



Doctoral Thesis in Electrical Engineering

# Resource-aware Wireless Process Control

TAKUYA IWAKI

# Resource-aware Wireless Process Control

TAKUYA IWAKI

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Friday the 26th of February 2021, at 2:00 p.m. in F3, Lindstedsvägen 26, Stockholm.

Doctoral Thesis in Electrical Engineering  
KTH Royal Institute of Technology  
Stockholm, Sweden 2021

© Takuya Iwaki

ISBN 978-91-7873-763-5

TRITA-EECS-AVL-2021:8

Printed by: Universitetsservice US-AB, Sweden 2021

## Abstract

To tackle the ever-growing demands on high-quality and cost-effective industrial production, recent developments in embedded sensing, wireless communication, and cloud computing offer great opportunities. Resource-aware reliable wireless communication and real-time control are needed to leverage these technologies. The thesis develops a new design framework for such wireless process control systems.

In the first part, an energy-aware multi-hop network scheduler for remote estimation and control is developed. Multiple sensors transmit their data to a remote estimator or controller through a shared multi-hop network. We develop scheduling algorithms determining which links of the network that should be activated and when to convey sensor data. For remote estimation, an optimization problem minimizing a linear combination of the averaged estimation error and network energy is formulated. We solve the problem by splitting it into tree planning and sensor selection subproblems, and show that an optimal periodic schedule can be obtained. The setting is then extended to an optimal control formulation, where an optimal solution minimizes the combination of the averaged linear quadratic Gaussian control cost and network energy consumption. Algorithms to reconfigure schedules and routes when network link outages are present are also introduced. The applicability of the proposed scheduler is demonstrated in numerical examples.

In the second part, event-triggered sensing, actuation, and control reconfiguration algorithms are developed. We derive stability conditions under event-triggered actuation for PID, cascade, decoupling, and delay-compensating control systems. Sensors sample and transmit their measurements periodically, while control commands are updated only when a certain event threshold is crossed. A tuning method for the threshold is proposed. We show that the approach yields setpoint tracking and disturbance rejection. Event-triggered sensing together with control reconfiguration is then considered for feedforward and cascade control, illustrating how wireless sensing can efficiently attenuate disturbances. Numerical examples demonstrate how the methods reduce information exchange without closed-loop performance degradation.



## Sammanfattning

För att möta den ständigt växande efterfrågan på högkvalitativ och kostnads-effektiv industriell produktion så erbjuder utvecklingen av inbyggda system, trådlös kommunikation och molntjänster stora möjligheter. Resursmedveten och pålitlig trådlös kommunikation tillsammans med realtidsreglering är nödvändigt för att fullt ut använda dessa teknologier. Den här avhandlingen utvecklar ett nytt ramverk för att designa sådana trådlösa processregleringssystem.

I den första delen av avhandlingen utvecklas en energimedveten schemaläggare för trådlös estimering och reglering. Flera sensorer sänder data till en trådlös estimator eller regulator genom ett delat multihoppnätverk. Schemaläggaren bestämmer vilka länkar i nätverket som ska vara aktiverade och när sensordata ska överföras. För trådlös estimering formuleras ett optimeringsproblem där målfunktionen är en linjärkombination av det genomsnittliga estimeringsfelet och den förbrukade energin i nätverket. Vi löser problemet genom att dela upp det i trädplanering och sensorval, och vi visar att ett optimalt periodiskt schema kan erhållas. Sedan formuleras ett optimalt styrproblem där målfunktionen är en kombination av den genomsnittliga linjärvadratiska Gaussiska reglerkostnaden och den förbrukade energin i nätverket. Algoritmer för omkonfigurering av scheman och rutter när länkavbrott kan hända presenteras också. Tillämpningar av den föreslagna schemaläggaren demonstreras i numeriska exempel.

I den andra delen av avhandlingen utvecklas händelsestyrd mätning, aktivering och omkonfigurering av styrsystemet. Vi härleder stabilitetsvillkor för händelsestyrd aktivering för PID-reglerade, kaskadreglerade, frikopplade och fördröjningskompenserande styrsystem. Sensorer samlar in och sänder data periodiskt medan styrkommandon uppdateras när en särskild händelsestyrd tröskelvärde överskrids. En metod för att ställa in tröskeln föreslås. Vi visar att tillvägagångssättet ger referensföljning och störningsdämpning. Händelsestyrd mätning tillsammans med omkonfigurering av regulatorn används sedan för framkoppling och kaskadreglering för att illustrera att trådlös mätning effektivt kan dämpa störningar. Numeriska exempel demonstrerar hur metoderna reducerar informationsutbyte utan att degradera prestandan hos det slutna styrsystemet.



## Acknowledgments

First and foremost, it is an immense pleasure to express my deepest gratitude to my supervisor, Professor Karl Henrik Johansson, for providing me the opportunity to work at KTH. He always welcomed me for discussion and gave me continuous inspiration and insightful guidance. I learned from him how to draw the big picture in research, how to challenge detailed questions, how to involve and motivate people towards the goal. It was the most invaluable experience for my career to work with him.

I am also profoundly grateful to my co-supervisor, Professor Henrik Sandberg, for giving me detailed support in my research and teaching. It was a very beneficial opportunity to teach basic control course to undergraduate students with him. I had much inspiration from him on how to convey the attraction of control theory to the students.

I could not have completed this work without the support from my collaborators. Sincere thanks to Professor Emilia Fridman for her valuable advice on my research. Her advice, as well as her excellent textbook and exciting course on time-delay systems, brought me new strong tools for my study. It was also truly a pleasure to work with everyone in the VINNOVA PiiA project and WASP program. Mostly, I am thankful to Professor Anders Ahlén, André Teixeira, Alf Isaksson, Johan Åkerberg, Markus Eriksson, Steffi Knorn, and Thomas Lindh for the enjoyable discussion in the PiiA project meetings.

I have received incredibly much help to complete this thesis. I would like to thank Professor Sophie Tarbouriech from LAAS-CNRS for being the opponent. I would also like to thank Professor Vijay Gupta from University of Notre Dame for being the opponent of my licentiate seminar and the committee member of the defense. I am grateful to Professor Subhrakanti Dey from National Univ of Ireland, Maynooth, Professor Mikael Gidlund from Mid Sweden University, for acting as the committee members. Special appreciation to Professor Elling W. Jacobsen for chairing the defense, and to Professors Dimos V. Dimarogonas and James Gross for willing to be the substitutes. Many thanks to Professor Cristian Rojas for being the advance reviewer of both my licentiate and doctoral theses. I would also like to thank Hampei Sasahara, Vahid Mamduhi, Xingkang He, and Yu Wang for proofreading this thesis, and Alexander Johansson for translating the abstracts of both my licentiate and doctoral theses into Swedish.

Starting my doctoral study would not have been possible without the sup-

port from the people at Tokyo Institute of Technology. I would like to express my sincere gratitude to my master supervisor, Professor Tomohisa Hayakawa, for his kind advice and support before and during my doctoral study. Heartfelt appreciation to Professor Jun-ichi Imura for his beneficial advice and support when I applied to the doctoral program. I would also thank him for his exciting undergraduate course, “Fundamental of Dynamical Systems”. It all started when I took this course. I am thankful to Professor Kenji Kashima for his support when I applied to the doctoral program.

I am thankful to all my current and former colleagues at the division of Decision and Control Systems for creating a friendly environment and an active working atmosphere, and for their continuous support for everything that I needed. Especially, I would like to express my heartfelt thanks to Junfeng Wu for his continued collaboration. I owe his patient support to my fruitful study in KTH. Special thanks to Yuchi Wu for the fruitful discussions and his careful attention to our work. I thank Kaito Ariu, José Mairton Barros da Silva Júnior, Dirk van Dooren, Kazumune Hashimoto, Yulong Gao, Jezdimir Milošević, Vahid Mamduhi, Ehsan Nekouei, Xiaoqiang Ren, Hampei Sashara, Anton Selivanov, Touraj Soleymani, Takashi Tanaka, Po-An Wang, Yu Wang, Jieqiang Wei, Pian Yu, Xinlei Yi, Han Zhang, for interesting discussions and supportive comments. I also want to thank the administrators in our department Gerd Franzon, Silvia Cardenas Svensson, Hanna Holmqvist, Anneli Ström, Felicia Gustafsson, Tord Christer Magnusson, and Karin Karlsson Eklund, for their kind assistance and support.

I am grateful to Yuta Aoki for our every-day chatting over the Messenger. It really motivated me a lot. I would like to give my heartfelt appreciation to the Mikami Karis family for their kind support to my family. The life in Stockholm with them is our lifelong memory.

Finally, I wish to dedicate this thesis to my family. To my parents and sisters, thank you for the unconditional support throughout my life. My deepest thanks with full of love goes to my wife Kyoko for always being with me, and my beautiful daughters Sakura and Kasumi for making my life colorful and full of smiles.

*Takuya Iwaki*  
Stockholm, Sweden  
January 2021

# Contents

<b>List of Abbreviations</b>	<b>xiii</b>
<b>Notation</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Starch cooker process example . . . . .	6
1.3 Problem formulation . . . . .	13
1.4 Thesis outline and contributions . . . . .	16
<b>2 Background</b>	<b>21</b>
2.1 Process control systems . . . . .	21
2.2 Wireless networks for industrial control systems . . . . .	29
2.3 Wireless process control systems . . . . .	34
<b>I Multi-hop Network Scheduling</b>	<b>45</b>
<b>3 Multi-hop Network Scheduling for Remote Estimation</b>	<b>47</b>
3.1 Problem formulation . . . . .	48
3.2 Structures of network scheduler . . . . .	56
3.3 Construction of suboptimal solutions . . . . .	66
3.4 Numerical examples . . . . .	68
3.5 Summary . . . . .	79
3.A Proof of Theorem 3.1 . . . . .	79
3.B Proof of Theorem 3.2 . . . . .	80
3.C Proof of Theorem 3.3 . . . . .	82

<b>4</b>	<b>Multi-hop Network Scheduling for Distributed Control</b>	<b>85</b>
4.1	Problem formulation . . . . .	87
4.2	Optimal controller and scheduler . . . . .	91
4.3	Link disconnection and route reconfiguration . . . . .	93
4.4	Numerical example . . . . .	93
4.5	Summary . . . . .	98
<b>II</b>	<b>Event-triggered Process Control</b>	<b>101</b>
<b>5</b>	<b>Event-triggered Actuation for Multi-loop Control Systems</b>	<b>103</b>
5.1	System model and problem formulation . . . . .	104
5.2	Main results . . . . .	109
5.3	Applications to PID, cascade, and decoupling control . . . . .	118
5.4	Summary . . . . .	133
5.A	Proof of Lemma 5.1 . . . . .	133
5.B	Proof of Theorem 5.1 . . . . .	135
<b>6</b>	<b>Event-triggered Actuation for Time-delay Systems</b>	<b>137</b>
6.1	Time-triggered PI control for time-delay systems . . . . .	138
6.2	Event-triggered control of time-delay systems . . . . .	141
6.3	Numerical examples . . . . .	143
6.4	Summary . . . . .	147
6.A	Proof of Theorem 6.1 . . . . .	147
6.B	Proof of Theorem 6.2 . . . . .	151
<b>7</b>	<b>Event-triggered Controller Switching</b>	<b>153</b>
7.1	Output feedback control with multiple sensors . . . . .	154
7.2	Event-triggered controller switching . . . . .	156
7.3	Applications to cascade and feedforward control . . . . .	161
7.4	Numerical examples . . . . .	163
7.5	Summary . . . . .	168
7.A	Proof of Lemma 7.1 . . . . .	169
<b>8</b>	<b>Conclusions and Future Research</b>	<b>171</b>
8.1	Conclusions . . . . .	171
8.2	Future research directions . . . . .	173

**Bibliography**

**175**



# List of Abbreviations

CA	Contention access
CAP	Contention access period
CF	Contention-free
CFP	Contention-free period
CSMA	Carrier-sense multiple access
CSMA/CA	Carrier-sense multiple access with collision avoidance
DCN	Distributed control node
DCS	Distributed control system
i.i.d.	Independent and identically distributed
I/O	Input/output
LEACH	Low energy adaptive clustering hierarchy
LMI	Linear matrix inequality
LQG	Linear quadratic Gaussian
LTI	Linear time-invariant
MAC	Medium access control
MAD	Maximum allowable delay
MATI	Maximum allowable transmission interval
MFR	MAC footer
MHR	MAC header
MDP	Markov decision process
MMSE	Minimum mean square error
MPDU	MAC protocol data unit
OSI	Open systems interconnection
PHR	Physical layer header
PHY	Physical layer
PI	Proportional integral
PID	Proportional integral derivative

PPDU	PHY protocol data unit
RR	Round-robin
RSSI	Received signal strength indicator
SHR	Synchronization header
TDMA	Time-division multiple access
TOD	Try-once-discard

# Notation

## Sets

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of real $n$ -dimensional vectors
$\mathbb{R}^{n \times m}$	Set of real $n$ -by- $m$ real matrices
$\mathbb{N}$	Set of positive integers
$\mathbb{N}_0$	Set of nonnegative integers
$\mathbb{S}_{++}^n$	Set of $n$ -by- $n$ positive definite matrices
$\mathbb{S}_+^n$	Set of $n$ -by- $n$ positive semi-definite matrices
$\emptyset$	Empty set
$ \mathcal{S} $	Cardinality of set $\mathcal{S}$
$2^{\mathcal{S}}$	Power set of $\mathcal{S}$
$\text{int}(\mathcal{S})$	Interior of set $\mathcal{S}$
$\partial\mathcal{S}$	Boundary of set $\mathcal{S}$

## Vectors

$x[i]$	The $i$ -th element of vector $x$
$x_{0:k}$	The sequence of all vectors $x_t, t = 0, \dots, k$
$\mathbf{1}_n$	The $n$ -by-1 vector of all ones
$\ x\ $	The euclidean norm of vector $x$
$\mathbb{E}[x]$	Expectation of a random vector $x$

## Matrices

$I_n$	The $n$ -by- $n$ identity matrix
$X^\top$	Transpose of real matrix $X$
$\lambda_{\max}(A)$	The largest eigenvalue of $A$ ; $A$ has real eigenvalues
$X > Y$	$X - Y$ is positive definite
$X \geq Y$	$X - Y$ is positive semi-definite

$\text{tr}(A)$ Trace of  $A$  $\begin{bmatrix} A & B \\ * & C \end{bmatrix}$ Symmetric matrix of the form  $\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}$ **Other notations** $\wedge$ 

Logical conjunction

 $\vee$ 

Logical disjunction

 $\lfloor \cdot \rfloor$ 

The floor function

# Chapter 1

## Introduction

The Industrial Revolution, which originated in the invention of the water frame and steam engine in the 1770s, initiated the development of our technically advanced society. Since then, the society has witnessed amazing technological progress. Along with this, the process industry, producing essential products for our daily life, such as oil, gas, chemicals, water, steel, and papers, has increased its capacity to satisfy the increasing societal demand.

Recent advances in wireless communication, sensing, and computation technologies have brought revolutionary changes to our life and society. Process control also has become highly digitalized in the past decades. Attempts are now made to integrate wireless communication into process control systems, as the usage of wireless communication enables more effective design, deployment, operation, and maintenance. Such improvements could have an enormous impact on our society, thanks to the critical importance of the process industry and industrial automation.

Despite recent progress, some essential challenges remain to realize wireless process control. The communication capacity and battery limitation of today's wireless networks introduce non-negligible delays and transmission failures, hindering stable and continuous operation of the control systems. These imperfections, if not suitably mitigated, may result in significant economic losses or may even threaten human safety. Wireless process control systems, therefore, must be designed to reduce these effects. In this thesis, we focus on two important problems of such resource-aware wireless process control systems. First, we investigate how to design scheduling and routing for multi-hop networks, integrated with estimation and control applications. In other words,

we study when, where, and how the information of the systems should be used and transmitted in process control systems. Second, we research applications of resource-aware communication to specific process control loops, which are often introduced to compensate for disturbances and time delays.

The rest of this chapter is organized as follows. In Section 1.1, we motivate resource-aware process control systems. We discuss why replacing conventional communication with wireless is profitable but challenging. The motivation is exemplified by the Iggesund paper mill case study in Section 1.2. Section 1.3 formulates the two problems considered in this thesis, important for realizing resource-aware wireless process control systems. Lastly, the structure of the thesis is presented in Section 1.4, together with a summary of the contributions.

## 1.1 Motivation

Our daily life cannot be maintained without stable supplies of a wide variety of industrial products. The process industry provides those products through a series of chemical and mechanical operations, involving oil, gas, chemicals, steel, paper, water, medicine, and food, etc. Examples of production facilities, called process plants, are shown in Figure 1.1. Figure 1.1a illustrates an oil refinery, which transforms crude oil into petrochemical products. Figure 1.1b is a water treatment facility, performing various treatments to increase water quality. Figure 1.1c is a paper mill, producing paper from raw materials such as wood pulp, old rags, and other ingredients.

These plants can usually be broken down into several subunits. Raw materials are transformed into final products through chemical processing carried out in subunits, and such transformations are managed by process control systems. The control systems manage the processes to satisfy the following requirements:

- *Safety*: Process states, such as flow rate, pressure, temperature, and concentration, must be maintained in given ranges to keep safe operation.
- *Specification*: A required amount of final product must be obtained, and it should satisfy the predetermined specifications.
- *Economy*: Plants must be operated with as little material, labor force, and energy as possible to maximize profit.



(a)



(b)



(c)

Figure 1.1: Three examples of process plants. (a) An oil refinery. *Source:* Ministry of Economy, Trade and Industry, Japan ([https://www.meti.go.jp/policy/safety\\_security/industrial\\_safety/sangyo/hipregas/sp-nintei/nintei/interview/jxtg\\_sakai.html](https://www.meti.go.jp/policy/safety_security/industrial_safety/sangyo/hipregas/sp-nintei/nintei/interview/jxtg_sakai.html)). (b) A water treatment facility. *Source:* Ministry of Land, Infrastructure, Transport and Tourism, Japan (<https://www.mlit.go.jp/mizukokudo/sewerage/index.html>). (c) A paper mill [1].

Various technical innovations have been made to increase the efficiency and flexibility of process control systems. When process plants were first constructed at the end of the 19th century, operators manually opened and closed control valves to operate the plant. Control of process plants was performed automatically only after mechanical and pneumatic PID regulators were introduced. Digital controllers replaced electronic analog PID controllers in the 1970s [2].

The revolutionary development of wireless communication and computation technologies, known as the Internet of Things and Industry 4.0, has further challenged process control systems [3, 4]. Wireless technologies and cloud computing allow removing cables and increasing accessibility from remote locations, resulting in cost-efficient and flexible configurations [2, 5, 6]. The benefits of introducing wireless process control systems can be summarized as follows:

- *Cost efficiency*: Using wireless communication reduces cable and installation costs. It is estimated that the reduction ranges from 300 to 6000 US dollars per meter [2].
- *Installation and maintenance flexibility*: Wireless devices can be deployed where the cabled devices cannot be located. For example, a wireless sensor can be installed on rotating machinery. This enables the control of processes more precisely. Furthermore, wireless devices can easily be added, replaced, and modified even after the plant is under operation. Wireless process control systems can, therefore, be easily installed, maintained, and expanded.
- *Resiliency*: Wireless process control systems can be more resilient against disasters. On March 11, 2011, a Japanese oil refinery plant was severely damaged by an earthquake and succeeding tsunami. While many cabled instruments were destroyed, wireless sensors could continue their operation. Even if damaged, a wireless network can be recovered faster without extensive replacements and installation crews [7].

With the increasing attention to wireless process control systems, two major wireless communication protocols, WirelessHART [8] and ISA100.11a [9], have been proposed. Despite their availability, only monitoring applications have mainly been deployed so far. Feedback control applications are more critical than monitoring and therefore require reliable communication and closed-

loop guarantees. Control over wireless communication is concerned with the following aspects:

- *Limited channel capacity:* Capacity limitation of wireless communication leads to packet dropout and delay. Since network nodes usually share network channels, data transmission may fail due to packet collisions or interference. Multiple channel access methods such as time-division multiple access (TDMA) and carrier-sense multiple access with collision avoidance (CSMA/CA) are available to handle data collision, but non-negligible delays may be introduced. Network-induced packet dropout and delay degrade control performance, sometimes resulting in serious performance losses [10].
- *Channel characteristic variation:* The channel characteristic of a wireless network varies over time [11]. In indoor factory environments, the characteristic varies over both short and long time horizons [12–14]. The required transmission power to send data consequently changes and some links between wireless nodes might even become unavailable. Transmission power must be adjusted according to the current channel characteristic, and network routing needs to be reconfigured to counteract any lost links.
- *Battery energy limitation:* Many wireless networks are battery-powered or have unreliable energy sources. Batteries of wireless nodes must be replaced periodically, resulting in increased maintenance costs. If a wireless node becomes unavailable because of a power shortage, the control system's performance is critically affected. Thus, energy consumption must be minimized. It is especially important to reduce data transmission and reception since they consume much energy of typical wireless sensors, see Figure 1.2 [15].

As discussed so far, while introducing wireless communication into process control systems may have tremendous merits, there remain important problems to be solved, particularly regarding reliability. As stated in [10], reliability has a significant impact on control system performance. This thesis seeks to develop the theoretical foundation of wireless process control systems to overcome the reliability challenge and to meet the required control performance.

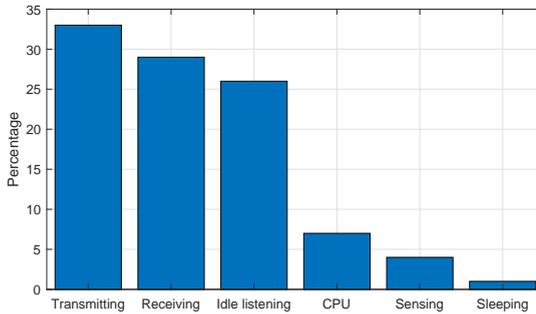


Figure 1.2: Typical energy consumption of a wireless sensor [15].

## 1.2 Starch cooker process example

This section discusses the starch cooker process of the Iggesund paper mill [1], to motivate wireless process control systems with a particular industrial example. In the Iggesund paper mill, there are two parallel paper machines (Figure 1.1c) to produce high-quality cardboard, which is mainly used for packing and graphic purposes. There is also a coating kitchen in the mill, delivering layers on the cardboard to make it smoother. The starch paste, produced by mixing starch powders with water at the starch cooker, is used as an ingredient for the cardboard coating. In this section, we first describe the starch cooker process and its current control system, followed by a proposal for a wireless control architecture. Some experimental data are presented together with a discussion on potential future improvements.

### 1.2.1 Starch cooker process

Figure 1.3 shows the process flow and control system architecture of the starch cooker process. The architecture consists of multiple feedback loops involving sensors ( $S_i$ ) and actuators ( $A_i$ ). First, the dry starch powder stored in the starch powder buffer (Figure 1.4a) is mixed with water at the mix funnel (Figure 1.4b). The water is stored at the mix water tank, in which the water tank level is regulated by the level control loop ( $S_1, A_1$ ). Two control loops ( $S_2, A_2$ ) and ( $S_3, A_3$ ) regulate the starch–water mixture concentration. The mixture is heated up at the steam ejector (Figure 1.4c), in which temperature is regulated by the temperature control loop ( $S_4, A_4$ ). The concentration is further

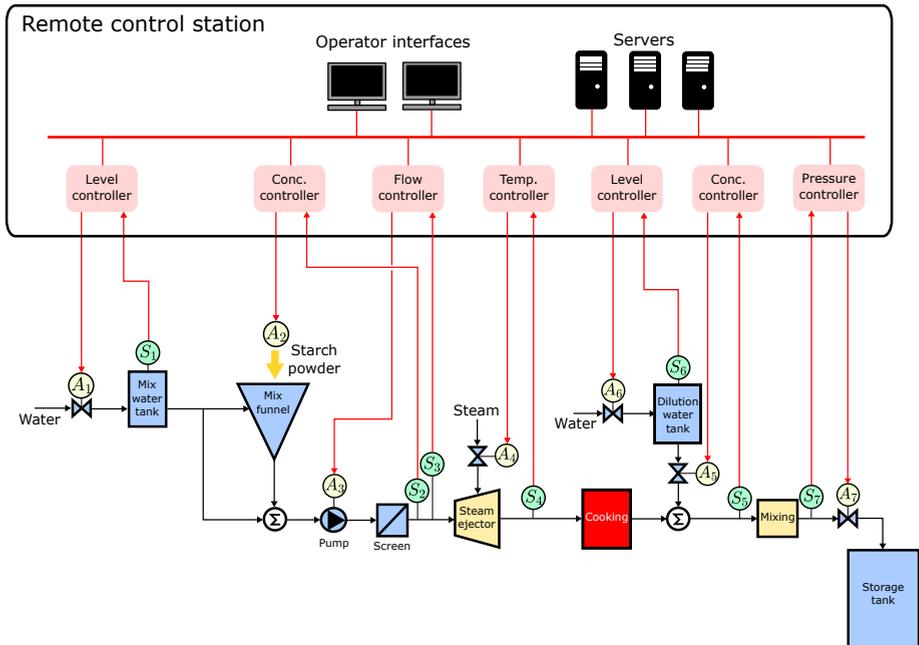


Figure 1.3: The starch cooker process flow and its control system architecture. The starch powder is mixed with water at first to the left. Then the mixture is heated up. The heated mixture is stored at the storage tank after a fine adjustment of its concentration. Seven feedback control loops are implemented and utilize pairs of sensors ( $S_i$ ) and actuators ( $A_i$ ). The corresponding control algorithms run at the remote control station.



(a) Starch powder buffer



(b) Mix funnel



(c) Steam ejector



(d) Storage tank

Figure 1.4: Pictures of the starch cooker process at the Iggesund paper mill [1].

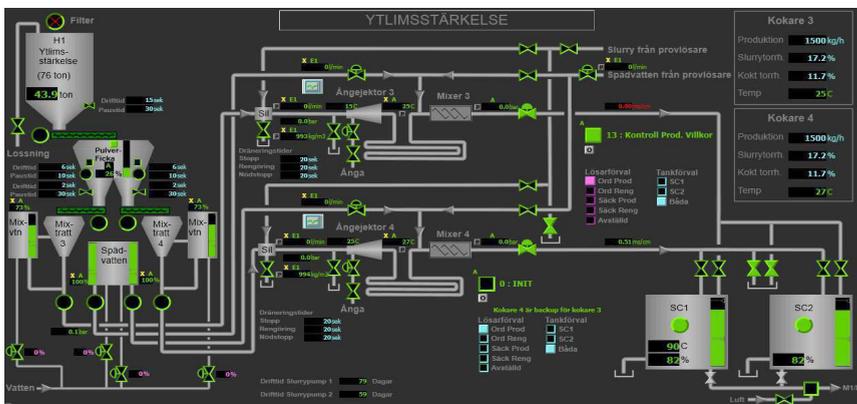


Figure 1.5: An operator interface panel of the starch cooker process. A sketch of the process flow is indicated. Operators can monitor the process and change its parameters.

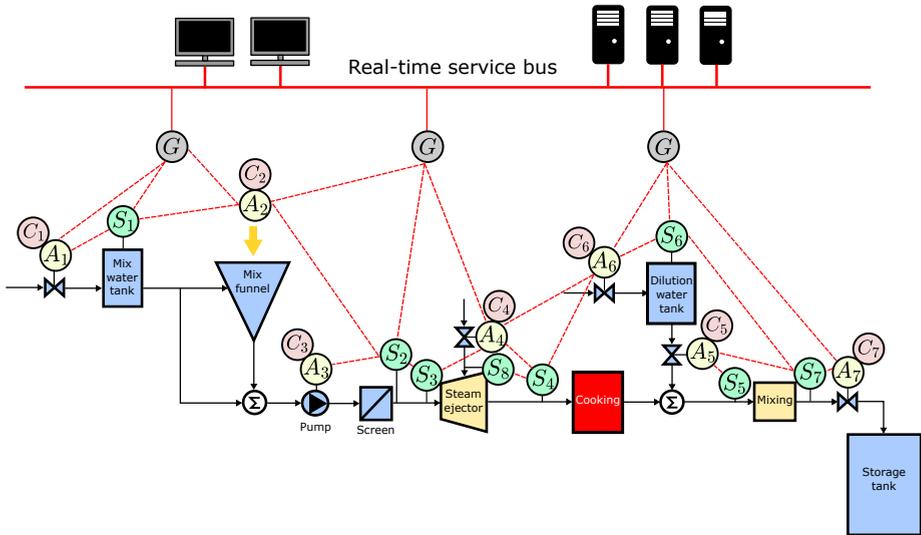


Figure 1.6: Wireless process control system for the starch cooker process, cf. wired architecture in Figure 1.3. Controllers are co-located with the actuators. Sensors and actuators communicate through a shared multi-hop network. The information from the sensors and actuators are sent to operators via gateways for monitoring and other applications. Note that compared to the architecture in Figure 1.3, a new sensor ( $S_8$ ) has been added to the steam line to monitor the steam flow variation.

adjusted by the concentration control loop ( $S_5$ ,  $A_5$ ) by adding water from the dilution water tank. The level is governed by the level control loop ( $S_6$ ,  $A_6$ ). The mixture is stored at the storage tank (Figure 1.4d) after its pressure is regulated by the pressure control loop ( $S_7$ ,  $A_7$ ).

