7.20}$$

$$\Psi_i \geq 0 \tag{7.21}$$

2. $\forall i, j \in \Omega$,

$$P_i \leq \lambda P_j, \; Q_i \leq \lambda Q_j, \; R_i \leq \lambda R_j \tag{7.22}$$

where Φ_i^{11}, Φ_i^{12}, Φ_i^2, Ψ_i, H_i *are defined in Theorem 7.1 and* D, F_A, F_B *are defined in* (7.8).

Proof The above theorem can be obtained following a standard analysis for systems with time-varying parameter uncertainties, as done in Theorem 7.3 in [117]. Therefore we omit the technical details for brevity.

Based on Theorem 7.1, the following stabilized controller design method can also be obtained in terms of LMIs.

Theorem 7.3 *Given* $\lambda \geq 1$. *The system in* (7.6) *is stabilizable if there exist* $L_i = L_i^T > 0$, $W_i = W_i^T > 0$, $M_i = M_i^T > 0$, $X_i = \begin{pmatrix} X_i^{11} & X_i^{12} \\ (X_i^{12})^T & X_i^{22} \end{pmatrix} \geq 0$, Y_i^1, Y_i^2, V_i *with appropriate dimensions such that*

1. $\forall i \in \Omega$,

$$\Pi_i = \begin{pmatrix} \Pi_i^{11} & \Pi_i^{12} & \lambda L(A-I)^T & \bar{\tau} L(A-I)^T \\ * & \Pi_i^{22} & \lambda (BV_i)^T & \bar{\tau}(BV_i)^T \\ * & * & -\lambda L_i & 0 \\ * & * & * & -\bar{\tau} M_i \end{pmatrix} < 0 \tag{7.23}$$

$$\Sigma_i = \begin{pmatrix} X_i^{11} & X_i^{12} & Y_i^1 \\ * & X_i^{22} & Y_i^2 \\ * & * & \frac{1}{\lambda} L_i M_i^{-1} L_i \end{pmatrix} \geq 0 \tag{7.24}$$

2. $\forall i, j \in \Omega$,

$$L_i \le \lambda L_j, M_i \le \lambda M_j, W_i \le \lambda W_j \tag{7.25}$$

where

$$\Pi_i^{11} = (\lambda - 1)L_i + W_i + 2\lambda(A - I)L_i + Y_i^1 + (Y_i^1)^T + iX_i^{11},$$

$$\Pi_i^{12} = \lambda B V_i - Y_i^1 + (Y_i^2)^T + iX_i^{12},$$

$$\Pi_i^{22} = -Y_i^2 - (Y_i^2)^T + iX_i^{22}.$$

Furthermore, the control law is defined in (7.5) *with* $K(i) = V_i L_i^{-1}$.

Proof The condition in (7.9) in Theorem 7.1 can be reformed as

$$\begin{pmatrix} \Phi_i^{11} & \Phi_i^{12} & \lambda(A-I)^T P_i & \bar{\tau}(A-I)^T R_i \\ * & \Phi_i^{22} & \lambda(BK(i))^T P_i & \bar{\tau}(BK(i))^T R_i \\ * & * & -\lambda P_i & 0 \\ * & * & * & -\bar{\tau} R_i \end{pmatrix} < 0 \tag{7.26}$$

Pre- and Post multiply (7.26) and (7.10) by $diag(P_i^{-1}, P_i^{-1}, P_i^{-1}, R_i^{-1})$ and $diag(P_i^{-1}, P_i^{-1}, P_i^{-1})$, respectively, and let $L_i = P_i^{-1}$, $M_i = R_i^{-1}$, $W_i = P_i^{-1} Q_i P_i^{-1}$, $X_i = diag(P_i, P_i) \cdot S_i \cdot diag(P_i, P_i)$, $Y_i = P_i^{-1} T_i P_i^{-1}$, $V_i = K(i) P_i^{-1}$. We then complete the proof.

Theorem 7.3 provides a way to design a stabilized controller for NCSs in (7.6). However, the condition in (7.24) in Theorem 7.3 is no longer LMI conditions due to the term $L_i M_i^{-1} L_i$. To deal with this difficulty, the cone complementarity technique is used in this chapter to derive a suboptimal solution for (7.24) [135], by transforming the problem to a minimization problem involving LMI conditions.

Corollary 7.1 *Given* $\lambda \ge 1$. *Define the following nonlinear minimization problem involving LMI conditions for* $i \in \Omega$,

$$\mathcal{P}_i : \begin{cases} \textit{Minimize } Tr(Z_i R_i + L_i P_i + M_i Q_i) \\ \textit{Subject to } (7.23), (7.25), L_i = L_i^T > 0, W_i = W_i^T > 0, \\ M_i = M_i^T > 0, X_i = \begin{pmatrix} X_i^{11} & X_i^{12} \\ (X_i^{12})^T & X_i^{22} \end{pmatrix} \ge 0, \\ \Sigma_i' \ge 0, \Theta_i^1 \ge 0, \Theta_i^2 \ge 0, \Theta_i^3 \ge 0, \Theta_i^4 \ge 0. \end{cases} \tag{7.27}$$

where $\Sigma_i^{'} = \begin{pmatrix} X_i^{11} & X_i^{12} & Y_i^1 \\ * & X_i^{22} & Y_i^2 \\ * & * & \frac{1}{\lambda}Z_i \end{pmatrix}$, $\Theta_i^1 = \begin{pmatrix} R_i & P_i \\ * & Q_i \end{pmatrix}$, $\Theta_i^2 = \begin{pmatrix} Z_i & I \\ * & R_i \end{pmatrix}$, $\Theta_i^3 = \begin{pmatrix} L_i & I \\ * & P_i \end{pmatrix}$, $\Theta_i^4 = \begin{pmatrix} M_i & I \\ * & Q_i \end{pmatrix}$.

If the solution of $\mathcal{P}_i = 3n, \forall i \in \Omega$, *the system in* (7.6) *is then stabilizable with the control law defined in Theorem 7.3.*

For the detailed algorithm based on Corollary 7.1, the reader is referred to [135, 138].

7.3 An Illustrative Example

Example 7.1 Consider an inverted pendulum system with delayed control input, first discussed in [133]. The discretized model for the system with the sampling period of 30 ms was given by

$$x(k+1) = \begin{pmatrix} 1.0078 & 0.0301 \\ 0.5202 & 1.0078 \end{pmatrix} x(k) + \begin{pmatrix} -0.0001 \\ -0.0053 \end{pmatrix} u(k),$$

and a state feedback gain was obtained in [133] as $K = [102.9100\ 80.7916]$, which is fixed for all network conditions.

In this example, let $\bar{\tau} = 12$ and thus the network-induced delay in the round trip is time-varying within the range [2 12]. In order to generate the delay sequence satisfying (7.4), a random delay sequence $\{\tau_k, k \geq 1\}$ is first produced within the range [2 12], which is then modified to obtain $\{\tau_k^*, k \geq 1\}$ according to (7.4). This is done by (1) let $\tau_1^* = \tau_1$; (2) for $k > 1$, if $\tau_{k+1} > \tau_k^* + 1$ then let $\tau_{k+1}^* = \tau_k^* + 1$; let $\tau_{k+1}^* = \tau_{k+1}$ otherwise. It is worth mentioning that this process of generating the round trip delay sequence $\{\tau_k^*, k \geq 1\}$ represents the reality in practical NCSs where only the latest information is used. A typical delay sequence of $\{\tau_k^*, k \geq 1\}$ is illustrated in Fig. 7.1, where it is seen that the growth rate of the round trip delay is upper bounded by the dashed lines with their slopes being 1.

Using Corollary 7.1, the following feedback gains are obtained with respect to different round trip delays,

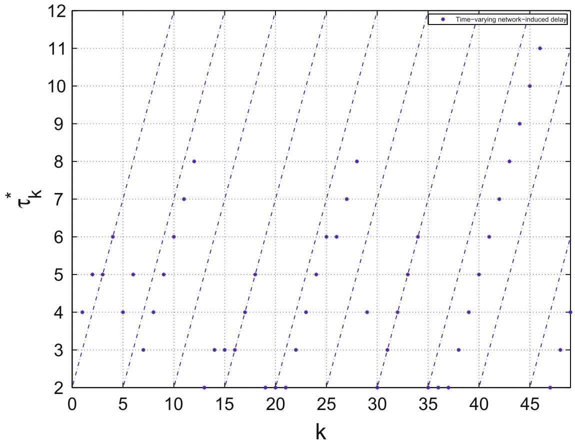


Fig. 7.1 The round trip delay τ_k^* which satisfies (7.4)

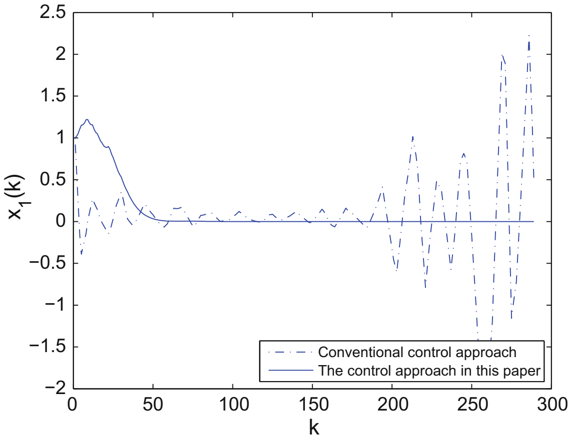


Fig. 7.2 The state responses using conventional control approach and the control approach in this chapter

$$\begin{pmatrix} K(2) \\ K(3) \\ K(4) \\ K(5) \\ K(6) \\ K(7) \\ K(8) \\ K(9) \\ K(10) \\ K(11) \\ K(12) \end{pmatrix} = \begin{pmatrix} 133.1456 & 32.2599 \\ 132.1299 & 31.9383 \\ 132.0470 & 31.9097 \\ 132.2548 & 31.9475 \\ 131.9716 & 31.8689 \\ 132.3325 & 31.9662 \\ 132.1224 & 31.8954 \\ 132.1417 & 31.9062 \\ 132.1821 & 31.9203 \\ 132.1446 & 31.9008 \\ 131.9822 & 31.8532 \end{pmatrix}.$$

The state responses with the control approaches in [133] and in this chapter are illustrated in Fig. 7.2, where it is shown that the system is unstable using the conventional approach in [133] whereas the control approach in this chapter can efficiently stabilize the system. This is due to two reasons. Firstly, the stabilized controller design method proposed in this chapter takes clear account of the delay constraint in (7.4). Secondly, the use of the time-dependent feedback gains in our model brings more freedom in designing the control law.

7.4 Summary

By recognizing the reality that only the latest information is used in practical NCSs, a new time delay system model for NCSs is proposed. This model takes account of both the specific characteristics of the network-induced delay in practical NCSs and the time-dependent feedback gain scheme. Stability and stabilization results are obtained based on this model in which less complex Lyapunov functional is used due to the new model. A numerical example illustrates the effectiveness of the proposed approach.

Chapter 8
Exploring the Different Delay Effects in Different Channels in Networked Control Systems

The sensor-to-controller and the controller-to-actuator delays in networked control systems are investigated for the first time with respect to their different effects on the system performance. The study starts with identifying the delay-independent and delay-dependent control laws in networked control systems, and confirms that only two delay-dependent control laws can cause different delay effects in different channels. The conditions under which the different delays in different channels can cause different effects, are then given for both delay-dependent control laws. The results are verified by numerical examples. Potentially these results can be regarded as important design principles in the practical implementation of networked control systems.

This chapter is organized as follows. The problem is formulated in Sect. 8.1. Existing control laws are then categorized in Sect. 8.2. The delay effects with respect to different categories of control laws are analyzed both qualitatively and quantitatively in Sect. 8.3. The obtained results are verified by numerical examples in Sects. 8.4 and 8.5 concludes the chapter.

8.1 Problem Formulation

One of the most distinct characteristics in NCSs is the network-induced delay, caused by the imperfect data transmission in NCSs. Most available works do not distinguish between the delay in either the sensor-to-controller or the controller-to-actuator channel, meaning that the majority of the existing models of NCSs simply assume those two delays affect the system performance in the same way. Although this assumption seems naturally true, further clarifications are necessary before regarding it as a general principle: Is it universally true that the delays in both channels are identical with respect to their effects on the system performance? This question is important

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_8

since, although the answer of "yes" could confirm the correctness of existing results, the possible answer of "no" will put all these existing results in an awkward position and open the gate for a more appropriate modeling approach to NCSs.

The first difficulty in answering the raised question is how the concerned system performance can be quantitatively represented in terms of different delays in different channels. This is addressed by defining an error of the control signals involving the different delays in different channels. Based on this definition, the concerned question can then be formulated appropriately.

Consider the typical system setting of NCSs illustrated in Fig. 2.1. Two delays exist in this system setting, i.e., the sensor-to-controller delay, α, and the controller-to-actuator delay, β, respectively. We assume that the delays are upper bounded, i.e.,

$$0 \leq \alpha \leq \bar{\tau}_{sc}, 0 \leq \beta \leq \bar{\tau}_{ca}$$

and consequently, $0 \leq \tau \leq \bar{\tau}$.

In order to focus mainly on the delay effects rather than the plant dynamics, the following linear nominal system model for the plant is adopted,

$$x(k+1) = Ax(k) + Bu(k) \tag{8.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$.

The following is an immediate observation from Fig. 2.1, which however is the foundation of our analysis on the different delay effects in NCSs: It is the different control laws that may cause different delay effects in NCSs. More specifically, the delay effects on the evolution of the system in (8.1) entirely rely on how the control signal $u(k)$ is obtained, as only $u(k)$ is directly affected by the delays (this will be more evident afterwards when we categorize existing control laws). This observation enables us to focus mainly on the analysis of different control laws in NCSs.

Given a control law, define the difference of the control signals between the one with sensor-to-controller delay α and controller-to-actuator delay β (denoted by $u_\tau(k : \alpha, \beta)$) and the one without any delay (denoted by $u_0(k)$), to be

$$e_\tau(k : \alpha, \beta) := |u_\tau(k : \alpha, \beta) - u_0(k)| \tag{8.2}$$

Since $u_0(k)$ is the control action achievable without any delay, $e_\tau(k : \alpha, \beta)$ can thus be interpreted as a measure of how different delays in different channels would affect the ability of the system to achieve this desirable control action $u_0(k)$. Based on $e_\tau(k : \alpha, \beta)$ we are able to give the index to evaluate the delay effects in different channels, as follows.

Definition 8.1 Given a control law (consequently the way of calculating the control signal $u(k)$). The control law is said to be "different-channel-delay-independent" (DCDI) if at any specific time k, for any fixed τ and any combinations of α and β satisfying $\tau = \alpha + \beta$,

$$e_\tau(k : \alpha, \beta) = \text{constant}$$

The control law is said to be "different-channel-delay-dependent" (DCDD) otherwise. Furthermore, "the degree of the DCDD dependence" is measured by $e_\tau(k : \alpha, \beta)$. For $\tau = \alpha_1 + \beta_1 = \alpha_2 + \beta_2$ and $\alpha_1 > \alpha_2$, if

$$e_\tau(k : \alpha_1, \beta_1) > e_\tau(k : \alpha_2, \beta_2)$$

the sensor-to-controller delay α is said to be affecting the system performance more severely, and vice versa.

Based on Definition 8.1, the general question raised in the Introduction section can then be stated as: 1. Are all the control laws in NCSs DCDI? 2. If there exists a control law to be DCDD, then what is the degree of its DCDD dependence?

8.2 Categorizing the Control Laws

It is realized that the system performance defined in Definition 8.1 is entirely dependent on the choice of the control laws. The existing control laws are therefore categorized as the necessary preparation for further analysis.

8.2.1 Two General Categories of the Control Laws

For simplicity of analysis, we concentrate merely on static state feedback for the system in (8.1). Two categories of control laws are observed, referred to as the "delay-independent" and "delay-dependent" control laws, respectively.

- Delay-independent control laws. This category of control laws can be seen in most conventional control methods, the general form of which can be written as follows,

$$u(k) = Kx(k - \tau) \tag{8.3a}$$

  where K is the constant feedback gain and plenty of methods have been proposed to design it [66, 139, 140]. Although the control signal in (8.3a) is still dependent on the round trip delay τ, the controller (feedback gain) is designed "independently" from the delay.
- Delay-dependent control laws. A general form of the control laws belonging to this category can be written as follows,

$$u(k) = K_k x(k - \tau) \tag{8.3b}$$

where the feedback gain K_k is essentially time-varying and delay-dependent. Note that a specific delay-dependent control law may not be expressed explicitly in the form of (8.3b), i.e., the delay-dependent control law is defined in the "equivalent" sense: whatever the specific form of a control law is, it is delay-dependent if and only if it is not delay-independent. See for example the control law in (8.4a) which is defined later.

8.2.2 The Delay-Dependent Control Laws

The delay-dependent law given in (8.3b) is only of its general form. It can be implemented in practice via several different control strategies, and one of the most important control strategies is the packet-based control approach.

In the early development of PBNCSs [107], the controller is designed using a model based control method. The idea is to first "estimate", or "predict" the current system state from the delayed sensing data and then use a constant feedback gain. We refer to this design method as the "prediction-based" approach and the control law can be written as

$$u(k) = K\hat{x}(k|k-\tau) \tag{8.4a}$$

where $\hat{x}(k|k-\tau)$ is the predicted state at time k based on the state at time $k-\tau$.

The more general packet-based control law is given as follows,

$$u(k) = K_\tau x(k-\tau) \tag{8.4b}$$

where the feedback gain K_τ is dependent on the round trip delay τ. In the absence of time synchronisation, the control law may be defined by

$$u(k) = K_{\alpha,\beta} x(k-\tau) \tag{8.4c}$$

where the feedback gain $K_{\alpha,\beta}$ is dependent on both the sensor-to-controller delay, α and the controller-to-actuator delay, β.

8.3 When and How the Delay Effects in Different Channels Are Different

Based on the definition of the system performance index in Sect. 8.1 and the categorization of existing control laws in NCSs in Sect. 8.2, we are now able to analyze the delay effects in different channels. A qualitative analysis is firstly conducted to clarify which categories of control laws are DCDI (DCDD) and then a quantitative analysis is conducted for the DCDD control laws.

8.3.1 When the Delay Effects Are Different: A Qualitative Analysis

A qualitative analysis of the delay-independent control law in (8.3a) and the delay-dependent control laws in (8.4a) (8.4b) (8.4c) leads to the following Proposition.

Proposition 8.1 *1. The delay-independent control law in* (8.3a) *and the delay-dependent control law in* (8.4b) *are DCDI;*
2. The delay-dependent control laws in (8.4a) *and* (8.4c) *are typically DCDD;*

Proof Notice that for any given k and τ, $u_\tau(k : \alpha, \beta)$ remains to be constant for both control laws in (8.3a) and (8.4b). The first part of the Proposition is thus correct by (8.2) and Definition 8.1.

As for the control law in (8.4c), $u_\tau(k : \alpha, \beta)$ is varying with $K_{\alpha,\beta}$ and is thus typically different for different combinations of α and β, since otherwise, it degenerates to the control law in (8.4b).

The predicted system state in the control law in (8.4a) is usually designed based on a model of the plant. This design procedure presents two factors that would affect the predicted system state, that is, the model inaccuracy and the error occurred in the model prediction. The sensor-to-controller and the controller-to-actuator delays are related to these two factors in different ways (which will be more evident in the quantitative analysis in Sect. 8.3.2.1), meaning that these two delays typically present different delay effects for the system.

Remark 8.1 As for the control law in (8.4a), it makes a difference in terms of the delay effects whenever a model-based controller is designed for the system. This means that besides the packet-based control approach, other model-based methods could also suffer from different delay effects for different sensor-to-control and controller-to-actuator delays, such as the approaches proposed in [141, 142]. On the other hand, the idea of using delay-dependent feedback gains has also been seen in other models used for NCSs, see, e.g., [56], despite the missing of the practical design support. Clearly the results obtained here and in what follows are also applicable to these models. This observation implies that the formulated problem and the obtained results are widely applicable to a large number of NCSs.

8.3.2 How the Delay Effects with (8.4a) and (8.4c) are Different: A Quantitative Analysis

The second part of Proposition 8.1, i.e., the different delay effects caused by the control laws in (8.4a) and (8.4c), are addressed quantitatively in this section. These obtained results provide important design principles for the practical implementation of NCSs.

8.3.2.1 The Prediction-Based Approach in (8.4a)

The design of the control law in (8.4a) can be various due to the different predictive methods used to obtain $\hat{x}(k|k-\tau)$. Examples of this variance can be seen in [107] for a model-based approach and in [37] for a receding horizon based approach. In what follows a quantitative analysis is done for the case in [107], whereas other cases can be analyzed similarly.

The fundamental idea of predicting the state $\hat{x}(k|k-\tau)$ in [107] is to use an estimated plant model at the controller side, which can give the predicted states based on delayed state information. The model used can be written as

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k) \tag{8.5}$$

where $\hat{A}$ and $\hat{B}$ are not equivalent to A and B in (8.1) in general due to the modeling error. Furthermore, the control signals $\hat{u}(k)$ may not be the same as the real ones, $u(k)$, since the latter is usually not fully accessible to the controller.

Notice that k in (8.4a) is the time at the actuator side. The time when the FCS is calculated at the controller side is thus $k-\beta$ and the FCS is calculated based on the delayed state information at time $k-\tau$. The prediction-based approach estimates the state $\hat{x}(k|k-\tau)$ based on the available delayed state $x(k-\tau)$, using the following two steps.

1. Estimate from $\hat{x}(k-\tau+1|k-\tau)$ to $\hat{x}(k-\beta|k-\tau)$. In this step it is assumed that the real control signals applied to the plant from $u(k-\tau)$ to $u(k-\beta-1)$ are available to the controller, that is, $\hat{u}(k-\tau+i) = u(k-\tau+i), i = 0, 1, \ldots, \beta+1$. It is realized later that this assumption is difficult to be implemented in practice and a better approach is proposed to deal with this difficulty [143]. However in this chapter we keep this assumption unchanged for simplicity of analysis. Based on this assumption and the predictive model in (8.5), the dynamics of the predictive model can be written as

$$\begin{aligned}&\hat{x}(k-\tau+i|k-\tau)\\&= \hat{A}\hat{x}(k-\tau+i-1|k-\tau) + \hat{B}u(k-\tau+i-1),\ i = 1, \ldots, \alpha\end{aligned} \tag{8.6}$$

   where $\hat{x}(k-\tau|k-\tau) = x(k-\tau)$. This yields

$$\hat{x}(k-\beta|k-\tau) = \hat{A}^{\alpha}x(k-\tau) + \sum_{j=1}^{\alpha} \hat{A}^{\alpha-j}\hat{B}u(k-\tau+j-1) \tag{8.7}$$

2. Estimate from $\hat{x}(k-\beta+1|k-\tau)$ to $\hat{x}(k|k-\tau)$. In this step the control signal is assumed to be given by $\hat{u}(k-\beta+i) = u(k-\beta+i|k-\tau) = K\hat{x}(k-\beta+i|k-\tau)$, as the real ones are clearly not available. Based on this assumption, the predictive model in (8.5) turns to be

$$\hat{x}(k-\beta+i|k-\tau) = (\hat{A}+\hat{B}K)\hat{x}(k-\beta+i-1|k-\tau),\ i=1,\ldots,\beta \tag{8.8}$$

which gives

$$\hat{x}(k|k-\tau) = (\hat{A}+\hat{B}K)^{\beta}\hat{x}(k-\beta|k-\tau) \tag{8.9}$$

As state feedback with a constant feedback gain is used in (8.4a), $e_\tau(k:\alpha,\beta)$ in (8.2) can thus be evaluated equivalently by the difference between the estimated state, $\hat{x}(k|k-\tau)$ and the real one, $x(k|k-\tau) = x(k)$, i.e.,

$$e(k|k-\tau) := x(k|k-\tau) - \hat{x}(k|k-\tau)$$

By (8.1) $x(k|k-\tau)$ is given by

$$x(k|k-\tau) = A^{\tau}x(k-\tau) + \sum_{j=1}^{\tau} A^{\tau-j}Bu(k-\tau+j-1) \tag{8.10}$$

which is based on the state at time $k-\tau$.

From (8.7), (8.9) and (8.10) $e(k|k-\tau)$ can be explicitly expressed, which in general is a function of the sensor-to-controller delay, α, (or the controller-to-actuator delay, β) given the fixed round trip delay, τ,

$$e(k|k-\tau) = \Gamma_{\tau,K}(\alpha) \tag{8.11}$$

Although it is possible to investigate the explicit expression of $e(k|k-\tau)$ in (8.11) directly, it is too complicated to derive any valuable results. As the main purpose of the chapter is to study the effects in the presence of different delays in different channels, it is thus possible to study the effects indirectly from two different dynamics of $e(k|k-\tau)$, based on (8.6) and (8.8). On the basis of this observation, the following result is obtained.

Proposition 8.2 *With the use of the prediction-based control law in [107], the sensor-to-controller delay, α, affects the system performance less than the controller-to-actuator delay, β, provided the predictive model in (8.5) is sufficiently precise.*

Proof In order to demonstrate the above result, the error dynamics $e(k|k-\tau)$ is analysed based on the aforementioned two steps in the prediction-based approach. From $k-\tau$ to $k-\beta$, the error dynamics is obtained for $i=1,\ldots,\alpha$, as follows, based on (8.6) and (8.10),

$$\begin{aligned} e_\alpha(i) :=& e(k-\tau+i|k-\tau) \\ =& (A-\hat{A})x(k-\tau+i-1|k-\tau) + \hat{A}e(k-\tau+i-1|k-\tau) \\ &+ (B-\hat{B})u(k-\tau+i-1) \end{aligned} \tag{8.12}$$

with $e(k-\tau|k-\tau)=0$.