Plant operators at the remote control station monitor and control the process through the operator interface. The interface panel shows a sketch of the process flow, indicating the sensor measurements and actuator statuses (Figure 1.5). The controllers are located at the remote control station, and the sensors and actuators communicate with the corresponding controllers through point-to-point hard-wired cables.

### 1.2.2 Wireless control system of starch cooker process

Communication cables and buses can be replaced by one or more wireless networks. Inspired by the ExxonMobil automation vision [5], a possible wireless process control system architecture for the starch cooker process is shown in Figure 1.6 [1, 4, 16]. In Figure 1.6, wireless sensors, actuators, and gateways ( $G$ ) are deployed as network nodes. The sensors communicate to the corresponding controllers, co-located with actuators, through a mesh-structured multi-hop network. This setup follows the available industrial wireless communication protocols such as WirelessHART and ISA100.11a, all of which have a mesh topology [17]. The information of the network nodes (sensors, controllers, and actuators) are sent through gateways and a real-time service bus to operator interfaces and other applications.

A real-time routing protocol [18] and time synchronization protocol [19] have been proposed and evaluated for the starch cooker process [1]. The protocols introduce flooding-based routing and precise TDMA to achieve reliable end-to-end communication. Five-days evaluation indicates that wireless starch cooker process control has no issues due to these protocols. Figure 1.7 shows a step response of the steam ejector temperature control. When the output reached the desired temperature of  $138^{\circ}\text{C}$  shortly after 21:00, the controller could maintain the temperature. The evaluation indicates that wireless networks have the potential for process control. However, much longer and more thorough studies are needed. For example, the influence of energy limitation of wireless network nodes should be considered.

Topology reconfiguration is an important property of industrial network protocols since channel characteristics in industrial environments change over time. Radio variations can be observed in the Iggesund paper mill caused by moving objects such as cranes and persons in the vicinity of the network nodes. The characteristics vary in the order of minutes or hours [20, 21]. Figure 1.8a shows the variation caused by a crane in the ceiling of the building. The crane moves over the floor to carry the finished paper to long-term storage, resulting in the channel gain variation in the order of hours. Figure 1.8b shows the shorter variations in the starch cooker process unit caused by personnel moving in narrow aisles around the unit.

The reliability of multi-hop networks despite energy limitations and channel characteristic changes needs to be guaranteed. New energy-efficient communication strategies and network reconfigurations are of practical importance.

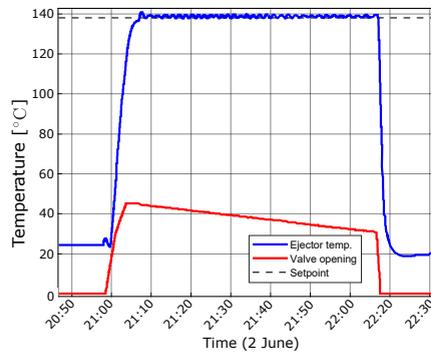


Figure 1.7: A step response of wireless steam ejector temperature control. The output temperature  $S_4$  (blue) increases to the setpoint temperature of  $138\text{ }^\circ\text{C}$  (red) at the beginning of the batch operation. After that, the temperature can be maintained around the setpoint. The red line indicates the valve opening  $A_4$  (percentage). Experimental data from [1].

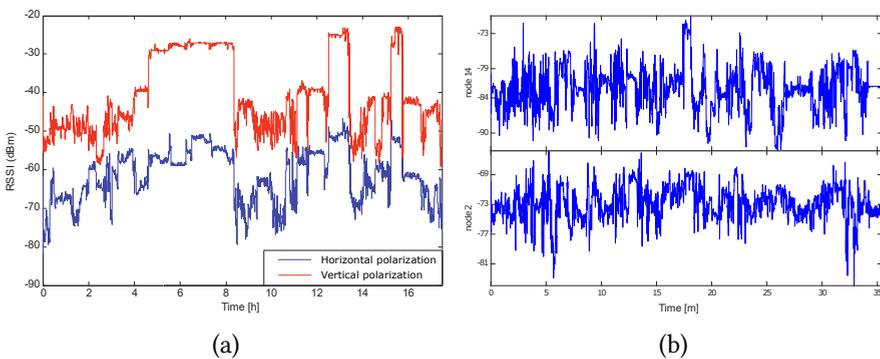
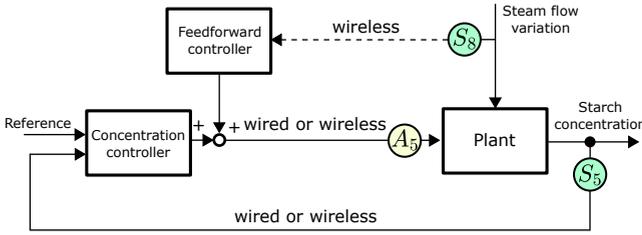
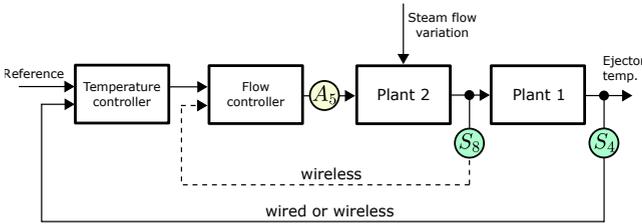


Figure 1.8: Received signal strength indicator (RSSI) of some radio links deployed at the Iggesund paper mill [1]. (a) Channel gain variation between two nodes located next to the paper machine finish line. The vertical polarization (red) and horizontal polarization (blue). (b) Channel gain variation between two nodes (node 14 and node 2) in the starch cooker process unit.



(a) Feedforward–PID configuration for starch concentration control. Steam flow variation is measured by the additional wireless sensor,  $S_8$ , which is used for feedforward control. The feedforward controller adjusts the control signal to the plant to take corrective action before the variation affects the starch concentration.



(b) Cascade configuration for steam ejector temperature control. The steam flow sensor and actuator introduce a tighter steam flow rate control. It reduces the effect of the steam flow variation.

Figure 1.9: Block diagrams of feedforward and cascade control loops. In both cases, a wireless steam flow sensor is added to the original feedback control loop.

### 1.2.3 Performance improvement using enhanced process control loops

Wireless communication enables the deployment of new sensors easily into control systems. For example, a new wireless sensor  $S_8$  is deployed to measure the steam flow rate in Figure 1.6. The final starch concentration, controlled by sensor  $S_5$  and actuator  $A_5$ , is possibly disturbed by the steam flow rate variation into the steam ejector. Since such a disturbance only slowly affects the final product, it is difficult to mitigate the influence effectively by feedback control. The feedforward controller adjusts the fine water flow rate as soon as the disturbance is detected by directly monitoring the steam flow rate us-

ing  $S_8$ , as illustrated in Figure 1.9a.

The sensor can also be used to improve the steam ejector temperature control by a cascade control configuration. Introducing a feedback control loop with the sensor–actuator pair  $(S_8, A_4)$  achieves a tighter steam flow rate control, which reduces the effects of the flow variation to the ejector temperature, see Figure 1.9b. These examples motivate us to consider feedforward, cascade, and other control architectures over a wireless network.

#### 1.2.4 Summary

We see from the starch cooker process example that wireless control can be cost-effective since it can remove cables between the field devices and controllers. Moreover, a new sensor can be flexibly deployed. We can realize more accurate concentration control by introducing feedforward–PID configuration or temperature control by cascade configuration with an additional sensor. While there are many reasons to use wireless control, some problems remain to be solved. One is how to cope with energy limitations of the network nodes and channel characteristic variation in the environment. Since wireless network nodes usually have no reliable energy sources, energy consumption must be carefully considered. Existing protocols are robust to communication-induced delay, packet dropout, and topology change due to channel gain variation, but how to reduce the energy consumption of the network nodes still needs to be investigated.

### 1.3 Problem formulation

This thesis proposes a design framework for resource-aware wireless communication and control, relevant to the realization of wireless process control systems. The idea is to introduce network communication strategies to reduce communication while maintaining system performance. In particular, we discuss two problems. The first problem is on the scheduling of control systems over multi-hop networks. A systematic way to obtain an energy-effective network schedule is considered to minimize the number of transmissions and thereby reduce energy consumption. The second problem is on resource-aware communication dedicated to enhanced process control loops. In what follows, we introduce the problems in detail.

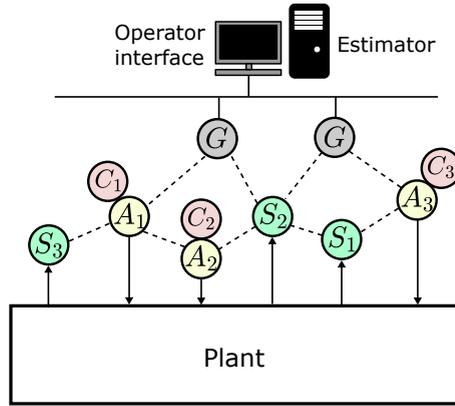


Figure 1.10: Illustration of a multi-hop wireless networked control system considered in Problem 1 with  $N = 3$  control loops. Sensors, actuators, and controllers communicate through a mesh-structured network. Controllers are co-located with actuators.

### Problem 1: Designing scheduler for wireless process control

In the first problem, we investigate a co-design framework of scheduling, routing, and control over a multi-hop network. Figure 1.10 depicts the system considered. Sensors  $S_i$  and actuators  $A_i$ ,  $i = 1, \dots, N$ , are distributed over a field, and form a multi-hop network. The controllers are co-located with the corresponding actuators. Sensor  $S_i$  transmits its measurements or its local plant state estimate to the corresponding controller  $C_i$ . The measurements are also sent to the operator for a monitoring purpose.

Consider the multi-hop network graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  is the node set consisting of  $S_i, A_i, i = 1, \dots, N$ , and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  the network link set. The information from each sensor is transmitted over the network  $\mathcal{G}$ . We need to find a scheduler that decides when and how the information is transmitted, i.e., at each time instance, which sensor data is transmitted through which link in  $\mathcal{E}$ , under the objective to reduce the energy consumption of the network nodes, while maintaining the control or estimation performances. In particular, we consider two scenarios: sensor data is transmitted to a remote estimator for monitoring and to the corresponding actuators for feedback control.

In summary, we address the following questions:

**Q1:** How to design a scheduler for remote estimation to reduce the energy

consumption of wireless sensors while maintaining the estimation performance?

**Q2:** How to design a scheduler for feedback control to reduce the energy consumption of wireless sensors and actuators while maintaining the control performance?

As the radio environment changes over time, the scheduler needs to reconfigure when a link in the network is disconnected due to channel variation. We address the following question:

**Q3:** How to reconfigure a schedule when a link in a multi-hop network becomes unavailable?

## **Problem 2: Resource-aware wireless communication for process control**

The application of wireless communication to feedforward, cascade, decoupling, and other traditional control architectures is investigated in the second problem. In particular, we study a way to limit sensor and actuator communications applied to such control loops. We address the following question:

**Q4:** How to reduce communications of wireless feedforward–PID, cascade, and decoupling control while maintaining control performance?

The Smith predictor is often introduced to compensate for a non-negligible time delays. Wireless control for time-delay systems is, therefore, to be investigated:

**Q5:** How to reduce communications of wireless time-delay control systems while maintaining control performance?

The questions above focus on fully wireless control systems. A reasonable and effective way to improve the control performance of an existing system is to retrofit a wireless sensor to a control loop and introduce feedforward or cascade control. We consider a feedback control loop with a potential disturbance that degrades the control performance. We introduce feedforward or cascade control by adding a wireless sensor. We address the following question:

**Q6:** How to reduce communications of a wireless sensor while keeping the effectiveness of feedforward or cascade control?

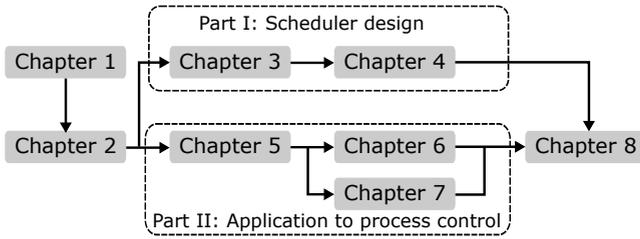


Figure 1.11: Illustration of the thesis structure.

## 1.4 Thesis outline and contributions

This section explains how the thesis is organized. The main results of this thesis are presented in Chapters 3–7 and are divided into two parts according to the problems considered. Part I addresses Problem 1, i.e., network scheduler designs for monitoring and distributed feedback control. Part II focuses on Problem 2, i.e., we discuss the applications of wireless communication to specific process control loops. The overall structure of the thesis is illustrated in Figure 1.11.

### Chapter 1: Introduction

This chapter motivates the thesis and states the problem illustrated using the starch cooker process of the Iggesund paper mill. The discussion is partially based on the following contribution:

- A. Ahlén, J. Åkerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, “Towards wireless control in industrial process automation: A case study at a paper mill,” *IEEE Control System Magazine*, vol. 20, no. 2, pp. 36–57, 2019.

### Chapter 2: Background

Chapter 2 gives the background for this thesis. First, we briefly overview process control systems. Second, we go through wireless communication protocols for industrial control systems. Third, we give a literature review on wireless process control.

### **Chapter 3: Multi-hop network scheduling for remote estimation**

Chapter 3 addresses **Q1**. We consider process state monitoring, in which sensors transmit their measurements to a remote estimator through a multi-hop network. We propose a design framework of a multi-hop sensor network scheduler for remote estimation. An optimization problem is formulated, and the optimal network schedule is searched to minimize estimation error under sensor energy considerations. It is shown that the optimal network schedule forms a tree with a root at the gateway node. From this observation, we manage to separate the optimization problem into two subproblems: tree planning and sensor selection. We solve the sensor selection subproblem by a Markov decision process, and show that the optimal solution admits a periodic structure when the transmission cost is sufficiently low. Efficient algorithms are proposed, and they are shown to reduce the computational complexity of the original optimization problem. Numerical studies illustrate the effectiveness of the proposed algorithms and show that they are scalable to large networks.

The covered material is based on the following contributions:

- T. Iwaki, Y. Wu, J. Wu, H. Sandberg, and K. H. Johansson, “Wireless sensor network scheduling for remote estimation under energy constraints,” in *Proceedings of IEEE Conference on Decision and Control*, pp. 3362–3367, 2017.
- T. Iwaki, Y. Wu, J. Wu, H. Sandberg, and K. H. Johansson, “Multi-hop sensor network scheduling for optimal remote estimation,” *Automatica*, To appear.

### **Chapter 4: Multi-hop network scheduling for distributed control**

Chapter 4 addresses **Q2** and **Q3**. We consider multiple feedback control loops, where each controller is co-located with the corresponding actuator. The sensors transmit their local estimates to the controllers through a multi-hop network. We propose a co-design framework of linear quadratic Gaussian (LQG) control, scheduling, and routing. An optimization problem is formulated, minimizing a linear combination of the averaged LQG control performance and the averaged transmission energy consumption. Optimal solutions are derived, and their performance is illustrated in a numerical example. In this chapter, algorithms to reconfigure routing between sensors and actuators in case of

link outage are also provided. The results are illustrated in a numerical example.

The covered material is based on the following contribution:

- T. Iwaki, and K. H. Johansson, “LQG control and scheduling co-design for wireless sensor and actuator networks,” in *Proceedings of IEEE International Workshop on Signal Processing Advances in Wireless Communications*, 2018.

## **Chapter 5: Event-triggered actuation for multi-loop control systems**

Chapter 5 tackles **Q4**. Particularly, we study periodic event-triggered actuation applied to PID, cascade, and decoupling control. We introduce an event-triggered output feedback controller, in which the control command is actuated only when it exceeds its previous value by a certain threshold. An exponential stability condition is derived in the form of linear matrix inequalities (LMIs) using a Lyapunov–Krasovskii functional based on Wirtinger’s inequality. It is shown that an observer-based controller can reject an unknown step disturbance. Using this result, we propose how to tune the event threshold subject to a given stability margin. We apply the proposed framework to PID, cascade, and decoupling control to illustrate how the event thresholds can be tuned in practice. Numerical examples show for these three control loops how communication can be reduced without performance degradation.

The covered material is based on the following contributions:

- T. Iwaki, and K. H. Johansson, “On setpoint tracking and disturbance rejection of event-triggered PI control,” in *Proceedings of SICE International Symposium on Control Systems*, 2020.
- T. Iwaki, E. Fridman, and K. H. Johansson, “Multi-loop periodic event-triggered actuation: Applications to PID, cascade, and decoupling control,” *International Journal of Control*, Submitted.

## **Chapter 6: Event-triggered control for time-delay systems**

Chapter 6 addresses **Q5**. We focus on event-triggered PI control for time-delay systems with parametric uncertainties. The systems are given by continuous-time linear systems with parameter uncertainty polytopes. We propose an

event-triggered PI controller, in which the controller transmits its signal to the actuator when its value goes beyond a threshold. A state-space formulation of the Smith predictor is used to compensate for the time delay. An asymptotic stability condition is derived in LMIs using a Lyapunov–Krasovskii functional. Numerical examples illustrate that our proposed controller reduces the communication load without performance degradation and despite plant uncertainties.

The covered material is based on the following contribution:

- T. Iwaki, E. Fridman, and K. H. Johansson, “Event-triggered PI control of time-delay systems with parametric uncertainties,” in *Proceedings of IFAC World Congress, 2020*.

## Chapter 7: Event-triggered controller switching

Chapter 7 addresses Q6. In this chapter, an event-triggered controller switching framework is proposed when the process state is monitored by multiple sensors. Asymptotic stability conditions for given sensor sampling intervals are derived. Based on these results, we propose an event-triggered controller switching, in which one sensor transmits its measurement to the controller with a fixed sampling rate while another sensor transmits with a send-on delta strategy. The proposed framework is applied to cascade and feedforward control. Numerical examples illustrate how our framework reduces the effect of disturbances for both cascade and feedforward control systems.

The covered material is based on the following contribution:

- T. Iwaki, E. Fridman, and K. H. Johansson, “Event-based switching for sampled-data output feedback control: Applications to cascade and feedforward control,” in *Proceedings of IEEE Conference on Decision and Control*, pp. 2592–2597, 2019.

Related publications are:

- T. Iwaki, J. Wu, and K. H. Johansson, “Event-triggered feedforward control subject to actuator saturation for disturbance compensation,” in *Proceedings of European Control Conference*, pp. 501–506, 2018.
- A. Ahlén, J. Åkerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, “Towards wireless control in industrial process automation: A case study at a paper mill,” *IEEE Control Systems Magazine*, vol. 20, no. 2, pp. 36–57, 2019.

## **Chapter 8: Conclusions and future research**

In Chapter 8, we present a summary of the results and discuss directions for future research.

### **The author's contribution and other publications**

In the aforementioned contributions, the author of the thesis had the most significant role in formulating the problems, solving them, and writing the articles. The order of the authors listed above indicates the relative contributions, except for the paper in Chapter 1. In this paper, the authors are ordered alphabetically.

The following publication by the author is not covered in the thesis, but contains related material:

- Y. Wu, T. Iwaki, J. Wu, K. H. Johansson, and L. Shi, "Sensor selection and routing design for state estimation over wireless sensor networks," in *Proceedings of Chinese Control Conference*, pp. 8008–8013, 2017.

We also remark that parts of Chapter 3–4 appear in the licentiate thesis:

- T. Iwaki, "Wireless sensor network scheduling and event-based control for industrial processes," KTH Royal Institute of Technology, 2018.

# Chapter 2

## Background

In this chapter, we establish the background concepts and literature for the thesis. We first briefly overview process control systems in Section 2.1. We introduce architectures of the current process control systems and some specific control loops. Section 2.2 presents the technologies that enable wireless process control systems. In this section, we survey industrial wireless communication protocols, particularly the literature on energy-effective wireless communication protocols. Finally, we review wireless process control systems in Section 2.3.

### 2.1 Process control systems

Process control was first performed manually by operating valves based on measured values by local sensors. Manual operation was harsh labor that required a large number of operators, and the operation was not accurate. Since the creation of the pneumatic PID controller in the 1930s, process control has been automated [22]. Eventually, controllers were located at a remote control station. However, since pneumatic signals were used as a signal transmission medium, there were restrictions on transmission distance and space. As the size of process plants increased and technology developed, there was a shift to electrical signals, resulting in more complex but intelligent control systems. In this section, we first briefly overview current process control system architectures. Next, we describe some common and specific control loops.

### 2.1.1 Process control system architectures

Process control systems based on microprocessors were introduced in the 1970s [2]. Each microprocessor serves input/output (I/O) processing and control computation for 8 to 32 control loops and forms the so-called distributed control system (DCS) [2]. Communications among controllers and field sensors and actuators were carried out through an analog electric current between 4 and 20 mA. While some digital communication technologies such as Foundation Fieldbus [23] and Profibus [24] are available now, the basic architecture of process control systems, as shown in Figure 2.1 [5], are fixed until now. Process control systems today have a hierarchical structure with five levels, see Figure 2.1. At Level 0, i.e., in the field level, sensors and actuators are distributed. At Level 1, basic controllers are located. Each controller collects sensor measurements and computes control commands. The commands are transmitted to the corresponding actuators. Operators can check the plant status and change setpoints and controller parameters through user interfaces. At Level 2, some other applications are located to monitor the field device conditions. At Level 3, advanced logic and applications such as model predictive control are implemented. Servers for business use are located at Level 4. This thesis focuses on control and communication at Level 0 and 1. In the next subsection, we describe control loops that appear at these levels.

### 2.1.2 Process control loops

PID control plays a central role in process control. Next, we give an overview of typical ways to modify the standard PID control loop to improve the overall control performance. In particular, we discuss feedforward control, cascade control, decoupling control, and the Smith predictor. These approaches are extended to wireless implementation in Part II of the thesis.

#### Feedforward control

Feedforward control is used together with PID control to attenuate external disturbances when the disturbances can be measured. Using the information from the disturbance sensor, the controller takes corrective action to avoid that the controlled variable deviates from its setpoint. Figure 2.2 shows the block diagram of a typical feedforward–PID control architecture. The disturbance affects the main plant (Plant 1) through a disturbance plant (Plant 2), which

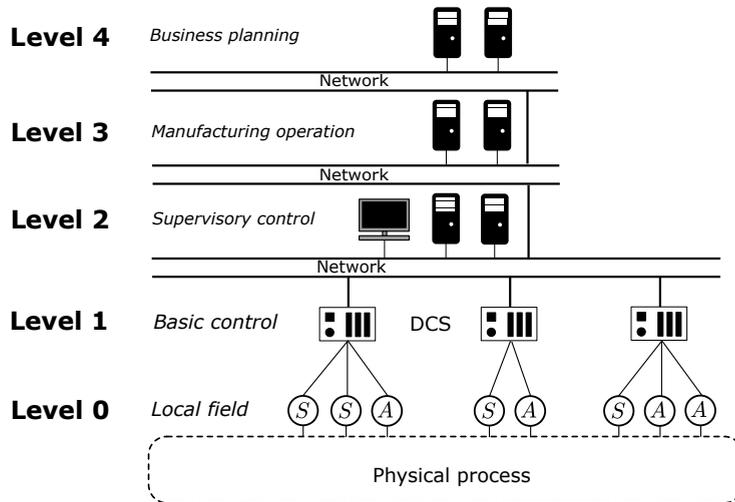


Figure 2.1: Architecture of industrial process control systems [5]. The overall system has a hierarchical structure divided into five levels, from field devices to business planning.

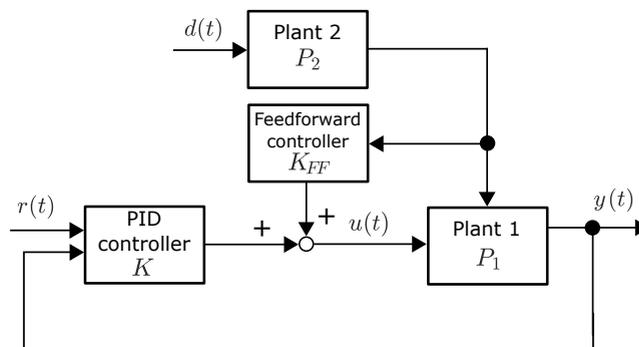


Figure 2.2: Block diagram of feedforward control.

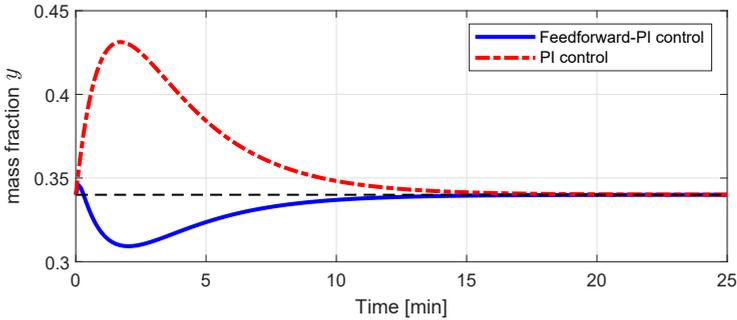


Figure 2.3: Comparison of feedforward-PI control (blue) and PI control (red).

can be an uncontrolled stable plant or even a closed-loop system. The disturbance sensor sends its measurement to the feedforward controller, adjusting the control signal.

This configuration is feasible only if the disturbance can be measured or estimated. We consider the case that the disturbance cannot be measured in Chapter 5 but introduce a disturbance observer. We consider that the disturbance can be measured in Chapter 7. In both chapters, we focus on wireless feedforward control. Let us illustrate the basic idea with an example.

**Example 2.1.** Consider a blending process with two chemical compounds [25]. The control objective is to ensure the desired mass fraction between the compounds. The main PI controller regulates the flow rate of one chemical. The disturbance sensor monitors the flow rate of another and sends its value for feedforward compensation. Plants 1 and 2 are given by

$$P_1(s) = \frac{0.0065}{0.392s^2 + 4.79s + 1}, \quad P_2(s) = 1,$$

respectively. We apply PI control and static feedforward control given by

$$K(s) = 21.9 + \frac{6.577}{s}, \quad K_{FF}(s) = -4.17.$$

Figure 2.3 shows the responses of feedforward-PI control and PI control, both with the setpoint  $r = 0.34$ . We see that the feedforward control reduces the deviation of the mass fraction. That is, feedforward-PI control has better disturbance rejection.

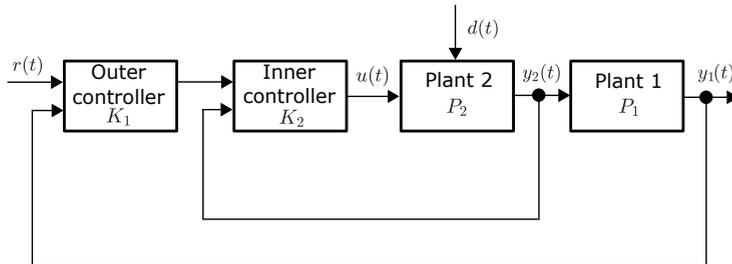


Figure 2.4: Block diagram of cascade control.

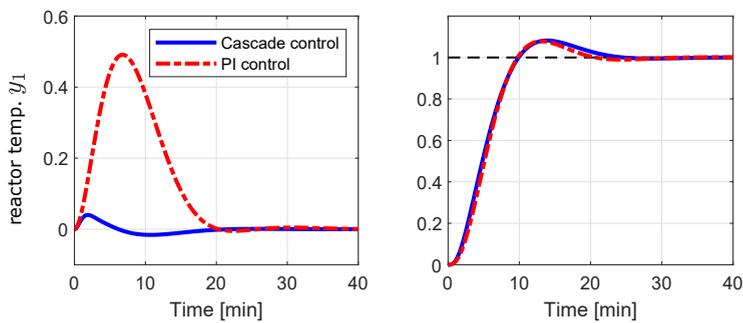


Figure 2.5: Comparison of cascade control (blue) and PI control (red). Left: Disturbance rejection. Right: Setpoint tracking.