On the other hand, from $k-\beta+1$ to k, the error dynamics is obtained for $j=1,\ldots,\beta$, based on (8.8) and (8.10), as follows,

$$\begin{aligned} e_\beta(j) :=& e(k-\beta+j|k-\tau) \\ =& (A-\hat{A}-\hat{B}K)x(k-\beta+j-1|k-\tau)+Bu(k-\beta+j-1) \\ &+(\hat{A}+\hat{B}K)e(k-\beta+j-1|k-\tau) \end{aligned} \tag{8.13}$$

It is noticed that the error $e_\alpha(\cdot)$ is purely dependent on the sensor-to-controller delay, α, and is accumulated with the increase of α. On the other hand, although $e_\beta(\cdot)$ is mainly affected by the controller-to-actuator delay, it is also affected by the sensor-to-controller delay, since its initial state, $e(k-\beta|k-\tau)$, is obtained in (8.12).

Now suppose we have an exact model of the plant, i.e., $A=\hat{A}$, $B=\hat{B}$. It immediately follows that $e_\alpha(i)\equiv 0, i=1,\ldots,\alpha$, and in particular the initial state for (8.13), $e(k-\beta|k-\tau)=e_\alpha(\alpha)=0$. Therefore, in this case the sensor-to-controller delay does not affect the system performance at all. On the other hand, it is readily seen that $e_\beta(i)\neq 0$ in general and will accumulate with the increase of β. Based on this observation, it is therefore fair to claim the statement made in this proposition.

Remark 8.2 Proposition 8.2 implies that, under certain conditions, it can result in a better system performance to place the controller as close to the actuator as possible, if the system allows us to do so. In this sense Proposition 8.2 has its practical guidance value. However, Proposition 8.2 is based on the nominal system and it could be wrong in the presence of large model inaccuracy, measurement error, or any other type of uncertainties in the system. Indeed, as stated in the proof, the sensor-to-controller delay affects both $e_\alpha(\cdot)$ and $e_\beta(\cdot)$ while the controller-to-actuator delay affects only $e_\beta(\cdot)$. Therefore, if the system setting allows the sensor-to-controller delay to take effect, it is very likely that this delay could affect the system performance more severely than that of the controller-to-actuator delay. This implies that Proposition 8.2 has its rigid conditions of applicability.

8.3.2.2 The Delay-Dependent Gain Based Approach in (8.4c)

Unlike the prediction-based approach in (8.4a) where the prediction of the current system state plays an essential role, the delay effects in the delay-dependent gain based approach in (8.4c) are purely dependent on the time-varying feedback gains. In order to specify $e_\tau(k:\alpha,\beta)$ in (8.2), we consider, for given round trip delay, τ, the unit error with respect to the delay in the sensor-to-controller channel, α,

$$\Delta e_\alpha := ||K_{\alpha+1,\beta-1}-K_{\alpha,\beta}||||x(k-\tau)|| = \Delta K_\alpha ||x(k-\tau)|| \tag{8.14}$$

and the unit error with respect to the delay in the controller-to-actuator channel, β,

$$\Delta e_\beta := ||K_{\alpha-1,\beta+1} - K_{\alpha,\beta}||||x(k-\tau)|| = \Delta K_\beta ||x(k-\tau)|| \tag{8.15}$$

where $||\cdot||$ is the norm of $\cdot$, $\Delta K_\alpha = ||K_{\alpha+1,\beta-1} - K_{\alpha,\beta}||$ and $\Delta K_\beta = ||K_{\alpha-1,\beta+1} - K_{\alpha,\beta}||$.

Recalling (8.4c), it is noticed that $K_{\alpha,\beta}x(k-\tau) = u(k)$ is the control signal actually applied to the plant at time k. Δe_α (Δe_β) can thus be interpreted as the difference between $u(k)$ and the control signal produced by increasing (decreasing) a unit delay in the sensor-to-controller channel and meanwhile decreasing (increasing) a unit delay in the controller-to-actuator channel. Therefore, to a certain extent Δe_α (Δe_β) is able to quantitatively measure the delay effects in the sensor-to-controller (controller-to-actuator) channel: The larger Δe_α (Δe_β) is, the more the delay in the sensor-to-controller (controller-to-actuator) channel affects the system performance. This fact is stated in the following Proposition.

Proposition 8.3 *In the delay-dependent gain based approach in* (8.4c)*, the effects of the delays in the sensor-to-controller channel and the controller-to-actuator channel are proportional to* ΔK_α *and* ΔK_β*, respectively.*

Remark 8.3 Notice that for the control law in (8.4b), for any given round trip delay the feedback gain remains fixed. This implies that for any given round trip delay, τ,

$$K_\tau = K_{\alpha+1,\beta-1} = K_{\alpha-1,\beta+1}, \forall \alpha + \beta = \tau$$

Therefore, in a certain sense the delay effects of the control law in (8.4b) can be deduced from Proposition 8.3.

Remark 8.4 It can be often seen in practice that either $\Delta K_\alpha > \Delta K_\beta$ or $\Delta K_\alpha < \Delta K_\beta$ is met by almost all round trip delays with only a few exceptions. In this case we should be confident to conclude that the sensor-to-controller delay or the controller-to-actuator delay plays more essential role than its counterpart although this conclusion can not be obtained directly from Proposition 8.3. In this sense the conditions in Proposition 8.3 can be too rigid to be actually applied in practice. To deal with this issue, we define the following global gain error, K_α and K_β, based on the partial gain error, ΔK_α and ΔK_β, respectively,

$$K_\alpha := \sum_\tau \sum_{\alpha+\beta=\tau} \Delta K_\alpha \tag{8.16}$$

$$K_\beta := \sum_\tau \sum_{\alpha+\beta=\tau} \Delta K_\beta \tag{8.17}$$

It is seen that K_α and K_β are the sum of ΔK_α and ΔK_β over all possible delays. Therefore, the former can be an effective measure for the delay effects in a global sense. Proposition 8.3 can also be modified accordingly to represent this global measure.

Table 8.1 The different delay effects in NCSs

Controller category	Which delay more affects the performance?	Implementation considerations
(8.3a) and (8.4b)	No difference[a]	N.A.
(8.4a)	The controller-to-actuator delay[b]	Minimize the controller-to-actuator delay[c]
(8.4c)	Dependent on different controller gains[d]	Adjust the control structure accordingly[e]

[a]Proposition 8.1; [b]Proposition 8.2; [c]Remark 8.2; [d]Proposition 8.3 and Remark 8.4; [e]Remark 8.5

Remark 8.5 Proposition 8.3 has its guidance value in the practical implementation of NCSs. After the feedback gains in (8.4c) have been designed, ΔK_α and ΔK_β in (8.14) and (8.15), and consequently the different delay effects, can then be determined by Proposition 8.3. The practical implementation can then be adjusted accordingly in favor of the system performance.

8.3.3 A Brief Summary and Discussion

We are now able to summarize the points scattered all over the chapter on the different delay effects in NCSs in Table 8.1.

Table 8.1 tells us that if the controller in a specific NCS is designed using a control law belonging to the controller category in (8.3a) or (8.4b), the sensor-to-controller and the controller-to-actuator delays are identical in terms of their effects on the system performance; However, if otherwise the control law belongs to the controller category in (8.4a) or (8.4c), the sensor-to-controller and the controller-to-actuator delays can possibly affect the system performance in different ways. As regards the practical implementing, we may therefore try to decrease as much as possible the delay which deteriorates the system performance more if the system allows us to do so. This can be served as an important design principle in the implementation of any NCSs.

8.4 Numerical Examples

Two numerical examples are considered to verify the conclusions made in the last section, regarding the delay effects of the prediction-based approach (Example 8.1) and the delay-dependent gain based approach (Example 8.2), respectively. All the simulations that follows are done using MATLAB.

Example 8.1 Consider the system in (8.1) with the following system matrices borrowed from [107],

$$A = \begin{pmatrix} 1.010 & 0.271 & -0.488 \\ 0.482 & 0.100 & 0.240 \\ 0.002 & 0.3681 & 0.7070 \end{pmatrix}, B = \begin{pmatrix} 5 & 5 \\ 3 & -2 \\ 5 & 4 \end{pmatrix}$$

As in [107], the initial state is set as $x_0 = [0.1\ 0.1\ 0.1]^T$ and the constant feedback gain is given by

$$K = \begin{pmatrix} 0.5858 & -0.1347 & -0.4543 \\ -0.5550 & 0.0461 & 0.4721 \end{pmatrix}$$

Different from the system setting in [107], it is assumed that the system states of the above system can be obtained exactly and therefore the measurement system and the observer are not necessary. The control signal is assumed to be zero before the arrival of the first FCS. In addition, in order to focus on the delay effects in different channels, the delays are all set to be time-invariant.

The simulations of the above system prove the statement made in Proposition 8.2 and Remark 8.2. Under the same round trip delay, $\bar{\tau} = 3$, Fig. 8.1 shows that the system is stable with $\tau_{ca} = 1$ while unstable with $\tau_{ca} = 2$. This proves the result in Proposition 8.2, that is, the smaller the controller-to-actuator delay is, the better the system performance will be. Further examples can be seen in Figs. 8.2 and 8.3. With $\tau_{ca} = 1$ and $\tau_{sc} = 1$ respectively, the system is stable even with $\tau_{sc} = 12$ (Fig. 8.2) while only stable for $\tau_{ca} < 2$ (Fig. 8.3). This clearly shows that the sensor-to-controller delay has a less negative effect on the system performance. In order to simulate the delay effects in the presence of the modeling error, a particular case is shown in Fig. 8.4, where the inaccurate system matrices are defined as $\hat{A} = (1 + \epsilon)A$ and $\hat{B} = (1 - \epsilon)B$ with $\epsilon = 0.16$. For this particular case it shows that the sensor-to-actuator delay could affect the system performance more severely. This proves the statement made in Remark 8.2. However, it is worth pointing out that with inaccurate models, the sensor-to-actuator delay could still be possible to affect the system performance more lightly. This implies that with the modeling error in presence, the delay effects in different channels are complicated and no general results exist.

Example 8.2 Consider the same system as in Example 8.1 but with the control law in (8.4c). In order to consider the different delay effects in this case we redefine the upper bounds of the delay as $\bar{\tau}_{sc} = \bar{\tau}_{ca} = 2$ and thus $\bar{\tau} = 4$. The delay-dependent feedback gains are then designed based on a receding horizon approach, as proposed in [37],

$$\begin{pmatrix} K_{0,0} \\ K_{0,1} \\ K_{0,2} \end{pmatrix} = \begin{pmatrix} -0.0312 & -0.0822 & -0.1719 \\ -0.1446 & 0.0568 & 0.2801 \\ 0.0583 & -0.0172 & -0.0976 \\ -0.1591 & 0.0427 & 0.2559 \\ 0.0474 & -0.0128 & -0.0765 \\ -0.1253 & 0.0338 & 0.2020 \end{pmatrix}$$

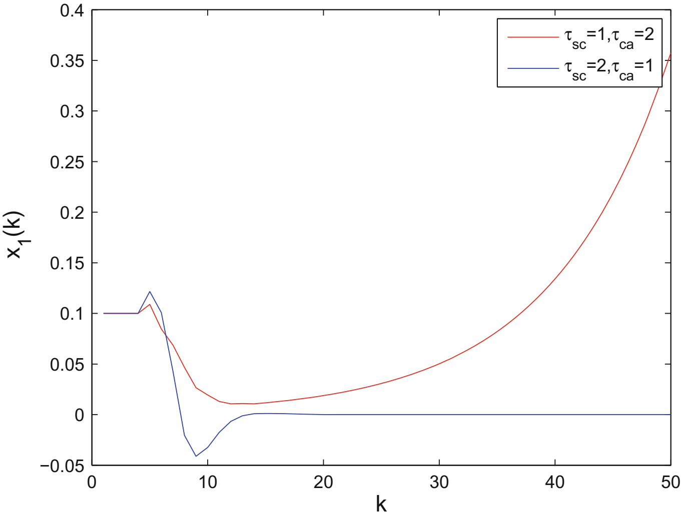


Fig. 8.1 Example 8.1. State responses with different delays in different channels

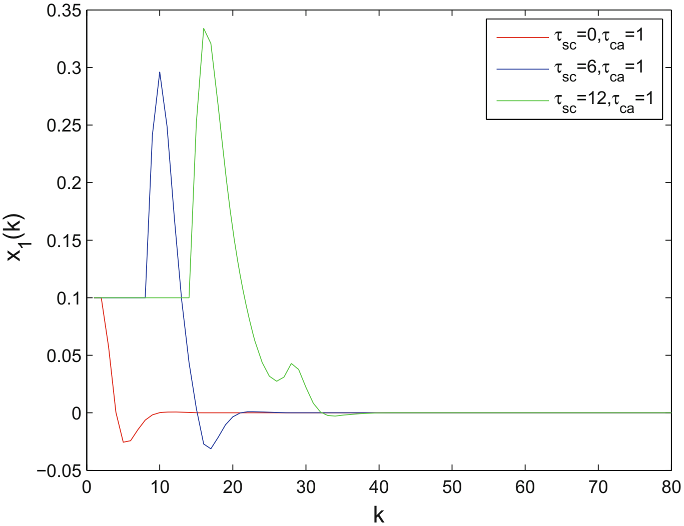


Fig. 8.2 Example 8.1. State responses with the same controller-to-actuator delay

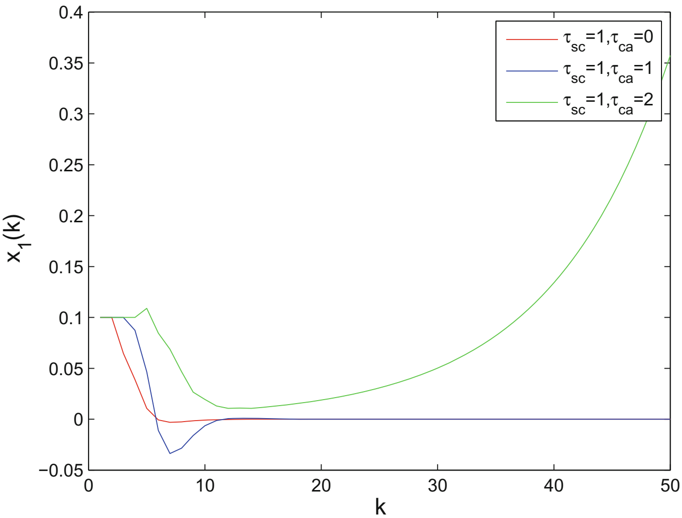


Fig. 8.3 Example 8.1. State responses with the same sensor-to-controller delay

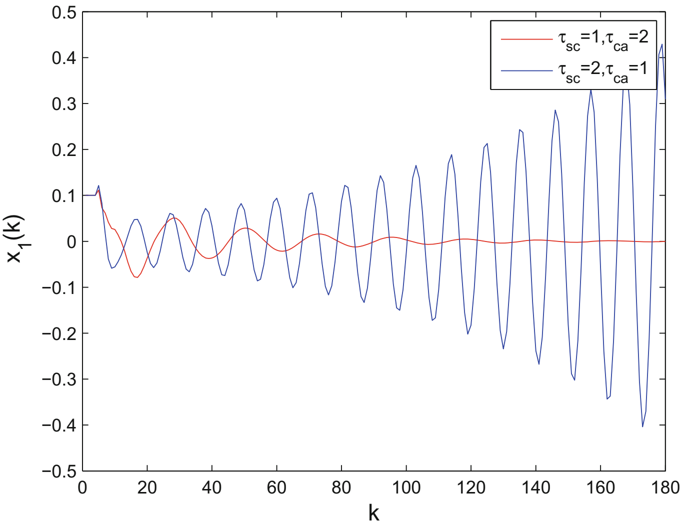


Fig. 8.4 Example 8.1. State responses in the presence of modeling error

$$\begin{pmatrix} K_{1,0} \\ K_{1,1} \\ K_{1,2} \end{pmatrix} = \begin{pmatrix} -0.1333 & 0.0018 & 0.1319 \\ -0.0772 & 0.0191 & 0.1235 \\ 0.0085 & -0.0027 & -0.0147 \\ -0.0290 & 0.0073 & 0.0455 \\ 0.0087 & -0.0022 & -0.0137 \\ -0.0230 & 0.0059 & 0.0363 \end{pmatrix}$$

$$\begin{pmatrix} K_{2,0} \\ K_{2,1} \\ K_{2,2} \end{pmatrix} = \begin{pmatrix} -0.1629 & 0.0137 & 0.1890 \\ -0.0256 & 0.0053 & 0.0405 \\ 0.0010 & -0.0006 & -0.0024 \\ -0.0104 & 0.0022 & 0.0152 \\ 0.0030 & -0.0006 & -0.0045 \\ -0.0081 & 0.0017 & 0.0120 \end{pmatrix}$$

As for the above delay-dependent gains, there is no general conclusion whether $\Delta K_\alpha > \Delta K_\beta$ or $\Delta K_\alpha < \Delta K_\beta$. The global gain error is then calculated by (8.16), which turns to be $K_\alpha = 0.0763$ and $K_\beta = 0.1688$. This indicates that in general the controller-to-actuator delay, β, affects the system performance more than the sensor-to-controller delay, α. This conclusion is verified by the state responses shown in Fig. 8.5, where the increase of the controller-to-actuator delay rapidly destabilizes the system, showing the more important role played by the controller-to-actuator delay.

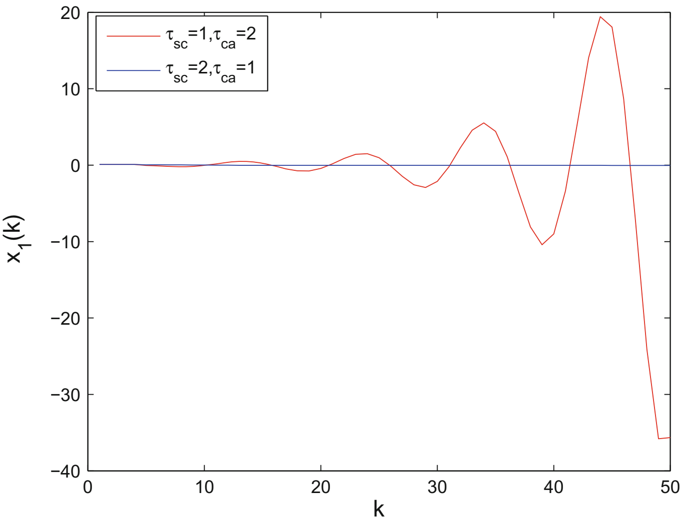


Fig. 8.5 Example 8.2. State responses with different delays in different channels

8.5 Summary

Delays play an important role in networked control systems. It is revealed for the first time that delays in different channels can possibly affect the system performance in very different ways. By categorizing existing control laws, qualitatively and quantitatively analyzing their roles in determining the delay effects in different channels, conditions and criteria are given to determine which delay can be more important under various conditions. These results can serve as important design principles in the practical implementation of networked control systems.

Part III
Extension

The packet-based control approach is based on the packet-based transmission in networked control systems. This part considers more communication characteristics in networked control systems and proposes corresponding solutions, including data packet disorder in Chap. 9, the reduction of the communication resource usage at the sensor side (Chap. 10) and at the controller side (Chap. 11), and the scheduling of the communication resource usage in Chap. 12, and so forth. These works are still within the packet-based control framework, but have greatly expanded the regime of the framework.

Chapter 9
Active Compensation for Data Packet Disorder in Networked Control Systems

Data packet disorder often occurs in Networked Control Systems which however has not been taken into account in most literature to date. In this chapter the cause and effect of data packet disorder are analyzed and an active compensation scheme is proposed to compensate for it. The proposed scheme is flexible to admit all the existing control approaches to be used, and also derives a novel closed-loop system model of NCSs which enables more reasonable and effective theoretical analysis of NCSs. The effectiveness of the proposed active compensation scheme is illustrated by a numerical example.

This chapter is organized as follows. In Sect. 9.1, we first analyze how data packet disorder occurs and then review the related work in this area. The active compensation scheme for data packet disorder is then presented in Sect. 9.2, which derives a novel system model for NCSs. A numerical example is also considered in Sect. 9.4, which illustrates the effectiveness of the proposed scheme. Section 9.5 concludes the chapter.

9.1 Data Packet Disorder and Related Work

In NCSs, the plant is controlled over some communication network by the controller. Due to the communication network inserted into the control system, network-induced delay is inevitable in NCSs, denoted by $\tau_{sc}(t)$ and $\tau_{ca}(t)$ for the delays in the sensor-to-controller channel and controller-to-actuator channel respectively. The plant dynamics, in general, is described by the following differential equation,

$$\dot{x}(t) = f(x(t), u(t)) \tag{9.1}$$

where u is the control input to the plant. In this chapter, the sensor is assumed to be time-driven whereas the controller and the actuator are event-driven, as assumed in

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_9

[88]. "Time-driven" for the sensor in this chapter implies an independent, constant time interval h between which the plant dynamics is sampled and the "event-driven" controller and actuator are trigged by newly arrived data (sampled data or control signals) but not at specific time instants.

9.1.1 Data Packet Disorder

Let us consider the typical data transmission process in NCSs illustrated in Fig. 9.1 and define "round trip delay" for a sampled data packet to be the time interval from sampling the system states to the control signal based on this sampled data being applied to the plant. Generally round trip delay consists mainly of two transmission delays: sensor-to-controller delay and controller-to-actuator delay, whereas in this chapter all the other potential delays such as computation delay of the controller are considered as part of the round trip delay, that is, round trip delay is the total delay in the system.

The time-driven sensor sends its sampled data every h seconds, as illustrated in Fig. 9.1 at time instants t^s_{k-1} and t^s_k respectively. However, due to the arbitrary network-induced delay, the sampled data packet sent at time instant t^s_{k-1} does not necessarily arrive at the actuator earlier than its subsequent data packet sent at time instant t^s_k. This occurs when, for example in Fig. 9.1, $\tau_{k-1} - \tau_{k11} > h$. Based on this analysis, Proposition 9.1 readily follows.

Proposition 9.1 *Given a constant sampling period h and arbitrarily variable network-induced delays, data packet disorder occurs if and only if*

$$\Delta\tau_m = \tau_{max} - \tau_{min} > h \tag{9.2}$$

where τ_{max} and τ_{min} are the upper and lower bounds of the round trip delay.

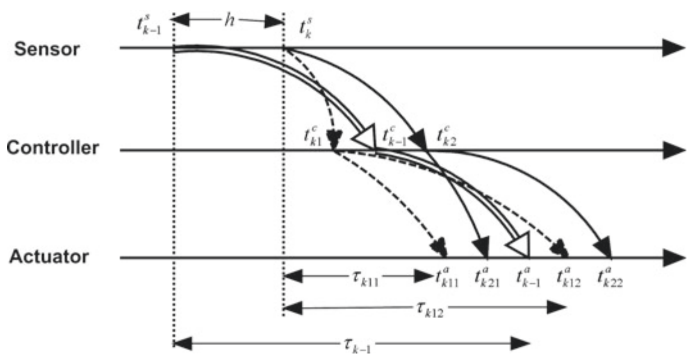


Fig. 9.1 Illustrating how data packet disorder occurs

Proof On the one hand, data packet disorder occurs when the round trip delays for the sampled data packet sent at time instants t_{k-1}^s and t_k^s in Fig. 9.1 are τ_{max} and τ_{min} respectively provided (9.2) holds; On the other hand, if $\Delta\tau_m \leq h$ then data packet sent at time instant t_k^s will never arrive at the actuator earlier than the one sent at time instant t_{k-1}^s, that is, no data packet disorder occurs.

Remark 9.1 If data packet disorder occurs in an NCS then we can conclude that (9.2) holds for this NCS; On the contrary, if (9.2) holds for an NCS, Proposition 9.1 implies that data packet disorder will inevitably occur for a certain data packet after the NCS runs for a sufficient long time but it does not mean every data packet will experience disorder in this case. From this point of view, it is readily seen that Proposition 9.1 is still valid in the presence of data packet dropout.

9.1.2 Related Work

If no special treatment is taken, the existence of data packet disorder will produce a situation where, older information is used instead of latest information available. Take Fig. 9.1 as an example where we assume the sampled data packet sent at time instant t_k^s arrives at the actuator at time instant t_{k11}^a (data packet disorder occurs in this case). According to conventional approaches without compensating for data packet disorder, the control signal based on sampled data at time instant t_k^s will be used between t_{k11}^a and t_{k-1}^a whereas after t_{k-1}^a the control signal based on older sampled data at time instant t_{k-1}^s will be used. This is obviously unreasonable and seriously degrades the system performance.

A possible solution for data packet disorder is the packet-based control approach. Though effective, the requirement of sending a sequence of forward control signals simultaneously in a single data packet may not be always available in NCSs, which thus restricts the application of this approach for certain conditions.

From the perspective of conventional TDS theory, there are still no effective approaches to deal with this issue to date. For example, in [88, 144] the following closed-loop system model for NCSs with a linear plant model was obtained,

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{9.3a}$$

$$u(t) = Kx(i_k h), t \in [i_k h + \tau_k, i_{k+1}h + \tau_{k+1}),\ k \geq 1 \tag{9.3b}$$

where $i_k h$ and $i_{k+1}h$ were the sampling time instants and the relationship $i_{k+1} > i_k$ was not required, that is, no compensation scheme for data packet disorder was considered. In [145], a similar model was considered and even the authors noticed that data packet disorder may occur, they unfortunately assumed $i_{k+1} > i_k$ artificially without providing any supportive design method. Similar situations can also be found in other works, for example, in [139, 146–148].

9.2 Actively Compensating for Data Packet Disorder in NCSs

In this section, an active compensation scheme for data packet disorder in NCSs is presented. As mentioned earlier, the derived system model considers the communication constraints in NCSs including network-induced delay, data packet dropout and data packet disorder simultaneously, compared with previously reported results where data packet disorder is excluded.

The schematic structure of the active compensation scheme for NCSs is illustrated in Fig. 9.2, which is seen to be distinct from conventional control systems in two aspects: the Time Stamp Generator (TSG) at the sensor side and the Control Action Selector (CAS) at the controller side.

As implied by the name, TSG is used to label each sampled data packet with a "time stamp" which contains the information of the corresponding sampling time instant of the sampled data packet. This time stamp remains in the control data packet after the control signal is calculated based on the sampled data, thus enabling the sampling time instant based on which each control data packet is calculated to be known by the CAS module in Fig. 9.2. This information is then used by CAS to actively compensate for data packet disorder.

CAS consists of a register and a logic comparator. The register is used to store only a single step of the control signal with the corresponding time stamp as mentioned above. When a control data packet arrives, the logic comparator compares the time stamps of both the newly arrived control data packet and the one already in the register of CAS, and only the latest control data packet is stored after the comparison process and then applied to the plant. In this way, the introduction of CAS can effectively deal with data packet disorder in NCSs with the help of TSG. For example, suppose in Fig. 9.1 that the control data packet based on sampled data at time instant t_k^s arrives at the actuator at time instant t_{k11}^a. Then at time instant t_{k-1}^a when the control data packet based on sampled data at time instant t_{k-1}^s arrives, CAS knows that the newly arrived control data packet is calculated based on older sampled data and thus

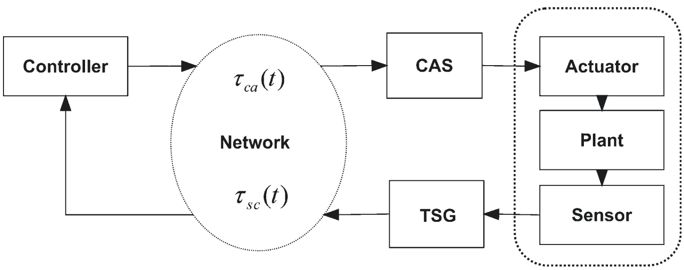


Fig. 9.2 The schematic structure of the active compensation scheme for NCSs

will discard this control data packet and the register of CAS remains unchanged. Thus, CAS avoids the existence of the aforementioned situation where older control action instead of the latest available is applied to the plant, that is, CAS successfully eliminates the effect of data packet disorder in NCSs.

Suppose the kth control data packet that arrives at the actuator successfully is based on the i_kth sampled data packet, and its corresponding round trip delay is τ_k. Then the sampled data packet based on which the kth "effective" control data packet is applied to the plant after the comparison process can be determined by the following comparison rule,

$$P_{i_k^*} = \begin{cases} P_{i_{k-1}^*}, & \text{if } i_{k-1}^* > i_{k'}; \\ P_{i_{k'}}, & \text{otherwise.} \end{cases} \tag{9.4}$$

where the control data packet $P_{i_{k-1}^*}$ which is based on sampled data at time instant $i_{k-1}^* h$ is already in the register of CAS and $P_{i_{k'}}$ which is based on $i_{k'}$ just arrives at CAS.

Based on the above analysis, the algorithm of the active compensation scheme can now be summarized as follows.

Algorithm 9.1 Active compensation for data packet disorder

S1. The sensor samples the system dynamics.
S2. TSG labels the sampled data with the time stamp and sends the sampled data packet over the network to the controller.
S3. The controller receives the sampled data packet and calculates the corresponding control signal which is then sent to the actuator with the time stamp.
S4. CAS compares the time stamps of the newly arrived control data packet and the one already in the register of CAS by (9.4). The latest control data packet is then sent to the actuator and also stored in the register.
S5. The control action from CAS is applied to the plant.

Using the proposed active compensation scheme, data packet disorder in NCSs can now be effectively dealt with. From the schematic structure in Fig. 9.2 and Algorithm 9.1, it is readily seen that this scheme additionally inserts two modules, namely TSG and CAS, into the control system but does not modify the original control components in the system, that is, the sensor, the controller and the actuator. This design approach therefore enables all the existing conventional control approaches to be applied to this control structure without any modification whilst data packet disorder can be effectively dealt with. This flexibility enables the proposed scheme to be readily deployed in practice.

Remark 9.2 It is noticed that data packet disorder may occur in both sensor-to-controller and controller-to-actuator channels, and the proposed active compensation scheme can effectively deal with data packet disorder no matter in which channel data packet disorder occurs. However, the existence of data packet disorder in the sensor-to-controller channel makes it unnecessary to calculate the control signal at certain time instants. For example, in Fig. 9.1, if the sampled data packet sent at time instant t_k^s arrives at the controller at time instant t_{k1}^c, then the calculation of control signal based on sampled data at time instant t_{k-1}^s (which arrives at the controller at time instant t_{k-1}^c) is unnecessary since this control action will definitely not be used by the actuator (actually it will be discarded by CAS). To deal with this issue, a Sampled Data Selector (SDS) similar to CAS, can be deployed at the controller side. SDS also consists of a register and a logic comparator like CAS, which can be used in a similar way to determine the latest sampled data. The controller will work as normal if the newly arrived sampled data packet contains the latest system information; otherwise the controller will be idle. In this way, the use of SDS is able to reduce both the computation burden of the controller and the communication burden in the controller-to-actuator channel without affecting the deployment of the active compensation scheme.

9.3 Modeling and Further Discussion

In this section, we show that the active compensation scheme derives a unified model for NCSs which can take the communication constraints including network-induced delay, data packet dropout and data packet disorder into account simultaneously. We further point out that the active compensation scheme also reduces the communication constraints in NCSs which is thus beneficial for the control performance.

9.3.1 A Unified Model for NCSs

With the active compensation scheme proposed in the last section, the control law for the plant in (9.1) can be obtained as follows,

$$u(t) = g(x(i_k^* h)),\ t \in [t_k^*, t_{k+1}^*),\ k \geq 1 \tag{9.5}$$

where i_k^* is defined in (9.4), τ_k^* is the round trip delay with respect to i_k^* and $t_k^* \triangleq \tau_k^* + i_k^* h$.

Compared with the system model in (9.3), where the sequence of the sampling time instants $\{i_k h : k = 1, 2, \ldots\}$ can be decreasing, in the above system model, we can guarantee that the sequence of the sampling time instants $\{i_k^* h : k = 1, 2, \ldots\}$ is increasing which implies that the effect of data packet disorder is effectively eliminated in this model.

It is worth mentioning that the control law in (9.5) has already taken data packet dropout into account since there is no more constraint on the increasing sequence $\{i_k^* h : k = 1, 2, \ldots\}$. Therefore, the system model in (9.1) and (9.5) can be regarded as a unified model for NCSs, which considers the communication constraints including network-induced delay, data packet dropout and data packet disorder simultaneously.

9.3.2 *Further Discussion: Reduced Communication Constraints*

It is observed that the active compensation scheme for data packet disorder not only effectively eliminates the negative effects of data packet disorder, but also modifies the characteristics of the communication constraints to the system as well. This can be observed in the following two aspects.

9.3.2.1 Reduced Delay Increase Rate

With the active compensation scheme, it is noticed that at $t_k^* + h$, the worst case would be using the control signal at time t_k^* which implies no new control signals arrive during $(t_k^*, t_k^* + h)$. On the other hand, using a new control signal at $t_k^* + h$ can only decrease the actual delay τ_k^* from the worst case. In view of this fact, we have the following relationship

$$\tau_{t_k^*+h} \leq \tau_k^* + h \tag{9.6}$$

where $\tau_{t_k^*+h}$ denotes the round trip delay of the system at time $t_k^* + h$. This fact further implies that the actual delay to the system can not grow too fast, i.e., during any time interval (t_1, t_2), the actual delay can only increase as much as $\Delta \triangleq t_2 - t_1$. This constraint on the delay increase rate does not exist in conventional models for NCSs and can potentially be used to derive less conservative controller design methods for NCSs.

9.3.2.2 Reduced Delay Bound

It is well know that burst traffic often occurs in Internet-based data transmission, which implies that in practice network-induced delay with large lower and upper bounds usually varies for the most time within a narrow range of relatively small delays. For such a case, designing a controller with respect to these large bounds is clearly conservative. Fortunately, it is noticed that the active compensation scheme can effectively reduce the actual delay bound, since it discards those data packets with a sudden change in delay. For example, in the NCS test rig used in [37], the round trip delay is bounded in 2–8 sampling periods, while for most of the time (above 80%) the delay is constrained to two values, i.e., 4 and 5 sampling periods.

Using the active compensation scheme, the delay bound can be effectively narrowed to 3–6 sampling periods which is certainly beneficial for the control performance, for now we can design the controller for a narrow delay bound.

From the above analysis, it is seen that the active compensation scheme is more like a communication protocol rather than a control strategy, since this scheme has effectively reduced the communication constraints but does not affect the control structure itself. In this sense, while the use of this scheme could contribute greatly to improve the system performance, as shown in the next section, the analysis of the control performance such as stability, stabilization, robustness, etc., can still be done separately based on the unified model in (9.1) and (9.5), for which there are plentiful results in the literature [139, 145, 146]. Therefore we exclude such analysis in this chapter.

9.4 A Numerical Example

Example 9.1 Consider the following continuous-time linear system borrowed from [35], which has also been studied in, for example, [88, 145],

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & -0.1 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} u(t) \tag{9.7}$$

In the following simulation, we use the same state feedback gain as designed in [35], that is, $K = [-3.75 - 11.5]$ and the plant is sampled with a constant period $h = 0.04$ s. The lower and upper bounds of the round trip delay are first set as $\tau_{min} = 0.24$ s and $\tau_{max} = 1.6$ s respectively, and both channels are assumed to have the same upper and lower bounds (0.12 s and 0.8 s respectively).

According to the analysis in Sect. 9.1, data packet disorder inevitably occurs in this case and, without compensation old information might be used instead of latest information available. However, with the active compensation scheme proposed in Sect. 9.2, it is seen from Fig. 9.3 that the sampling time instants based on which the control actions are applied (that is, $i_k^* h$) are non-decreasing, which implies that data packet disorder has been effectively dealt with.

The system state responses of both with the active compensation scheme and without it are illustrated in Fig. 9.4, which proves the effectiveness of the proposed approach. Another case is also considered in Fig. 9.5, where the upper bound of the round trip delay is increased to $\tau_{max} = 2.4$ s with τ_{min} remaining unchanged. In this case, it is seen that the system is still stable in the presence of the active compensation scheme while unstable without it.

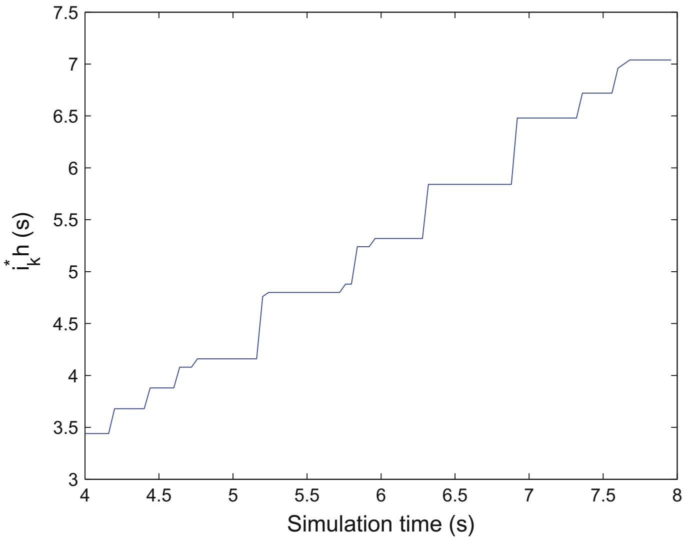


Fig. 9.3 The relationship between simulation time t and $i_k^* h$ in (9.5), which shows data packet disorder has been effectively dealt with

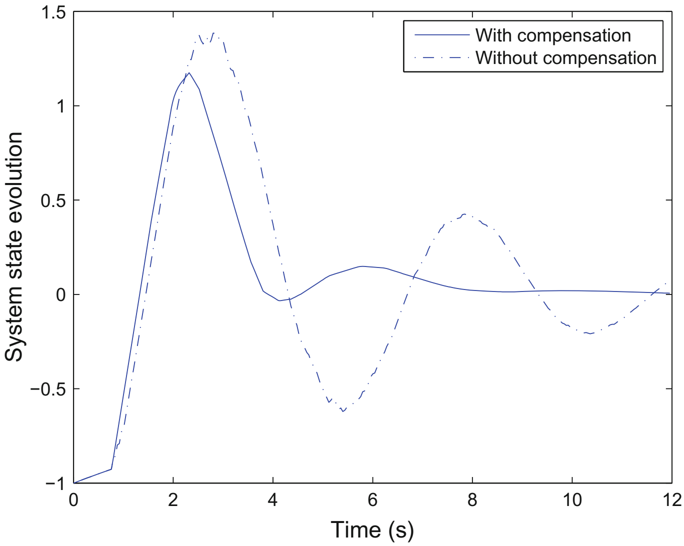


Fig. 9.4 The active compensation scheme results in a better system performance where $\tau_{max} = 1.6\,\text{s}$

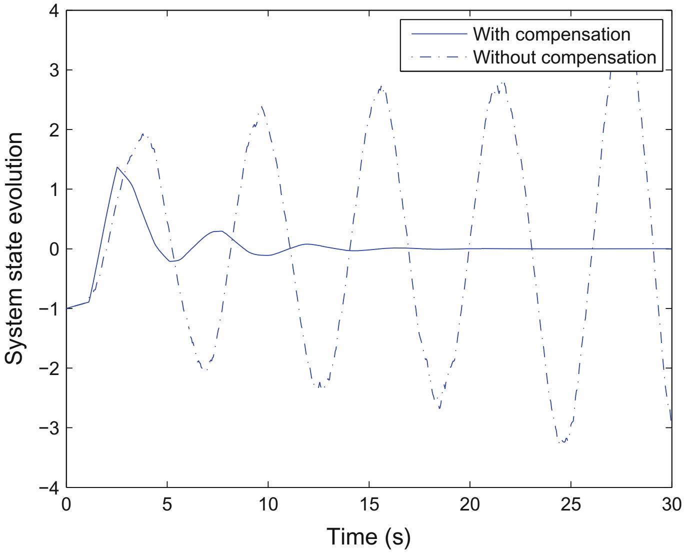


Fig. 9.5 The active compensation scheme maintains the stability of the closed-loop system where $\tau_{max} = 2.4\,\mathrm{s}$

9.5 Summary

Network-induced delay in NCSs has been widely explored in literature to date, while unfortunately the effect of data packet disorder is often neglected, despite its frequent presence in NCSs. In this chapter, the cause and effect of data packet disorder in NCSs are investigated in detail, and an active compensation scheme is also proposed to deal with the negative effect. The derived novel model for NCSs within this framework provides the foundation of more reasonable and effective theoretical analysis of NCSs. The effectiveness of the proposed approach is illustrated by a numerical example. Future research will be focused on controller design and performance analysis within the framework of data packet disorders.

Chapter 10
Error Bounded Sensing for Packet-Based Networked Control Systems

An error bounded sensing strategy is proposed within the packet-based control framework for networked control systems (NCSs). This strategy reduces the data transmissions in both the sensor-to-controller and the controller-to-actuator channels by allowing the transmissions of only the sensing and control data packets that satisfy some predetermined transmission rules. By fitting it into the packet-based control framework for NCSs, this strategy can achieve the goal of reducing the use of the communication resources while at the meanwhile maintaining the system performance at an acceptable level. Stabilized controllers are designed within this framework and the effects on the system stability brought by this approach are also investigated. Numerical and experimental examples illustrate the effectiveness of the proposed approach.

This chapter is organized as follows. In Sect. 10.1, the design details of the EBS strategy within the packet-based control framework are presented. The corresponding closed-loop system is then modeled and analyzed, and the effects brought by this approach are also investigated in Sect. 10.2. The effectiveness of the proposed approach is illustrated by both numerical and experimental examples in Sect. 10.3 and Sect. 10.4 concludes the chapter.

10.1 Error Bounded Sensing for PBNCSs

The block diagram of the networked control system considered in this chapter is illustrated in Fig. 2.1. Although not explicitly shown in the figure, it is usually the case that the communication network is shared with other applications but not private to the considered control system. The applications are also not limited solely to the control purpose. This system setting justifies the claim made earlier, that is, the dependence on the data exchanges in Internet-based NCSs ought to be reduced as much as possible, especially in the presence of heavy communication burdens, since the consuming of the communication resources can (1) affect the access to the communication resources of other applications and (2) increase the risk of causing

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_10

congestion in the communication network which can then degrade the overall system performance.

The EBS strategy proposed in this chapter is exactly intended for the very purpose of reducing the use of the communication resources. With the help of the packet-based control approach to NCSs, this strategy can achieve the goal of reducing the use of the communication resources while maintaining the system performance at an acceptable level at the same time. In what follows, the EBS strategy is discussed first, which is then fitted into the packet based control framework to form a complete solution to NCSs.

Before proceeding with the EBS strategy for PBNCSs, however, it is necessary to make the following assumption on the characteristics of the communication constraints in NCSs, which guarantees that the sensing data at the controller side and the control signals at the actuator side, are updated within finite time intervals. This assumption is reasonable in practice as well as important in theory.

Assumption 10.1 The network-induced delay in the sensor-to-controller channel and the controller-to-actuator channel are upper bounded by $\bar{\tau}_{sc}$ and $\bar{\tau}_{ca}$, respectively.

10.1.1 Error Bounded Sensing in the Sensor-to-controller Channel

As mentioned earlier, the implementation of the EBS strategy is based on the trade-off between the system performance and the use of the communication resources. Regardless of the specific implemental procedures, the goal of the strategy is clear, that is, it is aimed at reducing the sensing data transmissions as much as possible while at the same time guaranteeing the sensing error at the controller side being bounded by a predetermined bound, say, $\delta_s > 0$. Therefore, it is fairly clear that the key procedure of the EBS strategy is to determine whether the sensing signal at a specific time is sent to the controller or not. We refer to this key procedure as the "sensor transmission rule" (STR) which is discussed in detail as follows.

In order to present the STR in a precise manner, first define δ_k and $\sigma(\cdot) : \mathbb{N} \rightarrow \{0, 1\}$, as follows,

$$\delta_k \triangleq ||x(k) - x(k-1)|| \tag{10.1a}$$

$$\sigma(k) \triangleq \begin{cases} 1, & \text{if } x(k) \text{ is sent to the controller;} \\ 0, & \text{otherwise.} \end{cases} \tag{10.1b}$$

where $x(k)$ is the system state at time k and $|| \cdot ||$ indicates the Euclidian norm. For simplicity the system states are assumed to be fully accessible in this chapter. However, even this is not the case the system states can still be obtained (probably with error) by using an appropriate state observer and thus will not affect the discussions that follow. It is readily seen from the definitions in (10.1) that $\sigma(\cdot)$, as an indicator function, indicates whether a sensing signal at a specific time is sent to the controller or not. Therefore, the function $\sigma(\cdot)$ actually defines the STR in a mathematical manner, whose specific definition will be given as follows.

Suppose for some integers $k_0 > 1$ and $0 \leq j \leq N_s$, the sensing signal at time $k_0 - 1$ is sent to the controller while those from time k_0 to $k_0 + j - 1$ are not. The STR can then be defined at time $k_0 + j$, as follows,

$$\sigma(k_0 + j) = \begin{cases} 1, & \text{if } \delta_{k_0+j} > \alpha_j ||x(k_0 + j)|| \\ & \text{or } j = N_s; \\ 0, & \text{otherwise.} \end{cases} \tag{10.2a}$$

where $\alpha_j \triangleq \frac{\delta_s}{N_s+(N_s-j)\delta_s}$. Notice here that N_s, referred to as the "maximum transmission interval", is an integer being chosen to guarantee that the sensing signals at the controller side are updated within a finite time interval. In fact, by the STR defined in (10.2a), at least one sensing signal will be sent to the controller within N_s time steps and therefore the sensing data at the controller side will be updated no more than $\bar{\tau}_{sc}^* \triangleq \bar{\tau}_{sc} + N_s$ time steps. It is also noticed that the definition of the STR in (10.2a) is complete in the sense that it has been defined for all the time instants $k \geq 1$. To interpret this, for any $k \geq 1$, define $k_\sigma = \max\{j | 1 \leq j \leq k, \sigma(j) = 1\}$, and the STR in (10.2a) can then be reformed as

$$\sigma(k) = \begin{cases} 1, & \text{if } \delta_k > \alpha_{k-k_\sigma} ||x(k)|| \\ & \text{or } k - k_\sigma = N_s; \\ 0, & \text{otherwise.} \end{cases} \tag{10.2b}$$

which clearly is a complete definition for all $k \geq 1$.

The rationality of the STR defined in (10.2a) may not seem straightforward at the first sight, for one can readily propose a much simpler transmission rule by simply letting the sensing signal being sent at time k if δ_k is larger than a predetermined constant threshold. However, the transmission rule defined in (10.2a) is different from this simple rule in two aspects, the presence of the maximum transmission interval, N_s, and the use of variable thresholds, α_j, for good reasons. Firstly, as mentioned above, with the definition of N_s it is guaranteed that the sensing data at the controller side is updated no more than $\bar{\tau}_{sc}^*$ time steps while without it, a particular case could occur in principle where for a sufficient long time no sensing data is updated at the controller side which can destabilize the system readily. Secondly, with the carefully chosen variable thresholds, α_j, it is shown later that the sensing error at the controller side is always upper bounded by δ_s, which is essential for the sake of maintaining the system performance.

10.1.2 Packet-Based Control in the Controller-to-actuator Channel

With the use of the EBS strategy and the packet-based control approach, the FCS at time k can be constructed as follows,

$$\hat{U}(k) \triangleq [\hat{u}(k) \ \ldots \ \hat{u}(k+N-1)] \tag{10.3}$$

where $\hat{u}(k+i), i = 1, 2, \ldots, \bar{\tau}_{ca}$ are the forward control signals based on the sensing data $\hat{x}(k)$ at time k, and N is the number of the control signals that one data packet can contain. Note here that the symbol ˆ is used to indicate the fact that the control signals are calculated based on the sensing data with error, $\hat{x}(k)$, due to the use of the EBS strategy.

In conventional packet-based control approach to NCSs, the FCSs are sent to the actuator at every step. However, in view of the fact that the sensing data at the controller side is not updated at every step, it is therefore not necessary to send the FCS in the case of no sensing data being updated. This strategy, referred to as the "controller transmission rule" (CTR) , analogously to the STR discussed in the previous subsection, can considerably reduce the data transmissions in the controller-to-actuator channel. In fact, the total number of the FCS that is actually sent would be the same as that of the sensing data packets received by the controller. Therefore, analogously the upper bound of the delay in the controller-to-actuator channel after applying the CTR can be obtained as $\bar{\tau}^*_{ca} = \bar{\tau}_{ca} + N_s$.

10.1.3 The EBS Strategy for PBNCSs

Notice that with the EBS strategy, the sensing data at the controller side, $\hat{x}(k)$ at time k, is actually the real sensing signal at a previous time, $k - \tau^*_{sc,k}$, that is,

$$\hat{x}(k) = x(k - \tau^*_{sc,k}) \tag{10.4}$$

where $\tau^*_{sc,k} \leq \bar{\tau}^*_{sc}$ and $k - \tau^*_{sc,k}$ indicates the time when the sensing signal $\hat{x}(k)$ was sampled at the sensor side. This facts enables us to modify the conventional packet-based control for NCSs by reconstructing the FCS defined in (10.3),

$$U(k|k - \tau^*_{sc,k}) = [u(k|k - \tau^*_{sc,k}) \ \ldots \ u(k+N-1|k - \tau^*_{sc,k})] \tag{10.5}$$

where the sampling time of the sensing data based on which the FCS is calculated, is explicitly indicated. Note here that both the FCS and the forward control signals in (10.5) use a dual time indicator $(k_1|k_2)$ in which k_1 stands for the time instant of the control action while k_2 for the time instant of the sensing data that is used to produce the control signal. In light of (10.4) this FCS can be equivalent to the one in

(10.3) provided the same controller design methods are used. Their difference only relies on the different perspectives from which we look at the EBS strategy. That is, the effects brought by the EBS for PBNCSs can be interpreted by two different but equivalent ways, either sensing error without delay in the sensor-to-controller channel (10.3), or pure extra delay without sensing error (10.5).

Based on (10.5), the control signal that is actually applied to the plant at time k at the actuator side can be determined as follows. Denote the delay of the FCS from which the control signal is selected at time k by $\tau^*_{ca,k}$. This FCS was thus calculated based on the sensing data at time $\tau^*_k \triangleq \tau^*_{sc,k} + \tau^*_{ca,k}$ and therefore it should be $U(k - \tau^*_{ca,k}|k - \tau^*_k)$ based on the time at the actuator side. The control signal actually applied to the plant at time k can then be chosen as

$$u(k) = u(k|k - \tau^*_k) \tag{10.6a}$$

which can compensate for the current network-induced delay in a precise way. Let $\bar{\tau}^* \triangleq \bar{\tau}^*_{sc} + \bar{\tau}^*_{ca}$ be the modified upper bound of the delay in the round trip after the application of the EBS strategy and define $\Omega^* = \{2, 3, \ldots, \bar{\tau}^*\}$ as the set of all possible round trip delays, it is held that

$$\tau^*_k \in \Omega^*, \forall k \tag{10.6b}$$

which with (10.6a) defines the complete control law for the proposed approach in this chapter.

The algorithm of the EBS strategy for PBNCSs can now be organized as follows, the block diagram of which is illustrated in Fig. 10.1.