## Cascade control

Feedforward control improves disturbance rejection, but the usage is limited to when the disturbance can be measured or estimated reliably. If this is not the case, cascade control should sometimes be considered, see Figure 2.4. Cascade control employs an inner control loop, which is meant to compensate for the disturbance before it appears in the main controlled variable. In cascade control, the outer PID controller computes its control signal as a setpoint for the inner controller. We illustrate again with an example.

**Example 2.2.** Consider the exothermic reactor in [25]. Its temperature is regulated by an outer PI control loop. However, the temperature is affected by the reactor's jacket temperature. The inner control loop regulates it so that its variation can be reduced before it affects the reactor's temperature. Plants 1 and 2 are

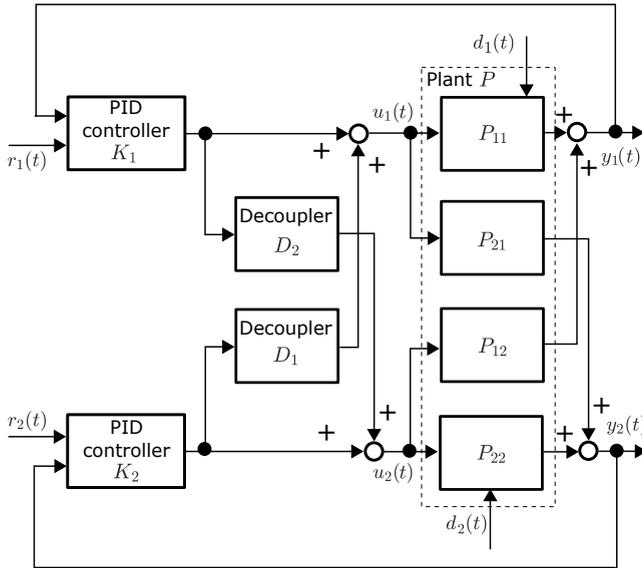


Figure 2.6: Block diagram of decoupling control.

given by

$$P_1(s) = \frac{1}{8s^2 + 6s + 1}, \quad P_2(s) = \frac{1}{s + 1},$$

respectively. We apply cascade control with

$$K_1(s) = 0.76 + \frac{0.22}{s}, \quad K_2(s) = 1.23 + \frac{3.73}{s},$$

and compare this to a simple PI controller

$$K_{PI}(s) = 0.94 + \frac{0.21}{s}.$$

Figure 2.5 shows the responses of cascade control and together PI control. From the plots, we can conclude that cascade control responds to the input disturbance promptly and much better than the PI control, while the setpoint tracking performance is almost the same.

## Decoupling control

Control signal variation of a feedback control loop may interact with other loops. In this case, introducing decouplers can be beneficial. Figure 2.6 shows

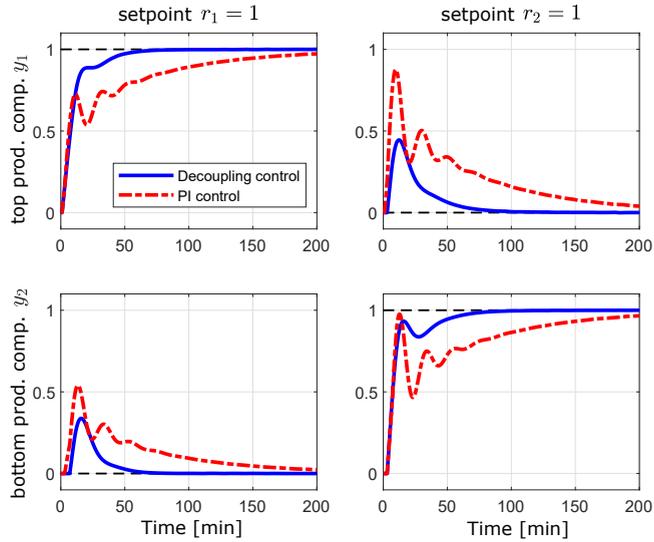


Figure 2.7: Comparison of decoupling control (blue) and PI control (red). Top left: Responses of the top production composition  $y_1$  to the setpoint change  $r_1 = 1$ . Top right: those of the top production composition  $y_1$  to  $r_2 = 1$ . Bottom left: those of  $y_2$  to  $r_1 = 1$ . Bottom right: those of  $y_2$  to  $r_2 = 1$ .

the block diagram of decoupling control, where the plant dynamics is given by

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}.$$

In decoupling control, decouplers adjust the corresponding control signals in a feedforward fashion so that the interactions are mitigated proactively, as illustrated in the following example.

**Example 2.3.** Consider a pilot-scale distillation column [26], given by

$$P(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s + 1} & \frac{-18.9e^{-3s}}{21s + 1} \\ \frac{6.6e^{-7s}}{10.9s + 1} & \frac{-19.4e^{-3s}}{14.4s + 1} \end{bmatrix}.$$

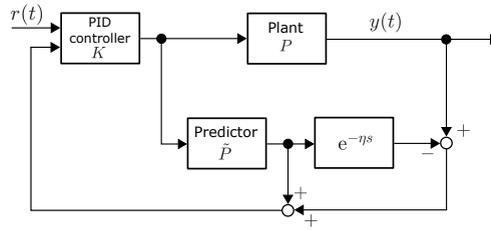


Figure 2.8: Block diagram of the Smith predictor.

We apply decoupling with PI control given by

$$\begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} = \begin{bmatrix} K_1(s) & D_1 K_2(s) \\ D_2 K_1(s) & K_2(s) \end{bmatrix} \begin{bmatrix} R_1(s) - Y_1(s) \\ R_2(s) - Y_2(s) \end{bmatrix}$$

where PI controllers  $K_1(s)$ ,  $K_2(s)$  and decouplers  $D_1$ ,  $D_2$  are given by

$$K_1(s) = 0.096 + \frac{0.014}{s}, \quad K_2(s) = -0.083 - \frac{0.008}{s},$$

$$D_1 = -1.477, \quad D_2 = 0.34.$$

Figure 2.7 shows decoupling control and PI control responses with the setpoints  $r_1 = 1$  and  $r_2 = 1$ . The left figures illustrate the responses to the setpoint change from  $r_1 = 0$  to 1 (the top shows  $y_1$  and bottom  $y_2$ ). The right shows those of  $r_2$ . We see that decoupling control reduces the deviation due to the setpoint changes for both setpoint changes.

## The Smith predictor

Process plants usually have time delays that are associated with physical movements of material or energy. The time delay may deteriorate the control performance or even make the system unstable. The Smith predictor is often introduced to compensate for time delays in process control applications. It was introduced in [27]. Since then, modifications were proposed to apply the predictor to integrator [28] and unstable systems [29, 30]. Figure 2.8 shows the block diagram of the Smith predictor. The Smith predictor introduces the plant model  $\hat{P}(s)$ , which predicts the plant output. By subtracting the predicted output from the actual output, the predictor can mitigate the effect of time delays on the closed loop, as illustrated next.

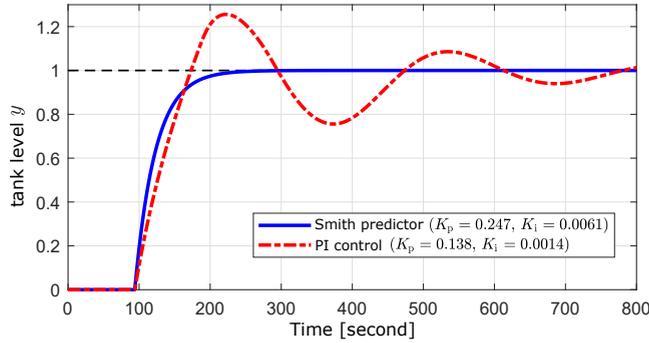


Figure 2.9: Comparison of the Smith predictor (blue) and PI control (red), where  $K_p$  and  $K_i$  is the proportional and integral gains, respectively.

**Example 2.4.** Consider the lab tank process introduced in [31] given by

$$P(s) = \frac{5.6}{40.2s + 1} e^{-93.9}.$$

We introduce the Smith predictor with PI controller

$$K(s) = 0.138 + \frac{0.0014}{s}, \quad \tilde{P}(s) = \frac{5.6}{40.2s + 1}.$$

A comparison between PI control with and without the Smith predictor is shown in Figure 2.9. We see that the Smith predictor yields no overshoot and almost the same rise time.

## 2.2 Wireless networks for industrial control systems

5G communication, cloud computing, and networked embedded devices are technologies supporting a wide range of applications, including mission-critical industrial control systems. WirelessHART [8] and ISA100.11a [9] have been explicitly proposed for the process industry. Both protocols employ the open systems interconnection (OSI) reference model [32], in which the physical layer (PHY) and the media access control (MAC) sublayer of the data link layer are specified by the low-rate wireless personal area network standard, IEEE 802.15.4 [33]. Wireless network behaviors, determined by such protocols,

affect control system performance. Any network protocol introduces packet dropout and delays. The network behaviors should, in some cases, be modeled and considered together with a control problem. In this section, we present a brief overview of 5G network communication and IEEE 802.15.4. Then we review work on communication protocols for industrial control systems.

### 2.2.1 5G network communication for industrial applications

5G network communication introduces wireless access to information and data “*anywhere and anytime for anyone and anything*” [34]. Its potential can meet the reliability and low-latency requirements of mission-critical industrial applications.

Two features of 5G are essential to industrial applications [35]:

- *Device-to-device communication*: Device-to-device communication refers to the technology that allows devices to communicate directly with each other, without involving network infrastructure such as access points and base stations [36]. Such direct communication reduces latency and increases data rate, resulting in reliable and efficient communication between sending and receiving devices.
- *Network slicing*: Network slicing is the technology that slices a physical network into several logical networks. A customized service becomes available for each application scenario while using the same physical network [37, 38]. This offers flexibility and scalability of the network. Industrial plants can be operated from a remote place and even from mobile devices. Information can be exchanged between plant sites and other parties.

5G network communication suggests the new architecture of industrial control systems shown in Figure 2.10 [4]. The current hierarchical structure for industrial control systems is dissolved and becomes seamless using 5G communication. In the right figure in Figure 2.10, any device can communicate with other devices regardless of the level. Thus, operators can monitor and control plants from remote places, and the management can easily access plant states and operation data, enabling rapid and flexible decision-making.

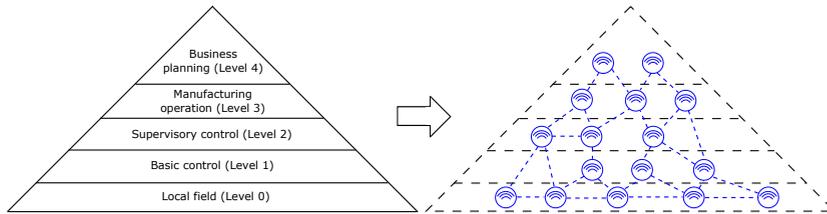


Figure 2.10: Dissolution of automation pyramid using 5G communication [4].

### 2.2.2 IEEE 802.15.4

IEEE 802.15.4 is the de-facto standard, specifying the physical layer and MAC sublayer for low-rate wireless personal area networks. Its application is widely expanding recently in many industries. The physical layer, the bottom layer in the OSI reference model, is responsible for transmitting and receiving a PHY protocol data unit (PPDU) and selecting operating channels. The PPDU consists of three parts: synchronization header (SHR), PHY header (PHR), and PHY payload, which is called MAC protocol data unit (MPDU), as depicted in Figure 2.11. The SHR is introduced for a receiving node to synchronize a stream of framed data, where the PHY includes the frame information. Three frequency bands, consisting of 27 channels in total, can be used for the data transmission. In particular, the frequency band 2400–2483.5 MHz, providing 16 channels, is available.

The MAC sublayer is responsible for transmitting and receiving the MPDU through the interaction with the physical layer. It also provides beacon management, channel access, and packet delivery mechanisms. The MPDU includes the main information (MAC payload) appended by the MAC layer between the MAC header (MHR) and MAC footer (MFR), as in Figure 2.11. The IEEE 802.15.4 MAC protocol introduces two channel access mechanisms: contention access period (CAP) and contention-free period (CFP) [15]. A superframe is an interval between two beacons, which is divided into usually 16 equally sized timeslots. The beacons are sent in the first timeslot to synchronize the network nodes. Timeslots in a superframe can be classified into two periods, as shown in Figure 2.12. The first period is CAP, where all network nodes are allowed to access the channel. Hence, collisions may occur. Carrier sense multiple access with collision avoidance (CSMA/CA) is introduced to handle possible collisions. The second period is the CFP. In the CFP, timeslots are allocated to devices that want to transmit data. This scheme is called

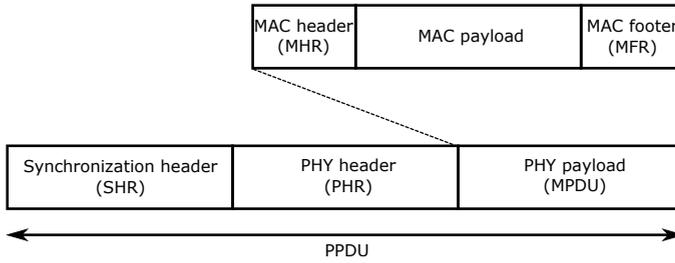


Figure 2.11: Frame structure of IEEE 802.15.4 PHY protocol. A packet consists of three parts: SHR, PHY, and PHY payload [33].

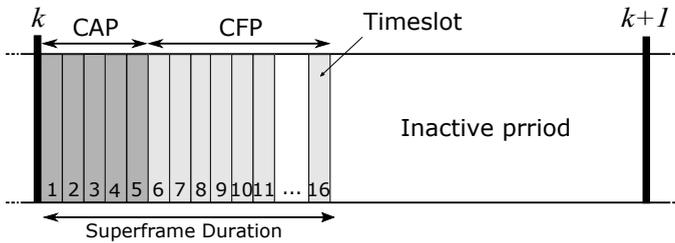


Figure 2.12: Superframe structure of IEEE 802.15.4 MAC protocol. Beacon interval between  $k$  and  $k + 1$  consists of three periods: CAP, CFP and inactive period.

time division multiple access (TDMA). At the end of all the transmissions, the network becomes inactive to save the batteries.

### 2.2.3 WirelessHART and ISA100.11a

Two major standards for wireless process control systems, WirelessHART and ISA100.11a, are available. In accordance with the IEEE 802.15.4, both WirelessHART and ISA100.11a have some common features, but there are also some differences [39, 40]. Both WirelessHART and ISA100.11a support star and mesh topologies. In WirelessHART, sensors and actuators can be used as relay nodes, while those in ISA100.11a are connected to routing devices in a star topology.

For the physical layer, both WirelessHART and ISA100.11a use the frequency band 2400–2483.5 MHz. Within this band, 15 channels with 2 MHz bandwidth spaced 5 MHz apart are available. Channel hopping is used in both

standards to decrease the effect of interference and noise. ISA100.11a applies more complicated channel hopping, such as slow hopping, slotted hopping, and a combination of them. In the MAC layer, WirelessHART uses TDMA for channel access, while ISA100.11a combines TDMA with CSMA. The duration of each time slot is 10 ms for both standards. Within the allocated time slot, the sending node can transmit a data packet to the receiving node. After the successful reception of a packet, the receiving node transmits an acknowledgment packet to the sending node.

Some researchers focus on developing, implementing, and validating control systems using WirelessHART [41, 42]. The authors of [43] discuss how to implement a WirelessHART mesh-structured network to satisfy a prespecified control performance. Scheduling of superframes is investigated in [44].

#### **2.2.4 Resource-aware communication protocols**

Much research has been devoted to wireless communication protocols based on the IEEE 802.15.4 standard to improve reliability and energy efficiency. Reliable and resource-aware wireless communication protocols for industrial control systems have been investigated in the communication community [10, 11, 17, 40, 45–50].

How to route data through a multi-hop network is an essential issue to reliable communication [51, 52]. A reliable real-time routing protocol and achieve synchronization TDMA protocol are proposed in [18] and [19], respectively. Both are evaluated in a paper mill [1]. Synchronization of a wireless sensor network using the Zigbee protocol [53] has been experimentally validated in a factory environment [54]. Several researchers investigate energy-efficient protocols. Routing algorithms for static wireless sensor networks are proposed in [55–57]. In [58], the Breath protocol, which minimizes the energy consumption subject to packet reliability and delay constraints, is developed. The protocol is experimentally evaluated and compared with the IEEE 802.15.4 protocol. An experimental testbed is built, and the evaluation of WirelessHART routing protocols is performed in [59]. Using the testbed, a reliable and energy-effective channel hopping algorithm is proposed. Time delay due to wireless communication defined by the IEEE 802.15.4 is modeled using a worst-case approach [60]. In [61], some adaptive sampling strategies for a wireless sensor network in a building environment are evaluated. Sensor scheduling for smart home applications is discussed in [62].

Data aggregation is an effective technique to reduce the energy consumption of wireless sensor networks [63]. The energy-effective protocol called the low energy adaptive clustering hierarchy (LEACH) protocol developed in [64] supports a data aggregation technique, where an energy consumption model of data sending and receiving is introduced. When a network node transmits data to another node, it consumes energy

$$E_s(p, d) = E_{\text{elec}}p + E_{\text{amp}}d^2p$$

where  $p$  bits is the amount of transmitted data and  $d$  the distance to the receiving node. The energy coefficient  $E_{\text{elec}}$  is determined by the electronics, coding, and other implementation aspects, and  $E_{\text{amp}}$  by the amplifier. While the consumption for sending is a function of the data amount and the distance, the energy consumption to receive data is given by

$$E_s(p) = E_{\text{elec}}p.$$

If a node collects multiple sensor data and sends them together in the same timeslot, the headers of the packet (Figure 2.11) can be shared. Assume that data from any sensors have  $c$  bits. Then the bits of information after aggregation is given by

$$p(q) = c[1 + (q - 1)(r - 1)]$$

where  $q \in \mathbb{N}$  is the number of sensor data and  $r \in [0, 1]$  is the data aggregation rate [65]. If  $r = 1$ , the data is aggregated perfectly, and the bits after aggregation is independent of the number of measurements. If  $r = 0$ , no packet aggregation is used. This model indicates that data aggregation, which reduces the size of a packet for each transmission, can reduce the amount of energy consumption. Many variations of the LEACH protocol have been proposed [66]. Data aggregation also has been considered in WirelessHART in [44].

## 2.3 Wireless process control systems

The ExxonMobil automation vision proposes a future process control architecture [5]. In this vision, control systems are expected to communicate more wirelessly and seamlessly. A simplified version of this architecture is illustrated in Figure 2.13. Operation and business platforms are implemented in the cloud, collecting data through a real-time service bus. By utilizing the powerful computation capacity of the cloud, various applications are available such as remote operation, data storage and analysis, and predictive maintenance.

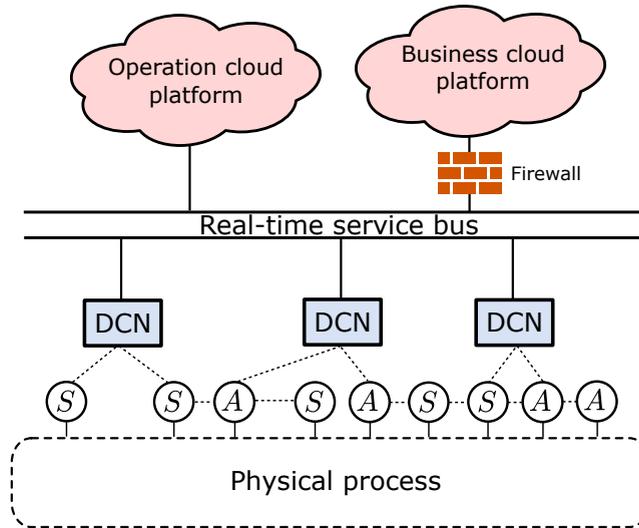


Figure 2.13: Future process control system architecture proposed in [5].

Basic control is carried out in the distributed control nodes (DCNs). Only a single control loop is assigned to each DCN, making the system more robust against controller failures. As DCNs only require simple control computation and I/O processing, they can be located closer to the field or even co-located at the sensors and actuators [4]. Sensors, controllers, and actuators are connected to their neighbor nodes, resulting in a multi-hop network. Various kinds of information from the nodes are exchanged through the network. Sensor data are sent to the corresponding controllers for feedback control and to operators for monitoring. Control signals are transmitted to the operators as well as to the corresponding actuators.

Although many protocols have been developed for wireless control systems, they are not yet widely adopted, mainly due to the strict performance and safety requirements of industrial control applications. Control over wireless networks is studied in the context of networked control theory, which in general focuses on control problems under network-induced constraints such as time-varying sampling intervals, delay, packet dropout, and channel access limitation [67, 68]. In networked control theory, various dynamical system models are introduced to express such network-induced phenomena. Time-varying sampling intervals and communication delays are modeled as hybrid

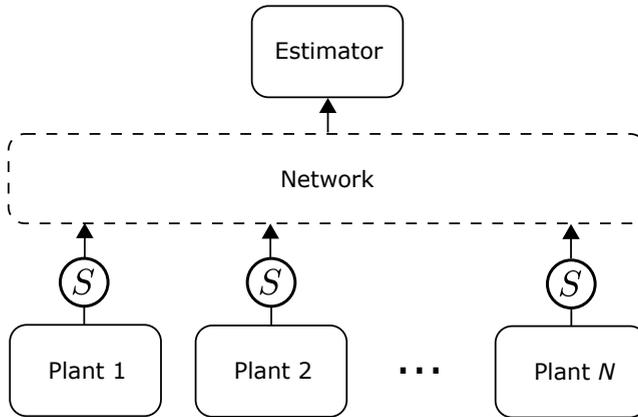


Figure 2.14: Block diagram of a monitoring system with remote estimation consisting of  $N$  plants and sensors. Each sensor monitors the corresponding plant and transmits its data (measurement or local estimate) to a remote estimator through a shared network.

systems in [69–71] and time-delay systems in [72, 73]. Estimation and control of noisy processes over a lossy communication channel are considered in the context of stochastic optimal control problems [74].

In this section, we go through an overview of this field of study. We first survey research on remote estimation. Second, studies on control over a network are summarized. Third, we discuss control over industrial wireless protocols. Finally, we overview the literature on event-triggered control.

### 2.3.1 Remote estimation for monitoring systems

In wireless process control systems, sensor data are transmitted to high-level services for monitoring, data storage, and analysis purposes, see Figure 2.14. Several research groups have investigated the remote estimation problem. Kalman filtering over unreliable communication channels is considered in [75]. It is shown that there exists a critical value for the failure probability, beyond which the error covariance becomes unbounded. Another performance metric is introduced in [76], where the proposed Kalman filter is given with the probability that the error covariance is bounded by a specific constant.

In [77], the stability of a scheduler under the trade-off between estimation

performance and communication cost is discussed. Random delay, as well as packet loss, is considered in [78]. A scheduling policy for multiples sensors measuring each independent plant is proposed in [79]. A stochastic sensor scheduling algorithm is proposed in [80], where multiple sensors monitor a plant, but the only one can access the estimator at every time instance.

Optimal estimation with a multiple time-step cost is introduced in [81, 82]. The authors consider a finite-time horizon and obtain a suboptimal schedule by introducing a relaxed convex optimization problem. The infinite horizon problem is considered in [83–85]. The authors of [86] derive conditions for the cost functions to be submodular to guarantee estimation performance. Schedules designed by greedy algorithms are studied in [87]. A minimum mean square error (MMSE) estimation schedule can be obtained in some special cases, for example, for two sensors in [88, 89] and more sensors in [90–92]. In [91, 92], optimal schedules are obtained by formulating Markov decision processes (MDPs). Remote estimation with variance-based triggering is proposed in [93], which yields a periodic transmission schedule. In this setup, sensors can directly communicate with the remote estimator through a common bus.

The studies above consider remote estimation under network constraints. Sensor energy consumption and packet dropout are explicitly considered for covariance-based state estimation in [94]. A threshold scheduling policy is obtained by formulating the optimal estimation problem as an MDP. Event-triggered estimator where a local sensor determines when to transmit its measurement is proposed in [95]. Deterministic online MMSE schedulers using feedback information from the remote estimator are proposed in [96, 97]. The scheduler in [98] introduces a time-out condition but uses only local information at each sensor. An MMSE stochastic event-trigger is proposed in [99]. An event-triggered estimators over shared communication channels with multiple sensors are considered in [100, 101]. Event-triggered estimation under unknown external disturbances is considered in [102]. A scheduler that determines transmission data based on the so-called Value of Information is proposed in [103]. Learning techniques are introduced to obtain a scheduler in [104, 105].

Sensor energy allocation problems are investigated in some studies. The energy consumption is then a control variable, which determines the probability of packet loss. Energy allocation for state estimation is discussed in [106–110]. In [111–113], energy harvesting sensors are considered for remote esti-

mation.

While the results above mainly consider single-hop networks, many industrial control systems need to be supported by multi-hop networks. Remote estimation over a multi-hop network is investigated in [114, 115]. The authors of [115] consider how to manage the remote estimator when the multi-hop network environment is changing. Then they propose a way to reconfigure the network under time-varying channel states.

In Chapter 3, we focus on a remote estimation problem over a multi-hop network. Motivated by the starch cooker example in Section 1.2, we consider remote estimation of a multiple-sensor system over a shared multi-hop network. The results differ from the literature in many ways. For instance, we introduce a superframe structure model to capture the features of many industrial communication protocols.

### 2.3.2 Control over wireless networks

Recently many results have been developed on control over wireless networks. Networked control systems are spatially distributed systems that use a shared communication network to exchange information from sensors to controllers and from controllers to actuators, see Figure 2.15. The results can be categorized into which types of networked-induced imperfections are considered.

#### **Control under variable sampling, delayed communication, and scheduling protocols**

Conventional control systems assume periodic sensing and actuation [116]. In networked control systems, however, this assumption cannot be preserved due to network-induced imperfections. Networks may impose variable sampling intervals and delayed communication. In [117–120], stability conditions for networked control systems with sampling interval smaller than the maximum allowable transmission interval (MATI) are derived. Communication delays are also considered in [69, 70]. Therein, stability conditions where the delays are smaller than the maximum allowable delay (MAD) are derived by introducing either a hybrid system [69] or a discrete-time system [70]. The authors of [121] consider channel access limitations, resulting in variable sensor sampling. The authors propose the try-once-discard (TOD) protocol as a way to decide the channel access schedule. The studies [69, 70, 117, 118, 120] discuss

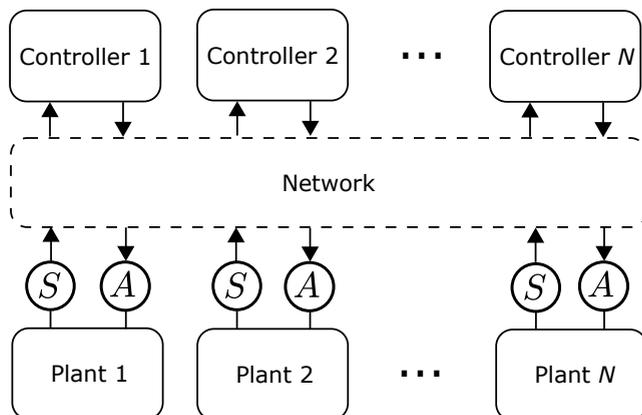


Figure 2.15: Block diagram of general networked control systems consisting of  $N$  control loops. In each loop, a sensor monitors the corresponding plant and transmits its data (measurement or local estimate) to the controller through a shared network. The controller sends the control command to the corresponding actuator.

the TOD protocols and the round-robin (RR) protocols. The TOD protocol is considered for systems with randomized sampling intervals and delays in [71] and with random process noise in [122].

Time-delay approaches are considered to model networked control systems with large communication delays, see [72, 73]. Time-varying sampling is discussed using a Lyapunov–Krasovskii functional in [123, 124]. The results are extended in [125, 126] based on Wirtinger’s inequality. Control systems with scheduling protocols have been investigated using time-delay approaches. In [127, 128], stability conditions are derived for systems under RR protocols. Both RR and TOD protocols are considered in [129].

In contrast to the results above, which focus on the way to handle time-varying intervals and delays induced by network communication, event-triggered control aims at reducing communication by intentionally varying the sampling interval. A significant number of results on event-triggered control have been obtained until now [130]. The details of these results are summarized in Subsection 2.3.4 below. In Part II, we introduce event-triggered control for process control loops. Stability analyses are given using the time-delay approach introduced in [125, 131].

## Design of networked controllers and schedulers

Designing controllers for networked control systems is investigated in many studies. Linear quadratic Gaussian (LQG) control through a lossy channel between sensor and controller is considered in [132]. The problem is divided into the standard linear quadratic regulator and the optimal encoder and decoder design problems. This result is extended to more general networks in [133] and to the case when the network between controller and actuator in [134]. LQG control with network-induced delays and access constraints is investigated in [135]. Network capacity is explicitly considered in [136–139], where the fundamental trade-offs between control performance and communication data-rate are investigated. LQG control subject to power constraints is studied in [140].