Remark 10.1 One may wonder why we do not construct a model of the plant at the controller side and update the system states using this model if the real sensing data is unavailable, as done in [53, 149], in which way the developed model seemingly can be used to reduced the sensing error. The reasons of not doing so are twofold. Firstly, the data transmission in both the sensor-to-controller and the controller-to-actuator channels can be effectively reduced using the EBS strategy within the packet-based

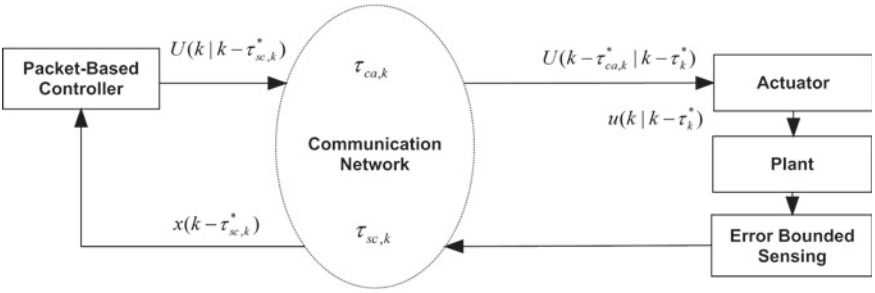


Fig. 10.1 Error bounded sensing for packet-based networked control systems

Algorithm 10.1 The EBS strategy for PBNCSs

S1. Initialization. Set $k = 1$, $k_\sigma = 1$. Sample the system state $x(1)$ and send it to the controller.
S2. Let $k = k + 1$. If $\delta_k > \alpha_{k-k_\sigma}||x(k)||$ or $k - k_\sigma = N_s$, sample the system state $x(k)$, send it to the controller and let $k_\sigma = k$.
S3. Check if the system state is updated at the controller side. If so then calculate the FCS by (10.5) and send it to the actuator.
S4. The control signal in (10.6) is applied to the plant. Return to S2.

control framework, which has not been considered in this model-based approach. Secondly, with the use of the packet-based control approach which is capable of producing forward control signals based on delayed sensing data, the reconstruction of the system states is thus not necessary which is however the main concern of using the model-based approach.

10.2 Stabilization and Further Discussion

In this section, the stability and stabilization issues of the proposed approach are considered first, and the effects on the system stability brought by the EBS strategy are then investigated by comparing it with conventional packet-based control approach. This analysis is based on two different models for the proposed approach, that is, in the former analysis the delay effect brought by the approach is explicitly formulated with the FCS in (10.3) while for the latter the focus is mainly on the sensing error introduced by the approach with the FCS in (10.5).

For simplicity the following linear plant in discrete time is considered in Fig. 2.1, however it is worth pointing out that the proposed approach is applicable to any systems but not limited to this particular type,

$$x(k+1) = Ax(k) + Bu(k) \tag{10.7}$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^m, A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$. The controller is assumed to be of the form of state feedback. In light of (10.6) the controller can be obtained as follows,

$$u(k) = u(k|k-\tau_k^*) = K_{\tau_k^*} x(k-\tau_k^*) \tag{10.8a}$$

Note here that the controller gains $K_{\tau_k^*}$ are delay dependent, which is one of the most important characteristics of the packet-based control approach. This characteristic distinguishes this approach from conventional control approaches to NCSs, where normally a constant controller gain is used for all network conditions [36, 37].

It is noticed that by (10.4) the control law in (10.8a) can also be written in the following way

$$u(k) = u(k|k-\tau_k^*) = K_{\tau_{ca,k}^*} \hat{x}(k-\tau_{ca,k}^*) \tag{10.8b}$$

Notice that the controller in (10.8b) is now based on sensing data $\tau^*_{ca,k}$ instead of τ^*_k as in (10.8a), meaning that the delay in the sensor-to-controller channel is eliminated in this model. However this is obtained at the cost of introducing sensing error to the system, which is defined as

$$e_s(k) \triangleq ||x(k) - \hat{x}(k)||, k \geq 1 \tag{10.9}$$

Although it is possible to define the same control gains in both (10.8a) and (10.8b), it is preferred, however, to define the controller gains based on the current delays, as done above. It is thus clear that the two controllers are not exactly equivalent, as will be illustrated later in Fig. 10.3 in Example 10.1.

10.2.1 Stabilization

It is noticed that the closed-loop system in (10.7) and (10.8a) is in its standard form within the packet-based control framework. As far as the model is concerned, the EBS strategy only increases the upper bound of the delay but does not affect the formulation of the system, meaning that the standard analysis techniques for PBNCSs can still be applied here. Therefore, for completeness the stability and stabilization results for the closed-loop system in (10.7) with the control law defined in (10.8a) are presented as follows without proving, since the proofs can be obtained following similar procedures as done in [109].

Theorem 10.1 (Stability) *Given $\lambda \geq 1$ and the feedback gains $K_i, i \in \Omega^*$. The system in (10.7) with the control law in* (10.8a) *is stable if there exist $P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \geq 0$, T_i^1, T_i^2 with appropriate dimensions such that*

1. $\forall i \in \Omega^*$,

$$\Phi_i = \begin{pmatrix} \Phi_i^{11} & \Phi_i^{12} & (A-I)^T H_i \\ * & \Phi_i^{22} & (BK_i)^T H_i \\ * & * & -H_i \end{pmatrix} < 0$$

$$\Psi_i = \begin{pmatrix} \lambda S_i^{11} & \lambda S_i^{12} & \lambda T_i^1 \\ * & \lambda S_i^{22} & \lambda T_i^2 \\ * & * & R_i \end{pmatrix} \geq 0$$

2. $\forall i, j \in \Omega^*$

$$P_i \leq \lambda P_j, Q_i \leq \lambda Q_j, R_i \leq \lambda R_j$$

where

$$\Phi_i^{11} = (\lambda - 1)P_i + Q_i + 2\lambda P_i(A - I) + T_i^1 + (T_i^1)^T + iS_i^{11},$$

$$\Phi_i^{12} = \lambda P_i BK_i - T_i^1 + (T_i^2)^T + iS_i^{12},$$

$$\Phi_i^{22} = -T_i^2 - (T_i^2)^T + iS_i^{22},$$

$$H_i = \lambda P_i + \bar{\tau}^* R_i.$$

Based on Theorem 10.1, the following stabilization result can then be obtained, which is computationally feasible due to the cone complementarity linearization technique [135].

Theorem 10.2 (Stabilization) *Given $\lambda \geq 1$. Define the following nonlinear minimization problem $\mathcal{P}_i$ involving LMI conditions for $i, j \in \Omega^*$,*

$$\mathcal{P}_i : \begin{cases} \text{Minimize } Tr(Z_i R_i + L_i P_i + M_i Q_i) \\ \text{Subject to:} \\ L_i = L_i^T > 0,\ W_i = W_i^T > 0,\ M_i = M_i^T > 0, \\ L_i \leq \lambda L_j,\ M_i \leq \lambda M_j,\ W_i \leq \lambda W_j, \\ X_i = \begin{pmatrix} X_i^{11} & X_i^{12} \\ (X_i^{12})^T & X_i^{22} \end{pmatrix} \geq 0, \\ \Phi_i' < 0,\ \Psi_i' \geq 0,\ \Theta_i^1 \geq 0,\ \Theta_i^2 \geq 0,\ \Theta_i^3 \geq 0,\ \Theta_i^4 \geq 0. \end{cases} \tag{10.10}$$

where

$$\Phi_i' = \begin{pmatrix} \Phi_i^{11'} & \Phi_i^{12'} & \lambda L(A - I)^T & \bar{\tau} L(A - I)^T \\ * & \Phi_i^{22'} & \lambda(BV_i)^T & \bar{\tau}(BV_i)^T \\ * & * & -\lambda L_i & 0 \\ * & * & * & -\bar{\tau} M_i \end{pmatrix},$$

$$\Psi_i' = \begin{pmatrix} \lambda X_i^{11} & \lambda X_i^{12} & \lambda Y_i^1 \\ * & \lambda X_i^{22} & \lambda Y_i^2 \\ * & * & Z_i \end{pmatrix},$$

$$\Theta_i^1 = \begin{pmatrix} R_i & P_i \\ * & Q_i \end{pmatrix},\ \Theta_i^2 = \begin{pmatrix} Z_i & I \\ * & R_i \end{pmatrix},$$

$$\Theta_i^3 = \begin{pmatrix} L_i & I \\ * & P_i \end{pmatrix}, \Theta_i^4 = \begin{pmatrix} M_i & I \\ * & Q_i \end{pmatrix}.$$

If the solution of $\mathcal{P}_i = 3n, \forall i \in \Omega$, the system in (10.7) *is then stabilizable with the control gains in* (10.8a) *being $K_i = V_i L_i^{-1}$.*

Remark 10.2 The above LMI-based stabilization approach is only one of the possible ways to design the controller within the proposed framework. Indeed, the designed EBS strategy for PBNCSs is intended to reduce the data transmissions and it does not affect directly the design of the controllers. This means all the previously designed controllers within the packet-based control framework can still be used for this modified framework. However, further improvement is still necessary, as it is straightforward that a design approach with the EBS strategy taking into consideration, rather than one without it, can potentially improve the system performance further.

10.2.2 The Effects of the EBS Strategy

In order to investigate the effects of the EBS strategy for PBNCSs, the upper bound of the sensing error defined in (10.9) is first discussed in the following proposition.

Proposition 10.1 *The sensing error $e_s(k)$ brought by the EBS strategy is upper bounded by δ_s, $\forall k$. That is,*

$$e_s(k) \leq \delta_s ||x(k)||, \forall k \tag{10.11}$$

Proof For simplicity of notation let $j \triangleq \tau^*_{sc,k}$ in (10.4). Noticing that $\sigma(k-i) = 0$ for $0 \leq i \leq j-1 < N_s$, the following inequality for $0 \leq i \leq j-1$ is thus held in light of (10.2),

$$||x(k-i) - x(k-i-1)|| \leq \alpha_{j-i} ||x(k-i)||$$

From above it is concluded that for $1 \leq i \leq j$

$$||x(k-i)|| \leq (1 + \alpha_{j-i+1}) ||x(k-i+1)||$$

Repeatedly using above yields

$$\alpha_{j-i} ||x(k-i)|| \leq \alpha_{j-i} \prod_{l=0}^{i-1} (1 + \alpha_{j-i+1+l}) ||x(k)||$$

Notice that by the definition of α_i we have $\alpha_i(1 + \alpha_{i+1}) = \alpha_{i+1}, 0 \leq i \leq N_s - 1$. Therefore

$$
\begin{aligned}
& \alpha_{j-i} \prod_{l=0}^{i-1}(1+\alpha_{j-i+1+l}) \\
\leq \; & \alpha_{j-i} \prod_{l=0}^{i-1+N_s-j}(1+\alpha_{j-i+1+l}) = \alpha_{N_s}
\end{aligned}
$$

Thus,

$$
\begin{aligned}
e_s(k) & \leq \sum_{i=0}^{j-1} ||x(k-i) - x(k-i-1)|| \\
& \leq \sum_{i=0}^{j-1} \alpha_{j-i} ||x(k-i)|| \\
& \leq j\alpha_{N_s} ||x(k)|| \\
& \leq N_s \alpha_{N_s} ||x(k)|| \\
& = \delta_s ||x(k)||
\end{aligned}
$$

which completes the proof.

With (10.11), the control law in (10.8b) can then be reformed as

$$
u(k) = K_{\tau^*_{ca,k}}(I + \Delta_k)x(k - \tau^*_{ca,k})
$$

with

$$
(I + \Delta_k)x(k - \tau^*_{ca,k}) \triangleq \hat{x}(k - \tau^*_{ca,k})
$$

where by (10.11) we have

$$
||\Delta_k|| \leq \delta_s, \forall k
$$

The closed-loop system can then be obtained as

$$
x(k+1) = Ax(k) + BK_{\tau^*_{ca,k}}(I + \Delta_k)x(k - \tau^*_{ca,k}) \tag{10.12}
$$

Correspondingly, without the EBS strategy the closed-loop system should be of the following form

$$
x(k+1) = Ax(k) + BK_{\tau_k}x(k - \tau_k), \tau_k \in \Omega \tag{10.13}
$$

where $\Omega = \{2, 3, \ldots, \bar{\tau}\}$.

Remark 10.3 From (10.12) and (10.13), it is seen that the EBS strategy modifies the system in two ways, the introduction of the bounded sensing error (represented by Δ_k)

and the modification of the delay to the system. The former invariably introduces negative effects to the system which is the cost that we have to pay in order to reduce the use of the communication resources. However, noticing that $\bar{\tau}_{ca}^* = \bar{\tau}_{ca} + N_s$ and $\bar{\tau} = \bar{\tau}_{sc} + \bar{\tau}_{ca}$, the latter can, at least in principle, reduce the delay bound to the system (when $N_s < \bar{\tau}_{sc}$) which can potentially be beneficial to the system performance.

To quantitatively interpret these effects, a special case of the closed-loop system in (10.12) is considered by letting $N_s = \bar{\tau}_{sc}$, thus making $\tau_{ca,k}^* = \tau_k$ and (10.12) being reformed to

$$x(k+1) = Ax(k) + BK_{\tau_k}(I + \Delta_k)x(k - \tau_k) \tag{10.14}$$

where $\tau_k \in \Omega$.

For the closed-loop systems in (10.13) and (10.14), their stability conditions are compared in the following theorem. It is seen that the stability conditions for both systems are closely related and the system in (10.14) requires relatively stronger conditions for stability due to the sensing error introduced, which makes sense in practice.

Theorem 10.3 *Given* $\lambda \geq 1$ *and the feedback gains* $K_i, i \in \Omega$. *The closed-loop system in (10.13) is stable if there exist* $P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \geq 0$, T_i^1, T_i^2 *with appropriate dimensions and a scalar* $\gamma > 0$ *such that*

1. $\forall i \in \Omega$,

$$\Phi_i^{''} < 0 \tag{10.15}$$

$$\Psi_i^{''} \geq 0 \tag{10.16}$$

2. $\forall i, j \in \Omega$,

$$P_i \leq \lambda P_j,\ Q_i \leq \lambda Q_j,\ R_i \leq \lambda R_j \tag{10.17}$$

where $\Phi_i^{''}$ *and* $\Psi_i^{''}$ *are similarly defined as in Theorem 10.1 by replacing* Ω^* *by* Ω. *Furthermore, The closed-loop system in (10.14) is stable if (10.16) and (10.17) are held and (10.15) is replaced by*

$$\begin{pmatrix} \Phi_i^{''} & \Upsilon_i^T \\ * & -\gamma I \end{pmatrix} < 0 \tag{10.18}$$

where $\Upsilon_i = [\lambda \delta P_i B\ 0\ \delta H_i B]$, $\delta = \delta_s \bar{K}$ *and* $\bar{K} = \max\{||K_i|||i \in \Omega\}$.

Proof The stability conditions for the system in (10.13) can be obtained directly from Theorem 10.1. From the definition of δ it is noticed that

$$BK_{\tau_k}(I + \Delta_k) = BK_{\tau_k} + \delta B \times K_{\tau_k}\Delta_k/\delta$$

with $||K_{\tau_k}\Delta_k/\delta|| \leq 1$. The closed-loop system in (10.14) can then be well treated as a time delay system with time-varying uncertainty and (10.18) can be obtained by replacing BK_{τ_k} in (10.15) by $BK_{\tau_k} + \delta B \times K_{\tau_k}\Delta_k/\delta$ and then follow a standard robust stability analysis, as done in [150]. The technical details are therefore omitted in this chapter.

10.3 Numerical and Experimental Examples

In this section, both numerical and experimental examples are considered to illustrate the effectiveness of the EBS strategy for PBNCSs.

Example 10.1 In this example, the system in (10.7) is considered with the following system matrices the same as in Example 2.1,

$$A = \begin{pmatrix} 0.98 & 0.1 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0.04 \\ 0.1 \end{pmatrix}.$$

This system is readily seen to be open-loop unstable. In the simulation, the initial state for the above system is set as $x_0 = [-1\ 1]^T$, the upper bound of the delay in both channels are $\bar{\tau}_{sc} = \bar{\tau}_{ca} = 3$ respectively, $N_s = 2$ and $\delta_s = 0.35$. Other parameters can then be obtained as follows: $\bar{\tau} = \bar{\tau}_{sc} + \bar{\tau}_{ca} = 6$, $\bar{\tau}^*_{sc} = \bar{\tau}_{sc} + N_s = 5$, $\bar{\tau}^*_{ca} = \bar{\tau}_{sc} + N_s = 5$, $\bar{\tau}^* = \bar{\tau}^*_{sc} + \bar{\tau}^*_{ca} = 10$ and α_j, $0 \leq j \leq N_s$ can also be obtained accordingly which are not listed here for simplicity of notations.

The main purpose of this example is to illustrate the effectiveness of the proposed EBS strategy within the packet-based control framework, by comparing it with conventional packet-based control approach. In order to eliminate possible effects on the system performance brought by different controller design methods, the controllers for both approaches are therefore designed using the same receding horizon approach, which yields the following feedback gain K for the packet-based control approach with $\bar{\tau} = 6$,

$$K = \begin{pmatrix} K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \end{pmatrix} = \begin{pmatrix} -0.4280 & -0.9753 \\ -0.3412 & -0.8704 \\ -0.2660 & -0.7739 \\ -0.2012 & -0.6853 \\ -0.1458 & -0.6040 \end{pmatrix},$$

and the following feedback gain K^* for the EBS strategy for PBNCSs with $\bar{\tau}^* = 10$,

$$K^* = \begin{pmatrix} K_2^* \\ K_3^* \\ K_4^* \\ K_5^* \\ K_6^* \\ K_7^* \\ K_8^* \\ K_9^* \\ K_{10}^* \end{pmatrix} = \begin{pmatrix} -0.4280 & -0.9753 \\ -0.3412 & -0.8704 \\ -0.2660 & -0.7739 \\ -0.2012 & -0.6853 \\ -0.1458 & -0.6040 \\ -0.0990 & -0.5296 \\ -0.0600 & -0.4616 \\ -0.0280 & -0.3996 \\ -0.0023 & -0.3432 \end{pmatrix}.$$

Four different cases are considered in the simulation: (1) conventional packet-based control approach with all the sensing and control data packets being sent; (2) the EBS strategy for PBNCSs with the control law in (10.8a) where the delay effect is explicitly considered; (3) the EBS strategy for PBNCSs with the control law in (10.8b) where the extra delay is explicitly considered and (4) conventional packet-based control approach with only partial sensing and control data packets being sent (with the same transmission ratio as using the EBS strategy). The last case is considered mainly to illustrate the effectiveness of the proposed EBS strategy by comparison in the presence of poor communication conditions and is simulated by applying zero control when no sensing data is available.

The state responses for the above four cases are illustrated in Fig. 10.2. It is seen that the system performance of case 1) is the EBSt which is reasonable since this case has used the most communication resources. Though slightly worse than case 1), the system performances with the EBS strategy (the solid line for case 2) and the dashed line for case 3)) are still maintained at a satisfactory level, which illustrates the effectiveness of the proposed approach. This result can be verified by looking into the comparison of the control inputs for these three cases shown in Fig. 10.3. All these control inputs are seen to be very close. It is worth mentioning that the acceptable system performances using the EBS strategy are achieved with a 65% reduction of the communication resources, meaning that only around 35% of the sensing data packets and the FCSs are actually sent.

The effectiveness of the EBS strategy can further be proven by comparing with case 4) (the dotted line in Fig. 10.2) where the same amount of the sensing data packets and FCSs are sent but conventional packet-based control approach gives rise to much worse system performance. This also proves the effectiveness of the EBS strategy in the presence of poor communication conditions.

As for the two control laws, (10.8a) and (10.8b), for the EBS strategy for PBNCSs, it is noticed that the control law in (10.8a) results in a little better system performance than that in (10.8b). This makes sense in this particular example, since the used controller design method in this example only takes delay effect into account but not the sensing error.

Example 10.2 In order to verify the effectiveness of the proposed approach in practice, the Internet-based test rig for NCSs introduced in Chap. 2 is considered.

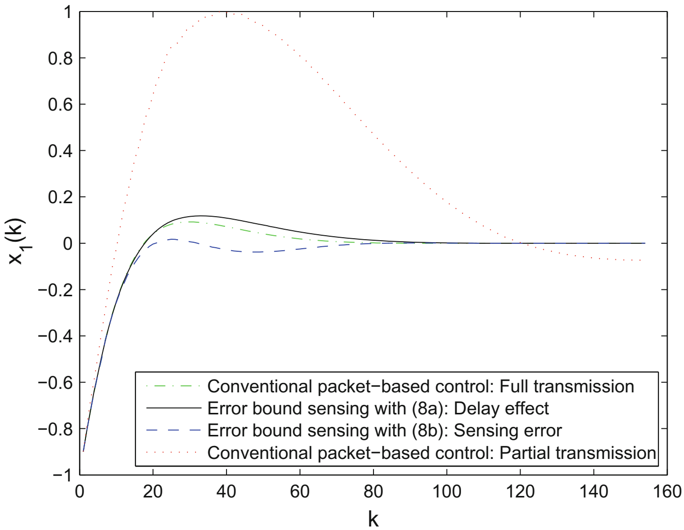


Fig. 10.2 Illustrating the effectiveness of the error bounded sensing strategy for packet-based control for networked control systems

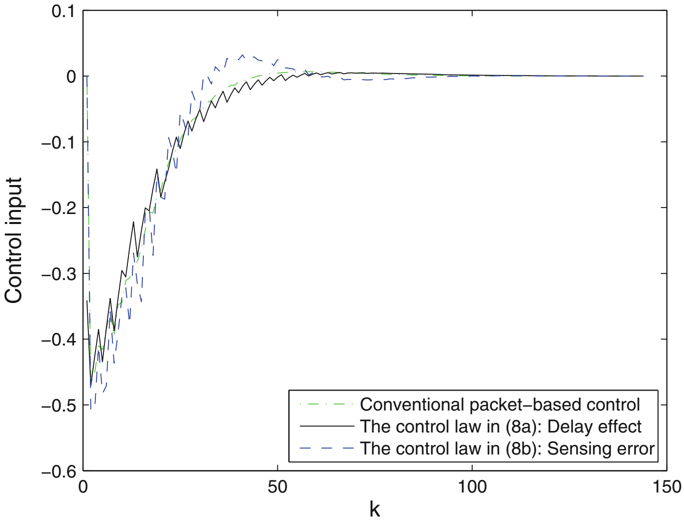


Fig. 10.3 Comparing the control signals with and without the error bounded sensing strategy

In the experiment we set $N_s = 4$, $\delta_s = 0.4$, and thus $\bar{\tau}_{sc}^* = \bar{\tau}_{ca}^* = 8$ sampling periods and $\bar{\tau}^* = 16$ sampling periods. The controller is designed using Theorem 10.2, as follows,

$$K = \begin{pmatrix} K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \\ K_7 \\ K_8 \\ K_9 \\ K_{10} \\ K_{11} \\ K_{12} \\ K_{13} \\ K_{14} \\ K_{15} \\ K_{16} \end{pmatrix} = \begin{pmatrix} -0.0735\ 0.0065\ 0.0294 \\ -0.0671\ 0.0057\ 0.0260 \\ -0.0626\ 0.0051\ 0.0236 \\ -0.0601\ 0.0052\ 0.0225 \\ -0.0579\ 0.0051\ 0.0215 \\ -0.0564\ 0.0048\ 0.0209 \\ -0.0536\ 0.0045\ 0.0198 \\ -0.0530\ 0.0045\ 0.0194 \\ -0.0524\ 0.0044\ 0.0191 \\ -0.0517\ 0.0043\ 0.0188 \\ -0.0506\ 0.0042\ 0.0181 \\ -0.0496\ 0.0041\ 0.0177 \\ -0.0491\ 0.0039\ 0.0175 \\ -0.0483\ 0.0040\ 0.0170 \\ -0.0481\ 0.0041\ 0.0169 \end{pmatrix}.$$

The system response is illustrated in Fig. 10.4, where it is seen that the system performance is fairly satisfactory. At the meanwhile, it is noticed that with the EBS strategy and the above parameters, in both channels only around 26% of the data pack-

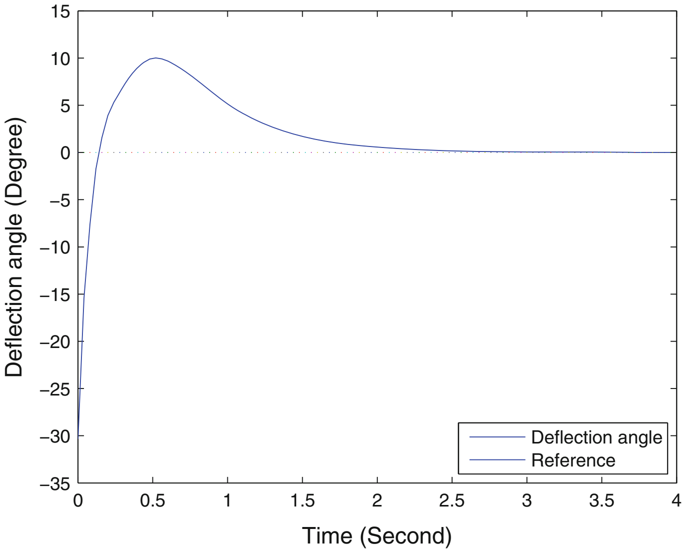


Fig. 10.4 Experimental response using the error bounded sensing strategy for packet-based networked control systems

ets are actually sent, meaning that the system performance in Fig. 10.4 is achieved with a reduction of 74% of the use of the communication resources in both channels. This reduction is beneficial for other applications that share the Internet and also beneficial for the considered system in the sense that it can still perform well in the case of poor communication conditions with the use of the EBS strategy for PBNCSs. This thus proves the effectiveness of both the EBS strategy as well as the stabilization controller designed in this chapter.

10.4 Summary

Reducing the use of the communication resources is one of the important design principles in NCSs, which is beneficial for other applications that share the same communication network and also of potential significant importance to the system itself in the presence of poor communication conditions. An error bounded sensing strategy is thus proposed in this chapter within the packet-based control framework for NCSs in order to do this reduction, by sending only the sensing and control data packets that are absolutely necessary for the purpose of maintaining the system performance. It is noticed that the efficient reduction of the use of the communication resources by the proposed approach is obtained at the cost of introducing bounded sensing error, or equivalently, extra delay, to the system. Theoretical analysis reveals that these negative effects can be well treated within the packet-based control framework and do not affect the system performance severely. Numerical and experimental examples verify the theoretical results and also illustrate its effectiveness in the presence of poor communication conditions. Therefore, in some sense this strategy completes the packet-based control approach and enables the latter to be an efficient and complete solution to NCSs.

Chapter 11
Packet-Based Deadband Control for Networked Control Systems

This chapter proposes a deadband control approach to packet-based networked control systems. Compared with standard PBNCSs, the deadband control approach takes full advantage of the packet-based data transmission in NCSs, and thus considerably reduces the use of the communication resources in NCSs whilst maintaining the system performance at a satisfactory level. A stabilized controller design method is obtained using time delay switched system theory, which has not been achieved in previously reported packet-based control approaches. The proposed deadband control strategy and the stabilized controller design method are verified using a numerical example as well as practical experiments based on the Internet-based test rig for NCSs.

This chapter is organized as follows. Within the packet-based control framework, Sect. 11.1 presents the packet-based deadband control approach to NCSs. The corresponding closed-loop system is then obtained, with stability analysis and a stabilized controller design method obtained based on LMIs in Sect. 11.2. In Sect. 11.3 both numerical and experimental examples are considered to illustrate the effectiveness of the theoretical results and Sect. 11.4 concludes the chapter.

11.1 Packet-Based Deadband Control for NCSs

The following linear plant in discrete time is considered in this chapter, which is controlled over the network by a remote controller, as shown in Fig. 2.1,

$$x(k+1) = Ax(k) + Bu(k) \tag{11.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$.

For the implementation of the packet-based deadband control approach to NCSs, we assume that the dynamics of the control system and the characteristics of the communication network in Fig. 2.1 satisfy the following assumptions as discussed in Chap. 2.

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_11

Assumption 11.1 (*Delay bound*) The sum of the network-induced delay and consecutive data packet dropout in both the sensor-to-controller and the controller-to-actuator channels (denoted by $\bar{\tau}_{sc}$ and $\bar{\tau}_{ca}$ respectively) are upper bounded, i.e.,

$$\bar{\tau}_{sc} \triangleq \max_{k\geq 1}\{\tau_{sc,k} + \bar{\chi}_{sc}\} < \infty \tag{11.2a}$$

$$\bar{\tau}_{ca} \triangleq \max_{k\geq 1}\{\tau_{ca,k} + \bar{\chi}_{ca}\} < \infty \tag{11.2b}$$

where $\tau_{sc,k}$, $\bar{\chi}_{sc}$ and $\tau_{ca,k}$, $\bar{\chi}_{ca}$ represent the network-induced delay and the upper bound of consecutive data packet dropout in the sensor-to-controller and the controller-to-actuator channels respectively.

As discussed in Chap. 2, using the packet-based control approach, the following FCS is calculated and sent in one data packet to the actuator,

$$U^p(k|k-\tau_{sc,k}) = [u(k|k-\tau_{sc,k}) \ \ldots \ u(k+\bar{\tau}_{ca}|k-\tau_{sc,k})] \tag{11.3}$$

Denote the effective load of the data packet being used in the NCS by B_p and the data size required for encoding a single step of the control signal by B_c. The number of control signals that one data packet can contain can then be obtained as

$$N = \lfloor \frac{B_p}{B_c} \rfloor \tag{11.4}$$

where $\lfloor \frac{B_p}{B_c} \rfloor = \max\{\varsigma | \varsigma \in \mathbb{N}, \varsigma \leq \frac{B_p}{B_c}\}$.

It is noticed that N is usually much larger than \bar{\tau}_{ca}. This observation thus motivates us to design the following modified FCS where the length of FCS is extended to the maximum of what a data packet can contain but not determined by the upper bound of the communication constrain in the controller-to-actuator channel,

$$U(k|k-\tau_{sc,k}) = [u(k|k-\tau_{sc,k}) \ \ldots \ u(k+N-1|k-\tau_{sc,k})] \tag{11.5}$$

The motivation of proposing the deadband control strategy is due to the fact that the communication constraints play a dominant role in the system performance of NCSs and for a better system performance, we have to decrease possible congestion in the network by reducing the use of the communication resources. On the other hand, much more redundant forward control signals are packed into one data packet using FCS in (11.5). This enables us to set a deadband for FCSs and send only those that have a sufficiently large change compared with the last sent FCS. In this way, the use of the communication resources can be significantly reduced and the system performance can still be maintained at a satisfactory level if the deadband is carefully chosen.

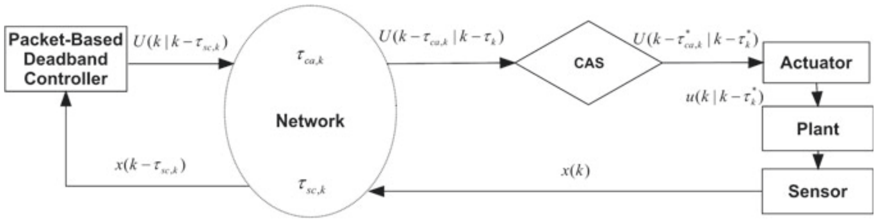


Fig. 11.1 The block diagram of packet-based deadband control for networked control systems where CAS represents the control action selector

The block diagram of the packet-based deadband control approach to NCSs is illustrated in Fig. 11.1, where it is seen that this structure is different from conventional control approaches mainly in two aspects: the packet-based deadband controller and the so-called Control Action Selector (CAS) at the actuator side. The latter consists of a register to store only the latest data packet and a logic comparator to determine which data packet contains the latest information and thus can be used to deal with data packet disorder and to actively compensate for network-induced delay, as discussed in Chap. 2.

The packet-based deadband controller is used to produce FCS in (11.5) and, different from standard packet-based control approach, also to determine whether a newly produced FCS should be sent or not. For this purpose, a register is present at the controller side to store the last sent FCS which is denoted by $U(k-\varrho_k|k-\varrho_k-\tau_{sc,k-\varrho_k})$ at time k at the controller side, where $k-\varrho_k$ is the time when the last FCS was sent. The newly produced FCS $U(k|k-\tau_{sc,k})$ at time k will be sent to the actuator if it has changed dramatically compared with the one last sent, i.e.,

$$\delta_k \triangleq \max_{0\leq i\leq N-\varrho_k-1} \frac{||\Delta u_{ki}||}{||u(k+i|k-\tau_{sc,k})||} > \delta \tag{11.6}$$

where δ is the deadband set for FCSs, $||\cdot||$ denotes the Euclidean norm and $\Delta u_{ki} = u(k+i|k-\tau_{sc,k}) - u(k+i|k-\varrho_k-\tau_{sc,k-\varrho_k})$. On the other hand, in order that there is always a control signal available at the actuator side, FCS has to be sent at least once within $N-\bar{\tau}_{ca}$ time steps, which also implies that $\varrho_k \leq N-\bar{\tau}_{ca}-1, \forall k$.

The algorithm of packet-based deadband control for NCSs can be organized as follows.

It is readily seen that this packet-based deadband control approach is different from standard packet-based control approach since not all the FCSs are sent to the actuator, but only those that have changed dramatically compared with the one last sent. This strategy reduces the demand on the communication resource in NCSs and, furthermore, can improve the system performance in the presence of heavy transmission load on the network being used in NCSs, as illustrated in Sect. 11.3.

Algorithm 11.1 Packet-based deadband control

Initiation. Set $k = 0$, $\varrho_k = 0$.
if At time k at the controller side, either 1) (11.6) is satisfied; or 2) $\varrho_k = N - \bar{\tau}_{ca} - 1$, the controller **then**
  sends the current FCS to the actuator
  updates the register of the controller to be this FCS
  lets $\varrho_{k+1} = 1$, $k = k + 1$
else
  Let $\varrho_{k+1} = \varrho_k + 1$, $k = k + 1$ and wait for the next time instant.
end if
if A new FCS is received **then**
  CAS compares its time stamp with the one already in its register and only the latest is stored.
end if
The appropriate control signal is selected from FCS by (11.7b) and applied to the plant.

11.2 Stability and Stabilization of Packet-Based Deadband Control

In this section, the control law using the packet-based deadband control approach is explicitly presented, with a comparative analysis with the previous packet-based control approach. The stability of the derived closed-loop systems is then investigated from a time delay switched system theory perspective [117, 146, 151], with also a comparison of the stability conditions for both approaches. Finally, an LMI-based stabilization result is obtained, which can be solved using the well-known cone complementarity technique [135].

11.2.1 The Control Laws

It is noticed that one major difference between the previous packet-based control approach and the packet-based deadband control approach in this chapter lies in the use of different FCSs, as presented in (11.3) and (11.5), respectively. Using FCS in (11.3), that is, with the use of the packet-based control approach, the control action taken at time k at the actuator side is determined by

$$u^p(k) = u(k|k - \tau_k^{*p}), \tau_k^{*p} \in \Omega^p \tag{11.7a}$$

where τ_k^{*p} denotes the round trip delay of the FCS being used at time k, $\bar{\tau}^p = \bar{\tau}_{sc} + \bar{\tau}_{ca}$ is the upper bound of the delay and consecutive data packet dropout for the round trip and $\Omega^p = \{2, 3, \ldots, \bar{\tau}^p\}$. It is worth mentioning that $\tau_k^{*p} \geq 2$ is due to the fact that the data packets in both the sensor-to-controller and the controller-to-actuator channels experience at least one step delay respectively in practice.

With the use of FCS in (11.5) and the corresponding deadband control strategy in (11.6), the control signal used may be based on older sampled data information with the control action taken at time k being

$$u(k) = u(k|k-\tau_k^*), \tau_k^* \in \Omega \tag{11.7b}$$

where τ_k^* denotes the round trip delay of the FCS being used in the packet-based deadband control case, $\bar{\tau} = \bar{\tau}_{sc} + N - 1$, $\Omega = \{2, 3, \ldots, \bar{\tau}\}$ and it is seen that $\tau_k^* \geq \tau_k^{*p}, \forall k$.

Though the control signal in (11.7b) may be based on older sampled data information, with the deadband control strategy in (11.6), the difference between $u(k)$ and $u^p(k)$ is however restrained within a small range, which helps to maintain the system performance using the packet-based deadband control approach at a satisfactory level,

$$||u(k) - u^p(k)|| \leq \delta ||u^p(k)||, \forall k \tag{11.8}$$

For simplicity, in this chapter state feedback is used and thus the control law in (11.7a) and (11.7b) can be explicitly represented by

$$u^p(k) = K^p_{\tau_k^{*p}} x(k - \tau_k^{*p}), \tau_k^{*p} \in \Omega^p \tag{11.9a}$$

and

$$u(k) = K_{\tau_k^*} x(k - \tau_k^*), \tau_k^* \in \Omega \tag{11.9b}$$

respectively, where the feedback gains $K^p_{\tau_k^{*p}}$, $K_{\tau_k^*}$ with respect to the corresponding round trip delays τ_k^{*p} and τ_k^*, are to be designed.

With the control laws defined in (11.9a) and (11.9b), the closed-loop system model with the packet-based control approach can be obtained as

$$x(k+1) = Ax(k) + BK^p_{\tau_k^{*p}} x(k - \tau_k^{*p}), \tau_k^{*p} \in \Omega^p \tag{11.10a}$$

and for the packet-based deadband control approach, it is obtained as

$$x(k+1) = Ax(k) + BK_{\tau_k^*} x(k - \tau_k^*), \tau_k^* \in \Omega \tag{11.10b}$$

In light of the relationship between both control laws in (11.8), the closed-loop system model for the packet-based deadband control approach in (11.10b) can also be represented by the following system model with time-varying uncertainty,

$$x(k+1) = Ax(k) + (B + \Delta B_k) K^p_{\tau_k^{*p}} x(k - \tau_k^{*p}), \tau_k^{*p} \in \Omega^p \tag{11.10c}$$

where ΔB_k satisfies

$$(B + \Delta B_k) K^p_{\tau_k^{*p}} x(k - \tau_k^{*p}) = B K_{\tau_k^*} x(k - \tau_k^*)$$

and $||\Delta B_k|| \leq \delta_B \triangleq \delta ||B||$ in light of (11.8).

11.2.2 Stability and Stabilization

In this subsection, we first investigate the stability of the closed-loop system in (11.10b) for the packet-based deadband control approach to NCSs, and then compare the stability conditions for both approaches, with and without the deadband control strategy.

Theorem 11.1 *Given $\lambda \geq 1$ and the feedback gains in (11.10b) for the packet-based deadband control approach K_i, $i \in \Omega$. The closed-loop system in (11.10b) is stable if there exist $P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \geq 0$, T_i^1, T_i^2 with appropriate dimensions such that*

1. $\forall i \in \Omega$,

$$\Phi_i = \begin{pmatrix} \Phi_i^{11} & \Phi_i^{12} & (A - I)^T H_i \\ * & \Phi_i^{22} & (BK_i)^T H_i \\ * & * & -H_i \end{pmatrix} < 0 \tag{11.11}$$

$$\Psi_i = \begin{pmatrix} \lambda S_i^{11} & \lambda S_i^{12} & \lambda T_i^1 \\ * & \lambda S_i^{22} & \lambda T_i^2 \\ * & * & R_i \end{pmatrix} \geq 0 \tag{11.12}$$

2. $\forall i, j \in \Omega$

$$P_i \leq \lambda P_j, \; Q_i \leq \lambda Q_j, \; R_i \leq \lambda R_j \tag{11.13}$$

where

$$\begin{aligned} \Phi_i^{11} &= (\lambda - 1) P_i + Q_i + 2\lambda P_i (A - I) \\ &\quad + T_i^1 + (T_i^1)^T + i S_i^{11}, \end{aligned}$$

$$\Phi_i^{12} = \lambda P_i B K_i - T_i^1 + (T_i^2)^T + i S_i^{12},$$

$$\Phi_i^{22} = -T_i^2 - (T_i^2)^T + i S_i^{22},$$

$$H_i = \lambda P_i + \bar{\tau} R_i.$$

Proof Suppose at time k, $\tau_k^* = i \in \Omega$. Let

$$z(l) = x(l+1) - x(l)$$

We then obtain

$$x(k) - x(k - \tau_k^*) - \sum_{l=k-\tau_k^*}^{k-1} z(l) = 0$$

Define the following Lyaponuv functional and notice that the choices of the matrices $P_{\tau_k^*}$, $Q_{\tau_k^*}$, $R_{\tau_k^*}$ at time k are dependent on the corresponding round trip delay $\tau_k^* \in \Omega$,

$$V_i(k) = V_i^1(k) + V_i^2(k) + V_i^3(k)$$

with

$$V_i^1(k) = x^T(k) P_i x(k)$$

$$V_i^2(k) = \sum_{m=-\bar{\tau}+1}^{0} \sum_{l=k+m-1}^{k-1} z^T(l) R_{\tau_l^*} z(l)$$

$$V_i^3(k) = \sum_{l=k-\tau_k^*}^{k-1} x^T(l) Q_{\tau_l^*} x(l)$$

Define $\Delta V_i(k) = V_{\tau_{k+1}^*}(k+1) - V_i(k)$. Then along the trajectory of the system in (11.10b), we have

$$\begin{aligned}
\Delta V_i^1(k) &= x^T(k+1) P_{\tau_{k+1}^*} x(k+1) - x^T(k) P_i x(k) \\
&\leq \lambda x^T(k+1) P_i x(k+1) - x^T(k) P_i x(k) \\
&= (\lambda - 1) x^T(k) P_i x(k) + 2\lambda x^T(k) P_i z(k) \\
&\quad + \lambda z^T(k) P_i z(k)
\end{aligned}$$

$$\begin{aligned}
\Delta V_i^2(k) &= \sum_{m=-\bar{\tau}+1}^{0} \Big(\sum_{l=k+m}^{k} - \sum_{l=k+m-1}^{k-1} \Big) z^T(l) R_{\tau_l^*} z(l) \\
&= \bar{\tau} z^T(k) R_i z(k) - \sum_{l=k-\bar{\tau}}^{k-1} z^T(l) R_{\tau_l^*} z(l) \\
&\leq \bar{\tau} z^T(k) R_i z(k) - \sum_{l=k-\tau_k^*}^{k-1} z^T(l) R_{\tau_l^*} z(l)
\end{aligned}$$

$$\begin{aligned}\Delta V_i^3(k) &= (\sum_{l=k-\tau_{k+1}^*+1}^{k} - \sum_{l=k-\tau_k^*}^{k-1})x^T(l)Q_{\tau_l^*}x(l) \\ &= (\sum_{l=k-\tau_{k+1}^*+1}^{k-1} - \sum_{l=k-\tau_k^*}^{k-1})x^T(l)Q_{\tau_l^*}x(l) \\ &\quad + x^T(k)Q_i x(k)\end{aligned}$$

In light of the fact that using the packet-based deadband control approach, data packet disorder has been effectively eliminated by CAS, that is, the actuator will never use an older control signal as long as the latest is available. Therefore we have the following relationship

$$k + 1 - \tau_{k+1}^* \geq k - \tau_k^*, \forall k \geq 1$$

and thus

$$\Delta V_i^3(k) \leq x^T(k)Q_i x(k)$$

Notice that

$$z(k) = (A - I)x(k) + BK_i x(k - \tau_k^*)$$

and

$$R_i \geq \frac{1}{\lambda}R_j, \; Q_i \geq \frac{1}{\lambda}Q_j, \forall i, j$$

We then obtain

$$\begin{aligned}\Delta V_i(k) \leq & x^T(k)((\lambda - 1)P_i + Q_i + 2\lambda P_i(A - I) + \\ & (A - I)^T H_i(A - I))x(k) \\ & + 2x^T(k)(\lambda P_i BK_i + (A - I)^T H_i BK_i) \\ & x(k - \tau_k^*) \\ & + x^T(k - \tau_k^*)(BK_i)^T H_i BK_i x(k - \tau_k^*) \\ & - \frac{1}{\lambda}\sum_{l=k-\tau_k^*}^{k-1} z^T(l)R_i z(l)\end{aligned} \qquad (11.14)$$

where $H_i = \lambda P_i + \bar{\tau} R_i$.

In addition, we have for any T_i^1, T_i^2 with appropriate dimensions,

$$2[x^T(k)T_i^1 + x^T(k-\tau_k^*)T_i^2] \times [x(k) - x(k-\tau_k^*) - \sum_{l=k-\tau_k^*}^{k-1} z(l)] = 0 \tag{11.15}$$

and for any S_i with appropriate dimensions,

$$i\zeta_1^T(k)S_i\zeta_1(k) - \sum_{l=k-\tau_k^*}^{k-1} \zeta_1^T(k)S_i\zeta_1(k) = 0 \tag{11.16}$$

where $\zeta_1(k) = [x^T(k)\ x^T(k-\tau_k^*)]^T$.

From (11.14), (11.15) and (11.16) we obtain

$$\Delta V_i(k) \le \zeta_1^T(k)\Xi_i\zeta_1(k) - \frac{1}{\lambda}\sum_{l=k-\tau_k^*}^{k-1} \zeta_2^T(k,l)\Psi_i\zeta_2(k,l)$$

where

$$\Xi_i = \begin{pmatrix} \Phi_i^{11} + \Pi_i^{11} & \Phi_i^{12} + \Pi_i^{12} \\ * & \Phi_i^{22} + \Pi_i^{22} \end{pmatrix}$$

$$\Pi_i^{11} = (A-I)^T H_i (A-I),$$

$$\Pi_i^{12} = (A-I)^T H_i B K_i,$$

$$\Pi_i^{22} = (BK_i)^T H_i B K_i,$$

and $\zeta_2(k,l) = [\zeta_1^T(k), z^T(l)]^T$. If $\Xi_i < 0$ and $\Psi_i \ge 0$, then we can guarantee that the system is stable. Furthermore, notice that by Schur complement, $\Xi_i < 0$ is equivalent to $\Phi_i < 0$. Thus we complete the proof.

The stability result for packet-based deadband control in Theorem 11.1 can be readily extended to the packet-based control approach since both of them have similar closed-loop models, as presented in (11.10a) and (11.10b), respectively.

Corollary 11.1 *Given $\lambda \ge 1$ and the feedback gains for the packet-based control approach $K_i^p, i \in \Omega^p$. The closed-loop system in (11.10a) is stable if there exist $P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \ge 0$, T_i^1, T_i^2 with appropriate dimensions such that*

1. $\forall i \in \Omega^p$,

$$\Phi_i^p = \begin{pmatrix} \Phi_i^{11} & \Phi_i^{12^p} & (A-I)^T H_i \\ * & \Phi_i^{22} & (BK_i^p)^T H_i \\ * & * & -H_i \end{pmatrix} < 0 \tag{11.17}$$

$$\Psi_i^p = \Psi_i \geq 0 \tag{11.18}$$

2. $\forall i, j \in \Omega^p$,

$$P_i \leq \lambda P_j,\ Q_i \leq \lambda Q_j,\ R_i \leq \lambda R_j \tag{11.19}$$

where Φ_i^{11}, Φ_i^{22}, H_i *and* Ψ_i *are defined in Theorem 11.1 and*

$$\Phi_i^{12^p} = \lambda P_i B K_i^p - T_i^1 + (T_i^2)^T + i S_i^{12}.$$

From Theorem 11.1 and Corollary 11.1 it is readily to obtain the following relationship of the closed-loop stability conditions between packet-based control and packet-based deadband control for NCSs.

Corollary 11.2 *If* $K_i^p = K_i, i \in \Omega^p$ *and the stability conditions in Theorem 11.1 for the closed-loop system in* (11.10b) *using the packet-based deadband control approach are satisfied, then the closed-loop system in* (11.10a) *using the packet-based control approach is stable.*

Consider the closed-loop system description in (11.10c) from the robust control perspective and let $\Delta B_k K_{\tau_k^*} = \delta_B \bar{K}^p \cdot \frac{\Delta B_k K_{\tau_k^*}}{\delta_B \bar{K}^p}$ where $\bar{K}^p = \max\{||K_i^p||\ |i \in \Omega^p\}$. It is readily seen that $||\frac{\Delta B_k K_{\tau_k^*}}{\delta_B \bar{K}^p}|| \leq 1$. The comparison of the stability conditions between packet-based control and packet-based deadband control for NCSs can then be revealed from the following theorem.

Theorem 11.2 *Given* $\lambda \geq 1$ *and the feedback gains for the packet-based control approach* $K_i^p, i \in \Omega^p$. *The closed-loop system with the packet-based deadband control approach in* (11.10c) *is stable if there exist* $P_i = P_i^T > 0$, $Q_i = Q_i^T > 0$, $R_i = R_i^T > 0$, $S_i = \begin{pmatrix} S_i^{11} & S_i^{12} \\ (S_i^{12})^T & S_i^{22} \end{pmatrix} \geq 0$, T_i^1, T_i^2 *with appropriate dimensions and a scalar* $\gamma > 0$ *such that*

1. $\forall i \in \Omega^p$,

$$\begin{pmatrix} \Phi_i^p & \Upsilon_i^T \\ * & -\gamma I \end{pmatrix} < 0 \tag{11.20}$$

$$\Psi_i^p \geq 0 \tag{11.21}$$

2. $\forall i, j \in \Omega^p$,

$$P_i \leq \lambda P_j, Q_i \leq \lambda Q_j, R_i \leq \lambda R_j \tag{11.22}$$

where Φ_i^p *and* Ψ_i^p *are defined in Corollary 11.1 and* $\Upsilon_i = [\lambda \delta_B \bar{K}^p \; 0 \; \delta_B \bar{K}^p H_i]$.

Proof The theorem can be obtained following a standard robust stability analysis for systems with time-varying uncertainty, as done in [117], and thus we omit the technical details.

Remark 11.1 Suppose $K_i^p = K_i, i \in \Omega^p$. With the use of the deadband control strategy in (11.6) we have $\delta \neq 0$. In this case it is readily seen that (11.20) in Theorem 11.2 is harder to be satisfied than (11.11) in Corollary 11.1, that is, the system with the deadband control strategy is more likely to be unstable than the system without it, which is true in reality. On the other hand, if $\delta = 0$, that is, no deadband control strategy is used, we have $\Delta B_k \equiv 0$ and thus the closed-loop system model in (11.10c) is equivalent to (11.10a). In this case, it is seen that $\Upsilon_i = 0$ in Theorem 11.2 and then (11.20) is equivalent to (11.11), thus enabling Theorem 11.2 and Corollary 11.1 to be equivalent. From this point of view, Theorem 11.2 effectively presents the effects of the deadband control strategy on the closed-loop stability of the system considered.

Based on Theorem 11.1, we obtain the following stabilized controller design method.

Theorem 11.3 *Given* $\lambda \geq 1$. *The system in* (11.10b) *is stabilizable if there exist* $L_i = L_i^T > 0$, $W_i = W_i^T > 0$, $M_i = M_i^T > 0$, $X_i = \begin{pmatrix} X_i^{11} & X_i^{12} \\ (X_i^{12})^T & X_i^{22} \end{pmatrix} \geq 0$, Y_i^1, Y_i^2, V_i *with appropriate dimensions such that*

1. $\forall i \in \Omega$,

$$\Phi_i' = \begin{pmatrix} \Phi_i^{11'} & \Phi_i^{12'} & \lambda L(A-I)^T & \bar{\tau} L(A-I)^T \\ * & \Phi_i^{22'} & \lambda (BV_i)^T & \bar{\tau}(BV_i)^T \\ * & * & -\lambda L_i & 0 \\ * & * & * & -\bar{\tau} M_i \end{pmatrix} < 0 \tag{11.23}$$

$$\Psi_i' = \begin{pmatrix} \lambda X_i^{11} & \lambda X_i^{12} & \lambda Y_i^1 \\ * & \lambda X_i^{22} & \lambda Y_i^2 \\ * & * & L_i M_i^{-1} L_i \end{pmatrix} \geq 0 \tag{11.24}$$

2. $\forall i, j \in \Omega$,

$$L_i \leq \lambda L_j, M_i \leq \lambda M_j, W_i \leq \lambda W_j \tag{11.25}$$

where

$$\Phi_i^{11'} = (\lambda - 1)L_i + W_i + 2\lambda(A - I)L_i + Y_i^1 + (Y_i^1)^T + iX_i^{11},$$

$$\Phi_i^{12'} = \lambda B V_i - Y_i^1 + (Y_i^2)^T + iX_i^{12},$$

$$\Phi_i^{22'} = -Y_i^2 - (Y_i^2)^T + iX_i^{22}.$$

Furthermore, the control law is defined in (11.9b) *with* $K_i = V_i L_i^{-1}$.