Scheduling data transmission for networked control systems has attracted attention. In [141], a joint optimization problem is presented where the problem can be separated into an optimal estimation, optimal control, and optimal scheduling problem. An event-triggering algorithm for control under packet dropout is considered in [142]. LQG control with packet dropouts and energy limitation is considered in [143], where the covariance-base sampling proposed in [94] is introduced. The authors of [144] investigate LQG control using multiple sensors. A co-design framework is proposed to minimize the number of sensors utilized. Inspired by the Wifi protocol, LQG control with adjustable bit-rate based on the signal-to-noise ratio is studied in [145]. Co-design of an LQG controller and scheduler, where the information is exchanged over a latency-varying network is studied in [146].

Scheduling among multiple control loops with a shared communication network is proposed in [147–149]. Scheduling under limited channel slots [147] and under a MAC-like protocol [148] are developed. Decentralized controllers are considered in [150] for an LQG problem, where a remote controller is introduced to cooperatively minimize a quadratic performance cost, while the uplinks from the controllers and the remote controller are unreliable. A latency-varying network shared by multiple control loops is considered in [151]. Learning techniques are introduced to obtain a scheduler for networked control systems with lossy communication channels in [152, 153].

Design of control systems over a multi-hop network is considered in some studies [154–156]. In [154], a mathematical model for representing and analyzing multi-hop networked control systems is proposed. Using the model, the authors of [157] investigate a co-design framework of the controller, schedul-

ing, and routing of multi-hop networked control systems. The result is extended to a robust design framework that is able to handle to permanent link failures [158] and resilient to malicious attacks [159]. Random packet dropout is considered in [160]. A co-design framework of multi-hop network scheduling and an optimal controller for a single process is proposed in [155]. A flexible control design framework is considered in [161], where control functionality can be allocated to any network node in a multi-hop network. A way to adaptively locate the controller over a lossy multi-hop network is proposed.

In Chapter 4, we focus on LQG control over a multi-hop network. This chapter differs from the previous work on LQG control over networks in that we consider sensor–actuator pairs and that the communication within each pair is carried out through a multi-hop network. Besides, we provide network reconfiguration algorithms handling when a link in the network is disconnected. Network reconfiguration needs to be investigated since factory environments are often exposed to variable channel characteristics.

### 2.3.3 Control over industrial wireless networks

Control over industrial wireless communication protocols, such as the IEEE 802.15.4 protocol, has been investigated by many researchers [10]. The authors of [162] propose a hybrid MAC protocol that switches between two modes: a Contention access MAC (CA-MAC) mode and a Contention-free MAC (CF-MAC) mode. The modes transit from CA-MAC to CF-MAC when a large disturbance occurs in the process. In [163], an aperiodic sampling algorithm and a MAC protocol with a scheduling algorithm are jointly designed. An experimental evaluation with water tanks is also conducted. A formal method is introduced to analyze control systems with IEEE 802.15.4 protocol in [164]. While these studies deal with both CSMA/CA and TDMA, the authors of [165–167] focus on CSMA/CA protocols.

The authors of [168–170] develop a network model that captures the WirelessHART protocol, and stability conditions are derived. In [171], co-design of the controller, scheduling, and routing algorithms with WirelessHART protocols is provided. The authors of [172] consider a multi-hop network protocol with faster sampling, together with a design of a stabilizing controller. The novelty of this work is to demonstrate fast sampling control over a multi-hop network by introducing a testbed consisting of synchronized multiple inverted pendulums. In [173], self-triggered communication is introduced for the multi-hop network with fast sampling.

In Chapters 3 and 4, we introduce a communication model that captures the features of WirelessHART. Sensors and actuators are also used as relay nodes, and the communications among them are carried out with a TDMA protocol. In particular, we focus on communication over a mesh-structured network. We employ the energy consumption model of [64], and energy-aware estimation and control over a TDMA network are developed.

### 2.3.4 Event-triggered control

Dealing with network-induced imperfections is an essential part of the studies of wireless control systems. One approach to such a problem is event-triggering sampling and actuation, so-called event-triggered control. In event-triggered control, sensors and/or controllers transmit their data only when a certain condition is satisfied. By doing so, communications among the system components can be reduced without sacrificing performance [174, 175]. This has motivated many researchers to study event-triggered control into a large variety of control applications [130, 176].

In [177], it is shown that the event trigger using an absolute value of the output generates an oscillatory behavior. The authors of [178] introduce a send-on-delta strategy, i.e., an event is generated when the output goes beyond a certain threshold from the last transmitted value. The strategy is considered under unknown constant disturbances in [179]. Relative threshold triggering for event-triggered actuation is introduced in [180]. In this strategy, the control signal is updated when a ratio between the absolute value of the measurement error and the actual measurement value deviates from the given threshold. The result is extended to output feedback control in [181]. The system consists of multiple sensors and actuators, and the events are generated by each individual component.

### Periodic event-triggered control

The above work requires continuous-time monitoring of the plants. To better suit practical implementation to such digitalized systems, self-triggered control is proposed in [182]. In self-triggered control, the event generator determines the next event time in advance already when the previous event was generated. The system does not need to monitor the plant continuously. Another approach is to check the event condition periodically, which is termed periodic event-triggered control, proposed in [183] for state feedback and

in [184] for output feedback control. Periodic event-triggered control is introduced for time-delay systems [185]. The authors of [186] consider time-delay systems, where time delays are artificially introduced to stabilize the plant. In [187], continuous monitoring is considered, but a minimum waiting time for two consecutive events is introduced to avoid the Zeno phenomenon.

### **Setpoint tracking and disturbance rejection**

Event-triggered control for setpoint tracking and disturbance rejection has been considered. In contrast to [179], which introduces a model-based approach to estimate the external disturbance, observer-based event-triggered control is introduced in [188, 189]. In [190], the authors introduce an observer-based event-triggered controller for constant reference signals. A co-design framework for the event trigger and controller is provided. The work [191] considers observer-based event-triggered control under parameter uncertainties as well as external disturbances. In [192], bounded slope nonlinearities are included.

### **Event-triggered PID control**

Event-triggered PID control has attracted the attention of many researchers since the early work [175]. In [193], a modified event-triggered PID controller is proposed and compared with the original setup in [175]. Some practical problems when introducing event-triggered PI control are discussed in [194, 195]. In [194], it is shown that event-triggered sampling may result in a sticking effect or large stationary oscillations. To overcome these problems, [194] proposes PIDPLUS [196–198]. Furthermore, [195, 199–203] focus on actuator saturation [204] for event-triggered control. Actuator saturation cannot be avoided since valves and pumps always have such a limitation. In [195], it is shown that an anti-windup technique can significantly improve the performance of event-triggered control systems under actuator saturation. The authors of [200] introduce an event-triggered anti-windup scheme, where the saturated control signal is fed back to the controller in an event-triggered fashion. While the works [195, 199] use a send-on-delta sampling strategy, and therefore only boundedness of the states are guaranteed, the authors of [201] derive asymptotic stability conditions by introducing a relative threshold strategy, as was proposed in [190]. External disturbances are included in [203]. In [205], a self-triggered control strategy is introduced for which the

event trigger is located at the controller. While both the setpoint tracking and disturbance rejection properties are investigated in this work, the application is limited to first-order systems. PI controller design problems are considered in [206, 207]. In [206], the authors introduce an LQ control design problem with event-triggered sampling. PI control synthesis with a relative threshold strategy is proposed in [207]. Event-triggered PID control is investigated for output feedback control with artificial delays in [186].

Some researchers focus on the validation of event-triggered control from both academic and industrial perspectives [208]. The event-triggered control proposed in [179] is evaluated using a chemical pilot plant in [209]. Event-triggered PI control with a send-on-delta strategy is validated on a double-tank system in [210], and with anti-windup compensation in [195]. Event-triggered PI control applied to DC-motors is considered in [207, 211]. In [212, 213], event-triggered PID control is evaluated on the industrial paper mill plant at Iggesund. To quantitatively evaluate event-triggered PID control, the authors of [214, 215] introduce benchmark problems.

In Chapter 5, we develop the event-triggered actuation for PID, cascade, and decoupling control. Event-triggered actuation is considered for setpoint tracking and disturbance rejection of multi-loop control systems. Chapter 6 extends this framework to time-delay systems. To compensate for time delays, we also consider the Smith predictor. Event-triggered sampling is considered in Chapter 7. We propose an event-triggered controller switching, in which one sensor transmits its measurements to the controller with a fixed sampling rate while another sensor transmits with a send-on-delta strategy. The switching framework can be applied to multi-sensor process control loops such as cascade and feedforward control and reduces sensor transmission while guaranteeing asymptotic stability.

## **Part I**

# **Multi-hop Network Scheduling**



## Chapter 3

# Multi-hop Network Scheduling for Remote Estimation

In process control systems, sensors transmit their measurements or estimates of the process states to the remote estimator at the control station for a monitoring purpose. Wireless process control systems employ multi-hop networks for communication among sensors and an estimator. Since battery limitation is imposed on the sensors, a systematic way to schedule communications among the elements must be considered. In other words, a scheduler that determines when and how to transmit the sensor information to the estimator under sensor energy considerations is desired.

This chapter considers a design framework of a scheduler for process state monitoring. We provide a framework of how to select and schedule a set of sensors to transmit their measurements efficiently over a time-synchronized multi-hop network. Our framework defines the links to be activated to transmit the sensor measurement for optimal remote estimation under sensor energy constraints when the sensors observe independent discrete-time linear time-invariant (LTI) systems. It is important to investigate estimation and control of multiple processes over a shared multi-hop network since previous works mainly deal with a single process. Different from the related work discussed in Chapter 2, the measurements are not directly sent to the estimator but through some intermediate nodes and a gateway. We consider a periodic superframe structure common to many existing wireless sensor net-

works for a medium access control (MAC) protocol. A superframe repeated every sampling interval is divided into timeslots. We assume only one point-to-point link is activated at a time. Then, by activating links in a certain order, the measurements of selected sensors can be efficiently conveyed to the estimator. The link activation is jointly determined with the sensor selection by considering data aggregation techniques and constrained by the energy consumption of the sensors.

The main contributions of this chapter are outlined as follows:

- We first find some structures of the multi-hop network schedule so that the problem can be decomposed into two subproblems. Then it is shown that this multi-hop network scheduling problem can be solved by formulating a Markov Decision Process (MDP).
- We exploit the MDP formulation to obtain a sufficient condition on the existence of a periodic optimal sensor network schedule.
- We provide algorithms to realize the periodic optimal schedule.
- We present algorithms to obtain suboptimal schedules to make our approach scalable for larger networks.
- The performance of the optimal and suboptimal algorithms are illustrated and evaluated in numerical examples. It is shown that the suboptimal algorithms effectively generate suboptimal schedules with slight performance degradation in small networks and is scalable to large networks.

The remainder of this chapter is organized as follows. Section 3.1 describes the system, including wireless network, process, communication, and energy consumption models, together with the remote estimator. The problem formulation is also presented. Section 3.2 presents the main result. Suboptimal schedules are obtained in Section 3.3. Numerical examples are provided in Section 3.4. Section 3.5 concludes this chapter.

### **3.1 Problem formulation**

In this chapter, we discuss an optimal remote estimation problem, where the estimator generates state estimates based on the received information from

sensors. The objective is to choose the network scheduler to minimize the estimation error subject to energy considerations. We elaborate on the main components of the system in the following subsections.

### 3.1.1 Wireless sensor network

A set of sensors  $\mathcal{V}_s \triangleq \{1, 2, \dots, N\}$  are deployed in an area, monitoring  $N$  decoupled discrete-time LTI processes. The sensors are interconnected via a wireless network and upload their measurements through the network to a remote estimator via a gateway. We denote the gateway as node 0, so the whole node set is given by  $\mathcal{V} \triangleq \mathcal{V}_s \cup \{0\}$ . The network is modeled by a directed graph  $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of communication links. The link  $(i, j)$  is included in  $\mathcal{E}$  if there is a link from node  $i$  to node  $j$ . For a link  $e = (i, j) \in \mathcal{E}$ , we introduce the maps to the sending node  $v_{\text{out}}(e) = i$  and to the receiving node  $v_{\text{in}}(e) = j$ . Let  $\mathcal{N}_i^{\text{in}}$  and  $\mathcal{N}_i^{\text{out}}$  denote the in- and out-neighbors of node  $i$ , respectively, i.e.,

$$\begin{aligned}\mathcal{N}_i^{\text{in}} &\triangleq \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}, \\ \mathcal{N}_i^{\text{out}} &\triangleq \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}.\end{aligned}$$

Furthermore, we denote  $d(e)$  as the distance between nodes  $i$  and  $j$ . By arranging an order for the links  $e_1, \dots, e_\ell, \dots, e_{|\mathcal{E}|}$ , the node-arc incidence matrix of the graph  $\mathcal{G}$  is defined as  $G \in \{-1, 0, 1\}^{(N+1) \times |\mathcal{E}|}$ , where  $(i, \ell)$ -th element of  $G$  is 1 if  $v_{\text{out}}(e_\ell) = i$ , and  $-1$  if  $v_{\text{in}}(e_\ell) = i$ , otherwise 0.

Figure 3.1 illustrates a network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $\mathcal{V} = \{0, 1, 2, 3\}$  and  $\mathcal{E} = \{(1, 0), (1, 3), (2, 0), (2, 3), (3, 1), (3, 2)\}$ . Assume that the links are arranged in ascending order,  $G$  is then given by

$$G = \begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 \end{bmatrix}.$$

For sensor 1, the distances to sensor 3 and to gateway 0 are expressed by  $d((1, 3)) = d((3, 1))$  and  $d((1, 0))$ , respectively. The in- and out-neighbors are given by  $\mathcal{N}_1^{\text{in}} = \{3\}$  and  $\mathcal{N}_1^{\text{out}} = \{0, 3\}$ .

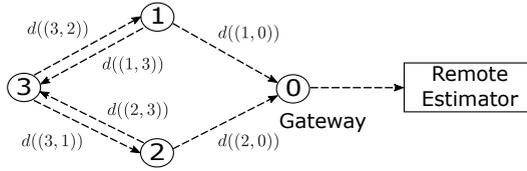


Figure 3.1: In a multi-hop wireless sensor network, each sensor transmits its data to a remote estimator through intermediate sensors and a gateway.

### 3.1.2 Process model

We consider  $N$  discrete-time LTI processes

$$x_{k+1}^{(i)} = A_i x_k^{(i)} + w_k^{(i)}, \quad i \in \mathcal{V}_s, \quad (3.1)$$

where  $x_k^{(i)} \in \mathbb{R}^n$  is the state of process  $i$  at time  $k$ ,  $w_k^{(i)} \in \mathbb{R}^n$  is process noise assumed to be Gaussian process with zero-mean independent and identically distributed with covariance  $W_i \triangleq \mathbb{E}[w_k^{(i)} (w_k^{(i)})^\top] > 0$ . The initial state  $x_0^{(i)}$ , independent of  $w_k^{(i)}$ ,  $k \in \mathbb{N}_0$ , is also assumed to be Gaussian with mean  $\mathbb{E}[x_0^{(i)}]$  and covariance  $\Sigma_0^{(i)}$ . Without loss of generality, we assume  $\mathbb{E}[x_0^{(i)}] = 0$ , as nonzero-mean can be translated into zero-mean by the coordinate change  $\tilde{x}_k^{(i)} = x_k^{(i)} - \mathbb{E}[x_0^{(i)}]$ . We assume that the state  $x_k^{(i)}$  can be observed directly by sensor  $i$ .

### 3.1.3 Communication model and network scheduling

The sensors communicate to the estimator through intermediate sensors and a gateway which define the underlying communication network. Time horizons of the sensors are partitioned into strips of identical time intervals, see Figure 3.2. Each time interval is divided into two phases: a sensing phase and a communication phase, where the former is a time period for sensor  $i$  to acquire the process state  $x_k^{(i)}$  and the latter is a time period for message delivery. The communication phase between time  $k$  and  $k + 1$ , which we call superframe at time instance  $k$ , is divided into  $L$  timeslots. Superframe structures are used in many industrial wireless communication protocols [8, 9, 53], built upon the IEEE 802.15.4 MAC layer [33]. These MAC schemes are characterized by time-division multiple access (TDMA) protocols. In TDMA protocols, some

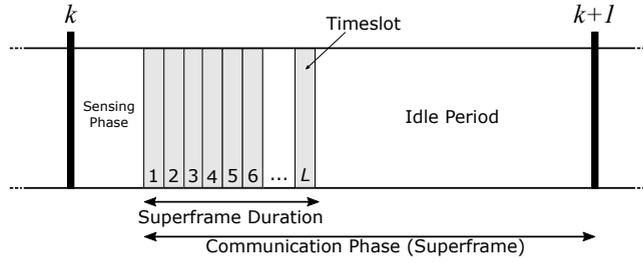


Figure 3.2: Each time interval is divided into a sensing and communication phase. The superframe duration is divided into timeslots. After the duration ends, sensors are in idle period [163].

given frequency channels are shared by the network nodes. At each timeslot, some links are allocated to the channels to transmit their data from a sending node to a receiving node. By repeating this, the data will finally arrive at the gateway. To model the protocols, we make the following natural assumptions for the communication.

**Assumption 3.1.** *We assume the following properties:*

- (i). *All sensors have the same sampling interval and are perfectly time-synchronized.*
- (ii). *Data can be transmitted among the nodes without failure, i.e., no packet dropout occurs.*
- (iii). *The number  $L$  of timeslots in a single superframe is sufficiently large for accommodating all links in  $\mathcal{G}$ .*

The data packet generated by sensor  $i$  is a tuple ('index', 'time', 'value'), where 'index' indicates the sensor index, 'time' the timestamp when the data is generated, 'value' the measurement value. Thus, formally we can describe the data from sensor  $i$  generated at time  $k$  as  $(i, k, x_k^{(i)})$ . In this chapter, for the sake of presentation simplicity, we assume that only one frequency channel is available<sup>1</sup>. Thus, at most one link is activated at each timeslot  $\ell$  of superframe  $k$ , i.e., a link  $e$  is determined by a pair  $(k, \ell)$ . To indicate this link, we

<sup>1</sup>Under Assumption 3.1, the results in this chapter can be straightforwardly extended to the case of multiple frequency channels.

denote the link activated at timeslot  $\ell$  of superframe  $k$  as  $e(k, \ell)$ . Let us denote  $\mathcal{I}^{(i)}(k, \ell)$  as the data set that sensor  $i$  holds at timeslot  $\ell \in \mathcal{L} \triangleq \{1, \dots, L\}$  in superframe  $k$ , and let  $\mathcal{D}(k, \ell) \subseteq \mathcal{I}^{(i)}(k, \ell)$  with  $i = v_{\text{out}}(e(k, \ell))$  and  $j = v_{\text{in}}(e(k, \ell))$  be the set of data transmitted by sensor  $i$ . That is,

$$\mathcal{D}(k, \ell) \triangleq \mathcal{I}^{(j)}(k, \ell + 1) \setminus \mathcal{I}^{(j)}(k, \ell).$$

The set of the sensor indices ('index') of which the data is to be transmitted through  $e(k, \ell)$  is expressed by

$$\mathcal{S}(k, \ell) \triangleq \left\{ i \in \mathcal{V}_s : (i, k', x_{k'}^{(i)}) \in \mathcal{D}(k, \ell), k' \leq k \right\}.$$

Then, given the initial data set  $\mathcal{I}^{(i)}(-1, L)$ ,  $\mathcal{I}^{(i)}(k, \ell)$  can be recursively written as

$$\mathcal{I}^{(i)}(k, \ell) = \begin{cases} \mathcal{I}^{(i)}(k-1, \ell) \oplus (i, k, x_k^{(i)}), & \text{if } \ell = 1, \\ \mathcal{I}^{(i)}(k, \ell-1) \oplus \mathcal{D}(k, \ell), & \text{if } \ell \geq 2, i = v_{\text{in}}(e(k, \ell-1)), \\ \mathcal{I}^{(i)}(k, \ell-1), & \text{if } \ell \geq 2, i \neq v_{\text{in}}(e(k, \ell-1)), \end{cases}$$

where the operation  $\mathcal{I} \oplus \mathcal{D}$  is union but only the data packet with large timestamp ('time') is preserved if  $\mathcal{I}$  and  $\mathcal{D}$  hold measurements from the same sensor. When  $L$  timeslots terminate, the gateway transmits all the measurement  $\mathcal{D}_k \triangleq \mathcal{I}^{(0)}(k, L)$  to the estimator. We denote the elapsed time of the data from sensor  $i$  in  $\mathcal{D}_k$  as  $\tau_k^{(i)}$  which can be calculated from the current time and the timestamp ('time'). Assuming that the data of sensor  $i$  in  $\mathcal{D}_k$  is generated at time  $k'$ , and therefore described as  $(i, k', x_{k'}^{(i)})$ , then we write  $\tau_k^{(i)} = k - k'$ .

The gateway is responsible for coordinating which sensors to be activated at which timeslots and which communication links to be established. This function is called network scheduling. That is, the gateway decides the network schedule

$$\mathcal{C}_k \triangleq ((e(k, 1), \mathcal{S}(k, 1)), \dots, (e(k, L), \mathcal{S}(k, L)))$$

given the available information after superframe duration  $k$  (after timeslot  $L$ ) denoted  $\mathcal{K}_k^{(0)} \triangleq \{\mathcal{C}_{0:k}, \mathcal{D}_{0:k}\}$ . The network scheduler chooses the schedule  $\mathcal{C}_{k+1}$  for the next superframe according to

$$\mathcal{C}_{k+1} = f_k(\mathcal{K}_k^{(0)})$$

where  $f_k$  is the map from the set of available information at the gateway to the set of network schedules. We also define  $\mathbf{f} \triangleq (f_k)_{k \in \mathbb{N}_0}$  as the network scheduling strategy.

### 3.1.4 Energy consumption

The sensors consume a certain amount of energy when they receive data from and transmit data to other sensors. Here we introduce an energy consumption model often employed in wireless communication protocols [64]. The energy consumption for receiving a packet, which contains  $p$  bits information, is

$$E_r(p) = E_{\text{elec}}p \quad (3.2)$$

where the energy coefficient  $E_{\text{elec}}$  is determined by the electronics, coding and other implementation aspects. The energy consumption for sending  $p$  bits information is

$$E_s(p, d) = E_{\text{elec}}p + E_{\text{amp}}d^2p \quad (3.3)$$

where  $E_{\text{amp}}$  is the energy coefficient for the amplifier and  $d$  is the distance to the receiving sensor or gateway. When transmitting multiple measurements, a sensor can aggregate them into a single packet in order to reduce the transmission overhead. This technology is called packet aggregation [63]. Assume that a single measurement from any sensor has  $c$  bits. Then the bits of information after aggregation is given by

$$p(q) = c[1 + (q - 1)(1 - r)] \quad (3.4)$$

where  $q \in \mathbb{N}$  is the number of measurements and  $r \in [0, 1]$  is the data aggregation rate [65]. If  $r = 1$  the data is aggregated perfectly and the bits after aggregation are independent of the number of measurements, which is, for instance, the case for the LEACH protocol [64]. If  $r = 0$ , no packet aggregation is used. Notice that it is difficult to aggregate collected data from different sensors perfectly, but some parts of the data such as header can be removed when aggregating.

Let  $q(k, \ell) \triangleq |\mathcal{S}(k, \ell)|$  be the number of measurements transmitted to node  $v_{\text{in}}(e(k, \ell))$ . Notice that  $q(k, \ell)$  is determined by the network schedule  $\mathcal{C}_k$ , so the total energy consumption for sensor  $i$  to receive and send packets in the superframe at time  $k$  is given by

$$\begin{aligned} E_k^{(i)}(\mathcal{C}_k) = & \sum_{\ell: v_{\text{in}}(e(k, \ell))=i} E_r(p(q(k, \ell))) \\ & + \sum_{\ell: v_{\text{out}}(e(k, \ell))=i} E_s(p(q(k, \ell)), d(e(k, \ell))). \end{aligned} \quad (3.5)$$

### 3.1.5 Remote estimation

After superframe duration  $k$ , the remote estimator computes an estimate

$$\hat{X}_k \triangleq \left( \hat{x}_k^{(1)}, \dots, \hat{x}_k^{(N)} \right)$$

where  $\hat{x}_k^{(i)}$  denotes the estimate of  $x_k^{(i)}$ . Let  $\mathcal{K}_k^{(R)}$  denote the information set at the estimator. Notice that the sensor measurements sent by node 0 and the estimation history are accessible to the remote estimator. In other words, the information available to the remote estimator is

$$\mathcal{K}_k^{(R)} \triangleq \left\{ \hat{X}_{0:k-1}, \mathcal{D}_{0:k} \right\}.$$

In this chapter, as a metric of the estimator performance, we use the mean square error  $\mathbb{E}[(\epsilon^{(i)})^\top \epsilon^{(i)}]$  with  $\epsilon^{(i)} \triangleq x_k^{(i)} - \hat{x}_k^{(i)}$ . Note that the optimal estimate for process  $i$  is computed recursively following the modified Kalman filter [75, 76]:

$$\begin{aligned} \hat{x}_k^{(i)} &= \mathbb{E}[x_k^{(i)} | \mathcal{K}_k^{(R)}] \\ &= \mathbb{E}[x_k^{(i)} | \mathcal{K}_{k-1}^{(R)}, \hat{X}_{k-1}, \mathcal{D}_k] \\ &= A^{\tau_k^{(i)}} x_{k-\tau_k^{(i)}}^{(i)}, \end{aligned} \quad (3.6)$$

with initial estimate  $\hat{x}_0^{(i)} = 0$ . Correspondingly, the error covariance of  $x_k^{(i)}$  is denoted as

$$P_k^{(i)} \triangleq \mathbb{E} \left[ (x_k^{(i)} - \hat{x}_k^{(i)})(x_k^{(i)} - \hat{x}_k^{(i)})^\top | \mathcal{K}_k^{(R)} \right].$$

Note that possible values of the error covariance are included in a set

$$P_k^{(i)} \in \{0, h_i(0), h_i^2(0), \dots\}, \quad i \in \mathcal{V}_s, \quad \forall k \in \mathbb{N}_0, \quad (3.7)$$

where  $h_i : \mathbb{S}_n^+ \rightarrow \mathbb{S}_n^+$  is the operator  $h_i(X) = A_i X A_i^\top + W_i$ , and  $h_i^n(X)$  is the  $n$ -hold composition of  $h_i(\cdot)$  with  $h_i^0(X) = X$  since  $P_k^{(i)}$  evolves with  $h_i(\cdot)$  from 0 once the estimator receives the measurement [94]. Then the error conariance is computed as  $P_k^{(i)} = h_i^{\tau_k^{(i)}}(0)$  and we have

$$\mathbb{E}[(\epsilon^{(i)})^\top \epsilon^{(i)}] = \text{tr} \left( h_i^{\tau_k^{(i)}}(0) \right). \quad (3.8)$$

With this, the estimation error (3.8) is determined only by  $\tau_k^{(i)}$ , which is included in  $\mathcal{D}_k$ .

### 3.1.6 Problem formulation

The problem of interest is to find an optimal network scheduling strategy that minimizes long-term estimation errors penalized by sensor transmission energy usage. We define the cost at time  $k$  as

$$C(\mathcal{C}_k, \mathcal{D}_k) \triangleq \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau_k^{(i)}}(0) \right) + E(\mathcal{C}_k)$$

where  $E(\mathcal{C}_k) \triangleq \sum_{i \in \mathcal{V}_s} \beta_i E_k^{(i)}(\mathcal{C}_k)$  with  $\beta_i > 0$ . We formulate the following problem:

**Problem 3.1.**

$$\min_{\mathbf{f}=(f_0, f_1, \dots)} J(\mathbf{f}) \triangleq \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} C(\mathcal{C}_k, \mathcal{D}_k). \quad (3.9)$$

**Remark 3.1.** *Problem 3.1 jointly optimizes a weighted average of the estimation error and sensor energy consumption. Minimization of (3.9) with given values of  $\beta_i$  corresponds to a minimum-cost schedule with energy consumption constraint given by some  $\alpha_i > 0$ :*

$$\begin{aligned} \min_{\mathbf{f}} \quad & \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} \sum_{i \in \mathcal{V}_s} \mathbb{E} \left[ (\epsilon^{(i)})^\top \epsilon^{(i)} \right], \\ \text{s.t.} \quad & \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} E_k^{(i)}(\mathcal{C}_k) \leq \alpha_i, \quad i \in \mathcal{V}_s. \end{aligned}$$

For Problem 3.1 to be well-posed, we make the following assumption.