Proof Stability condition (11.11) in Theorem 11.1 can be reformed as

$$\begin{pmatrix} \Phi_i^{11} & \Phi_i^{12} & \lambda(A-I)^T P_i & \bar{\tau}(A-I)^T R_i \\ * & \Phi_i^{22} & \lambda(BK_i)^T P_i & \bar{\tau}(BK_i)^T R_i \\ * & * & -\lambda P_i & 0 \\ * & * & * & -\bar{\tau} R_i \end{pmatrix} < 0 \tag{11.26}$$

Pre- and Post multiply (11.26) and (11.12) by $diag(P_i^{-1}, P_i^{-1}, P_i^{-1}, R_i^{-1})$ and $diag(P_i^{-1}, P_i^{-1}, P_i^{-1})$, respectively, and let $L_i = P_i^{-1}$, $M_i = R_i^{-1}$, $W_i = P_i^{-1} Q_i P_i^{-1}$, $X_i = diag(P_i^{-1}, P_i^{-1}) \cdot S_i \cdot diag(P_i^{-1}, P_i^{-1})$, $Y_i^j = P_i^{-1} T_i^j P_i^{-1}$, $j = 1, 2$, $V_i = K_i P_i^{-1}$. We then complete the proof.

It is noticed that (11.24) in Theorem 11.3 is no longer LMI conditions due to the term $L_i M_i^{-1} L_i$. There are several techniques available to deal with this difficulty, among which the cone complementarity linearization technique is one of the most commonly used [135]. In the following corollary, this technique is used to derive a suboptimal solution for (11.24) by transforming it to a nonlinear minimization problem involving LMI conditions.

Corollary 11.3 *Given* $\lambda \geq 1$. *Define the following nonlinear minimization problem involving LMI conditions for* $i \in \Omega$,

$$\mathcal{P}_i : \begin{cases} \textit{Minimize } Tr(Z_i R_i + L_i P_i + M_i Q_i) \\ \textit{Subject to } (11.23), (11.25), L_i = L_i^T > 0, W_i = W_i^T > 0, \\ M_i = M_i^T > 0, X_i = \begin{pmatrix} X_i^{11} & X_i^{12} \\ (X_i^{12})^T & X_i^{22} \end{pmatrix} \geq 0, \\ \Psi_i^{''} \geq 0, \Theta_i^1 \geq 0, \Theta_i^2 \geq 0, \Theta_i^3 \geq 0, \Theta_i^4 \geq 0. \end{cases} \tag{11.27}$$

where

$$\Psi_i^{''} = \begin{pmatrix} \lambda X_i^{11} & \lambda X_i^{12} & \lambda Y_i^1 \\ * & \lambda X_i^{22} & \lambda Y_i^2 \\ * & * & Z_i \end{pmatrix},$$

$$\Theta_i^1 = \begin{pmatrix} R_i & P_i \\ * & Q_i \end{pmatrix}, \Theta_i^2 = \begin{pmatrix} Z_i & I \\ * & R_i \end{pmatrix},$$

$$\Theta_i^3 = \begin{pmatrix} L_i & I \\ * & P_i \end{pmatrix}, \Theta_i^4 = \begin{pmatrix} M_i & I \\ * & Q_i \end{pmatrix}.$$

If the solution of $\mathcal{P}_i = 3n, \forall i \in \Omega$, *the system in* (11.10b) *is then stabilizable with the control law being defined in Theorem 11.3.*

Remark 11.2 In this chapter LMI-based stability and stabilization results are obtained which are feasible in practice (Corollary 11.3) and will be proven to be effective by both numerical and experimental examples in the next section. However, it is worth mentioning that as a control framework, the performance of the packet-based deadband control approach to NCSs can certainly be investigated by any appropriate control theories and the controller can be designed according to the closed-loop system models in (11.10b) and (11.10c), independently from the deployment of the packet-based deadband control strategy. In this sense further theoretical analysis and improvement are still needed, in order to reduce the conservativeness of the LMI-based results presented in this chapter.

11.3 Numerical and Experimental Examples

In this section, both numerical and experimental examples are considered to illustrate the effectiveness of the proposed packet-based deadband control approach to NCSs and the stabilized controller design method within this framework.

Example 11.1 Consider the system in (11.1) with the following system matrices as seen in Example 2.1, which is seen to be open-loop unstable,

$$A = \begin{pmatrix} 0.98 & 0.1 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0.04 \\ 0.1 \end{pmatrix}.$$

In the simulation, the initial state for the system in (11.1) is set as $x_0 = [-1\ 1]^T$, the upper bound of the delay and consecutive dropout for the round trip is $\bar{\tau} = 4$ and the deadband for the packet-based deadband approach is chosen as $\delta = 0.1$.

In this example, our main purpose is to illustrate the effectiveness of the proposed deadband control strategy within the packet-based control framework, by comparing it with the standard packet-based control approach in Chap. 2. In order to eliminate possible effects on the system performance brought by different controller design methods, in this example the controllers for both cases are designed using the same receding horizon approach as in Chap. 2, which yields the following feedback gain for the packet-based control approach,

$$K^p = \begin{pmatrix} K_2^p \\ K_3^p \\ K_4^p \end{pmatrix} = \begin{pmatrix} -0.6438 & -1.4748 \\ -0.5242 & -1.3079 \\ -0.4198 & -1.1549 \end{pmatrix}$$

and the feedback gain for the packet-based deadband control approach with $N = 9$,

$$K = \begin{pmatrix} K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \\ K_7 \\ K_8 \\ K_9 \\ K_{10} \end{pmatrix} = \begin{pmatrix} -0.4371 & -1.0510 \\ -0.3428 & -0.9334 \\ -0.2615 & -0.8251 \\ -0.1921 & -0.7257 \\ -0.1334 & -0.6346 \\ -0.0843 & -0.5515 \\ -0.0439 & -0.4759 \\ -0.0114 & -0.4074 \\ 0.0142 & -0.3454 \end{pmatrix}$$

where both of the feedback gains K_i^p and K_i are designed from $i = 2$ due to the fact that the round trip delays in both cases are not less than 2 sampling periods, as stated in (11.7a).

It is seen from the comparison of the state responses in Fig. 11.2 that the system performance with the deadband control strategy is still maintained at a satisfactory level. This can also be verified by looking into the comparison of the control inputs for both cases shown in Fig. 11.3 where it is seen that the control inputs to both systems are very close. It is worth mentioning, however, only around 60% of FCSs

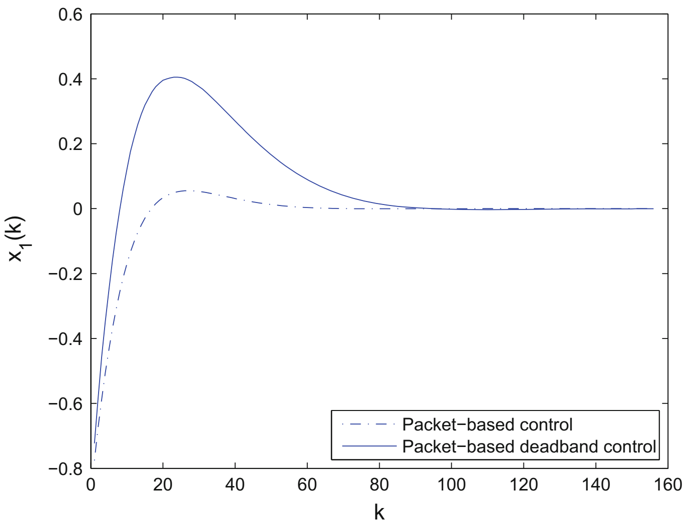


Fig. 11.2 Example 11.1. Comparison of the state responses between with and without the deadband control strategy

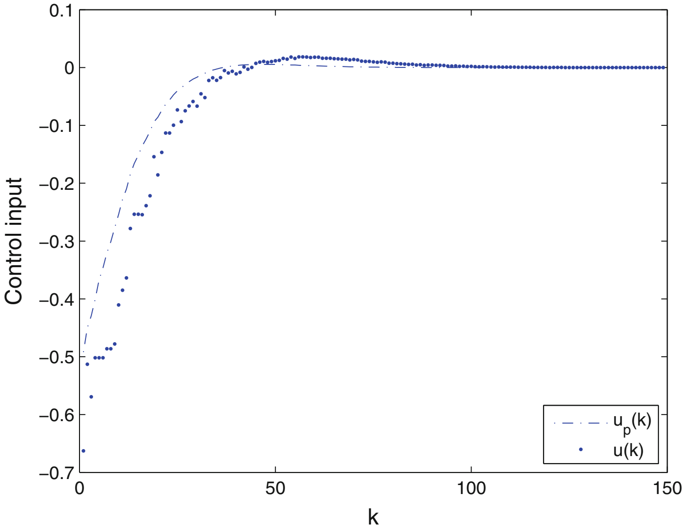


Fig. 11.3 Example 11.1. Comparison of the control inputs between with and without the deadband control strategy

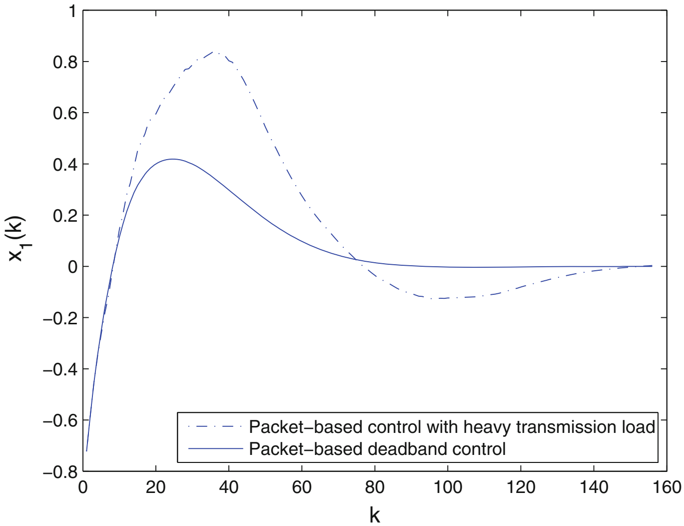


Fig. 11.4 Example 11.1. Comparison of the state responses with heavy transmission load

are sent to the actuator using the deadband control strategy. The effectiveness of the packet-based deadband control approach can also been seen from Fig. 11.4, where the packet-based deadband control approach yields a far better system performance than the packet-based control approach, when the latter also transmit only around 60% of its FCSs.

Example 11.2 The Internet-based test rig for NCSs as discussed in Chap. 2 is used to illustrate the effectiveness of the proposed packet-based deadband control approach and the stabilized controller design method.

In the experiment, the round trip delay between UK and China is found to be typically upper bounded by 0.32 s which is 8 sampling periods. For the implementation of the packet-based deadband control approach, an FCS containing 20 forward control signals is used, with the feedback gains being the following, designed using Corollary 11.3,

$$K = \begin{pmatrix} K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \\ K_7 \\ K_8 \\ K_9 \\ K_{10} \\ K_{11} \\ K_{12} \\ K_{13} \\ K_{14} \\ K_{15} \\ K_{16} \\ K_{17} \\ K_{18} \\ K_{19} \\ K_{20} \\ K_{21} \end{pmatrix} = \begin{pmatrix} -0.0643 & 0.0039 & 0.0249 \\ -0.0589 & 0.0035 & 0.0221 \\ -0.0547 & 0.0032 & 0.0202 \\ -0.0527 & 0.0030 & 0.0191 \\ -0.0495 & 0.0025 & 0.0180 \\ -0.0494 & 0.0029 & 0.0177 \\ -0.0485 & 0.0032 & 0.0175 \\ -0.0466 & 0.0027 & 0.0168 \\ -0.0458 & 0.0029 & 0.0165 \\ -0.0460 & 0.0027 & 0.0164 \\ -0.0459 & 0.0030 & 0.0164 \\ -0.0456 & 0.0031 & 0.0164 \\ -0.0445 & 0.0026 & 0.0158 \\ -0.0440 & 0.0024 & 0.0154 \\ -0.0439 & 0.0025 & 0.0153 \\ -0.0437 & 0.0025 & 0.0152 \\ -0.0429 & 0.0022 & 0.0149 \\ -0.0430 & 0.0023 & 0.0149 \\ -0.0434 & 0.0026 & 0.0150 \\ -0.0437 & 0.0028 & 0.0151 \end{pmatrix}$$

Using a deadband of $\delta = 0.14$, it is seen from Fig. 11.5 that only around 25% of the FCSs are sent to the actuator. In other words, the deadband control strategy used here reduces around 75% of the control data transmissions.

On the other hand, with the feedback gains defined above and the packet-based deadband control approach in Sect. 11.1, the output response of the DC servo system which is remotely controlled via the Internet is illustrated in Fig. 11.6. The results show that the output responses converge quickly which proves the effectiveness of both the packet-based deadband control approach and stabilized controller design method.

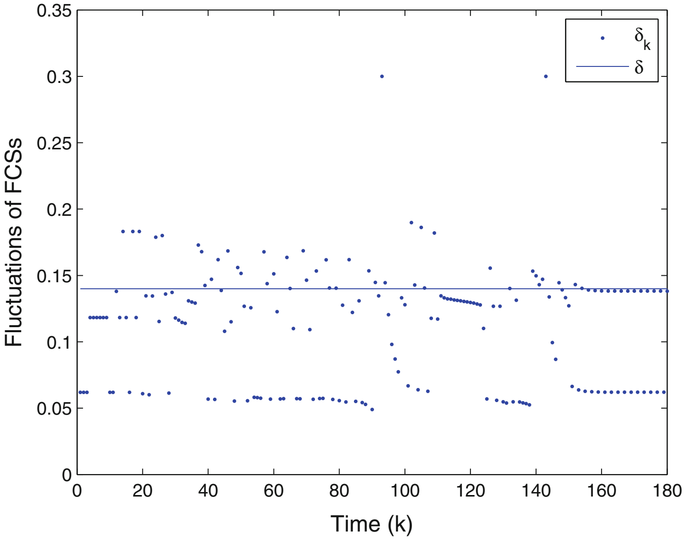


Fig. 11.5 Example 11.2. Using deadband to reduce data transmissions in NCSs

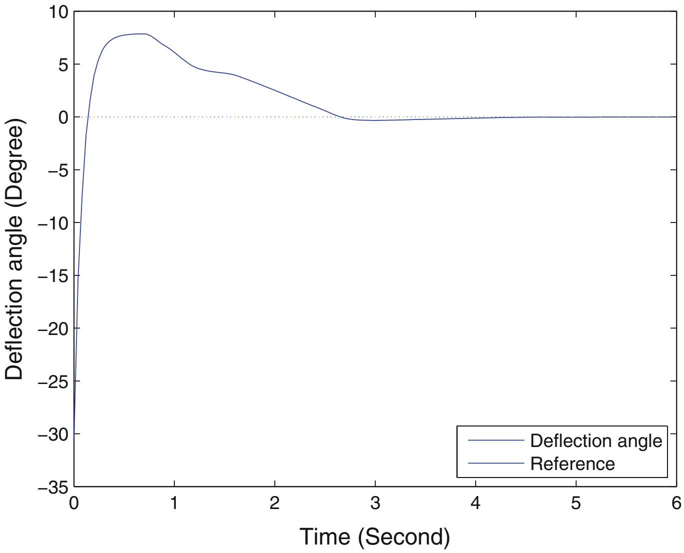


Fig. 11.6 Example 11.2. Experimental response using the packet-based deadband control approach

11.4 Summary

Within the packet-based control framework for NCSs, a packet-based deadband control approach is proposed, with also a stabilized controller design method obtained using time delay switched system theory. This approach exploits more fully of the packet structure in the network being used in NCSs, by sending a larger forward control sequence and then setting a deadband on the sequences which allows transmission only in the presence of a sufficiently large change between the current sequence and the one last sent. As a modified packet-based control approach to NCSs, this work expands the application of the approach when the reduction of the communication resources is necessary.

Chapter 12
Packet-Based Control and Scheduling Co-Design for Networked Control Systems

Different from previous chapters where only one NCS occupies the network resource, in this chapter the design and analysis of the system setting where a set of linear NCSs (i.e., subsystems) share the limited network resources is considered. In this system setting, the packet-based control approach is applied to each subsystem and two scheduling algorithms, the existing static Rate Monotonic (RM) algorithm and a newly proposed Dynamic Feedback Scheduling (DFS) algorithm, are considered to schedule the network resource allocations among those subsystems, to achieve the objective that all the subsystems are stable under the limited network resources.

This chapter is organized as follows. The problem being studied is first described in Sect. 12.1, and then the packet-based controller for each subsystem is obtained in Sect. 12.2, from which an important definition for the subsystems is derived which is the supremum of round trip delay under which the stability of the subsystems is guaranteed. With this definition, two scheduling algorithms are presented in Sect. 12.3, and a numerical example is presented in Sect. 12.4. Section 12.5 concludes the chapter.

12.1 Problem Statement

A set of $\mathcal{N}$ continuous-time LTI systems $(\mathcal{S}_i^c)_{1\leq i\leq \mathcal{N}}$ are considered which share the network resource as shown in Fig. 12.1,

$$\mathcal{S}_c^i : \begin{cases} \dot{x}_i^c(t) = A_i^c x_i^c(t) + B_i^c u_i^c(t) & (12.1a) \\ y_i^c(t) = C_i^c x_i^c(t) & (12.1b) \end{cases}$$

where $x_i^c(t) \in \mathbb{R}^{n_i}$, $u_i^c(t) \in \mathbb{R}^{m_i}$, and $y_i^c(t) \in \mathbb{R}^{r_i}$.

In a digital control environment, a discrete-time representation $\mathcal{S}_d^i$ of system $\mathcal{S}_c^i$ is obtained using a sampling period T_i,

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6_12

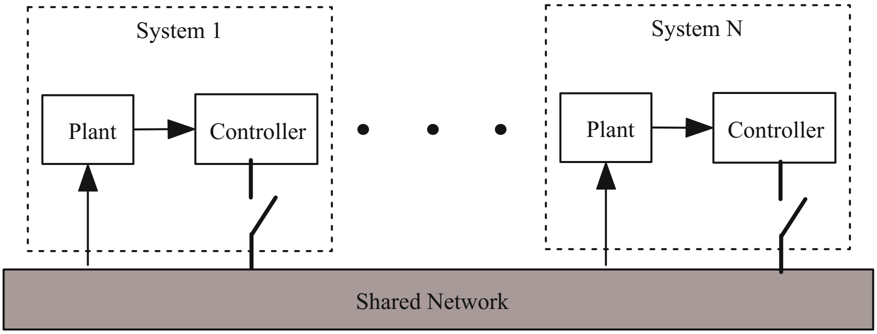


Fig. 12.1 Multiple networked control systems share the communication channel

$$\mathcal{S}_d^i : \begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) & (12.2a) \\ y_i(k) = C_i x_i(k) & (12.2b) \end{cases}$$

where $x_i(k) = x_i^c(kT_i)$, $u_i(k) = u_i^c(kT_i)$, $y_i(k) = y_i^c(kT_i)$, $A_i = e^{A_i^c T_i}$, and $B_i = \int_0^{T_i} e^{A_i^c s} ds B_i^c$.

Suppose that the backward channel delays of all the subsystems are random but bounded and the transmissions from the controllers to the actuators share a communication network with limited resource. The communication resource is limited in the sense that, at each time instant, only one controller can access the network for transmission. Therefore the forward channel delay for each subsystem depends on not only the time during which the data is transmitted over the network but the time taken for waiting for the permission of network access which is determined by the used scheduling algorithm, see Fig. 12.1.

Thus the problem here is not only to design a controller for each subsystem $\mathcal{S}_d^i$ but also to design the scheduling scheme for the network resource allocations for all the subsystems $(\mathcal{S}_d^i)_{1 \le i \le \mathcal{N}}$, in an environment of network-induced delay, data packet dropout and data packet disorder. To this end, a co-design approach is proposed with the integration of the packet-based control approach and the scheduling algorithm. In the following section, the packet-based controller for each subsystem is first determined, and two different scheduling algorithms, the existing static RM algorithm and a novel DFS algorithm, are then adopted to schedule the transmissions of FCSs, with the guarantee of the stability of all the subsystems.

12.2 Packet-Based Control for Subsystems

For each subsystem $(\mathcal{S}_d^i)_{1 \le i \le \mathcal{N}}$, exactly the same packet-based control approach proposed in Sect. 2.2 is applied, with the following objective function,

$$J^i_{k,\tau_{sc,k}} = \sum_{j=N_1^i}^{N_2^i} q_j(\hat{y}^i(k+j|k-\tau_{sc,k}) - \omega(k+j))^2 + \sum_{j=1}^{N_u^i} r_j \Delta u^2(k+j-1) \quad (12.3)$$

where the definitions of the parameters are referred to (3.6).

Following the same procedure as in Sect. 3.2.1.2, FCIS for system $\mathcal{S}_d^i$ is then obtained as

$$\Delta U(k|k-\tau_{sc,k}) = M_{\tau_{sc,k}}(\varpi_k - E_{\tau_{sc,k}}\bar{x}(k-\tau_{sc,k}))$$

where $\Delta U(k|k-\tau_{sc,k}), M_{\tau_{sc,k}}, \varpi_k$ and $E_{\tau_{sc,k}}$ can be similarly defined as in Sect. 3.2.1.2.

With this FCIS, the following FCS from k to $k+N_u-1$ is readily obtained as

$$U(k|k-\tau_{sc,k}) = Gu(k-\tau_{sc,k}-1) + H\Delta U(k|k-\tau_{sc,k}) \quad (12.4)$$

where $G = [I_m \cdots I_m]^T_{mN_u \times m}$, I_m is the identity matrix with rank m and

$$H = \begin{pmatrix} I_m & \cdots & I_m & 0 & \cdots & 0 \\ I_m & \cdots & I_m & I_m & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ I_m & \cdots & I_m & I_m & \cdots & I_m \end{pmatrix}_{mN_u \times m(N_u+\tau_{sc,k})}.$$

Thus, for each subsystem with the packet-based control approach and the aforementioned FCS, following the same procedure as in Sect. 2.3.1, a stability theory similar to Theorem 2.1 can then be obtained using switched system theory.

We now explore a little further on Theorem 2.1. As a matter of fact, Theorem 2.1 implies that system $\mathcal{S}_d^i$ with the packet-based control approach and the aforementioned FCS in (12.4), is stable under certain conditions if the round trip delay is less than a fixed value. In other words, given a linear system, the least upper bound, or the supremum of the round trip delay, under which the system is stable can be found from Theorem 2.1. We call this supremum of round trip delay that guarantees the stability of the system the "Stability-guaranteed Supremum of Round Trip Delay (SSRTD)", which is an inherent characteristic of a given system. The techniques such as the LMI tool-box, are useful to find the SSRTD for a given system. It is also necessary to point out that if other performance constraints besides stability are considered, a smaller supremum of round trip delay than SSRTD is needed. For convenience, denote the SSRTDs of the systems $(\mathcal{S}_d^i)_{1 \le i \le \mathcal{N}}$ by $\hat{D}_i > 0$, $1 \le i \le \mathcal{N}$.

Remark 12.1 Note that the notion of SSRTD here is similar to MADB (see Sect. 1.3.1) which has been used in a number of publications, see spsciteKim2003 for an overview. We prefer SSRTD to MADB in this thesis since the former can better express the particular requirement of the round trip delay in Theorem 2.1 for the stability of the system.

12.3 Scheduling

In this section, scheduling theory is applied to allocate the limited network resources for the transmission tasks of the PBNCSs that are derived from the subsystems $(\mathcal{S}_d^i)_{1\le i\le\mathcal{N}}$. The static, priority-based scheduling algorithm RM is applied to the set of subsystems under a private network environment first, and then the dynamic, feedback-based scheduling algorithm DFS is presented to extend the application to the public network.

When a scheduling algorithm is applied to schedule the transmission tasks of a set of PBNCSs, stability of the subsystems have to be guaranteed as a precondition. To ensure this, we define "Stable Schedulability" as follows.

Definition 12.1 (*Stable Schedulability*). A set of PBNCSs sharing the network resources is said to be stable schedulable by a scheduling algorithm if the transmissions of all the subsystems can be scheduled so that all the subsystems are stable.

12.3.1 Static Scheduling

In the static scheduling case, the network is assumed to be used only by the subsystems $(\mathcal{S}_d^i)_{1\le i\le\mathcal{N}}$, i.e. is private to the set of subsystems. In the analysis, the transmissions of the FCSs for the subsystem $(\mathcal{S}_d^i)_{1\le i\le\mathcal{N}}$ are regarded as real-time tasks in scheduling theory, which are defined by analyzing the formation of the network-induced delay. The RM algorithm is then adopted over these transmission tasks and the feasibility theorem is obtained as well.

12.3.1.1 The Transmission Tasks of the PBNCSs

As shown in spsciteLian2001b, the forward channel delay τ_{ca} is mainly composed of the following three parts.

1. The propagation delay, which is the time from when a packet is put onto the network till it successfully arrives at its destination. Since the network is private to the subsystems $(\mathcal{S}_d^i)_{1\le i\le\mathcal{N}}$, the propagation delay depends merely on the speed of signal transmission and the distance between the source and the destination, which are assumed to be fixed for all the subsystems. Therefore this delay is assumed to be known as a constant τ_{ca}^0 in the static scheduling case.
2. The frame time delay, which is the time for the source to place a packet on the network. Suppose that the size of the packet which contains the FCS is B_cN_u, where B_c is data size required for encoding a single step control signal as defined in Assumption 2.2, which can be assumed to be the same for all the subsystems. The frame time delay is then obtained as

$$e_i = \frac{B_c N_u^i}{B_N} \tag{12.5}$$

where B_N is the bandwidth of the network, N_u^i is the control horizon of subsystem $\mathcal{S}_d^i$.

It is natural to assume that all the subsystems use the same control horizon N_u, since the selection of N_u mainly depends on the round trip delay of the network and all the subsystems endure similar network-induced delays and data packet dropouts by sharing the network. Hence,

$$e_i = e = \frac{B_c N_u}{B_N}, i = 1, 2, ..., \mathcal{N} \tag{12.6}$$

The frame network-induced delay e serves as the execution time in the transmission tasks of the PBNCSs.