**Assumption 3.2.** *The graph  $\mathcal{G}$  contains a spanning tree with the root being the gateway node 0.*

Assumption 3.2 guarantees that persistently exciting protocols [118] can be configured over the network  $\mathcal{G}$ . Therefore, Problem 3.1 is well-posed as long as  $\mathcal{G}$  contains a spanning tree.

## 3.2 Structures of network scheduler

In this section, we discuss structural properties of the network scheduler solving Problem 3.1. First, we show that an optimal schedule requires the network to carry sensor data within a superframe through a tree network formed by a set of activated links, by which the search space for an optimal strategy can be reduced. With this finding, we manage to separate Problem 3.1 into two subproblems: network routing and sensor selection.

### 3.2.1 Necessary conditions for network schedule optimality

We consider all communication links within a single superframe jointly and analyze the resulting graph by treating these links as a whole, where the notion of joint graph arises. Let us define the joint graph for a superframe  $k$  under a network scheduling strategy  $\mathbf{f}$  in the following way. Denote  $\mathcal{E}_k \triangleq (e(k, 1), \dots, e(k, L))$  the sequence of communication links in the superframe  $k$  selected from  $\mathcal{E}$  of the underlying graph  $\mathcal{G}$ . Then we call  $\mathcal{G}_k \triangleq (\mathcal{V}, \mathcal{E}_k)$  the joint graph of the superframe  $k$ . Let us also denote  $\mathcal{S}_k \subseteq \mathcal{V}_s$  as the set of sensor indices that the latest data  $(i, k, x_k^{(i)})$  departs sensor  $i$  at the one of timeslots in superframe  $k$ , i.e.,  $i \in \mathcal{S}_k$  if and only if there exists  $\ell \in \mathcal{L}$  such that the sending node of  $e(k, \ell)$  is  $i$  and its data are included in this transmission. That is,

$$\mathcal{S}_k \triangleq \left\{ i \in \mathcal{V}_s : \exists \ell \in \mathcal{L} \text{ s.t. } i = v_{\text{out}}(e(k, \ell)), i \in \mathcal{S}(k, \ell) \right\}.$$

For an optimal scheduling strategy  $\mathbf{f}^* = (f_0^*, \dots, f_k^*, \dots)$ , denote the optimal network schedule at time  $k$  as  $\mathcal{C}_k^* = (e^*(k, \ell), \mathcal{S}^*(k, \ell))_{\ell=1}^L$ , and the optimal set of the communication links and joint graph as  $\mathcal{E}_k^*$  and  $\mathcal{G}_k^*$ , respectively. Furthermore, under a given optimal network schedule  $\mathcal{C}_k^*$ , we denote the index set  $\mathcal{S}_k$  and the data set  $\mathcal{D}_k$  as  $\mathcal{S}_k^*$  and  $\mathcal{D}_k^*$ , respectively. Then we have the following lemma.

**Lemma 3.1.** *Suppose that Problem 3.1 has an optimal solution  $\mathbf{f}^*$ . Then the followings hold:*

- (i). *If  $i \in \mathcal{S}_k^*$ , then  $(i, k, x_k^{(i)}) \in \mathcal{D}_k^*$ .*
- (ii). *The joint graph  $\mathcal{G}_k^*$  is a tree with node 0 being its unique root.*

*Proof.* Suppose that there exists  $\ell \in \mathcal{L}$  such that  $i = v_{\text{out}}(e^*(k, \ell)) \in \mathcal{S}^*(k, \ell)$ . Obviously, the data  $(i, k, x_k^{(i)})$  arrives at the gateway through a single path without a circle path from sensor  $i$  to the gateway. Let the arrival time of the data  $(i, k, x_k^{(i)})$  be  $k + m, m \in \mathbb{N}_0$ . We show that  $m = 0$  for an optimal network schedule. The proof is by contradiction. Suppose that  $m > 0$ . Consider a sequence of graphs  $\tilde{\mathcal{G}}_{k:k+m} \triangleq (\tilde{\mathcal{G}}_k, \dots, \tilde{\mathcal{G}}_{k+m})$  which is the same as  $\mathcal{G}_{k:k+m}^* \triangleq (\mathcal{G}_k^*, \dots, \mathcal{G}_{k+m}^*)$  except that links  $e \in \mathcal{G}_{k:k+m}^*$  that are used to transmit  $x_k^{(i)}$  and the measurements aggregated into  $x_k^{(i)}$  are removed, but rescheduled in  $\tilde{\mathcal{G}}_{k+m}$  with the latest data  $x_{k+m}^{(i)}$ . Notice that  $\mathcal{G}_{k:k+m}^*$  and  $\tilde{\mathcal{G}}_{k:k+m}$  consume the same or smaller amount of energy, but  $\tilde{\mathcal{G}}_{k:k+m}$  has smaller estimation error due to the monotonicity of  $h_i(\cdot)$  starting from  $X = 0$  [89]. This contradicts the optimality of  $\mathcal{G}_{k:k+m}$ . Thus,  $m = 0$ , hence  $(i, k, x_k^{(i)}) \in \mathcal{D}_k$ . The second statement is obvious from  $m = 0$ . The proof is now completed.  $\square$

**Remark 3.2.** *Lemma 3.1 suggests that data  $(i, k, x_k^{(i)})$  will arrive at the remote estimator within superframe  $k$  through tree graph  $\mathcal{G}_k^*$  if it departs from sensor  $i$  in superframe  $k$ .*

We give another lemma that indicates the order of link activations in an optimal network schedule. Suppose that the network schedule satisfies (i) and (ii) of Lemma 3.1. We introduce a partial order to the links in tree graph  $\mathcal{G}_k$ . That is, for any  $e, e' \in \mathcal{E}_k$ , we say  $e \succeq e'$  if there exists a directed path from  $v_{\text{in}}(e)$  to  $v_{\text{out}}(e')$ . It defines a partial order on  $\mathcal{E}_k$  since we can readily show that it is reflexive, antisymmetric, and transitive.

**Lemma 3.2.** (*Upstream-first rule*) *Suppose that Problem 3.1 has an optimal solution  $\mathbf{f}^*$ . Then,  $\ell_1 \leq \ell_2$  if  $e^*(k, \ell_1) \succeq e^*(k, \ell_2)$  for  $e^*(k, \ell_1), e^*(k, \ell_2) \in \mathcal{E}_k^*$ .*

*Proof.* By letting each sensor  $i$  in  $\mathcal{G}_k^*$  send  $x_k^{(i)}$  following upstream-first order, all measurements sampled and sent within the superframe  $k$  reach node 0 free of delays. Otherwise, a part of measurements received by node 0 will arrive with delays. In other words, any strategy  $\mathbf{f}$  in this case can never be optimal.  $\square$

**Remark 3.3.** *The upstream-first rule requires each sensor  $i$  in  $\mathcal{G}_k$  to wait until all the scheduled upstream sensor data arrive. After their arrival, sensor  $i$  transmits its data to its downstream neighbor node. It is immaterial in what order of the upstream branches of node  $i$  are activated for transmission.*

Lemmas 3.1 and 3.2 jointly suggest that, to construct a network schedule, it is essential to select which sensors need to transmit data to the remote estimator and to plan communication paths. The sensor selection fully determines the estimation error while the communication paths fully determines the communication cost. To investigate an optimal network scheduling strategy, we only need to focus on the path planning of data communication and the sensor selection. These two steps are separably studied in the sense that given a selected sensor set, we only need to account for the communication cost when we plan the communication paths. Therefore, in the sequel, we will investigate two subproblems: tree planning and sensor selection. The tree planning is studied with respect to sensor energy cost when a subset of sensors is selected. Then the sensor selection is investigated given the optimal communication paths.

### 3.2.2 Tree planning subproblem

In the previous subsection, we see that  $\mathcal{G}_k^*$  should be always a tree with the unique root node 0 and the links are activated according to the upstream-first rule. In this subsection, we introduce a necessary condition to satisfy the statements (i) and (ii) of Lemma 3.1. Imposing this condition to  $\mathcal{E}_k$ , we formulate an integer linear problem called the tree planning problem, which gives a tree  $\mathcal{G}_k$  minimizing the energy consumption  $E(\mathcal{C}_k)$ .

Let  $z_k^{(i)}(e) \in \{0, 1\}$  be an index function for any  $i \in \mathcal{V}_s$ , denoting whether  $(i, k, x_k^{(i)})$  is transmitted through link  $e \in \mathcal{E}$  at time  $k$ . That is,  $z_k^{(i)}((j, m)) = 1$  if there exists  $\ell \in \mathcal{L}$  such that  $e(k, \ell) = (j, m)$  and  $i \in \mathcal{S}(k, \ell)$ , otherwise 0. To fulfill conditions (i) and (ii) of Lemma 3.1, it is necessary to satisfy the following constraints:

- (i). Each node in  $\mathcal{S}_k$  has outgoing flow of its own measurement. That is, for  $i \in \mathcal{S}_k$ ,

$$\sum_{m \in \mathcal{N}_i^{\text{out}}} z_k^{(i)}((i, m)) - \sum_{m \in \mathcal{N}_i^{\text{in}}} z_k^{(i)}((m, i)) = 1. \quad (3.10)$$

- (ii). The gateway has only incoming flow. That is, for  $i \in \mathcal{S}_k$ ,

$$\sum_{m \in \mathcal{N}_0^{\text{in}}} z_k^{(i)}((m, 0)) = 1. \quad (3.11)$$

(iii). Intermediate nodes of a path obey the flow balance. That is, for  $i \in \mathcal{S}_k$  with  $j \neq i$ ,

$$\sum_{m \in \mathcal{N}_j^{\text{out}}} z_k^{(i)}((j, m)) - \sum_{m \in \mathcal{N}_j^{\text{in}}} z_k^{(i)}((m, j)) = 0. \quad (3.12)$$

(iv). The nodes that are not in  $\mathcal{S}_k$  obey the flow balance. That is, for  $i \notin \mathcal{S}_k$ , the constraint (3.12) hold.

Let  $z_k^{(i)} = [z_k^{(i)}(e_1), \dots, z_k^{(i)}(e_{|\mathcal{E}|})]^\top \in \{0, 1\}^{|\mathcal{E}|}$  be the vector of index functions for node  $i$ , where links are aligned in an appropriate order, and

$$z_k = [z_k^{(1)\top}, \dots, z_k^{(N)\top}]^\top \in \{0, 1\}^{|\mathcal{E}| \cdot N}. \quad (3.13)$$

Then, using the node–arc incidence matrix  $G$ , the constraints (3.10)–(3.12) can be written in a compact form

$$Gz_k^{(i)} = b^{(i)}(\mathcal{S}_k), \quad i \in \mathcal{V}_s, \quad (3.14)$$

where  $b^{(i)}(\mathcal{S}_k) \in \mathbb{R}^{N+1}$  is a vector with elements taking one of the values  $0, \pm 1$  according to the right terms of (3.10)–(3.12).

**Example 3.1.** As an example of the flow constraint (3.14), consider the network shown in Figure 3.1. For node 1, we denote

$$z_k^{(1)} = \begin{bmatrix} z_k^{(1)}((1, 0)) \\ z_k^{(1)}((1, 3)) \\ z_k^{(1)}((2, 0)) \\ z_k^{(1)}((2, 3)) \\ z_k^{(1)}((3, 1)) \\ z_k^{(1)}((3, 2)) \end{bmatrix}.$$

Then, by (3.10)–(3.12), we obtain (3.14) when  $1 \in \mathcal{S}_k$  as

$$\begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 \end{bmatrix} z_k^{(1)} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

where the left term matrix corresponds the node–arc incidence matrix  $G$  and the right term vector  $b^{(1)}(\mathcal{S}_k)$ .

Since  $E(\mathcal{C}_k)$  is only a function of the network schedule at the current time  $k$ , the variable  $z_k$  that satisfies the constraints (3.10)–(3.12) can be pre-calculated with fixed  $\mathcal{S}_k$ . That is, at time  $k$ , given the set  $\mathcal{S}_k$ , recalling the energy consumption model (3.2)–(3.5) and the definition (3.13) of  $z_k$ , we obtain a tree network graph  $\mathcal{G}_k$  by solving the following problem:

**Problem 3.2.** (*Tree planning subproblem*)

$$\begin{aligned} E_{\min}(\mathcal{S}_k) \triangleq \min_{z_k} \quad & E(\mathcal{C}_k) = \sum_{e \in \mathcal{E}} c\eta(e) \left[ (1-r) \sum_{i \in \mathcal{V}_s} z_k^{(i)}(e) + r \max_{i \in \mathcal{V}_s} z_k^{(i)}(e) \right] \\ \text{s.t.} \quad & Gz_k^{(i)} = b^{(i)}(\mathcal{S}), \quad i \in \mathcal{V}_s, \\ & z_k^{(i)}(e) \in \{0, 1\}, \quad i \in \mathcal{V}_s, \quad e \in \mathcal{E}, \end{aligned}$$

where

$$\eta(e) \triangleq \begin{cases} \beta_{v_{\text{out}}(e)}(E_{\text{elec}} + E_{\text{amp}}d^2(e)) + \beta_{v_{\text{in}}(e)}E_{\text{elec}}, & \text{if } v_{\text{in}}(e) \in \mathcal{V}_s, \\ \beta_{v_{\text{out}}(e)}(E_{\text{elec}} + E_{\text{amp}}d^2(e)), & \text{if } v_{\text{in}}(e) \in \{0\}. \end{cases}$$

Problem 3.2 is a binary integer problem, which is in general NP-hard. Nevertheless, due to a special algebraic property of the constraints, we manage to find the global minimizer of Problem 3.2 by solving a relaxed problem. The result is formally presented as follows.

**Theorem 3.1.** *A vector  $z^* \in \{0, 1\}^{N|\mathcal{E}|}$  is a minimizer of Problem 3.2 if and only if it is a minimizer of the following problem:*

**Problem 3.3.**

$$\begin{aligned} \min_{z_k, t} \quad & \sum_{e \in \mathcal{E}} c\eta(e) \left[ (1-r) \sum_{i \in \mathcal{V}_s} z_k^{(i)}(e) + rt(e) \right] \\ \text{s.t.} \quad & Gz_k^{(i)} = b^{(i)}(\mathcal{S}_k), \quad i \in \mathcal{V}_s, \end{aligned} \tag{3.15}$$

$$0 \leq z_k^{(i)}(e) \leq 1, \quad i \in \mathcal{V}_s, \tag{3.16}$$

$$z_k^{(i)}(e) \leq t(e), \quad i \in \mathcal{V}_s, \quad e \in \mathcal{E}, \tag{3.17}$$

$$t(e) \in \{0, 1\}, \quad e \in \mathcal{E}.$$

*Proof.* See Appendix 3.A. □

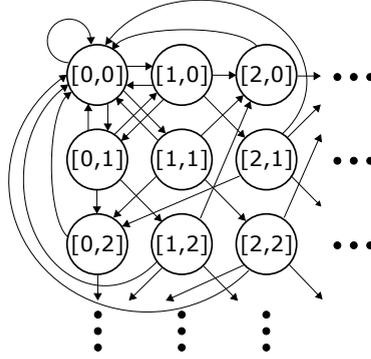


Figure 3.3: MDP  $\mathcal{M}$  states and their transitions for two sensors, where  $[u_1, u_2]$  in a circle indicates the MDP state  $[\tau[1], \tau[2]]^\top = [u_1, u_2]^\top$ . The arrows indicate state transitions.

**Remark 3.4.** *In general, binary integer problems can be solved by a branch and bound algorithm. Theorem 3.1 shows that an optimal solution can be obtained by a relaxed problem without loss of performance. This extremely reduces the number of iterations in the algorithm—the number of possible branches are reduced to  $2^{|\mathcal{E}|}$  from  $2^{N|\mathcal{E}|}$ .*

### 3.2.3 Sensor selection subproblem

In the previous subsection, we saw that  $\mathcal{C}_k$  is determined by solving Problem 3.3 with given  $\mathcal{S}_k$  and applying the upstream-first rule to the resulted graph. We can rewrite the immediate cost using  $\tau_k \triangleq [\tau_k^{(1)}, \dots, \tau_k^{(N)}]^\top$  and  $\mathcal{S}_k$  as

$$C(\tau_k, \mathcal{S}_k) = \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau_k^{(i)}}(0) \right) + E_{\min}(\mathcal{S}_k).$$

Due to the necessary condition (i) of Lemma 3.1,  $\tau_k$  is determined by  $\mathcal{S}_k$ . To obtain the network scheduler, we need to find a map from  $\tau_k$  to  $\mathcal{S}_{k+1}$ . This problem is called the sensor selection problem formulated as an MDP.

Define the MDP  $\mathcal{M} \triangleq (\mathcal{Q}, \mathcal{A}, F(\cdot, \cdot), C(\cdot, \cdot))$  as follows:

(i). The state space is given by

$$\mathcal{Q} \triangleq \{ \tau \in \mathbb{N}_0^N : \tau[i] \in \mathbb{N}_0, i \in \mathcal{V}_s \}.$$

(ii). The action space is given by

$$\mathcal{A} \triangleq \{\mathcal{S} : \mathcal{S} \in 2^{\mathcal{V}_s}\}.$$

(iii). The deterministic transition function from state  $\tau$  to  $\tau'$  with action  $\mathcal{S} \in \mathcal{A}$  is defined as  $F(\tau, \mathcal{S}) = \tau'$  where

$$\tau'[i] = \begin{cases} 0, & \text{if } i \in \mathcal{S}, \\ \tau[i] + 1, & \text{otherwise.} \end{cases}$$

(iv). The immediate cost for a transition from  $\tau$  to  $\tau'$  with action  $\mathcal{S} \in \mathcal{A}$  is given by

$$C(\tau, \mathcal{S}) = \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau'[i]}(0) \right) + E_{\min}(\mathcal{S}).$$

Figure 3.3 illustrates the MDP  $\mathcal{M}$  for a two-sensor case. A state  $[\tau[1], \tau[2]]^\top = [u_1, u_2]^\top$  corresponds to sensors 1 and 2 transmitted  $u_1$  and  $u_2$  time units ago, respectively.

With this setup, let us introduce a policy  $\pi_k : \mathcal{Q} \rightarrow \mathcal{A}$  for the MDP  $\mathcal{M}$  and  $\pi \triangleq (\pi_0, \pi_1, \dots)$ . We are interested in a policy that minimizes the average cost by choosing the sensor set to be transmitting:

**Problem 3.4.** (*Sensor selection problem*)

$$\rho^* \triangleq \min_{\pi \in \Pi} \rho_\pi(\tau_0, \mathcal{S}_0) = \min_{\pi \in \Pi} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} C(\tau_k, \mathcal{S}_k) \quad (3.18)$$

where  $\mathcal{S}_k = \pi_{k-1}(\tau_{k-1})$  for  $k \in \mathbb{N}$  given the initial state and action  $(\tau_0, \mathcal{S}_0)$ , and  $\Pi$  the set of all possible policies.

Now we will show that the MDP  $\mathcal{M}$  has an optimal stationary deterministic policy  $\pi^* \in \Pi$ . The idea is to check the sufficient conditions for the existence of an optimal solution for countable infinite-state MDPs [216] as discussed in [92, 94].

**Theorem 3.2.** *Consider the MDP  $\mathcal{M}$ . There exists a constant  $\rho^*$  and a relative-value function  $H(\cdot)$  satisfying the Bellman equation*

$$\rho^* + H(\tau) = \min_{\mathcal{S} \in \mathcal{A}} \left\{ C(\tau, \mathcal{S}) + H(F(\tau, \mathcal{S})) \right\}. \quad (3.19)$$

*Proof.* See Appendix 3.B.  $\square$

Theorem 3.2 shows that there exists a stationary deterministic optimal policy. To find such a policy, next we show that  $\mathcal{M}$  can be restricted into a finite-state MDP without loss of performance under the assumption that the estimation error is expensive compared to the communication cost. For the finite-state MDP, we can use classical algorithms such as value iteration. We make the following assumption.

**Assumption 3.3.** *Each process  $i \in \mathcal{V}_s$  either satisfies:*

(i).  $\lambda_{\max}(A_i) \geq 1$ , or

(ii).  $\lambda_{\max}(A_i) < 1$  and  $\text{tr}(X_i) > E_{\min}(\{i\})$ , where  $X_i \in \mathbb{S}_{++}^n$  is the unique solution to the Lyapunov equation  $A_i^\top X_i A_i + W_i - X_i = 0$ .

**Lemma 3.3.** *Suppose that Assumption 3.3 holds. Then there exists a constant*

$$\delta_i \triangleq \min_{\kappa} \{ \kappa \in \mathbb{N}_0 : \text{tr}(h_i^\kappa(0)) > E_{\min}(\{i\}) \}$$

for all  $i \in \mathcal{V}_s$ .

*Proof.* It is immediate from the monotonicity of  $h_i^n(X)$  along  $n$  starting from  $X = 0$  [89] and Assumption 3.3.  $\square$

Finally, we have the following theorem.

**Theorem 3.3.** *Suppose that Assumption 3.3 holds. Consider the MDP  $\mathcal{M}$ . If  $\tau[i] \geq \delta_i$ , then  $i \in \pi^*(\tau)$ .*

*Proof.* See Appendix 3.C.  $\square$

Let us define the finite-state MDP as

$$\mathcal{M}_f \triangleq (\mathcal{Q}_f, \mathcal{A}_f(\cdot), F(\cdot, \cdot), C(\cdot, \cdot))$$

with

$$\mathcal{Q}_f \triangleq \{ \tau \in \mathbb{N}_0^N : \tau[i] \leq \delta_i, i \in \mathcal{V}_s \}$$

and  $\mathcal{A}_f(\tau) \triangleq \mathcal{A} \setminus \bar{\mathcal{A}}_f(\tau)$  where  $\bar{\mathcal{A}}_f(\tau) \triangleq \{ \mathcal{S} \in \mathcal{A} : \exists i \in \mathcal{V}_s, \tau[i] = \delta_i, i \notin \mathcal{S} \}$ . That is, sensor  $i$  is always selected at state  $\tau$  when  $\tau[i] = \delta_i$ . In the optimal policy of the MDP  $\mathcal{M}$ , the state will move into  $\mathcal{Q}_f$  in the next transition even if

its initial state  $\tau_0$  is outside of  $\mathcal{Q}_f$ . After that, the states never leave  $\mathcal{Q}_f$ . Thus, the initial cost will be ignorable since its contribution to the average cost is reduced to zero as  $T$  tends to infinity [217]. Consequently, we can derive the optimal policy of the MDP  $\mathcal{M}$  by solving the finite state MDP  $\mathcal{M}_f$  without loss of performance. We show that the optimal sensor selection is periodic.

**Corollary 3.1.** *Suppose that Assumption 3.3 holds. Then there exists an optimal periodic schedule generated by an optimal policy  $\pi^*$ .*

*Proof.* Since the MDP  $\mathcal{M}_f$  is deterministic, we can fix an arbitrary action as an optimal one at any state in  $\mathcal{Q}_f$ . Furthermore, since  $\mathcal{Q}_f$  is finite, there exists a recurrent state over  $\pi^*$ . Thus, if the system reaches the recurrent state again, the state transition will repeat. Hence the result follows.  $\square$

### 3.2.4 Two-step value iteration algorithm

Previously, we showed that the set of optimally selected sensors over time is periodic under Assumption 3.3 and can be obtained by solving the finite-state MDP  $\mathcal{M}_f$  with pre-calculated  $E_{\min}(\mathcal{S})$ . We present a two-step algorithm based on relative value iteration [218]:

1. Calculate an optimal tree network for each candidate set of sensor selection (Algorithm 3.1).
2. Calculate an optimal policy of the MDP  $\mathcal{M}_f$  (Algorithm 3.2).

---

**Algorithm 3.1** Computation of an optimal tree network and energy cost

---

- 1: **INPUT:**  $\eta(e), r$
  - 2: **OUTPUT:**  $E_{\min}(\mathcal{S})$
  - 3: **for**  $\mathcal{S} \in \mathcal{A}$  **do**
  - 4:     Compute  $E_{\min}(\mathcal{S})$  in Problem 3.3
  - 5: **end for**
- 

Algorithm 3.2 has in general high computational complexity. The reasons are twofold: first, the number of states of  $\mathcal{M}_f$  depends on  $\delta_i$ , which increases exponentially by the number of sensors. Second, since we allow to pick any sensor at every time instance, the size of action space is  $2^N$  at every iteration. These issues motivate us to construct suboptimal schedules in the next section.

---

**Algorithm 3.2** Computation of an optimal schedule
 

---

- 1: **INPUT:**  $\mathcal{M}_f, E_{\min}(\mathcal{S}), v^0, \epsilon > 0, \bar{\tau}$
- 2: **OUTPUT:**  $\pi^*(\tau)$
- 3:  $v^0 \leftarrow v^0 - v^0(\bar{\tau}) \cdot \mathbf{1}_{|\mathcal{Q}_f|}$  and  $k = 0$
- 4: **for**  $\tau \in \mathcal{Q}_f$  **do**
- 5:   Compute

$$v^{k+1}(\tau) = \min_{\mathcal{S} \in \mathcal{A}_f} \left\{ C(\tau, \mathcal{S}) + v^k(F(\tau, \mathcal{S})) \right\} \quad (3.20)$$

- 6: **end for**
- 7:  $v^{k+1} \leftarrow v^{k+1} - v^{k+1}(\bar{\tau}) \cdot \mathbf{1}_{|\mathcal{Q}_f|}$
- 8: **if**  $\max(v^{k+1}(\tau) - v^k(\tau)) - \min(v^{k+1}(\tau) - v^k(\tau)) \leq \epsilon$  **then**
- 9:   Go to Step 13
- 10: **else**
- 11:    $k \leftarrow k + 1$  and return to Step 4
- 12: **end if**
- 13: For each  $\tau \in \mathcal{Q}_f$ , set

$$\pi^*(\tau) = \arg \min_{\mathcal{S} \in \mathcal{A}_f} \left\{ C(\tau, \mathcal{S}) + v^k(F(\tau, \mathcal{S})) \right\}$$


---

### 3.3 Construction of suboptimal solutions

In this section, we introduce algorithms to compute a suboptimal solution in an efficient way.

#### 3.3.1 Reduced MDP schedule

The first algorithm solves an approximate MDP by restricting the size of the state and action spaces. To do this, we introduce some sets of sensors and assume that the sensors in the same set are always scheduled to transmit together. The reduced MDP (R-MDP),  $\tilde{\mathcal{M}} \triangleq (\tilde{\mathcal{Q}}, \mathcal{A}_f(\cdot), F(\cdot, \cdot), \tilde{C}(\cdot, \cdot))$ , is obtained as follows:

- (i). Split  $\mathcal{V}_s$  into  $M$  disjoint subsets  $\tilde{\mathcal{V}} \triangleq \{\mathcal{V}_1, \dots, \mathcal{V}_M\}$ .
- (ii). Define the bounds  $\tilde{\delta}_j \triangleq \min\{\delta_i : i \in \mathcal{V}_j\}$ ,  $j = 1, \dots, M$ .
- (iii). Define the state space

$$\tilde{\mathcal{Q}} \triangleq \{\tau \in \mathbb{N}_0^M : \tau[j] = 0, \dots, \tilde{\delta}_j, j = 1, \dots, M\}$$

and the action space  $\mathcal{A}_f(\tau)$ .

- (iv). Define the cost

$$\tilde{C}(\tau, \mathcal{S}) \triangleq \sum_{i=1}^N \text{tr} \left( h_i^{\tau[j(i)]} (0) \right) + E_{\min}(\mathcal{S})$$

where  $j(i)$  indicates the subset  $j$  in  $\tilde{\mathcal{V}}$  to which sensor  $i$  belongs.

We compute the R-MDP schedule by calling Algorithms 3.1 and 3.2 with  $\mathcal{M}$  replaced by  $\tilde{\mathcal{M}}$ .

#### 3.3.2 Fixed-period algorithm

The idea of our second algorithm is to fix the transmission period of each sensor obtained by solving smaller MDPs. Then the whole schedule is obtained by combining all such schedules. The procedure is given by the fixed period algorithm (FPA) in Algorithm 3.3.

Let us denote the sensor selection obtained by Algorithm 3.3 as  $\mathcal{S}_{\text{FPA},k}$ . For this algorithm, we have the following result.

**Algorithm 3.3** Fixed Period Algorithm

- 
- 1: **INPUT:**  $\eta(e)$
  - 2: **OUTPUT:**  $\{\mathcal{S}_{\text{FPA},k}\}_{k=0}^D$
  - 3: **for**  $i \in \mathcal{V}_s$  **do**
  - 4:   Compute  $E_{\min}(\{i\})$
  - 5:   Set  $\mathcal{M}_i = (\mathcal{Q}_i, \mathcal{A}_i, F(\cdot, \cdot), C(\cdot, \cdot))$  with  $\mathcal{Q}_i = \{\tau_i \in \mathbb{N}_0 : \tau_i = 0, \dots, \delta_i\}$  and  $\mathcal{A}_i = \{\emptyset, i\}$
  - 6:   Solve  $\mathcal{M}_i$  and compute a period  $D_i$
  - 7:   Set
 
$$\mathcal{S}_{\text{FPA},k}^{(i)} = \begin{cases} \{i\}, & \text{if } k \equiv 0 \pmod{D_i} \\ \emptyset, & \text{if } k \not\equiv 0 \pmod{D_i} \end{cases}$$
  - 8: **end for**
  - 9: Compute  $D$ , the least common multiple of  $D_i, i = 1, \dots, N$
  - 10: **for**  $k = 0, 1, \dots, D$  **do**
  - 11:   Set  $\mathcal{S}_{\text{FPA},k} = \bigcup_{i \in \mathcal{V}_s} \mathcal{S}_{\text{FPA},k}^{(i)}$
  - 12:   Compute  $E_{\min}(\mathcal{S}_{\text{FPA},k})$
  - 13: **end for**
- 

**Proposition 3.1.** *Suppose that the data aggregation rate  $r = 0$ . Then the schedule obtained by Algorithm 3.3 is optimal, i.e.,*

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} C(\tau_k, \mathcal{S}_{\text{FPA},k}) = \rho^*.$$

*Proof.* We have  $E(\mathcal{C}_k) = \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{V}_s} \eta(e) z_k^{(i)}(e)$ , which means that the energy consumption  $E(\mathcal{C}_k)$  is a linear combination of  $z_k^{(i)}(e)$ . That is, we have

$$E_{\min}(\mathcal{S}_{\text{FPA},k}) = \sum_{i \in \mathcal{V}_s} \sigma_i(\mathcal{S}_{\text{FPA},k}^{(i)}) E_{\min}(\{i\})$$

where  $\sigma_i(\mathcal{S}) = 1$  if  $i \in \mathcal{S}$ , otherwise 0. Then we have

$$\begin{aligned} C(\tau, \mathcal{S}_{\text{FPA},k}) &= \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau[i]}(0) \right) + E_{\min}(\mathcal{S}_{\text{FPA},k}) \\ &= \sum_{i \in \mathcal{V}_s} \left[ \text{tr} \left( h_i^{\tau[i]}(0) \right) + \sigma_i(\mathcal{S}_{\text{FPA},k}^{(i)}) E_{\min}(\{i\}) \right]. \end{aligned}$$

Thus, minimization of

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} \left[ \text{tr} \left( h_i^{\tau^{[i]}}(0) \right) + \sigma_i(\mathcal{S}_{\text{FPA},k}^{(i)}) E_{\min}(\{i\}) \right]$$

for each  $i \in \mathcal{V}_s$  yields the minimum cost. This completes the proof.  $\square$

### 3.4 Numerical examples

In this section, we present three numerical examples to illustrate our results in this chapter. In the first example, we evaluate the performance of the R-MDP and FPA schedule by comparing them with the optimal one for the small network depicted in Figure 3.1. In the second example, we provide a larger network, and show that the two suboptimal algorithms can obtain the schedules efficiently even if the size of the original MDP is too large to efficiently compute the optimal schedule. The third example shows that these suboptimal schedules are scalable to networks consisting of a hundred nodes.

#### 3.4.1 Optimal and suboptimal schedules for a small network ( $N = 3$ )

To see the performances of the proposed algorithms, we consider the small network depicted in Figure 3.1. The system parameters of the three plants are

$$A_1 = \begin{bmatrix} 1.3 & 1.2 \\ 0 & 1.4 \end{bmatrix}, A_2 = \begin{bmatrix} 1.5 & 0.8 \\ 0 & 1.2 \end{bmatrix}, A_3 = \begin{bmatrix} 3.5 & 2.0 \\ 0 & 3.1 \end{bmatrix},$$

with  $W_i = 0.1I_2$ , for  $i = 1, 2, 3$  where  $I_2$  is the  $2 \times 2$  identity matrix. For communication parameters, we assume that  $E_{\text{elec}} = E_{\text{amp}} = 1$ ,  $c = 1$ ,  $\beta_i = 1$  for  $i = 1, 2, 3$ ,  $d((1, 0)) = d((2, 0)) = d((1, 3)) = d((2, 3)) = 1$ , and  $r = 0.5$ . The action set consists of every possible subset of sensors selected to transmit accompanied by all possible routes as shown in Table 3.1. Algorithm 3.1 yields the optimal paths and their energy costs for each sensor selection, see Table 3.2.

#### Optimal schedule

First, we derive the optimal schedule. By the value of  $E_{\min}(\{i\})$ ,  $i = 1, 2, 3$ , we obtain the bounds of the MDP state space as  $\delta_1 = 3$ ,  $\delta_2 = 4$ ,  $\delta_3 = 3$

Action index	Sensor selection	Path
0	$\emptyset$	-
1	1	$1 \rightarrow 0$
2	1	$1 \rightarrow 3 \rightarrow 2 \rightarrow 0$
3	2	$2 \rightarrow 0$
4	2	$2 \rightarrow 3 \rightarrow 1 \rightarrow 0$
5	3	$3 \rightarrow 1 \rightarrow 0$
6	3	$3 \rightarrow 2 \rightarrow 0$
7	1,2	$1 \rightarrow 0, 2 \rightarrow 0$
8	1,2	$1 \rightarrow 3 \rightarrow 2 \rightarrow 0$
9	1,2	$2 \rightarrow 3 \rightarrow 1 \rightarrow 0$
10	2,3	$3 \rightarrow 2 \rightarrow 0$
11	2,3	$3 \rightarrow 1 \rightarrow 0, 2 \rightarrow 0$
12	2,3	$2 \rightarrow 3 \rightarrow 1 \rightarrow 0$
13	3,1	$3 \rightarrow 1 \rightarrow 0$
14	3,1	$3 \rightarrow 2 \rightarrow 0, 1 \rightarrow 0$
15	3,1	$1 \rightarrow 3 \rightarrow 2 \rightarrow 0$
16	1,2,3	$3 \rightarrow 1 \rightarrow 0, 2 \rightarrow 0$
17	1,2,3	$3 \rightarrow 2 \rightarrow 0, 1 \rightarrow 0$
18	1,2,3	$1 \rightarrow 3 \rightarrow 2 \rightarrow 0$
19	1,2,3	$2 \rightarrow 3 \rightarrow 1 \rightarrow 0$

Table 3.1: All possible sensor selections and their routes to the gateway for the network in Figure 3.1.

Sensor selection	Optimal path	Action	Energy cost
$\emptyset$	-	0	0
1	$1 \rightarrow 0$	1	1
2	$2 \rightarrow 0$	3	1
3	$3 \rightarrow 1 \rightarrow 0$	5	3
1,2	$1 \rightarrow 0, 2 \rightarrow 0$	7	2
2,3	$3 \rightarrow 2 \rightarrow 0$	10	3.5
1,3	$3 \rightarrow 1 \rightarrow 0$	13	3.5
1,2,3	$3 \rightarrow 1 \rightarrow 0, 2 \rightarrow 0$	16	4.5

Table 3.2: The optimal paths for each sensor selection.

with Theorem 3.3. Then we can find the optimal schedule by Algorithm 3.2. The result is shown in Figure 3.4 (top). The period of the optimal schedule is 8 in which actions 0, 1, 3, 10, and 13 from Table 3.1 are taken. Figure 3.5 depicts the periodic optimal schedule with the corresponding paths. Timeslot allocations are given according to the upstream-first rule, namely, a sensor waits to transmit until all the scheduled sensor data from its upstream arrive. For example, if action 13 is taken at time  $k$ , the timeslots are allocated into  $e(k, 1) = (3, 1)$  and  $e(k, 2) = (1, 0)$ . If action 10, we have  $e(k, 1) = (3, 2)$  and  $e(k, 2) = (2, 0)$ .

### R-MDP schedule

Next, we formulate an R-MDP by setting  $\mathcal{V}_1 = \{1\}$  and  $\mathcal{V}_2 = \{2, 3\}$ . We then have  $\delta_1 = 3$  and  $\delta_2 = 3$ . The obtained schedule is shown in Figure 3.4 (middle) and Figure 3.6. It has period 6 with actions 0, 1, 10, and 16. We can see that sensors 2 and 3 are always selected together in actions 10 and 16. The timeslots allocated for action 16 can be obtained by the upstream-first rule. For example,  $e(k, 1) = (3, 1)$ ,  $e(k, 2) = (1, 0)$ , and  $e(k, 3) = (2, 0)$ .

### FPA schedule

We derive an FPA schedule by Algorithm 3.3. Now we have  $E_{\min}(\{1\}) = E_{\min}(\{2\}) = 2$ , and  $E_{\min}(\{3\}) = 5$ , with which we formulate MDP  $\mathcal{M}_i$  for  $i = 1, 2, 3$ . Then we obtain the fixed activation period for each sensor:

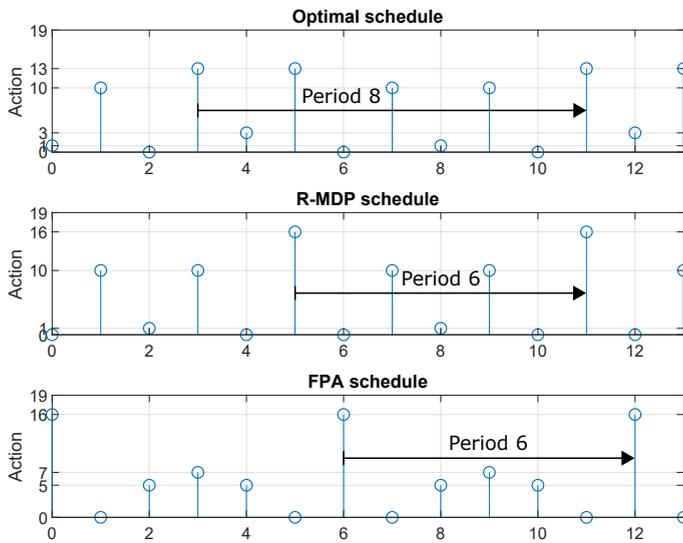


Figure 3.4: Three schedules obtained by the proposed optimal and suboptimal algorithms. Top: Optimal schedule, middle: R-MDP schedule, bottom: FPA schedule.

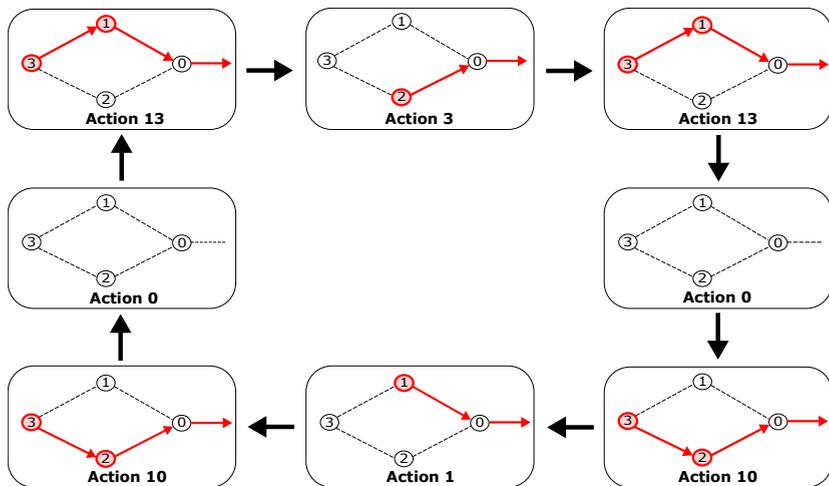


Figure 3.5: The 8-period optimal schedule for the small network. In each action, the timeslot allocation can be given based on the upstream-first rule.

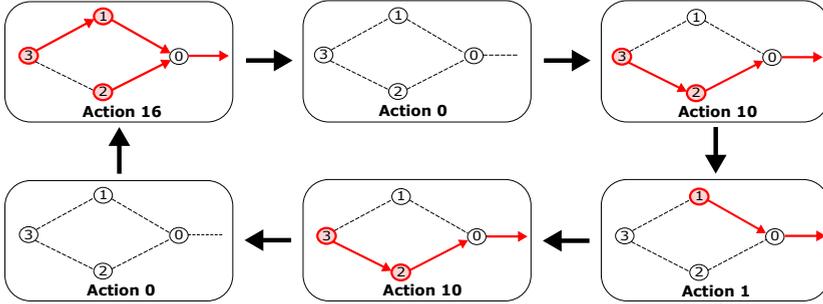


Figure 3.6: The 6-period R-MDP schedule for the small network. In this schedule, sensors 2 and 3 are always selected together in actions 10 and 16.

	Averaged cost	$ Q $	$ A $
Optimal	4.09	80	8
R-MDP	4.17	16	4
FPA	4.35	$\leq 5$	$\leq 2$

Table 3.3: Averaged costs and the sizes of MDP.

$D_1 = 3$ ,  $D_2 = 3$ , and  $D_3 = 2$ , which yields the period 6 schedule as shown in Figure 3.4 (bottom) and Figure 3.7. The drawback of this algorithm is that it may result in inefficient sensor selection. In fact, the obtained FPA schedule includes action 5, in which only sensor 3 is selected even though the measurement is transmitted through sensor 1. It results in missing an opportunity to data aggregation with sensor 1's data.

### Performance evaluation

The averaged cost and the sizes of the MDPs for each schedule are summarized in Table 3.3. We can see that the R-MDP and FPA schedule obtain similar performances compared to the optimal one even though the sizes of the MDPs are considerably reduced. The estimation performance  $\sum_{i \in \mathcal{V}_s} \epsilon_k^{(i)\top} \epsilon_k^{(i)}$  is plotted in Figure 3.8. We can conclude that the proposed suboptimal schedules obtain well-performing schedules.

We show the averaged cost of these three schedules with respect to the

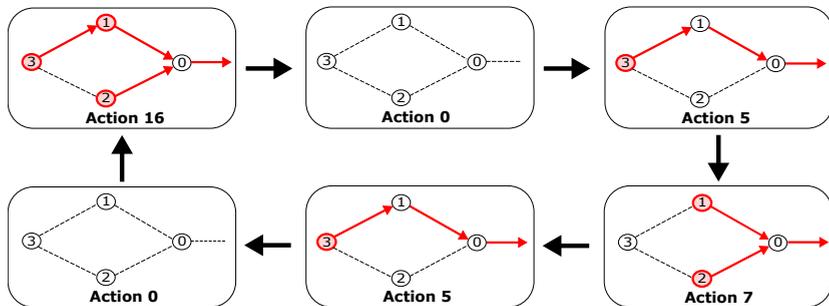


Figure 3.7: The 6-period FPA schedule for the small network. In this schedule, sensor 3 is selected alone in action 7, resulting in an inefficient schedule in terms of energy consumption.

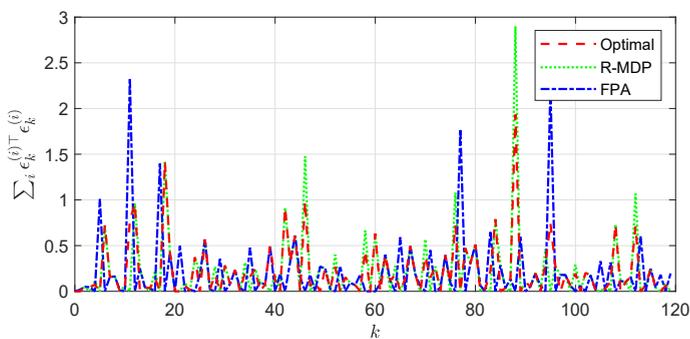


Figure 3.8: Estimation performance comparison of the three schedules: optimal (red), R-MDP (green), and FPA (blue).

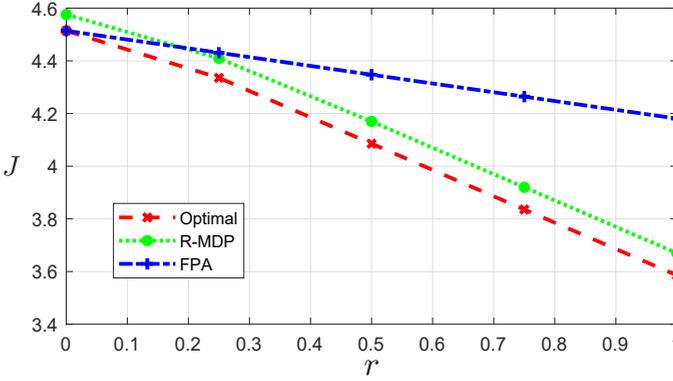


Figure 3.9: The averaged cost  $J$  of the optimal, R-MDP and FPA schedules with respect to the data aggregation rate  $r$  for the network with  $N = 3$ .

data aggregation rate  $r$  in Figure 3.9. It confirms Proposition 3.1, i.e., the averaged cost of the FPA schedule is optimal when  $r = 0$  and this averaged cost is the upper bound of the FPA averaged cost for any  $r$ . The difference of the averaged costs of the optimal and FPA schedules increases with increasing  $r$  since the optimal schedule receives benefit of the data aggregation. In Figure 3.4, the FPA schedule takes action 5 once in a period, i.e., sensor 3 is selected alone. However, this is not effective in terms of the energy cost since the data cannot be aggregated even though it passes through sensor 1. In the optimal schedule in Figure 3.4, sensor 3 is always selected together with sensor 1 (action 13) or with sensor 2 (action 10). The R-MDP schedule results in larger costs for any  $r$ . However, the cost is close to that of the FPA schedule if  $r = 1$  since the R-MDP still tries to take advantage of the benefit of the data aggregation.

### 3.4.2 Suboptimal schedules for a large network ( $N = 9$ )

To see the performances of the proposed suboptimal scheduling algorithms in a more realistic situation, we consider the network shown in Figure 3.10. The network consists of  $N = 9$  sensors distributed over a square field and a gateway at the origin. The sensors can communicate with the other sensors when the distances are shorter than  $d_{\max} = 4$ . The plants are given by

$$A_1 = \begin{bmatrix} 2.3 & 1.2 \\ 0 & 1.9 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1.6 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 3 & 2.4 \\ 2.2 & 3.5 \end{bmatrix},$$

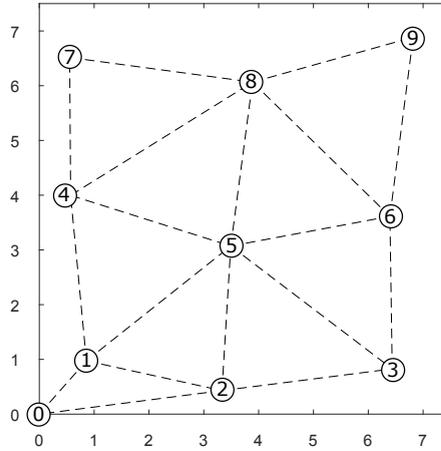


Figure 3.10: A sensor network with  $N = 9$  sensors.

$$\begin{aligned}
 A_4 &= \begin{bmatrix} 1.4 & 0.2 \\ 0.5 & 1.5 \end{bmatrix}, \quad A_5 = \begin{bmatrix} 2.3 & 0.5 \\ 0.2 & 1.4 \end{bmatrix}, \quad A_6 = \begin{bmatrix} 2.2 & 0 \\ 0 & 2 \end{bmatrix}, \\
 A_7 &= \begin{bmatrix} 2.5 & 0.2 \\ 1.2 & 2.2 \end{bmatrix}, \quad A_8 = \begin{bmatrix} 2.1 & 1.2 \\ 0 & 1.5 \end{bmatrix}, \quad A_9 = \begin{bmatrix} 3.5 & 3.6 \\ 2.3 & 3.5 \end{bmatrix},
 \end{aligned}$$

with  $W_i = 0.1I_2$ , for  $i = 1, \dots, 9$ ,  $E_{\text{elec}} = E_{\text{amp}} = 1$ ,  $c = 4$ ,  $\beta_i = 1$  for  $i = 1, \dots, 9$ , and  $r = 0.5$ . The bounds of the MDP states are obtained as  $\delta_1 = 4$ ,  $\delta_2 = 6$ ,  $\delta_3 = 3$ ,  $\delta_4 = 6$ ,  $\delta_5 = 5$ ,  $\delta_6 = 6$ ,  $\delta_7 = 4$ ,  $\delta_8 = 5$ , and  $\delta_9 = 3$ . This means that the original MDP problem is computationally expensive to solve as the size of its state space is of the order of  $\prod_{i=1}^N \delta_i \sim 10^6$ . The averaged cost and size of the MDPs are summarized in Table 3.4. R-MDP1 uses R-MDP algorithm with grouping sensors based on their locations in order to take advantage of the data aggregation, i.e., we include sensors placed in the near distance into the same set. We take  $\mathcal{V}_1 = \{1, 2\}$ ,  $\mathcal{V}_2 = \{3, 5\}$ ,  $\mathcal{V}_3 = \{4, 7\}$ , and  $\mathcal{V}_4 = \{6, 8, 9\}$ . For R-MDP2, we make sensor sets based on the bound  $\delta_i$  to avoid too many or too few transmissions with respect to the divergence speed of each error covariance, i.e., sensors with close bounds are included in the same sets. We use  $\mathcal{V}_1 = \{1, 7\}$ ,  $\mathcal{V}_2 = \{3, 9\}$ ,  $\mathcal{V}_3 = \{5, 8\}$ , and  $\mathcal{V}_4 = \{2, 4, 6\}$ . The

	Averaged cost	Period	$ \mathcal{Q} $	$ \mathcal{A} $
R-MDP1	180.63	12	400	16
R-MDP2	180.76	12	840	16
FPA	184.20	60	$\leq 6$	$\leq 2$

Table 3.4: Averaged costs, periods, sizes of MDP.

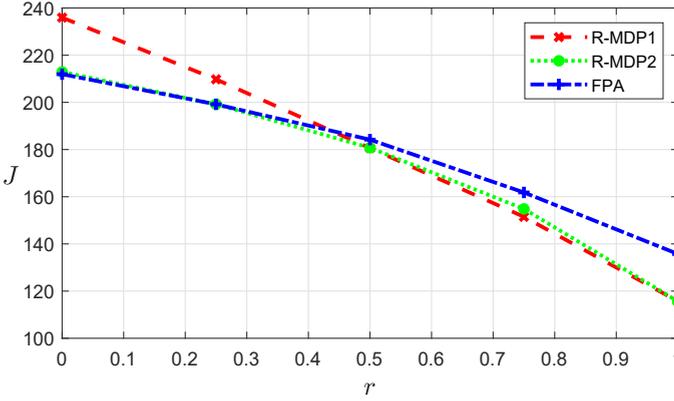


Figure 3.11: The averaged cost of the two R-MDP schedules and the FPA schedule with respect to the data aggregation rate  $r$  for the network with  $N = 9$  sensors.

FPA schedule is obtained from small MDPs  $\mathcal{M}_i$ ,  $i = 1, \dots, 9$ . The obtained FPA schedule generated by  $D_1 = 4$ ,  $D_2 = 6$ ,  $D_3 = 3$ ,  $D_4 = 6$ ,  $D_5 = 5$ ,  $D_6 = 6$ ,  $D_7 = 4$ ,  $D_8 = 5$ , and  $D_9 = 3$  has period 60.

Figure 3.11 shows the averaged costs of the three schedules with respect to  $r$ . As in Proposition 3.1, the FPA schedule is optimal when  $r = 0$ . Thus, it has a near optimal performance if  $r$  is small. The performance further degrades compared to the R-MDPs when  $r$  is large. Both approaches for the R-MDP schedules reduce cost when  $r$  is large. The R-MDP2 has a comparatively better performance regardless of the value of  $r$ . It implies that a way group sensors influences the performance.

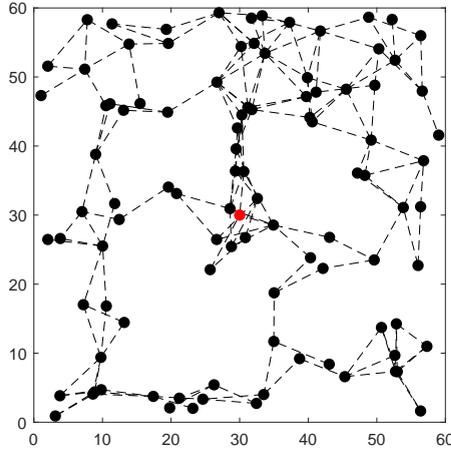


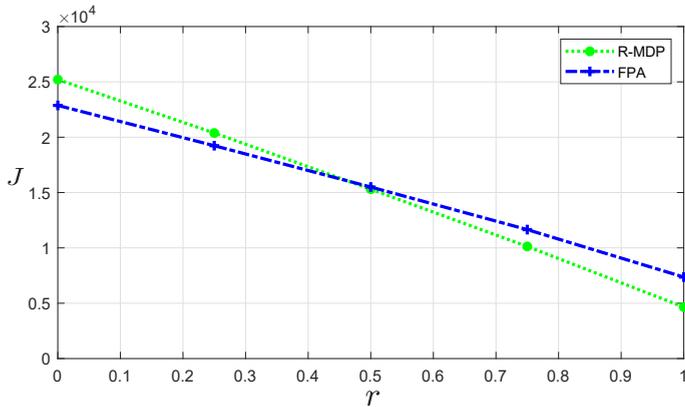
Figure 3.12: A sensor network with  $N = 99$  sensors. Black dots represent sensors and the red dot the gateway.

### 3.4.3 Suboptimal schedules for a large network ( $N = 99$ )

To see the scalability of the proposed suboptimal scheduling algorithms, we consider the larger network shown in Figure 3.12. The network consists of  $N = 99$  sensors distributed over a square field (black dots) and a gateway at the center of the field (red dot). For the R-MDP algorithm, we divide sensors into three subgroups, so that the number of states are reduced to 252 when  $r = 0.5$  (Table 3.5). The FPA schedule can also be obtained by solving 99 small MDPs where the maximum number of the states is 10. The period of the FPA schedule is 2520. However, the period may further increase in some cases since it is derived by taking the least common multiple among  $D_i, i \in \mathcal{V}_s$ . In this case, it is difficult to obtain the actual performance  $J$  since we need to recompute  $E_{\min}(\mathcal{S}_{\text{FPA},k})$  in step 12 of Algorithm 3.3. Figure 3.13 shows the averaged costs of the two schedules with respect to  $r$ . Similar to previous examples, the FPA algorithm results in better schedules when  $r$  is small while the R-MDP does better when  $r$  is large.

	Averaged cost	Period	$ \mathcal{Q} $	$ \mathcal{A} $
R-MDP	15300.9	5	252	8
FPA	15498.6	2520	$\leq 10$	$\leq 2$

Table 3.5: Averaged costs, periods, sizes of MDP.

Figure 3.13: The averaged cost of the two R-MDP and the FPA schedules with respect to the data aggregation rate  $r$  for the network with  $N = 99$ .

### 3.5 Summary

In this chapter, we proposed a design framework of multi-hop network scheduling for remote estimation. We formulated an optimization problem minimizing an infinite-time averaged estimation error covariance with sensor energy consumption. We showed that the problem could be divided into two subproblems by exploiting the necessary conditions for network scheduling optimality. The first one was a tree planning subproblem, which gives routes from sensors to the estimator. It was shown that the subproblem could be solved efficiently. The second one was a sensor selection subproblem. By formulating an MDP, this subproblem can be solved by the value iteration, and an existence condition for a periodic optimal schedule was derived. We proposed two alternative algorithms to obtain suboptimal schedules to reduce computational complexity. It was demonstrated how the proposed algorithms are effective in numerical examples.

### 3.A Proof of Theorem 3.1

To prove Theorem 3.1, we need the following definition and supporting lemmas.

**Definition 3.1** (Total unimodular matrix [219]). *A square integer matrix is unimodular if it has determinant  $+1$  or  $-1$ . A matrix is totally unimodular if every square non-singular submatrix of it is unimodular.*

**Lemma 3.4.** *Let  $X$  be a totally unimodular matrix. Then the following matrices are also totally unimodular:*

- (i).  $\text{diag}(X, \dots, X)$ ,
- (ii).  $\begin{bmatrix} X \\ I \end{bmatrix}$  and  $\begin{bmatrix} X \\ -I \end{bmatrix}$ ,
- (iii).  $\begin{bmatrix} X & -X \end{bmatrix}$  and  $\begin{bmatrix} X \\ -X \end{bmatrix}$ .