3. The waiting network-induced delay, is defined as the time a FCS has to wait for queuing and network availability before actually being sent. From earlier discussion, the SSRTD for system $\mathcal{S}_d^i$ is $\hat{D}_i$, therefore, to ensure the stability of all the subsystems, the waiting delay for each system should not be larger than

$$D_i = \hat{D}_i - \tau_{ca}^0 - \bar{\tau}_{sc} - e \tag{12.7}$$

Note that the upper bound of the backward channel delay $\bar{\tau}_{sc}$ is used since the RM algorithm assigns the priority of each task statically and the stability of the subsystems needs to be guaranteed under the worst case. It is only the stability of the system that we care about in the RM algorithm in this chapter, and therefore the transmission period of subsystem $\mathcal{S}_d^i$ needs to be no longer than D_i. As a result, the transmission period h_i of subsystem $\mathcal{S}_d^i$ is assumed to be equal to D_i in the static RM scheduling algorithm, i.e.

$$h_i = D_i, i = 1, 2, ..., \mathcal{N} \tag{12.8}$$

h_i is chosen by (12.8) so that the FCS of subsystem $\mathcal{S}_d^i$ is sent every h_i seconds no matter what the sampling period is or how fast the controller can generate FCSs.

Thus from the analysis of τ_{ca}, the transmission tasks of the set of PBNCSs can now be described as follows: All the tasks have the same execution time e, the deadline of each task equals its period h_i, and the first release time ϑ_i of task i is the time when subsystem $\mathcal{S}_d^i$ first operates. We denote the tasks by

$$\mathscr{T}_i = \mathscr{T}(\vartheta_i, e, h_i), i = 1, 2, \cdots, \mathcal{N} \tag{12.9}$$

12.3.1.2 Scheduling of PBNCSs by RM

RM is a widely used scheduling algorithm, where tasks with shorter periods have higher priorities. It is a fixed-priority assignment: priorities are assigned to tasks before execution and do not change over time. spsciteLiu1973 have shown that RM is superior to other fixed-priority assignments in the sense that no other fixed-priority algorithm can schedule a task set that cannot be scheduled by RM.

Consider the set of real-time transmission tasks $\mathscr{T}_i$, $1 \leq i \leq \mathcal{N}$ defined in (12.9). These tasks are periodic, independent, non-preemptive, and the period of each task equals its deadline. These characteristics are just what the operation of the RM algorithm needs. Therefore, the RM scheduling algorithm can be applied to schedule the set of transmission tasks in PBNCSs, by which the transmission with shorter deadlines (or periods) are assigned higher priorities, and thus the corresponding FCSs can be transmitted first if the network is idle, i.e.

$$\text{if } h_i < h_j, \text{ then } \Upsilon_i > \Upsilon_j, i, j = 1, 2, \cdots, \mathcal{N} \tag{12.10}$$

where Υ_i represents the priority of the transmission task of subsystem $\mathcal{S}_d^i$.

Theorem 12.1 *A set of $\mathcal{N}$ PBNCSs sharing the network resource in their forward channel (indexed by the increasing order of their transmission periods, i.e. $h_i \leq h_{i+1}, i = 1, 2, ...\mathcal{N} - 1$) are stable schedulable if for all $i = 1, ..., \mathcal{N}$*

$$\mathscr{U}(i) \leq f(i) \tag{12.11}$$

where $f(i) = i(2^{1/i} - 1)$ and

$$\mathscr{U}(i) = \begin{cases} e(\sum_{j=1}^{i} \frac{1}{h_j} + \frac{1}{h_i}) & i = 1, 2, ..., \mathcal{N} - 1 \quad (12.12a) \\ e(\sum_{j=1}^{i} \frac{1}{h_j}) & i = \mathcal{N} \quad (12.12b) \end{cases}$$

Proof From Theorem 16 in [153], a set of nonpreemptive periodic real-time tasks are schedulable if

$$\frac{e_1}{h_1} + \frac{e_2}{h_2} + \cdots + \frac{e_i}{h_i} + \frac{\bar{b}_{l,i}}{h_i} \leq i(2^{1/i} - 1) \tag{12.13}$$

where e_i is the frame time, h_i is the transmission period, each for the ith task, and $\bar{b}_{l,i}$ is task i's worst-case blocking time by the lower priority tasks, i.e.,

$$\bar{b}_{l,i} = \max_{j=i+1,...,\mathcal{N}} e_j. \tag{12.14}$$

As has been pointed out earlier, for the transmission tasks $\mathscr{T}_i$, $1 \le i \le \mathcal{N}$, (12.6) holds, and therefore $\bar{b}_{l,i} = e$, $i = 1, 2, ..., \mathcal{N} - 1$ and $\bar{b}_{l,\mathcal{N}} = 0$ from (12.14). Hence the theorem holds.

Corollary 12.1 *If 1) $h_{i+1} \le 2h_i$, $i = 1, 2, ...\mathcal{N} - 2$, and 2) $\frac{h_\mathcal{N}}{h_{\mathcal{N}-1}} \le \frac{\mathcal{N}-1}{\mathcal{N}} \frac{2^{\frac{1}{\mathcal{N}-1}}-1}{2^{\frac{1}{\mathcal{N}}}-1}$, then the set of tasks of PBNCSs is stable schedulable if*

$$e \sum_{i=1}^{\mathcal{N}} \frac{1}{h_i} \le \mathcal{N}(2^{1/\mathcal{N}} - 1) \tag{12.15}$$

Proof From 1) we obtain for $i = 1, 2, ..., \mathcal{N} - 2$ that

$$\mathscr{U}(i+1) - \mathscr{U}(i) = e(\frac{2}{h_{i+1}} - \frac{1}{h_i}) \ge 0$$

It is obvious that $\mathscr{U}(\mathcal{N}) \ge \mathscr{U}(\mathcal{N} - 1)$ from 2). Thus we obtain

$$\mathscr{U}(\mathcal{N}) = \max_{1 \le i \le \mathcal{N}} \mathscr{U}(i)$$

On the other hand, it is easy to show that function $f(\cdot)$ is nonincreasing, and therefore $\mathscr{U}(\mathcal{N}) \le f(\mathcal{N})$ implies $\mathscr{U}(i) \le f(i)$, $i = 1, 2, ..., \mathcal{N} - 1$, which completes the proof by Theorem 12.1.

Corollary 12.2 *If the transmission periods of all the subsystems are the same, i.e., $h_i = h$, $i = 1, 2, ..., \mathcal{N}$, then $(\mathcal{S}_i^d)_{1 \le i \le \mathcal{N}}$ are stable schedulable if*

$$\frac{e}{h} \le 2^{\frac{1}{\mathcal{N}}} - 1 \tag{12.16}$$

Proof It can be obtained directly from Corollary 12.1.

12.3.2 Dynamic Feedback Scheduling

In the static RM scheduling scheme presented earlier, the transmission periods h_i for all the subsystems are assigned a priori to ensure the stability of the subsystems and do not change any more. In the case of the network being shared only by $(\mathcal{S}_d^i)_{1 \le i \le \mathcal{N}}$, this method works though the performance of the subsystems may not be optimum because the network is not fully used. However, if the network is not private to these subsystems, i.e., there are other components occupying the network, it can not be assumed that the propagation delay is constant due to the change of the network loads. Based on this reality, DFS scheme is designed. In this scheme, a higher level feedback scheduler is proposed, which gets the information of the network utilization from the network and the control performances from all the subsystems as well, and

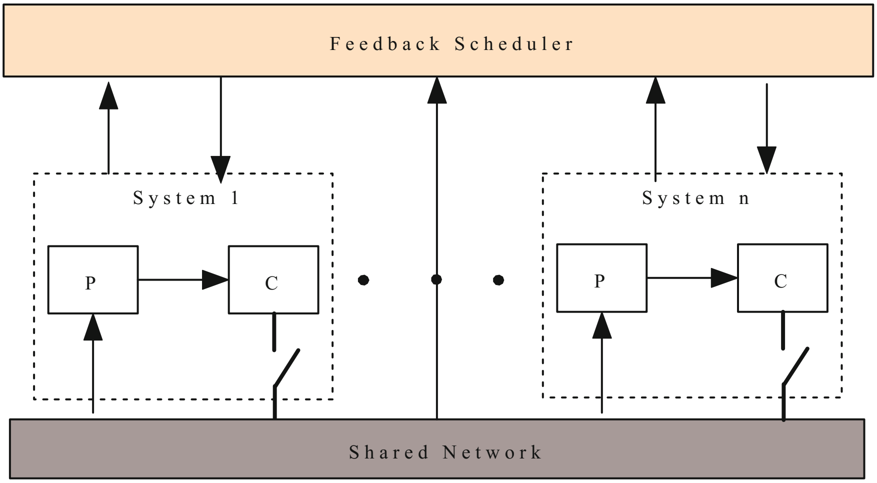


Fig. 12.2 Dynamic feedback scheduling of multiple systems

then regularly calculates and reassigns the transmission period for each subsystem. During the interval of two successive reassignments of periods, the RM algorithm still works. The framework of the DFS scheme is depicted in Fig. 12.2.

In order to implement DFS, such issues as the selection of the period of DFS, the measurement of the network utilization and the reassignment of the transmission periods of the PBNCSs, need to be dealt with first.

12.3.2.1 The Period of DFS

This period, noted by T_{DFS}, has to be chosen carefully. Generally, its value depends on the speed at which the condition of the network changes. A small T_{DFS} is needed if the network condition changes rapidly, while a larger one can still guarantee the performance of the system without overloading the network if the parameters of the network do not change much over a long time. However, in any case, T_{DFS} should be always not less than the transmission periods of all the subsystems, i.e. $T_{DFS} \geq \max_{i=1}^{\mathcal{N}} h_i$.

12.3.2.2 The Measurement of the Network Utilization

To obtain the utilization information of the network, a packet containing this information is sent to the feedback scheduler using the period of T_{DFS}. This information is mainly reflected by the propagation delays of the subsystems. The propagation time will increase to a certain extent with the increase of the network load. Another factor affecting the stability of the subsystems is the change of the backward channel delay. In order to take this factor into account and for simplicity, we assume that the upper bound of the backward channel delay during the kth period of DFS (denoted by $\bar{\tau}^i_{sc}(k)$ for subsystem $\mathcal{S}^i_d$) can be obtained from the network and the network does not change too much during this period thus enabling us to use $\bar{\tau}^i_{sc}(k)$ to estimate its value during the $(k+1)$th period. Then the deadline h_i of the task $\mathscr{T}_i$ will be recalculated by updating the propagation time τ^o_{ca} and the upper bound of the backward channel delay every T_{DFS} seconds as follows

$$D_i(k+1) = \hat{D}_i - \mu(\tau^0_{ca}(k) + \bar{\tau}^i_{sc}(k)) - e \tag{12.17}$$

where μ close to 1 is a smoothing factor satisfying

$$\mu(\tau^0_{ca}(k) + \bar{\tau}^i_{sc}(k)) \geq \tau^0_{ca}(k+1) + \bar{\tau}^i_{sc}(k+1), \forall k$$

12.3.2.3 The Reassignment of the Transmission Periods for All Subsystems

In order to obtain the control performance of the subsystems, an obvious idea is to use the predictive Quality of Performance (QoP) during the next DFS period. This QoP during the kth period of DFS can be defined for subsystem $\mathcal{S}^i_d$ as

$$\hat{\mathcal{P}}_i(k) = \sum_{j=\lceil kT_{DFS}/h_i \rceil}^{\lceil (k+1)T_{DFS}/h_i \rceil} (\hat{y}_i(j|j-\tau_{sc,j}) - \omega_i(j))^2 \tag{12.18}$$

The calculation of the new transmission periods for all the subsystems can then be modeled as an optimization problem $\mathscr{P}$ as follows:

$$\mathscr{P}: \begin{cases} \text{Select } h_i, i = 1, 2, \ldots \mathcal{N}, \text{ s.t.} \\ \min\limits_{h_i} \sum\limits_{i=1}^{\mathcal{N}} \hat{\mathcal{P}}_i, \\ \text{subject to} \\ \mathscr{U}(i) \leq f(i), i = 1, 2, \ldots, \mathcal{N}, \\ h_i \geq T_i, i = 1, 2, \ldots, \mathcal{N}. \end{cases}$$

where $\mathscr{U}(i)$ and $f(i)$ are defined in Theorem 12.1 and T_i is the sampling period for system $\mathcal{S}^i_d$ defined in (12.2).

In practice, the predictive outputs $\hat{y}(j|j-\tau_{sc,j})$ of the subsystems can be obtained using the open-loop prediction, whereas the online operation of the optimization problem $\mathscr{P}$ is not a simple one. Therefore, not the predictive QoP but the previous QoP, $\bar{J}_i(k)$, i for subsystem $\mathcal{S}^i_d$ and k for the kth period of DFS, is used to represent the performance of the system, which is defined as follows and can be easily obtained,

$$\bar{J}^i_{k,\tau_{sc,k}} = \sum_{J^i_{k,\tau_{sc,k}} \in \Pi_k} J^i_{k,\tau_{sc,k}} \tag{12.19}$$

where $J^i_{k,\tau_{sc,k}}$ is the objective function of subsystem $\mathcal{S}^i_d$ defined in (12.3), Π_k is the set of objective functions during the kth period of DFS, or from the $\lceil (k-1)T_{DFS}/h_i \rceil$th transmission period of subsystem $\mathcal{S}^i_d$ to the $\lceil kT_{DFS}/h_i \rceil$th.

Let the new transmission periods chosen in this way be

$$\frac{1}{h_i(k+1)} = \kappa(k+1)\frac{\bar{J}^i_{k,\tau_{sc,k}}}{\sum_{j=1}^{\mathcal{N}} \bar{J}^j_{k,\tau_{sc,k}}} = \kappa(k+1)\theta_i(k) \tag{12.20}$$

where κ is a proportion factor and can be chosen as follows to include the constraints of stable schedulability in Theorem 12.1,

$$\kappa(k+1) = \max_{i=1,\ldots,\mathcal{N}-1}\{\frac{f(i)}{e(\sum_{j=1}^{i}\theta_j(k)+\theta_i(k))}, \frac{f(\mathcal{N})}{e}\}$$

Considering the fact that the network load may change greatly between two periods of DFS, a smoothing factor $\rho(0 < \rho \leq 1)$ is introduced to avoid network overload. Also taking account of the fact that the transmission period h_i can never exceed D_i for the stability of the system, then the transmission periods are obtained as

$$h_i(k+1) = \min\{\frac{\rho}{\kappa(k+1)\theta_i(k)}, D_i(k+1)\} \tag{12.21}$$

The algorithm of DFS can then be summarized as follows.

Algorithm 12.1 Dynamic feedback scheduling

Initialization. $T_{DFS}, k = 1, t = 0, h_i = \hat{D}_i - e, 1 \leq i \leq \mathcal{N}$.
while $t \in [(k-1)T_{DFS}\ kT_{DFS}]$ **do**
  Calculate the FCSs $U_i(k|k-\tau_{sc,k}), 1 \leq i \leq \mathcal{N}$ using (12.4) for all the subsystems;
  Apply RM algorithm in Sect. 12.3.1 to determine the order of the transmission tasks of the PBNCSs;
  Transmit the FCS.
end while
if $t = kT_{DFS}$ and the network is in full use **then**
  Let $k = k + 1$
else
  The DFS module calculates the new transmission periods using (12.20) and reassigns the priorities for all the subsystems;
  Let $k = k + 1$.
end if

12.3.2.4 Stability of DFS

Theorem 12.2 $(\mathcal{S}_i^c)_{1\le i\le \mathcal{N}}$ *with the packet-based control approach are stable under DFS if the transmission tasks of the PBNCSs are always stable schedulable.*

Proof It is noticed that the use of DFS to the set of PBNCSs does not change the backward channel delay, since the DFS module is at the controller side, while it does change the forward channel delay by reassigning the transmission period $h_i, i = 1, 2, \cdots, \mathcal{N}$ for the subsystems. However, from (12.17) and (12.21), we obtain

$$h_i(k+1) \le D_i(k+1) \le \hat{D}_i - \tau_{ca}^0(k+1) - \bar{\tau}_{sc}^i(k+1) - e, \quad \forall k, i = 1, 2, \cdots, \mathcal{N} \tag{12.22}$$

which implies,

$$\sup_k \{h_i(k) + \tau_{ca}^0(k) + \bar{\tau}_{sc}^i(k) + e\} \le \hat{D}_i, \quad i = 1, 2, \cdots, \mathcal{N} \tag{12.23}$$

Note that the left side of (12.23) is the effective maximum of round trip delay for subsystem $\mathcal{S}_d^i$, which is always no more than SSRTD. Thus the theorem is valid by Theorem 12.1.

12.4 Numerical Examples

Three second order linear subsystems in $(\mathcal{S}_i^c)_{1\le i\le \mathcal{N}}$ are considered in the examples, whose system matrices are as follows

$$A_1^c = \begin{pmatrix} -11.1572 & -106.0132 \\ -110.3637 & -5.2680 \end{pmatrix}, A_2^c = \begin{pmatrix} -23.7783 & -48.9313 \\ -107.2959 & -34.0550 \end{pmatrix},$$

$$A_3^c = \begin{pmatrix} -91.6291 & -160.9438 \\ -69.3147 & -35.6675 \end{pmatrix}, B_1^c = \begin{pmatrix} -0.1295 \\ 2.6890 \end{pmatrix}, B_2^c = \begin{pmatrix} 4.3801 \\ 2.3278 \end{pmatrix},$$

$$B_3^c = \begin{pmatrix} 9.1471 \\ 4.0444 \end{pmatrix}, C_1^c = C_2^c = C_3^c = 1,$$

The sampling periods are set as $T_1 = 0.02$ s, $T_2 = 0.015$ s, $T_3 = 0.01$ s, respectively. The corresponding discrete-time subsystems $(\mathcal{S}_i^d)_{1\le i\le \mathcal{N}}$ can then be obtained with the following system matrices

$$A_1 = \begin{pmatrix} 0.8 & 0.12 \\ 0.11 & 0.9 \end{pmatrix}, A_2 = \begin{pmatrix} 0.7 & 0.48 \\ 0.2 & 0.6 \end{pmatrix}, A_3 = \begin{pmatrix} 0.4 & 0.2 \\ 0.5 & 0.7 \end{pmatrix},$$

$$B_1 = \begin{pmatrix} 0.02 \\ 0.05 \end{pmatrix}, B_2 = \begin{pmatrix} 0.08 \\ 0.06 \end{pmatrix}, B_3 = \begin{pmatrix} 0.08 \\ 0.1 \end{pmatrix},$$

$$C_1 = C_2 = C_3 = 1.$$

For the simplicity of simulation, assume for all the three subsystems that the set point $\omega = 0$, weighting factors $W_1 = I$, $W_2 = I$, and the state vector can be obtained directly. Other parameters of the simulation are shown in Table 12.1. A Gaussian white noise with standard deviation 0.1 is also introduced as the disturbance of the state (Tables 12.1 and 12.2).

Example 12.1 (RM algorithm)
Using the LMI toolbox in Matlab, the SSRTD $\hat{D}_i$ for the subsystem $(\mathcal{S}_i^d)_{1\le i\le \mathcal{N}}$ can be obtained by Theorem 2.1, thus enabling the transmission periods h_i to be calculated according to (12.7) and (12.8), as shown in Table 2.

Note that the execution time of each job is $e = 0.008$ s. The value of the utilization function $\mathscr{U}(\cdot)$ can then be obtained as $\mathscr{U}(i) = 0.2, 0.3133, 0.34, i = 1, 2, 3$ while $f(i) = 1, 0.8284, 0.7798, i = 1, 2, 3$ respectively. It is readily seen that in this case (12.11) holds and by Theorem 12.1, the set of NCSs is stable schedulable under RM.

Table 12.1 Simulation parameters

	System 1	System 2	System 3
T	0.02	0.015	0.01
ϑ	0	0.05	0.06
x_0	$[-1\ \ -1]^T$	$[-1\ \ -1]^T$	$[-1\ \ -1]^T$
P_1	$[1\ 30\ 20\ 0\ I\ I]$		
P_2	$[0.008\ 0.002\ \ 0.01\ \ 10]$		

T is the sampling period.
ϑ is the first release time.
$x_0 = [x_{01}\ x_{02}]^T$ is the initial state.
$P_1 = [N_1\ N_2\ N_u\ \omega\ W_1\ W_2]$ is the predictive parameters
$P_2 = [e\ \tau_{ca}^0\ \bar{\tau}_{sc}\ T_{sim}]$, T_{sim} is the simulation time.

Table 12.2 SSRTD and Transmission periods

	System 1	System 2	System 3
$\hat{D}_i$	5	7	8
h_i	4	5	6

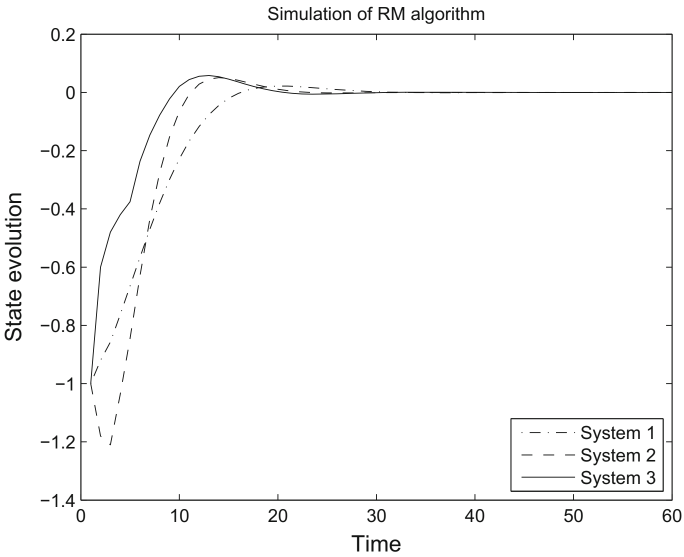


Fig. 12.3 Example 12.1. State evolution using RM algorithm. Only the first state is illustrated

The state evolution of the first state of the three subsystems under RM is shown in Fig. 12.3.

Example 12.2 (DFS algorithm)

It is noted that the SSRTD obtained in Theorem 2.1 is conservative. In the simulation of DFS, the deadlines of the three subsystems are set to be 8, 10 and 12 steps respectively, and the propagation delays of the subsystems in the forward channel are set to be randomly changing under the constraint that the real round trip delay are no more than the new SSRTD, in order to simulate the changes of the network loads. All the other parameters remain the same as in RM algorithm.

The simulation result (Fig. 12.4) shows that the subsystems are still stable under this larger SSRTD and with fluctuating propagation delays.

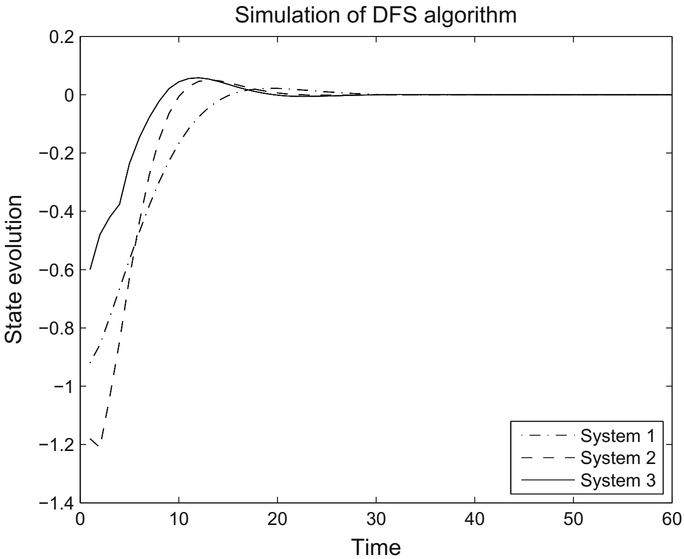


Fig. 12.4 Example 12.2. State evolution using DFS algorithm. Only the first state is illustrated

12.5 Summary

Different from the previous chapters where only one NCS occupies the network resources, in this chapter a situation where multiple NCSs share the network resources to transmit the FCS was considered. The packet-based control approach was still applied to the subsystems, and scheduling theory was also considered to schedule the network resources to guarantee the stability of all the subsystems. Two scheduling algorithms, the existing RM algorithm and a novel designed DFS algorithm were discussed, the validity of which were also illustrated by numerical examples.