**Lemma 3.5** ([220]). *If  $A$  is totally unimodular, then all the vertices of the polyhedron  $\{x : Ax \leq b\}$  are integer for any integer vector  $b$ .*

**Proof of Theorem 3.1.** First, we transform Problem 3.2 into an integer linear problem by introducing  $t(e) = \max_{i \in \mathcal{V}_s} z_k^{(i)}(e)$  and the constraint  $z_k^{(i)}(e) \leq t(e)$  for  $i \in \mathcal{V}_s$  and  $e \in \mathcal{E}$ . We show that relaxing  $0 \leq z_k^{(i)}(e) \leq 1$  still obtains a binary integer solution.

The constraints (3.15)–(3.17) can be written in a compact form

$$\{z_k : \mathbf{G}z_k \leq b\}$$

where  $b = [b(\mathcal{S}_k)^\top, \dots, -b(\mathcal{S}_k)^\top, 1, \dots, 1, 0, \dots, 0]^\top$  with  $b(\mathcal{S}_k)^\top = [b^{(1)}(\mathcal{S}_k)^\top, \dots, b^{(N)}(\mathcal{S}_k)^\top]^\top$  and

$$\mathbf{G} \triangleq \begin{bmatrix} \text{diag}(G, \dots, G) \\ -\text{diag}(G, \dots, G) \\ I \\ -I \end{bmatrix}.$$

The matrix  $G$  is the node–arc incidence matrix of  $\mathcal{G}$ , therefore it is totally unimodular. Then by Lemma 3.4,  $\mathbf{G}$  is totally unimodular. Fixing  $t(e)$  to 0 or 1 for all  $e \in \mathcal{E}$ ,  $z_k$  obtains the integer solution if the corresponding linear problem is feasible (Lemma 3.5). Thus, a minimizer of Problem 3.2 is equal to that of Problem 3.3. This completes the proof.  $\square$

### 3.B Proof of Theorem 3.2

To prove Theorem 3.2, we define a standard policy. Consider a Markov chain with countable infinite state space  $\mathcal{Q}$ . Let us denote  $p_{q_1, q_2}^n$  by the probability that the state that is currently at  $q_1$  will be  $q_2$  for the first time exactly after  $n \geq 1$  transitions. That is,

$$p_{q_1, q_2}^n = \Pr(\tau_k \neq q_2, k = 1, \dots, n-1, \tau_n = q_2 | \tau_0 = q_1).$$

The expected first passage time [216],  $t_{\tau, z}$ , is denoted as

$$t_{q_1, q_2} = \sum_{n=1}^{\infty} n p_{q_1, q_2}^n,$$

and the corresponding averaged total cost, called the expected first passage cost [216] is denoted by  $c_{q_1, q_2}$ .

**Definition 3.2** ([216]). *A randomized stationary policy  $\pi$  is a standard policy if there exists a state  $z \in \mathcal{Q}$  such that the expected first passage time  $t_{\tau,z}$  from  $\tau$  to  $z$  satisfies  $t_{\tau,z} < \infty$  for all  $\tau \in \mathcal{Q}$ , and the expected first passage cost  $c_{\tau,z}$  from  $\tau$  to  $z$  satisfies  $c_{\tau,z} < \infty$  for all  $\tau \in \mathcal{Q}$ .*

**Lemma 3.6** ([216, Corollary 7.5.10]). *Assume that the following conditions hold:*

(i). *There exist a standard policy  $\pi$  such that the positive recurrent class induced by  $\pi$  is equal to  $\mathcal{Q}$ .*

(ii). *Given  $U > 0$ , the set  $\mathcal{Q}_U = \{\tau : C(\tau, \mathcal{S}) \leq U \text{ for some } \mathcal{S}\}$  is finite.*

*Then there exists a solution to the Bellman equation (3.19) for the average cost problem with countable infinite state space  $\mathcal{Q}$ .*

**Proof of Theorem 3.2.** The proof follows [94] by considering a randomized policy  $\pi$  such that at any states sensor  $i$  transmits its measurement with probability  $\theta_i$  and  $\theta \triangleq \prod_{i \in \mathcal{V}_s} \theta_i$  satisfies  $1 - 1/\lambda_{\max}^2(A_i) < \theta < 1$  for all  $i \in \mathcal{V}_s$ . Let  $z = [0, \dots, 0] \in \mathbb{N}_0^N$ , then at any states it comes back to  $z$  with probability  $\theta$ . Thus,

$$t_{\tau,z} = \theta + 2(1 - \theta)\theta + 3(1 - \theta)^2\theta + \dots = \frac{1}{\theta} < \infty.$$

Notice that  $c_{\tau_1,z} \leq c_{\tau_2,z}$  if  $\tau_1[i] \leq \tau_2[i], \forall i$  and  $E_{\min}(\mathcal{S}_1 \cup \mathcal{S}_2) \leq E_{\min}(\mathcal{S}_1) + E_{\min}(\mathcal{S}_2)$ , the expected average cost is

$$\begin{aligned} c_{\tau,z} &\leq \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau[i]}(0) \right) + (1 - \theta) \left[ c_{\tau+1_N,z} + \sum_{i \in \mathcal{V}_s} E_{\min}(\{i\}) \right] \\ &\quad + \theta \sum_{i \in \mathcal{V}_s} E_{\min}(\{i\}) \\ &= \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau[i]}(0) \right) + \sum_{i \in \mathcal{V}_s} E_{\min}(\{i\}) + (1 - \theta)c_{\tau+1_N,z} \\ &= \sum_{n=0}^{\infty} (1 - \theta)^n \left[ \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau[i]+n}(0) \right) + \sum_{i \in \mathcal{V}_s} E_{\min}(\{i\}) \right] \\ &= \sum_{n=0}^{\infty} (1 - \theta)^n \sum_{i \in \mathcal{V}_s} \text{tr} \left( h_i^{\tau[i]+n}(0) \right) + \frac{1}{\theta} \sum_{i \in \mathcal{V}_s} E_{\min}(\{i\}) \\ &< \infty \end{aligned}$$

The boundedness of the last inequality holds from the assumption  $1 - 1/\lambda_{\max}^2(A_i) < \theta < 1$  [74, 77, 78]. Hence,  $\pi$  is a standard policy.

Next, we show that the positive recurrent class is equal to  $\mathcal{Q}$ . Consider an arbitrary state  $\tau \in \mathcal{Q}$ . This state is reachable from state  $z$  after  $\tau_{\max} \triangleq \max_i \{\tau[i]\}$  by letting sensor  $i$  transmit its data for the first  $\tau_{\max} - \tau[i]$  transitions and not transmit  $\tau[i]$  transitions after that. Let us denote the probability of this realization as  $\theta'$ . Then, for any state  $\tau$ , one can return to this state with probability equal to or higher than  $\theta'' \triangleq \theta \cdot \theta'$  after  $\tau_{\max}$  transitions. Thus, the probability that one returns to the state  $\tau$  is

$$\theta'' + (1 - \theta'')\theta'' + (1 - \theta'')^2\theta'' + \dots = 1,$$

which shows that the recurrent class is equal to  $\mathcal{Q}$ , hence the first condition is verified.

The second condition is verified as  $C(\tau, \mathcal{S})$  is monotonically increasing in  $\tau$ .  $\square$

### 3.C Proof of Theorem 3.3

To prove Theorem 3.3, we first present the following lemma.

**Lemma 3.7.** *Suppose that Assumption 3.3 holds. Consider MDP  $\mathcal{M}$ . Then, for all  $i \in \mathcal{V}_s$ , there exists a time instance  $k \in \mathbb{N}_0$  such that  $i \in \pi^*(\tau_k)$ .*

*Proof.* It is obvious since no transmission policy is never optimal if the process is unstable or the process is stable but the transmission cost is lower than the steady-state estimation error.  $\square$

Next, we introduce a partial order over the state space  $\mathcal{Q}$ . For the states  $\tau, \tau' \in \mathcal{Q}$ , we say  $\tau \succ \tau'$  if  $\tau[i] > \tau'[i]$  for all  $i \in \mathcal{V}_s$ .

**Proof of Theorem 3.3.** Proof is given by contradiction. Suppose that  $\pi^*$  is an optimal policy with  $i \notin \pi^*(\tau)$  where  $\tau[i] \geq \delta_i$ . Consider a policy  $\pi'$  such that  $i \in \pi'(\tau)$ , but all the other sensors are selected as same as  $\pi^*(\tau)$ . Let  $\tau_1^* \triangleq F(\tau, \pi^*(\tau))$  and  $\tau_1' \triangleq F(\tau, \pi'(\tau))$  be the next state of  $\tau$  according to the policy  $\pi^*$  and  $\pi'$ , respectively. Obviously,  $\tau_1^* \succ \tau_1'$  since  $\tau_1^*[j] = \tau_1'[j]$  for  $j \neq i$ , and  $\tau_1^*[i] = \tau^*[i] + 1$ ,  $\tau_1'[i] = 0$ . Define  $V(\tau, \mathcal{S}) \triangleq C(\tau, \mathcal{S}) + H(F(\tau, \mathcal{S}))$ , then by the Bellman principle, the optimal policy  $\pi^*$  needs to satisfy  $V(\tau, \pi^*(\tau)) -$

$V(\tau, \pi'(\tau)) < 0$ . We will show that  $\pi^*$  contradicts this principle. Now, we have

$$\begin{aligned}
& V(\tau, \pi^*(\tau)) - V(\tau, \pi'(\tau)) \\
&= C(\tau, \pi^*(\tau)) + H(\tau_1^*) - C(\tau, \pi'(\tau)) - H(\tau_1') \\
&= \sum_{j \in \mathcal{V}_s} \text{tr} \left( h_j^{\tau_1^*[j]}(0) \right) - \sum_{j \in \mathcal{V}_s} \text{tr} \left( h_j^{\tau_1'[j]}(0) \right) \\
&\quad + E_{\min}(\pi^*(\tau)) - E_{\min}(\pi'(\tau)) + H(\tau_1^*) - H(\tau_1') \\
&= \text{tr} \left( h_i^{\tau_1^*[i]+1}(0) \right) + E_{\min}(\pi^*(\tau)) - E_{\min}(\pi^*(\tau) \cup \{i\}) + H(\tau_1^*) - H(\tau_1') \\
&\geq \text{tr} \left( h_i^{\delta_i}(0) \right) - E_{\min}(\{i\}) + H(\tau_1^*) - H(\tau_1') \\
&> H(\tau_1^*) - H(\tau_1'). \tag{3.21}
\end{aligned}$$

The first inequality holds due to the subadditivity of  $E_{\min}(\cdot)$ . Let  $\mathcal{S}_1^* \triangleq \pi^*(\tau_1^*)$  be the action given by the optimal policy at state  $\tau_1^*$ . Consider a policy  $\pi_1'$  such that  $\pi_1'(\tau_1') = \mathcal{S}_1^*$ . Also let  $\tau_2^* \triangleq F(\tau_1, \pi^*(\tau_1^*))$  and  $\tau_2' \triangleq F(\tau_1', \pi_1'(\tau_1'))$  be the next state of  $\tau_1^*$  according to  $\pi^*$  and that of  $\tau_1'$  according to  $\pi_1'$ , respectively. The inequality (3.21) continues

$$\begin{aligned}
& H(\tau_1^*) - H(\tau_1') \\
&= C(\tau_1^*, \pi^*(\tau_1^*)) + H(F(\tau_1^*, \pi^*(\tau_1^*))) - C(\tau_1', \pi_1'(\tau_1')) - H(F(\tau_1', \pi_1'(\tau_1'))) \\
&\geq C(\tau_1^*, \pi^*(\tau_1^*)) + H(F(\tau_1^*, \pi^*(\tau_1^*))) - C(\tau_1', \pi_1'(\tau_1')) - H(F(\tau_1', \pi_1'(\tau_1'))) \\
&\geq C(\tau_1^*, \mathcal{S}_1^*) - C(\tau_1', \mathcal{S}_1^*) + H(\tau_2^*) - H(\tau_2') \\
&= \text{tr} \left( h_i^{\tau_2^*[i]}(0) \right) - \text{tr} \left( h_i^{\tau_2'[i]}(0) \right) + H(\tau_2^*) - H(\tau_2') \\
&> H(\tau_2^*) - H(\tau_2'). \tag{3.22}
\end{aligned}$$

If  $i \in \mathcal{S}_1^*$ , then we have  $\tau_2^*[i] = \tau_2'[i] = 0$ , i.e.,  $\tau_2^* = \tau_2'$ . Then we obtain  $V(\tau, \pi^*(\tau)) - V(\tau, \pi'(\tau)) > 0$ . If  $i \notin \mathcal{S}_1^*$ , repeating (3.22), we have

$$H(\tau_1^*) - H(\tau_1') > \dots > H(\tau_n^*) - H(\tau_n') > \dots .$$

By Lemma 3.7, we have a time instance  $k$  such that  $i \in \mathcal{S}_k^*$ . Thus, we have  $V(\tau, \pi^*(\tau)) - V(\tau, \pi'(\tau)) > 0$  and this contradicts the optimality of  $\pi^*$ .  $\square$



## Chapter 4

# Multi-hop Network Scheduling for Distributed Control

Wireless communication makes process control systems more seamless and distributed. In contrast to the current control systems in which controllers are deployed at geometrically the same place, those of wireless control systems can be located at the field level as multi-hop network nodes. Since the control loops under such a configuration share a network, its availability affects their performances. Thus, how to use the network resources needs to be considered together with the controller design problem.

This chapter addresses a co-design framework of controllers, scheduling, and routing of a wireless multi-hop sensor and actuator network. The network consists of sensors co-located with estimators and actuators with controllers, see Figure 4.1. Sensors and actuators are distributed over a field and can communicate with their neighborhoods. Figure 4.1 shows a four-plant (red, yellow, blue, and green) case, controlled by the corresponding local control loops consisting of sensors and actuators. We assume that the sensors and actuators are smart enough to carry out regular estimation and control. Sensors observe multiple decoupled discrete-time linear systems and transmit their estimates to the controllers co-located with the actuators. We formulate an optimization problem consisting of control performance and transmission energy consumption. By solving the problem, we obtain the communication schedules, routings, and optimal controllers for each control loop. We also consider how

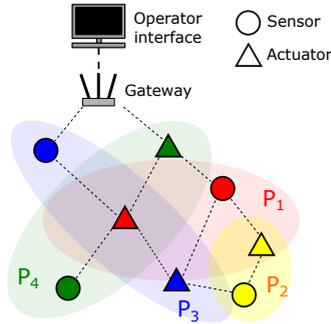


Figure 4.1: Process control over wireless sensor and actuator network discussed in Chapter 1. Four pairs of wireless sensors and actuators are controlling independent processes:  $P_1$  (red),  $P_2$  (yellow),  $P_3$  (blue), and  $P_4$  (green). Sensors and actuators are connected with neighborhood nodes and configures a multi-hop wireless sensor and actuator network. Operators can check the status of the plants by the information taken from the network.

to reconfigure schedules and routings in case of a network link outage.

The main contributions of this chapter are as follows:

- We formulate an optimization problem, minimizing a linear combination of the averaged linear quadratic Gaussian (LQG) control performance and the averaged transmission energy consumption.
- Under the assumption that the network is relatively small, we show that the problem can be divided into scheduling, routing, and optimal control problems. The optimal solution gives periodic schedules and the standard LQG controllers. The solution implies that one can automatically determine a sampling time of the system, which otherwise is usually chosen by a heuristic [116].
- We offer algorithms implemented in the sensors and the actuators, which can detect a network link disconnection and reroute its path when other paths are available.
- A numerical example is provided to illustrate the applicability of the results.

The remainder of the chapter is organized as follows. Section 4.1 describes the system, including process and energy consumption models, and formulates the optimization problem. The optimal solution is discussed in Section 4.2. Algorithms for route reconfiguration are offered in Section 4.3. A numerical example is provided in Section 4.4. Section 4.5 presents a summary.

## 4.1 Problem formulation

In this section, we formulate the problem considered in this chapter. First, we introduce the system model, which includes the plants, smart sensors, and actuators. Next, the energy consumption model of the multi-hop network nodes is provided. Then we formulate the problem.

### 4.1.1 System model

Figure 4.2 shows the block diagram discussed in this chapter, where  $N$  independent control loops are introduced to regulate each plant. Sensors have functionality of an estimator, and smart actuators have that of a controller. Control loops share a multi-hop network to transmits their information. Consider  $N$  discrete-time linear plants

$$x_{k+1}^{(i)} = A_i x_k^{(i)} + B_i u_k^{(i)} + w_k^{(i)}, \quad i \in \mathcal{N}, \quad (4.1)$$

where  $x_k^{(i)} \in \mathbb{R}^{n_i}$  is the state vector at time  $k$ ,  $u_k^{(i)} \in \mathbb{R}^{m_i}$  the input,  $w_k^{(i)} \in \mathbb{R}^{n_i}$  zero-mean independent and identically distributed (i.i.d.) Gaussian noise with covariance  $W_i$ , and  $\mathcal{N} = \{1, \dots, N\}$  the plant index set. Each plant is monitored and controlled by a sensor-actuator pair  $\mathcal{C}_i = \{s_i, a_i\}$ . The sensors have measurements

$$y_k^{(i)} = C_i x_k^{(i)} + v_k^{(i)}, \quad i \in \mathcal{N}, \quad (4.2)$$

where  $y_k^{(i)} \in \mathbb{R}^{p_i}$  is the output, and  $v_k^{(i)} \in \mathbb{R}^{p_i}$  is zero-mean i.i.d. Gaussian noise with covariance  $V_i$ . The pairs of  $N$  sensors and actuators are distributed over a field and connected through an underlying communication network denoted  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \bigcup_{i=1}^N \mathcal{C}_i$  is the sensor and actuator node set, and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of communication links.

Define the information set available at sensor  $i$  at time  $k$  as

$$\mathcal{I}_{s,k}^{(i)} = \{y_0^{(i)}, \dots, y_k^{(i)}, u_0^{(i)}, \dots, u_{k-1}^{(i)}, \nu_0^{(i)}, \dots, \nu_k^{(i)}\}$$

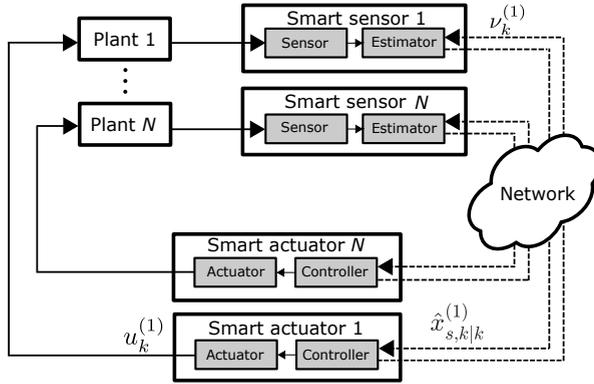


Figure 4.2: Block diagram of the system considered. The  $N$  control loops share a communication network. Smart sensors have the functionality of an estimator, and smart actuators have that of a controller. Smart sensor  $i$  transmits its estimated value  $\hat{x}_{s,k|k}^{(i)}$  to the corresponding smart actuator. The data update request (decision variable)  $\nu_k^{(i)}$  is sent back from the smart actuator  $i$  to the smart sensor  $i$ .

where  $\nu_k^{(i)} \in \{0, 1\}$  is the decision variable such that  $\nu_k^{(i)} = 1$  when the state estimate  $\hat{x}_{s,k|k}^{(i)}$  is transmitted to actuator  $a_i$ . We assume that the transmission is carried out without failure until a link outage occurs. Decisions for transmission are made by each actuator and fed back to the corresponding sensor. This enables the actuator to detect a link outage, which is discussed later. Note that actuators are not required to transmit their decisions at every time instance since the transmission is perfect and then the sensors can emulate the controllers.

The state estimate and the corresponding error covariance at sensor  $i$  are given by

$$\begin{aligned}\hat{x}_{s,k|k-1}^{(i)} &\triangleq \mathbb{E}[x_k^{(i)} | \mathcal{I}_{s,k-1}^{(i)}], \\ \hat{x}_{s,k|k}^{(i)} &\triangleq \mathbb{E}[x_k^{(i)} | \mathcal{I}_{s,k}^{(i)}], \\ P_{s,k|k-1}^{(i)} &\triangleq \mathbb{E}[(x_k^{(i)} - \hat{x}_{s,k|k-1}^{(i)})(x_k^{(i)} - \hat{x}_{s,k|k-1}^{(i)})^\top | \mathcal{I}_{s,k-1}^{(i)}], \\ P_{s,k|k}^{(i)} &\triangleq \mathbb{E}[(x_k^{(i)} - \hat{x}_{s,k|k}^{(i)})(x_k^{(i)} - \hat{x}_{s,k|k}^{(i)})^\top | \mathcal{I}_{s,k}^{(i)}].\end{aligned}$$

In the same way, define the information set at actuator  $i$  at time  $k$  as

$$\mathcal{I}_{a,k}^{(i)} = \{\nu_0^{(i)}, \dots, \nu_k^{(i)}, \nu_0^{(i)} \hat{x}_{s,0|0}^{(i)}, \dots, \nu_k^{(i)} \hat{x}_{s,k|k}^{(i)}, u_0^{(i)}, \dots, u_{k-1}^{(i)}\}$$

and the state estimate and the error covariance

$$\begin{aligned} \hat{x}_{a,k|k-1}^{(i)} &\triangleq \mathbb{E}[x_k^{(i)} | \mathcal{I}_{a,k-1}^{(i)}], \\ \hat{x}_{a,k|k}^{(i)} &\triangleq \mathbb{E}[x_k^{(i)} | \mathcal{I}_{a,k}^{(i)}], \\ P_{a,k|k-1}^{(i)} &\triangleq \mathbb{E}[(x_k^{(i)} - \hat{x}_{a,k|k-1}^{(i)})(x_k^{(i)} - \hat{x}_{a,k|k-1}^{(i)})^\top | \mathcal{I}_{a,k-1}^{(i)}], \\ P_{a,k|k}^{(i)} &\triangleq \mathbb{E}[(x_k^{(i)} - \hat{x}_{a,k|k}^{(i)})(x_k^{(i)} - \hat{x}_{a,k|k}^{(i)})^\top | \mathcal{I}_{a,k}^{(i)}]. \end{aligned}$$

#### 4.1.2 Energy consumption

We introduce the energy consumption model used in [64]. For data receiving and sending, a node consumes, respectively, the following amount of energy

$$\begin{aligned} E_r &= E_{\text{elec}} p, \\ E_s &= E_{\text{elec}} p + E_{\text{amp}} d^2 p, \end{aligned}$$

where  $p$  bits is an amount of data receiving or sending and  $d$  is the distance to a downstream node. Note that the energy consumption for sending depends on the link used. Denote  $\theta_k^{(i)}((j, \ell)) : \mathcal{E} \rightarrow \{0, 1\}$  as the indicator function whether the data of sensor  $i$  is sent through link  $(j, \ell)$  at time  $k$ . If link  $(j, \ell)$  is used, then  $\theta_k^{(i)}((j, \ell)) = 1$ , otherwise 0. Then the energy consumption of node  $j \in \mathcal{V}$  at time  $k$  is given by

$$\begin{aligned} E_k^{(j)} &= \sum_{\ell: (l,j) \in \mathcal{E}} \left[ E_{\text{elec}} \sum_{i \in \mathcal{N}} c_i \theta_k^{(i)}((j, \ell)) \right] \\ &+ \sum_{\ell: (j,\ell) \in \mathcal{E}} \left[ (E_{\text{elec}} + E_{\text{amp}} d_{j\ell}^2) \sum_{i \in \mathcal{N}} c_i \theta_k^{(i)}((j, \ell)) \right] \end{aligned} \quad (4.3)$$

where  $c_i$  [bit] is a constant amount of data transmitted from sensor  $i$  to actuator  $i$ . It is reasonable to assume that data flow is conserved such that for all  $i \in \mathcal{N}$  and  $k \geq 0$ :

$$\sum_{\ell: (j,\ell) \in \mathcal{E}} \theta_k^{(i)}((j, \ell)) - \sum_{\ell: (\ell,j) \in \mathcal{E}} \theta_k^{(i)}((\ell, j)) = 0, \text{ if } j \neq s_i, a_i, \quad (4.4a)$$

$$\sum_{\ell:(j,\ell)\in\mathcal{E}} \theta_k^{(i)}((j,\ell)) - \sum_{\ell:(\ell,j)\in\mathcal{E}} \theta_k^{(i)}((\ell,j)) = \nu_k^{(\ell)}, \text{ if } j = s_i, \quad (4.4b)$$

$$\sum_{\ell:(j,\ell)\in\mathcal{E}} \theta_k^{(i)}((j,\ell)) - \sum_{\ell:(\ell,j)\in\mathcal{E}} \theta_k^{(i)}((\ell,j)) = -\nu_k^{(\ell)}, \text{ if } j = a_i, \quad (4.4c)$$

in order to guarantee that sensor data can reach the corresponding actuator.

**Remark 4.1.** *In Chapter 3, we considered data aggregation, a data compression technique, combining data from different sources into a packet and transmitting it in the same timeslot. We assume in this chapter that the data aggregation rate equals to zero, i.e.,  $r = 0$ .*

### 4.1.3 Optimization problem

We formulate an optimization problem as LQG control with network node energy consumption to find the optimal feedback control, scheduling, and routing. Assuming that the links are arranged in a given order, we define the vector  $\theta_k^{(i)} = [\dots, \theta_k^{(i)}((j,l)), \dots]^\top \in \{0, 1\}^{|\mathcal{E}|}$ . With a weight factor  $\beta_i > 0$  and some vectors  $\nu_k = [\nu_k^{(1)}, \dots, \nu_k^{(N)}]^\top$ ,  $u_k = [u_k^{(1)\top}, \dots, u_k^{(N)\top}]^\top$ , and  $\theta_k = [\theta_k^{(1)\top}, \dots, \theta_k^{(N)\top}]^\top$ , the problem is given by:

**Problem 4.1.**

$$\min_{\{\nu_k, u_k, \theta_k\}_{k=0}^\infty} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} \left[ \sum_{i=1}^N (x_k^{(i)\top} Q_i x_k^{(i)} + u_k^{(i)\top} R_i u_k^{(i)}) + \sum_{j \in \mathcal{V}} \beta_j E_k^{(j)} \right] \quad (4.5a)$$

$$\text{s.t. } (4.4), \quad i \in \mathcal{N}, \quad k \geq 0. \quad (4.5b)$$

Note that controllers cannot access all the variables  $\{\nu_k\}_{k=0}^\infty$ ,  $\{u_k\}_{k=0}^\infty$ , and  $\{\theta_k\}_{k=0}^\infty$ , but we will show in the next section that the optimal solution can be found by distributed optimization at each controller without loss of performance.

## 4.2 Optimal controller and scheduler

In this section, we discuss the optimality of Problem 4.1. By equation (4.3), the last term of (4.5a) can be rewritten as

$$\begin{aligned}
\sum_{j \in \mathcal{V}} \beta_j E_k^{(j)} &= \sum_{i \in \mathcal{N}} \left[ \sum_{\ell: (\ell, j) \in \mathcal{E}} \left[ E_{\text{elec}} \sum_{i \in \mathcal{N}} c_i \theta_k^{(i)}((j, \ell)) \right] \right. \\
&\quad \left. + \sum_{\ell: (j, \ell) \in \mathcal{E}} \left[ (E_{\text{elec}} + E_{\text{amp}} d_{j\ell}^2) \sum_{i \in \mathcal{N}} c_i \theta_k^{(i)}((j, \ell)) \right] \right] \\
&= \sum_{i \in \mathcal{N}} \left[ \sum_{(j, \ell) \in \mathcal{E}} (\beta_j E_{\text{elec}} + \beta_\ell E_{\text{elec}} + \beta_j E_{\text{amp}} d_{j\ell}^2) c_i \theta_k^{(i)}((j, \ell)) \right] \\
&\triangleq \sum_{i \in \mathcal{N}} \left[ \sum_{(j, \ell) \in \mathcal{E}} \alpha_{j\ell} c_i \theta_k^{(i)}((j, \ell)) \right] \\
&\triangleq \sum_{i \in \mathcal{N}} E_k^{(i)} \tag{4.6}
\end{aligned}$$

where  $E_k^{(i)}$  is a weighted total energy consumption of the communication from sensor  $i$  to actuator  $i$ . Now, we have the following lemma.