References

1. G.C. Walsh, O. Beldiman, L. Bushnell, Asymptotic behavior of networked control systems, in Proceedings 1999 IEEE International Conference on Control Applications, pp. 1448–1453 (1999)
2. J. Araujo, M. Mazo, A. Anta, P. Tabuada, K. Johansson, System architectures, protocols and algorithms for aperiodic wireless control systems. IEEE Trans. Ind. Inform. **10**(1), 175–184 (2014)
3. A. Banos, F. Perez, J. Cervera, Network-based reset control systems with time-varying delays. IEEE Trans. Ind. Electron. **10**(1), 514–522 (2014)
4. F. Farokhi, K. Johansson, Stochastic sensor scheduling for networked control systems. IEEE Trans. Autom. Control **59**(5), 1147–1162 (2014)
5. G.C. Goodwin, M.E. Cea Garrido, A. Feuer, D.Q. Mayne, On the use of one bit quantizers in networked control. Automatica **50**(4), 1122–1127 (2014)
6. M.A. Kafi, D. Djenouri, J. Ben-Othman, N. Badache, Congestion control protocols in wireless sensor networks: A survey. IEEE Commun. Surv. Tut. **16**(3), 1369–1390 (2014)
7. Y. Halevi, A. Ray, Integrated communication and control systems: Part I- analysis. J. Dyn. Syst. Meas. Control-Trans. ASME **110**(4), 367–373 (1988)
8. A.E. Gil, Stability analysis of network-based cooperative resource allocation strategies, PhD thesis, Ohio State University (2003)
9. X. Meng, J. Lam, H. Gao, Network-based H1 control for stochastic systems. Int. J. Robust Nonlinear Control **49**(3), 295–312 (2009)
10. A.E. Gil, K.M. Passino, Stability analysis of network-based cooperative resource allocation strategies. Automatica **42**(2), 245–250 (2006)
11. N.J. Ploplys, P.A. Kawka, A.G. Alleyne, Closed-loop control over wireless networks. IEEE Control Syst. Mag. **24**(3), 58–71 (2004)
12. S. Tatikonda, S. Mitter, Control over noisy channels. IEEE Trans. Autom. Control, **49**(7), 1196–1201 (2004). 0018–9286
13. Y. Tipsuwan, C. Mo-Yuen, On the gain scheduling for networked pi controller over ip network. IEEE/ASME Trans. Mechatronics **9**(3), 491–498 (2004), 1083–4435
14. R. Gupta, M.-Y. Chow, Networked control system: Overview and research trends. IEEE Trans. Ind. Electron. **57**(7), 2527–2535 (2010)
15. J.P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems. IEEE Proc. **95**(1), 138–162 (2007)
16. M. Kranz, P. Holleis, A. Schmidt, Embedded interaction: Interacting with the Internet of Things. IEEE Internet Comput. **14**(2), 46–53 (2010)
17. F. Mattern, C. Floerkemeier, From the internet of computers to the internet of things. in From Active Data Management to Event- Based Systems and More, ser. Lecture Notes in Computer

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6

Science, K. Sachs, I. Petrov, and P. Guerrero, Eds., vol. 6462, Springer Berlin Heidelberg, pp. 242–259 (2010)
18. J. Davis, T. Edgar, J. Porter, J. Bernaden, M. Sarli, Smart manufacturing, manufacturing intelligence and demand-dynamic performance. Comput. Chem. Eng. **47**, 145–156 (2012)
19. E. Commission, Intelligent transport system (2015), Available: http://ec.europa.eu/transport/themes/its/index_en.htm
20. IBM. (2015). A vision of smarter cities, Available: http://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03227usen/GBE03227USEN.PDF
21. B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, A survey of research on cloud robotics and automation. IEEE Trans. Autom. Sci. Eng. **99**, 1–12 (2015)
22. L. Lao, M. Ellis, P.D. Christofides, Smart manufacturing: Handling preventive actuator maintenance and economics using model predictive control. AICHE J. **60**(6), 2179–2196 (2014)
23. P. Antsaklis, J. Baillieul, Special issue on technology of networked control systems. IEEE Proc. **95**(1), 5–8 (2007)
24. M. Mahmoud, M. Sabih, Experimental investigations for distributed networked control systems. IEEE Syst. J. **8**(3), 717–725 (2014)
25. B. Xue, N. Li, S. Li, Q. Zhu, Moving horizon scheduling for networked control systems with communication constraints. IEEE Trans. Ind. Electron. **60**(8), 3318–3327 (2013)
26. R. Postoyan, D. Nesic, A framework for the observer design for networked control systems. IEEE Trans. Autom. Control **57**(5), 1309–1314 (2012)
27. J.-N. Li, Y.-J. Pan, H.-Y. Su, C.-L. Wen, Stochastic reliable control of a class of networked control systems with actuator faults and input saturation. Int. J. Control Autom. Syst. **12**(3), 564–571 (2014)
28. M. Pajic, S. Sundaram, G. Pappas, R. Mangharam, The wireless control network: A new approach for control over networks. IEEE Trans. Autom. Control **56**(10), 2305–2318 (2011)
29. N.S. Mkondweni, R. Tzoneva, Labns2 as co-simulation, coemulation, and real-time control toolkit for investigation of network induced time delays and packet loss in networked control systems. Int. J. Innov. Comp. Inf. Control **10**(1), 1–18 (2014)
30. K.-Y. You, L.-H. Xie, Survey of recent progress in networked control systems. Acta Automatica Sinica **39**(2), 101–117 (2013)
31. L. Zhang, H. Gao, O. Kaynak, Network-induced constraints in networked control systems: A survey. IEEE Trans. Ind. Inform. **9**(1), 403–416 (2013)
32. Y. Ge, L. Tian, Z. Liu, Survey on the stability of netowked control systems. J. Contr. Theory Appl. **5**(4), 374–379 (2007)
33. Y. Tipsuwan, M.-Y. Chow, Control methodologies in networked control systems. Control Eng. Practice **11**(10), 1099–1111 (2003)
34. W. Stallings, *Data and Computer Communications*, 6th edn. (Prentice Hall, Englewood Cliffs, NJ, 2000)
35. W. Zhang, M.S. Branicky, S.M. Phillips, Stability of networked control systems. IEEE Control Syst. Mag. **21**(1), 84–99 (2001)
36. Y.-B. Zhao, G.-P. Liu, D. Rees, Integrated predictive control and scheduling co-design for networked control systems. IET Control Theor. Appl. **2**(1), 7–15 (2008)
37. Y.-B. Zhao, G.-P. Liu, D. Rees, Design of a packet-based control framework for networked control systems. IEEE Trans. Control Syst. Technol. **17**(4), 859–865 (2009)
38. W. Hu, G. Liu, D. Rees, Event-driven networked predictive control. IEEE Trans. Ind. Electron. **54**(3), 1603–1613 (2007)
39. A. Pawlowski, A. Cervin, J.L. Guzmán, M. Berenguel, Generalized predictive control with actuator deadband for event-based approaches. IEEE Trans. Ind. Inform. **10**(1), 523–537 (2014)
40. D. Yue, E. Tian, Q.-L. Han, A delay system method for designing event-triggered controllers of networked control systems. IEEE Trans. Autom. Control **58**(2), 475–481 (2013)
41. H. Xu, S. Jagannathan, A cross layer approach to the novel distributed scheduling protocol and event-triggered controller design for cyber physical systems, in 37th Annual IEEE Conference on Local Computer Networks, Clearwater, Florida, USA, pp. 232–235 (2012)

42. J. Baillieul, P.J. Antsaklis, Control and communcation challenges in networked real-time systems. IEEE Proc. **95**(1), 9–28 (2007)
43. R.M. Murray, Ed., Control in An Information-Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems (2003)
44. R.M. Murray, K.J. Astrom, S.P. Boyd, R.W. Brockett, G. Stein, Future directions in control in an information-rich world. IEEE Control Syst. Mag. **23**(2), 20–33 (2003)
45. J. Nilsson, B. Bernhardsson, B. Wittenmark, Stochastic analysis and control of real-time systems with random time delays. Automatica **34**(1), 57–64 (1998)
46. L.-S. Hu, T. Bai, P. Shi, Z. Wu, Sampled-data control of networked linear control systems. Automatica **43**(5), 903–911 (2007)
47. Y. Fan, Z.-P. Jiang, H. Zhang, Network flow control under capacity constraints: A case study. Syst. Control Lett. **55**(8), 681–688 (2006)
48. F. Yang, Z. Wang, Y.S. Hung, M. Gani, H1 control for networked systems with random communication delays. IEEE Trans. Autom. Control **51**(3), 511–518 (2006)
49. F.-L. Lian, Analysis, design, modeling, and control of networked control systems. PhD thesis, University of Michigan, USA (2001)
50. F.-L. Lian, J.R. Moyne, D.M. Tilbury, Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. IEEE Control Syst. Mag. **21**(1), 66–83 (2001)
51. S. Hu, Q. Zhu, Stochastic optimal control and analysis of stability of networked control systems with long delay. Automatica **39**(11), 1877–1884 (2003)
52. F.-L. Lian, J. Moyne, D. Tilbury, Modelling and optimal controller design of networked control systems with multiple delays. Int. J. Control **76**(6), 591–606 (2003)
53. L.A. Montestruque, P.J. Antsaklis, On the model-based control of networked systems. Automatica **39**(10), 1837–1843 (2003)
54. A. Sala, Computer control under time-varying sampling period: An LMI gridding approach. Automatica **41**(12), 2077–2082 (2005)
55. D. Yue, Q.-L. Han, Delayed feedback control of uncertain systems with time-varying input delay. Automatica **41**(2), 233–240 (2005)
56. L. Zhang, Y. Shi, T. Chen, B. Huang, A new method for stabilization of networked control systems with random delays. IEEE Trans. Autom. Control **50**(8), 1177–1181 (2005)
57. X. Fan, M. Arcak, Delay robustness of a class of nonlinear systems and applications to communication networks. IEEE Trans. Autom. Control **51**(1), 139–144 (2006)
58. X. Liu, J. Shen, Stability theory of hybrid dynamical systems with time delay. IEEE Trans. Autom. Control **51**(4), 620–625 (2006)
59. P.L. Tang, C. de Silva, Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control. IEEE Trans. Control Syst. Technol. **14**(4), 707–718 (2006)
60. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, S.S. Sastry, Foundations of control and estimation over lossy networks. IEEE Proc. **95**(1), 163–187 (2007)
61. B. Azimi-Sadjadi, Stability of networked control systems in the presence of packet. in Proceedings 42nd IEEE Conference Decision and Control, vol. 1, pp. 676–681 (2003)
62. O.C. Imer, S. Yuksel, T. Basar, Optimal control of LTI systems over unreliable communication links. Automatica **42**(9), 1429–1439 (2006)
63. S. Hu, W.-Y. Yan, Stability robustness of networked control systems with respect to packet loss. Automatica **43**(7), 1243–1248 (2007)
64. J. Xiong, J. Lam, Stabilization of linear systems over networks with bounded packet loss. Automatica **43**(1), 80–87 (2007)
65. S. Sun, Linear optimal state and input estimators for networked control systems with multiple packet dropouts. Int. J. Innov. Comp. Inf. Control **8**(0(B)), 7289–7305 (2012)
66. Y.-L. Wang, G.-H. Yang, H1 control of networked control systems with time delay and packet disordering. IET Control Theory Appl. **1**(5), 1344–1354 (2007)
67. Y.-B. Zhao, J. Kim, G.-P. Liu, D. Rees, Compensation and stochastic modeling of discrete-time networked control systems with data packet disorder. Int. J. Control Autom. Syst. **10**(5), 1055–1063 (2012)

68. Y.-B. Zhao, G.-P. Liu, D. Rees, Actively compensating for data packet disorder in networked control systems. IEEE Trans. Circuits Syst. II-Express Briefs **57**(11), 913–917 (2010)
69. H. Zhang, G. Hui, Y. Wang, Stabilization of networked control systems with piecewise constant generalized sampled-data hold function. Int. J. Innov. Comp. Inf. Control **9**(3), 1159–1170 (2013)
70. M. Ljesnjanin, D.E. Quevedo, D. Nesić, Packetized mpc with dynamic scheduling constraints and bounded packet dropouts. Automatica **50**(3), 784–797 (2014)
71. S. Longo, G. Herrmann, P. Barber, Robust scheduling of sampleddata networked control systems. IEEE Trans. Control Syst. Technol. **20**(6), 1613–1621 (2012)
72. N. Elia, S.K. Mitter, Stabilization of linear systems with limited information. IEEE Trans. Autom. Control **46**(9), 1384–1400 (2001), 0018–9286
73. D. Liberzon, Quantization, time delays, and nonlinear stabilization. IEEE Trans. Autom. Control **51**(7), 1190–1195 (2006)
74. C. Peng, Y.-C. Tian, Networked H1 control of linear systems with state quantization. Inf. Sci. **177**(24), 5763–5774 (2007)
75. E. Tian, D. Yue, X. Zhao, Quantised control design for networked control systems. IET Control Theory Appl. **1**(6), 1693–1699 (2007)
76. R. Lu, H. Li, A. Xue, J. Zheng, Q. She, Quantized H1 filtering for different communication channels. Circuits Syst. Signal Process. **31**(2), 501–519 (2012)
77. M. Cea, G. Goodwin, Stabilization of systems over bit-rateconstrained networked control architectures. IEEE Trans. Ind. Inform. **9**(1), 357–364 (2013)
78. A. Chamaken, L. Litz, Joint design of control and communication in wireless networked control systems: A case study. Am. Control Conf. (ACC) **2010**, 1835–1840 (2010)
79. H. Lee, D.C. Cox, Admission and congestion control for largescale wireless mesh access networks. in Wireless Communications and Networking Conference (WCNC), 2010 IEEE, 1–6 (2010)
80. D. Hristu-Varsakelis, Stabilization of networked control systems with access constraints and delays. in Proceedings 45th IEEE Conference Decision and Control (2006)
81. J.-P. Richard, Time-delay systems: An overview of some recent advances and open problems. Automatica **39**(10), 1667–1694 (2003)
82. L. Zhang, Access scheduling and controller design in networked control systems. PhD thesis, University of Maryland, College Park, USA (2005)
83. J. Colandairaj, G. Irwin, W. Scanlon, Wireless networked control systems with QoS-based sampling. IET Control Theory Appl. **1**(1), 430–438 (2007)
84. G.P. Liu, Y. Xia, D. Rees, W. Hu, Design and stability criteria of networked predictive control systems with random network delay in the feedback channel. IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev. **37**(2), 173–184 (2007)
85. F. Mazenc, S.-I. Niculescu, Lyapunov stability analysis for nonlinear delay systems. Syst. Control Lett. **42**(4), 245–251 (2001)
86. J.K.C. So, Delay modeling and controller design for networked control systems. PhD thesis, University of Toronto, Canada (2003)
87. Z.-H. Guan, W.-H. Chen, J. Xu, Delay-dependent stability and stabilizability of uncertain jump bilinear stochastic systems with mode-dependent time-delays. Int. J. Syst. Sci. **36**(5), 275–285 (2005)
88. D. Yue, Q.-L. Han, C. Peng, State feedback controller design of networked control systems. IEEE Trans. Circuits Syst. II-Express Briefs **51**(11), 640–644 (2004)
89. D.-S. Kim, Y.S. Lee, W.H. Kwon, H.S. Park, Maximum allowable delay bounds of networked control systems. Control Eng. Practice **11**(11), 1301–1313 (2003)
90. P. Seiler, R. Sengupta, An H1 approach to networked control. IEEE Trans. Autom. Control **50**(3), 356–364 (2005)
91. M. Basin, J. Rodriguez-Gonzalez, Optimal control for linear systems with multiple time delays in control input. IEEE Trans. Autom. Control **51**(1), 91–97 (2006)
92. D. Georgiev, D.M. Tilbury, Packet-based control: The H2 optimal solution. Automatica **42**(1), 137–144 (2006)

93. M. Sahebsara, T. Chen, S.L. Shah, Optimal H1 filtering in networked control systems with multiple packet dropouts. Syst. Control Lett. **57**(9), 696–702 (2008)
94. G.C. Goodwin, H. Haimovich, D.E. Quevedo, J.S. Welsh, A moving horizon approach to networked control system design. IEEE Trans. Autom. Control **49**(9), 1427–1445 (2004)
95. G.P. Liu, J.X. Mu, D. Rees, S.C. Chai, Design and stability analysis of networked control systems with random communication time delay using the modified MPC. Int. J. Control **79**(4), 288–297 (2006)
96. Y.-B. Zhao, G.-P. Liu, D. Rees, Networked predictive control systems based on Hammerstein model. IEEE Trans. Circuits Syst. II-Express Briefs **55**(5), 469–473 (2008)
97. –, A predictive control based approach to networked Hammerstein systems: Design and stability analysis. IEEE Trans. Syst. Man Cybern. Part B-Cybern. **38**(3), 700–708 (2008)
98. –, A predictive control based approach to networked Wiener systems. Int. J. Innov. Comp. Inf. Control **4**(11), 2793–2802 (2008)
99. Z.-H. Guan, D.J. Hill, X. Shen, On hybrid impulsive and switching systems and application to nonlinear control. IEEE Trans. Autom. Control **50**(7), 1058–1062 (2005)
100. P.A. Kawka, A.G. Alleyne, Stability and feedback control of wireless networked systems. in Proceedings American Control Conference, vol. 4. Portland, OR, USA **2005**, 2953–2959 (2005)
101. M. Yu, L. Wang, G. Xie, A switched system approach to stabilization f networked control systems **4**(1), 86–95 (2006)
102. Y. Xia, G.P. Liu, P. Shi, D. Rees, E.J.C. Thomas, New stability and stabilization conditions for systems with time-delay. Int. J. Syst. Sci. **38**(1), 17–24 (2007)
103. X.-M. Sun, D. Wu, G.-P. Liu, W. Wang, Input-to-state stability for networked predictive control with random delays in both feedback and forward channels. IEEE Trans. Ind. Electron. **61**(7), 3519–3526 (2014)
104. M.S. Branicky, V. Liberatore, S.M. Phillips, Networked control system co-simulation for co-design. in Proceedings 2003 American Control Conference **4**, 3341–3346 (2003)
105. J.R. Hartman, Networked control system co-simulation for co-design: Theory and experiments. PhD thesis, Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, USA (2004)
106. M.E.M.B. Gaid, A. Cela, Y. Hamam, Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system. IEEE Trans. Control Syst. Technol. **14**(4), 776–787 (2006)
107. G.P. Liu, Y. Xia, D. Rees, W. Hu, Networked predictive control of systems with random network delays in both forward and feedback channels. IEEE Trans. Ind. Electron. **54**(3), 1282–1297 (2007)
108. G.C. Goodwin, D.E. Quevedo, E.I. Silva, Architectures and coder design for networked control systems. Automatica **44**(1), 248–257 (2008)
109. Y.-B. Zhao, G.-P. Liu, D. Rees, Packet-based deadband control for Internet-based networked control systems. IEEE Trans. Control Syst. Technol. **18**(5), 1057–1067 (2010)
110. Y.-B. Zhao, J. Kim, G.-P. Liu, Error bounded sensing for packetbased networked control systems. IEEE Trans. Ind. Electron. **58**(5), 1980–1989 (2011)
111. D. Nesic, A.R. Teel, Input-to state stability of networked control systems. Automatica **40**(12), 2121–2128 (2004)
112. M. Tabbara, D. Nesić, A. Teel, Stability of wireless and wireline networked control systems. IEEE Trans. Autom. Control **52**(9), 1615–1630 (2007)
113. D. Hristu-Varsakelis, P.R. Kumar, Interrupt-based feedback control over a shared communication medium. in Proceedings 41st IEEE Conference Decision & Control, vol. 3, pp. 3223–3228 (2002)
114. R.W. Brochett, Stabilization of motor networks. in Proceedings 34th IEEE Conference Decision and Control, vol. 2, pp. 1484–1488 (1995)
115. M.S. Branicky, S.M. Phillips, W. Zhang, Scheduling and feedback co-design for networked control systems. in Proceedings 41st IEEE Conference Decision and Control, vol. 2, Las Vegas, USA, pp. 1211–1217, Dec. 2002

116. J. Wu, T. Chen, Design of netowked control systems with packet dropouts. IEEE Trans. Autom. Control **52**(7), 1314–1319 (2007)
117. Y. He, Delay-dependent robust stability and stabilization based on free-weighting matrices (in chinese). PhD thesis, Institute of Information Science and Engineering, Central South University, Changsha, China (2004)
118. W. Hu, Design of networked control systems and global web-based control laboratory. PhD thesis, Faculty of Advanced Technology, University of Glamorgan, Pontypridd, U.K (2008)
119. D.K. Rollins, N. Bhandari, A.M. Bassily, G.M. Colver, S. Chin, A continuous-time nonlinear dynamic predictive modeling method for Hammerstein processes. Industrial Eng. Chem. Res. **42**(4), 860–872 (2003)
120. B. Ding, Y. Xi, A two-step predictive control design for input saturated Hammerstein systems. Int. J. Robust Nonlinear Control **16**(7), 353 (2006)
121. J. Vörös, Identification of Hammerstein systems with time-varying piecewise-linear characteristics. IEEE Trans. Circuits Syst. II-Express Briefs **52**(12), 865–869 (2005)
122. H.H.J. Bloemen, B. Tjjvd, H.B. Verbruggen, Model-based predictive control for Hammerstein-Wiener systems. Int. J. Control **74**(5), 482–495 (2001)
123. K.P. Fruzzetti, A. Palazoglu, K.A. McDonald, Nonlinear model predictive control using Hammerstein models. J. Process Control **7**(1), 31–41 (1997)
124. B. Kouvaritakis, M. Cannon, Nonlinear Predictive Control: Theory and Practice, ser. IEEE control engineering series. London: Institution of Electrical Engineers (2001)
125. E.F. Camacho, C. Bordons, *Model Predictive Control*, 2nd edn. (Springer, London, 2004)
126. B. Ding, S.Y. Li, Y. Xi, Stability analysis of generalized predictive control with input nonlinearity based on Popov theorem **29**(4), 582–588 (2003)
127. G. Tao, P. Kokotovic, *Adaptive Control of Systems with Actuator and Sensor Nonlinearities* (Wiley, New York, 1996)
128. W.J. Rugh, Nonlinear System Theory, The Volterra & Wiener Approach (1981)
129. H.H.J. Bloemen, C.T. Chou, T.J.J. van den Boom, V. Verdult, M. Verhaegen, T.C. Backx, Wiener model identification and predictive control for dual composition control of a distillation column. J. Process Control **11**(6), 601–620 (2001)
130. H. Al-Duwaish, W. Naeem, Nonlinear model predictive control of hammerstein and wiener models using genetic algorithms. in Proceedings of the 2001 IEEE International Conference on Control Applications, 2001 (CCA'01), pp. 465–469 (2001)
131. A.L. Cervantes, O.E. Agamennoni, J.L. Figueroa, A nonlinear model predictive control system based on wiener piecewise linear models. J. Process Control **13**(7), 655–666 (2003)
132. D. Liberzon, Switching in Systems and Control. Birkhauser (2003)
133. H. Gao, T. Chen, New results on stability of discrete-time systems with time-varying state delay. IEEE Trans. Autom. Control **52**(2), 328–334 (2007)
134. X.-M. Sun, W. Wang, Integral input-to-state stability for hybrid delayed systems with unstable continuous dynamics. Automatica **48**(9), 2359–2364 (2012)
135. Y. Moon, P. Park, W. Kwon, Y. Lee, Delay-dependent robust stabilization of uncertain state-delayed systems. Int. J. Control **74**(14), 1447–1455 (2001)
136. P. Billingsley, Probability and Measure, 3rd. Wiley (1995)
137. L.E. Ghaoui, F. Oustry, M. AitRami, A cone complementarity linearization algorithm for static output-feedback and related problems. IEEE Trans. Autom. Control **42**(8), 1171–1176 (1997)
138. L. Zhang, P. Shi, M. Basin, Robust stability and stabilisation of uncertain switched linear discrete time-delay systems. IET Control Theory Appl. **2**(7), 606–614 (2008)
139. H. Gao, T. Chen, J. Lam, A new delay system approach to network-based control. Automatica **44**(1), 39–52 (2008)
140. J. Xiong, J. Lam, Stabilization of networked control systems with a logic ZOH. IEEE Trans. Autom. Control **54**(2), 358–363 (2009)
141. L.A. Montestruque, Model-based networked control systems, PhD thesis, University of Notre Dame (2004)

142. L.A. Montestruque, P.J. Antsaklis, Stability of model-based networked control systems with time-varying transmission times. IEEE Trans. Autom. Control **49**(9), 1562–1572 (2004)
143. Y.-B. Zhao, G.-P. Liu, D. Rees, Improved predictive control approach to networked control systems. IET Control Theory Appl. **2**(8), 675–681 (2008)
144. D. Yue, Q.-L. Han, J. Lam, Network-based robust H1 control of systems with uncertainty. Automatica **41**(6), 999–1007 (2005)
145. X. Jiang, Q.-L. Han, S. Liu, A. Xue, A new H1 stabilization criterion for netowked control systems. IEEE Trans. Autom. Control **53**(4), 1025–1032 (2008)
146. Y. He, Q. Wang, C. Lin, M. Wu, Delay-range-dependent stability for systems with time-varying delay. Automatica **43**(2), 371–376 (2007)
147. H. Ishii, H1 control with limited communication and message losses. Syst. Control Lett. **57**(4), 322–331 (2008)
148. I.G. Polushin, P.X. Liu, C.-H. Lung, On the model-based approach to nonlinear networked control systems. Automatica **44**, 2409–2414 (2008)
149. L.A. Montestruque, P.J. Antsaklis, Static and dynamic quantization in model-based networked control systems. Int. J. Control **80**(1), 87–101 (2007)
150. Y. He, M. Wu, J.-H. She, G.-P. Liu, Delay-dependent robust stability criteria for uncertain neutral systems with mixed delays. Syst. Control Lett. **51**(1), 57–65 (2004)
151. X. He, Z. Wang, D. Zhou, Robust H1 filtering for networked systems with multiple state delays. Int. J. Control **80**(8), 1217–1232 (2007)
152. C.L. Liu, J.W. Layland, Scheduling algorithm for multiprogramming in a hard-real-time environment. J. ACM **20**(1), 46–61 (1973)
153. L. Sha, R. Rajkumar, J. Lehoczky, Priority inheritance protocols: An approach to real-time synchronization. IEEE Trans. Comput. **39**(9), 1175–1185 (1990)

Index

C
Control action selector, 18
Controller transmission rule, 130
Control over (through) networks, 1
Cyclic service networks, 5

D
Data packet disorder, 6, 118
Data packet dropout, 5
Delay-dependent control laws, 101
Delay-independent control laws, 101
Direct structure NCSs, 2
Dynamic feedback scheduling, 167

E
Event driven, 4

F
Forward control increment sequence, 35
Forward control sequence, 19

G
Generalized predictive control, 25

H
Hammerstein systems, 33
Hierarchical structure NCSs, 2

I
Integrated Communication and Control Networks, 1
Internet-based test rig, 29

M
Multi-packet transmission, 7

N
Network access constraint, 9
Network-based control systems, 1
Networked control systems, 1
Network-induced delay, 5

P
Popov criterion, 41
Priority order networks, 5

Q
Quantization, 8

R
Random access networks, 5
Rate monotonic, 166
Round trip delay, 118

Y.-B. Zhao et al., *Packet-Based Control for Networked Control Systems*,
DOI 10.1007/978-981-10-6250-6

S
Scheduling, 9
Sensor transmission rule, 128
Stability-guaranteed supremum of round trip delay, 163
Stable schedulability, 164
Stochastic stability, 80

T
Time driven, 4
Time-synchronization, 4

V
Varying feedback gain, 17

W
Wiener system, 51