**Lemma 4.1.** *The optimal solution to Problem 4.1 is obtained by solving the distributed optimization problem:*

$$\min_{\{\nu_k^{(i)}, u_k^{(i)}, \theta_k^{(i)}\}_{k=0}^{\infty}} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} \left[ x_k^{(i)\top} Q_i x_k^{(i)} + u_k^{(i)\top} R_i u_k^{(i)} + \nu_k^{(i)} \tilde{E}_i \right] \tag{4.7}$$

where  $\tilde{E}_i$  is the minimum-cost path for loop  $i$  when  $\nu_k^{(i)} = 1$ , i.e.,  $\tilde{E}_i$  is the optimal value of the problem:

$$\tilde{E}_i \triangleq \min_{\theta_k^{(i)}} \pi^\top \theta_k^{(i)} \quad \text{s.t. (4.4)} \tag{4.8}$$

where  $\pi = [\dots, \pi_{j\ell}, \dots]^\top \in \mathbb{R}^{|\mathcal{E}|}$  is given by  $\pi_{j\ell} = \alpha_{j\ell} c_i$ .

*Proof.* Using (4.6), the objective function (4.5a) is equivalent to the sum of the function

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^{T-1} \left[ x_k^{(i)\top} Q_i x_k^{(i)} + u_k^{(i)\top} R_i u_k^{(i)} + E_k^{(i)} \right]$$

up to  $i = 1, \dots, N$ . Since  $E_k^{(i)}$  is only a function of  $\theta_k^{(i)}$ , and  $x_k^{(i)}$  and  $u_k^{(i)}$  are not affected by  $\theta_k^{(i)}$ , we can take any  $\theta_k^{(i)}$  provided that (4.4) is satisfied. Thus, the optimal value of  $E_k^{(i)}$  when  $\nu_k^{(i)} = 1$  can be obtained by solving problem (4.8).  $\square$

By Lemma 4.1, distributed optimization can achieve the optimality of Problem 4.1.

**Remark 4.2.** Problem (4.8) is the shortest path problem which can be solved by polynomial-time algorithms [220]. The transmission paths are pre-calculated before starting the operation.

**Remark 4.3.** Problem (4.7) is a special case in [143] where the energy consumption is determined by (4.8) and where there is no packet drop.

To see the optimal solution of the distributed optimization problem (4.7), we state the following theorem.

**Theorem 4.1.** There exists a stationary solution to (4.7), and the solution  $\{u_k^{(i)*}\}_{k=0}^\infty$  is given by

$$u_k^{(i)*} = -(B_i^\top S_i B_i + R_i)^{-1} B_i^\top S_i A_i \hat{x}_{a,k|k}^{(i)} \triangleq L_i^{(i)} \hat{x}_{a,k|k}^{(i)} \quad (4.9)$$

with

$$\hat{x}_{a,k|k}^{(i)} = \begin{cases} A_i \hat{x}_{a,k-1|k-1}^{(i)} + B_i u_{k-1}^{(i)}, & \text{if } \nu_k^{(i)} = 0, \\ \hat{x}_{s,k|k}^{(i)} & \text{if } \nu_k^{(i)} = 1, \end{cases} \quad (4.10)$$

$$P_{a,k|k}^{(i)} = \begin{cases} A_i P_{a,k-1|k-1}^{(i)} A_i^\top + W_i, & \text{if } \nu_k^{(i)} = 0, \\ \bar{P}_i, & \text{if } \nu_k^{(i)} = 1, \end{cases} \quad (4.11)$$

where  $S_i \in \mathbb{S}_{++}^n$  is a solution of the algebraic Riccati equation

$$S_i = A_i^\top S_i A_i + Q_i - A_i^\top S_i B_i (B_i^\top S_i B_i + R_i)^{-1} B_i^\top S_i A_i,$$

and  $\bar{P}_i \in \mathbb{S}_{++}^n$  is a solution of the algebraic Riccati equation for the standard Kalman filter at sensor  $i$ . In addition, the stationary solution  $\{\nu_k^{(i)*}\}_{k=0}^\infty$  is given by a threshold policy

$$\nu_k^{(i)*} = \begin{cases} 0, & \text{if } P_{a,k-1|k-1}^{(i)} < P_i^*, \\ 1, & \text{otherwise,} \end{cases} \quad (4.12)$$

where  $P_i^* \in \mathbb{S}_{++}^n$  is the threshold matrix.

*Proof.* Follows from the proof of Theorem 3 in [143]. This is a special case when  $\gamma_k = 1$  in [143].  $\square$

**Remark 4.4.** *The schedule of  $\nu_k^{(i)*}$  converges to the periodic solution when  $A_i$  is unstable. This follows from the fact that there exists  $t_i \in \mathbb{N}$  such that  $h_i^{t_i}(\bar{P}_i) = P_i^*$ , where  $h_i(X) \triangleq A_i X A_i^\top + W_i$  and  $h_i^m(\cdot)$  is the  $m$ -hold composition of  $h_i(\cdot)$ . As in [94, 143], the optimal cost of problem (4.7) is given by*

$$\text{tr}(S_i Q_i) + \frac{1}{t_i + 1} \left[ \text{tr}((A_i^\top S_i A_i + W_i - S_i) \sum_{m=0}^{t_i} h_i^m(\bar{P}_i)) + \tilde{E}_i \right].$$

### 4.3 Link disconnection and route reconfiguration

Theorem 4.1 shows that Problem 4.1 can be divided into optimal scheduling, minimum-cost path, and optimal control problems, which are solved separately. Since the obtained optimal schedule is periodic, and therefore can be fixed offline, both a sensor and an actuator know when the data are communicated. The controller recognizes a link disconnection when it fails to receive the new data from the sensor despite that  $\nu_k^{(i)*} = 1$ . In this case, the path is re-configured by searching for a new one. Let  $\mathcal{P}_i = \{p_1^{(i)}, \dots, p_j^{(i)}, \dots, p_{M_i}^{(i)}\}$  be a set of possible paths from  $s_i$  to  $a_i$  where  $p_j^{(i)} = ((s_i, \cdot), \dots, (\cdot, a_i))$  is the  $j$ -th minimum-cost path. Furthermore, let  $\mathcal{M}_i = \{P_1^{(i)*}, \dots, P_{M_i}^{(i)*}\}$  be the set of threshold matrices induced by each path. The sets  $\mathcal{P}_i$  and  $\mathcal{M}_i$  are pre-set in  $s_i$  and  $a_i$ .

Algorithms 4.1 and 4.2 provide the implementation of the proposed controller, scheduling, and routing reconfiguration. In the algorithms, if the controller detects a link disconnection, it changes its path to the second-best one. If no other paths are available, the control loop switches to fail-safe mode.

### 4.4 Numerical example

To illustrate our results, we consider a small network with  $N = 3$ , where sensors and actuators are distributed over a square field shown in Table 4.1 and Figure 4.3.

---

**Algorithm 4.1** Iterative algorithm for smart sensor  $i$ 


---

Calculate  $\hat{x}_{s,k|k}^{(i)}$   
**if**  $\nu_k^{(i)} = 1$  **then**  
    Send  $\hat{x}_{s,k|k}^{(i)}$  along path  $p_j^{(i)}$   
**end if**  
**if** New  $p_{j+\ell}$  received **then**  
    Calculate  $P_{a,k|k}^{(i)} = f^l(P_{a,k-1|k-1}^{(i)})$   
    Set a new path  $p_{j+\ell}^{(i)}$  and a threshold  $P_{j+\ell}^{(i)*}$   
**else**  
    Calculate  $P_{a,k|k}^{(i)}$  by (4.11)  
**end if**  
Calculate  $\nu_{k+1}^{(i)}$  by (4.12)  
 $k \leftarrow k + 1$

---



---

**Algorithm 4.2** Iterative algorithm for smart actuator  $i$ 


---

**if**  $\nu_k^{(i)} = 1$  **and**  $\hat{x}_{s,k|k}^{(i)}$  not received **then**  
    Calculate  $\hat{x}_{a,k|k}^{(i)} = A_i \hat{x}_{a,k-1|k-1}^{(i)} + B_i u_{k-1}^{(i)}$   
    Calculate  $P_{a,k|k}^{(i)} = f(P_{a,k-1|k-1}^{(i)})$   
    **if**  $j = M_i$  **then**  
        Go to fail safe mode  
    **else**  
        Set a new path  $p_{j+1}^{(i)}$  a threshold  $P_{j+1}^{(i)*}$   
        Send new path  $p_{j+1}^{(i)}$  along a backward route of  $p_{j+1}^{(i)}$   
    **end if**  
**else**  
    Calculate  $\hat{x}_{a,k|k}^{(i)}$  by (4.10)  
    Calculate  $P_{a,k|k}^{(i)}$  by (4.11)  
**end if**  
Calculate  $u_k^{(i)}$  by (4.9)  
Calculate  $\nu_{k+1}^{(i)}$  by (4.12)  
 $k \leftarrow k + 1$

---

	Sensor	Actuator
Loop 1	(5.7521, 1.5978)	(0.4302, 2.6899)
Loop 2	(2.3478, 4.5316)	(8.4912, 8.3172)
Loop 3	(8.2119, 1.1540)	(6.4775, 5.5092)

Table 4.1: Sensor and actuator locations.

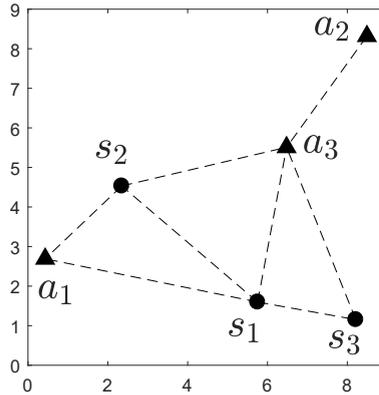


Figure 4.3: Network with three sensor–actuator pairs over a square field.

The system parameters of the three plants are given by

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 1.3 & 0.5 \\ 0.2 & 0.9 \end{bmatrix}, & B_1 &= \begin{bmatrix} 1 \\ 2 \end{bmatrix}, & C_1 &= [1 \quad 1], \\
 A_2 &= \begin{bmatrix} 1.2 & 0 \\ 0 & 1.4 \end{bmatrix}, & B_2 &= \begin{bmatrix} 1 \\ 2 \end{bmatrix}, & C_2 &= [1 \quad 1], \\
 A_3 &= \begin{bmatrix} 1.3 & 1.2 \\ 0 & 1 \end{bmatrix}, & B_3 &= \begin{bmatrix} 1 \\ 2 \end{bmatrix}, & C_3 &= [1 \quad 1],
 \end{aligned}$$

with  $W_i = 0.01I_2$ ,  $V_i = 1$ ,  $Q_i = I_2$ , and  $R_i = 1$  for all  $i = 1, 2, 3$ . For communication parameters, we assume that  $c_i = 32$ , and  $\beta_i = 10$  for  $i = 1, 2, 3$ , and  $E_{elec} = E_{amp} = 1$ . Under the given network parameters, we can derive the minimum-cost path for each control loop as in Table 4.2 and Figure 4.4 (left).

The optimal schedules are shown in Figure 4.5. We see that the solutions are periodic, as stated in Remark 4.4. Sensor  $s_1$  transmits its new estimate ev-

	Path	Energy cost $\tilde{E}_i$
Loop 1	$s_1 \rightarrow s_2 \rightarrow a_1$	$1.000 \times 10^4$
Loop 2	$s_2 \rightarrow a_3 \rightarrow a_2$	$1.086 \times 10^4$
Loop 3	$s_3 \rightarrow a_3$	$0.767 \times 10^4$

Table 4.2: Optimal paths and their energy costs of each control loop.

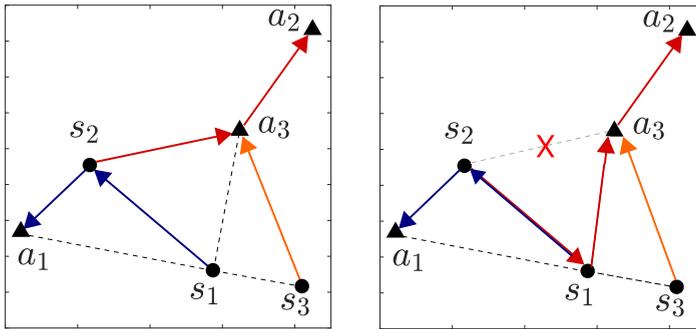


Figure 4.4: The minimum-cost paths for each loop before disconnection (left) and after disconnection (right). The path of loop 1: blue, loop 2: red, and loop 3: orange. The link between  $s_2$  and  $a_3$  is disconnected in the right network which results in the re-routing of loop 2.

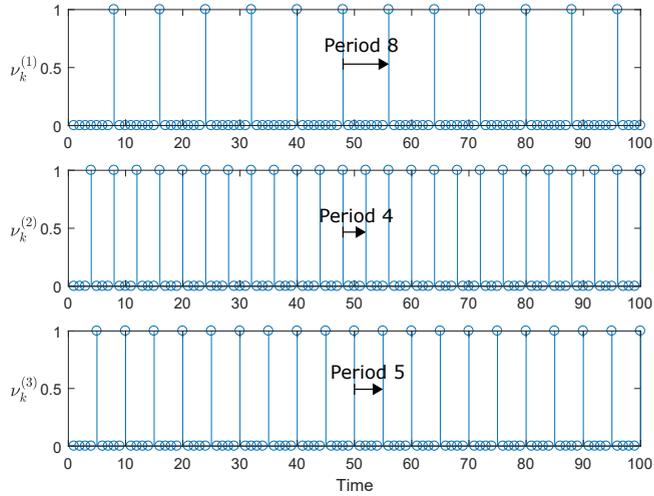


Figure 4.5: Optimal schedules of three loops. Loop 1 has a period 8 schedule, loop 2 has a period 4, and loop 3 has a period 5.

	<b>Proposed method</b>	<b>Every-time communication</b>
Averaged cost	$0.861 \times 10^4$	$3.022 \times 10^4$

Table 4.3: Comparison of averaged cost between the proposed method and the every-time communication strategy.

every eighth-time instance, sensor  $s_2$  every fourth-time instance, and sensor  $s_3$  every fifth-time instance, respectively. The difference of the periods among the loops comes from the relation of the eigenvalues of  $A_i$  and the energy costs for transmission. The optimal averaged cost of the proposed method is shown in Table 4.3 compared to the case that all the sensors communicate with the actuators at every time instance, i.e.,

$$\sum_{i \in \mathcal{N}} \left[ \text{tr}(S_i Q_i + (A_i^\top S_i A_i + W_i - S_i) \bar{P}_i) + \tilde{E}_i \right].$$

We find that the proposed method obtains lower cost than the every-time transmission strategy.

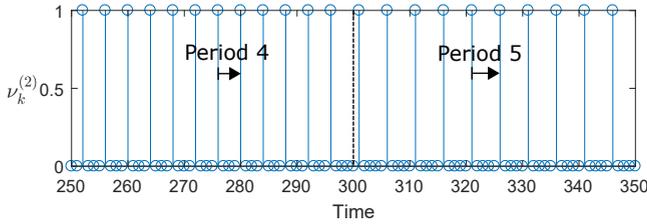


Figure 4.6: Optimal schedule of loop 2 around rerouting at  $k = 300$ . Disconnection leads to a longer period schedule with period 5.

We simulate link disconnection between  $s_2$  and  $a_3$  at time  $k = 300$ , which leads to reroute the path between  $s_2$  and  $a_2$  to  $((s_2, s_1), (s_1, a_3), (a_3, a_1))$  as shown in Figure 4.4 (right). The optimal schedule of loop 2 obtained by Algorithms 4.1 and 4.2 is indicated in Figure 4.6. Since it leads to more energy consumption, the period of loop 2 becomes five, which is longer than the period before the disconnection. The averaged energy consumptions of sensors and actuators from  $k = 0$  to  $k = T$ , i.e.,

$$E_j^{\text{ave}}(T) \triangleq \frac{1}{T+1} \sum_{k=0}^T E_{j,k}, \quad j \in \mathcal{V},$$

are shown in Figure 4.7. We found that the averaged energy consumption of  $s_1$  increases after  $k = 300$  since it is used as a new intermediate node for loop 2. The energy consumption of  $s_2$  and  $a_3$  decreases since the period of loop 2 activation becomes longer.

## 4.5 Summary

This chapter investigated a co-design framework of LQG control, sensor scheduling, and routing over a shared multi-hop sensor-actuator network when the multiple controllers regulated decoupled plants. We formulated an optimization problem, minimizing an infinite-time averaged LQG control performance and energy consumption. It was shown that the problem could be separated into optimal scheduling, routing, and control problems, which were solved independently. Each optimal schedule was periodic, the route was the minimum-cost path, and the controller was provided in the form of a standard

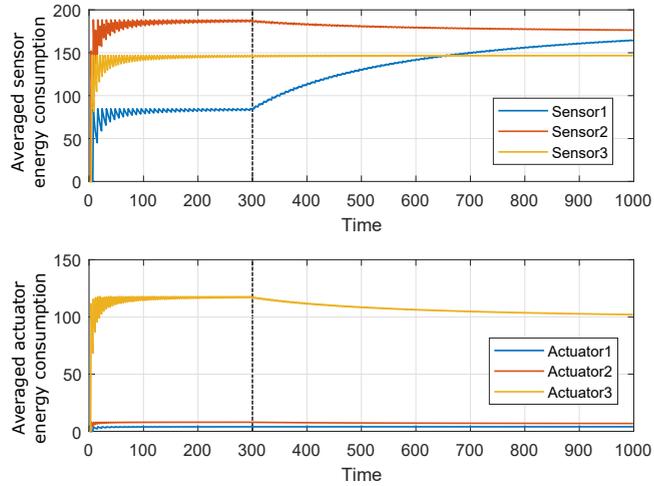


Figure 4.7: Averaged energy consumption for sensor and actuator communication. Averaged energy consumption of Sensor 1 increases after the disconnection at  $k = 300$  while those of Sensor 2 and Actuator 3 decrease.

LQG controller. We proposed algorithms for sensors and actuators to let them configure a new path when a link was disconnected. A numerical example was provided to see how the scheduling and the routing were designed with the LQG controller. The route reconfiguration algorithms were also illustrated in the numerical example.



**Part II**

**Event-triggered Process  
Control**



## Chapter 5

# Event-triggered Actuation for Multi-loop Control Systems

In process control applications, multi-loop control systems, such as cascade and decoupling control, are often introduced to enhance the control performance. Cascade control employs another controller so that it provides a tighter inner control loop. Decoupling control is introduced between two independent PID control loops when their control signal variations interact with each other. They adjust the corresponding control signals so that the interactions are mitigated proactively. The main purpose of this chapter is to investigate a resource-aware strategy of PID, cascade, and decoupling control over wireless networks.

This chapter proposes an event-triggered actuation framework for PID control, cascade, and decoupling control. We consider event-triggered output feedback control for a continuous-time linear system, where the plant state is measured periodically by multiple sampled-data sensors. The control signal is sent to the actuator in a periodic event-triggered fashion. Since the control signal is transmitted through a multi-hop network, the sensors and actuators consume energy as relay nodes. Our strategy effectively reduces the communication load in the network but also the number of control command changes. The controller updates the signal to the actuator when its value goes beyond a given threshold.

The main contributions of this chapter are outlined as follows:

- We introduce a general event-triggered output feedback control system with delayed sampling. An exponential stability condition is derived us-

ing a Lyapunov–Krasovskii functional based on Wirtinger’s inequality (Theorem 5.1).

- By modifying the event-triggering condition and introducing an observer, we show that the proposed controller achieves setpoint tracking and disturbance rejection (Theorems 5.2 and 5.3).
- A tuning procedure is provided for the event threshold under a given stability margin (Corollary 5.1). The optimal threshold is obtained by solving a semi-definite programming (SDP) problem.
- We apply this framework to PID control as well as cascade and decoupling control. Numerical examples for each case are presented to illustrate how to tune the event thresholds. The examples show that our proposed framework reduces the communication loads without performance degradation.

The remainder of this chapter is organized as follows. Section 5.1 introduces the plant model and formulates the problem considered. In Section 5.2, we derive the stability condition of the proposed event-triggered control. Setpoint tracking and disturbance rejection properties are also investigated. Applications to PID, cascade, and decoupling control are discussed in Section 5.3. Numerical examples for each controller are also provided. The conclusion is presented in Section 5.4.

## 5.1 System model and problem formulation

In this section, we formulate the problem considered. We first introduce a plant model given by a continuous-time linear system. A PID controller is then introduced together with cascade and decoupling controllers. Finally, the event trigger problem studied in this chapter is formulated.

### 5.1.1 System model

Consider a continuous-time linear plant measured by  $N$  sensors given by

$$\dot{x}_p(t) = A_p x_p(t) + B_p \tilde{u}(t_k) + B_d d(t), \quad t \in [t_k, t_{k+1}), \quad (5.1)$$

$$y_i(t) = C_p^i x_p(t), \quad i \in \mathcal{N} \triangleq \{1, \dots, N\}, \quad (5.2)$$

where  $x_p(t) \in \mathbb{R}^{n_p}$ ,  $\tilde{u}(t) \in \mathbb{R}^m$ ,  $d(t) \in \mathbb{R}^{n_d}$  and  $y_i(t) \in \mathbb{R}$  are the state, event-triggered control signal applied to the actuator, disturbance, and output of sensor  $i$ , respectively. We assume that the controller updates its state and control signal (defined below for each controller) every  $h_0$  time interval, and let  $t_k$  be the time of update  $k \in \mathbb{N}_0$  of the controller, i.e.,  $t_k = kh_0$  for all  $k \in \mathbb{N}_0$ . Sensor  $i$  samples and transmits its measurement every  $h_i$  time interval. The control signal is computed based on the measurements from sensor  $y_i(s_i(t_k))$ ,  $i \in \mathcal{N}$  and  $r(t_k)$  where  $s_i(t_k)$  is the last transmission time of sensor  $i$  at time  $t_k$ . That is, we have  $s_i(t_k) = \ell_i h_i$  where  $\ell_i = \lfloor t_k/h_i \rfloor$ .

We especially consider three controllers: PID, cascade, and decoupling controllers. Their state-space formulations are given as follows.

### PID control

The block diagram of event-triggered PID control is shown in Figure 5.1a. Note that PID control includes a single sensor, i.e.,  $N = 1$ . A standard PID controller can be written as

$$\dot{x}_c(t) = r(t_k) - y_1(s_1(t_k)), \quad (5.3)$$

$$\begin{aligned} u(t) = & K_p(br(t_k) - y_1(s_1(t_k))) + K_i x_c(t_k) \\ & + K_d(c\Delta r(t_k) - \Delta y_1(s_1(t_k))), \end{aligned} \quad (5.4)$$

where  $K_p, K_i, K_d$  are proportional, integral, and derivative gains, respectively, and  $b, c$ , weight parameters. For the derivative term, we use the backward Euler method, i.e.,

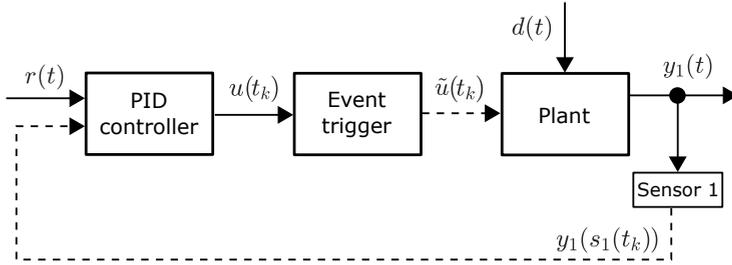
$$\begin{aligned} \Delta y_i(s_i(t_k)) &= \frac{1}{h_i} (y_i(s_i(t_k)) - y_i(s_i(t_k) - h_i)), \quad i \in \mathcal{N}, \\ \Delta r(t_k) &= \frac{1}{h_0} (r(t_k) - r(t_k - h_0)). \end{aligned}$$

The derivative term is usually implemented with a first-order filter [221]. In this case, the controller state, consisting of the integrator and derivative filter, is given by

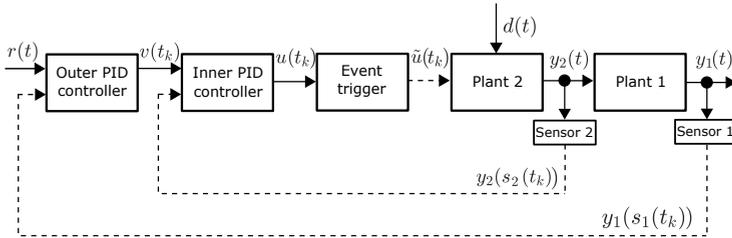
$$\dot{x}_{c,1}(t) = r(t_k) - y_1(s_1(t_k)), \quad (5.5)$$

$$\dot{x}_{c,2}(t) = -\frac{1}{T_1} x_{c,2}(t_k) + \frac{1}{T_1} (c\Delta r(t_k) - \Delta y_1(s_1(t_k))), \quad (5.6)$$

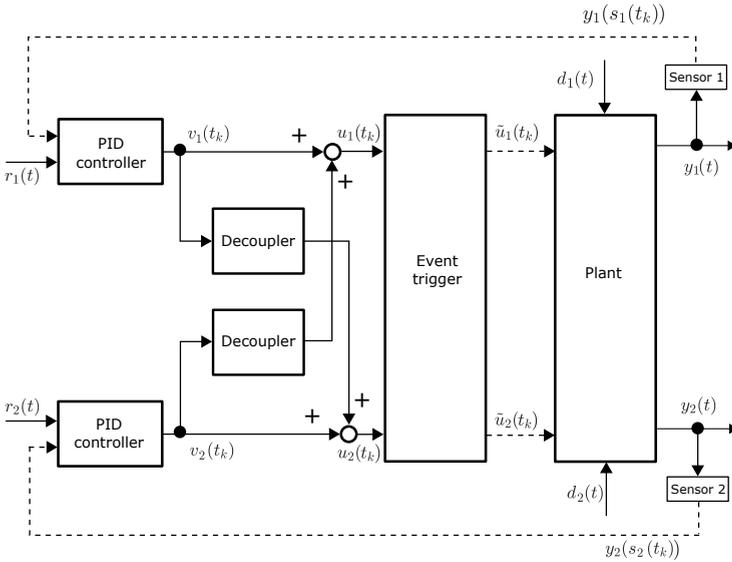
$$u(t) = K_p(br(t_k) - y_1(s_1(t_k))) + K_i x_{c,1}(t_k) + K_d x_{c,2}(t_k), \quad (5.7)$$



(a) PID control



(b) Cascade control



(c) Decoupling control

Figure 5.1: Block diagrams of event-triggered PID, cascade, and decoupling control.

where  $x_{c,1}(t), x_{c,2}(t)$  is the integrator and the filter states, respectively, of the PID controller, and  $T_1$  is the filter's time constant.

### Cascade control

In cascade control (Figure 5.1b), the outer PID controller computes its control signal for reference of the inner controller. The inner controller sends its signal to the actuator, i.e.,

$$\begin{aligned}\dot{x}_{c,1}(t) &= r(t_k) - y_1(s_1(t_k)), \\ \dot{x}_{c,2}(t) &= -\frac{1}{T_1}x_{c,2}(t_k) + \frac{1}{T_1}(c\Delta r(t_k) - \Delta y_1(s_1(t_k))), \\ \dot{x}_{c,3}(t) &= v(t_k) - y_2(s_2(t_k)), \\ \dot{x}_{c,4}(t) &= -\frac{1}{T_2}x_{c,4}(t_k) - \frac{1}{T_2}\Delta y_2(s_2(t_k)), \\ v(t) &= bK_p^1(r(t_k) - y_1(s_1(t_k))) + K_i^1x_{c,1}(t_k) + K_d^1x_{c,2}(t_k), \\ u(t) &= K_p^2(v(t_k) - y_2(s_2(t_k))) + K_i^2x_{c,3}(t_k) + K_d^2x_{c,4}(t_k),\end{aligned}$$

where  $x_{c,1}(t), x_{c,2}(t)$  are the integrator and filter states, respectively, of the outer PID controller,  $x_{c,3}(t), x_{c,4}(t)$  those of the inner PID controller,  $K_p^i, K_i^i, K_d^i, T_i, i = 1, 2$ , are the corresponding proportional, integral, and derivative gains, and the filter's time constants.

**Remark 5.1.** *The inner controller is given by setting  $b = 1$  and  $c = 0$  in PID control (5.5)–(5.7). This form is suitable for rejecting input disturbance.*

### Decoupling control

In decoupling control (Figure 5.1c), two PID controllers are interconnected. The controller dynamics is given by

$$\begin{aligned}\dot{x}_{c,1}(t) &= r_1(t_k) - y_1(s_1(t_k)), \\ \dot{x}_{c,2}(t) &= -\frac{1}{T_1}x_{c,2}(t_k) + \frac{1}{T_1}(c_1\Delta r(t_k) - \Delta y_1(s_1(t_k))), \\ \dot{x}_{c,3}(t) &= r_2(t_k) - y_2(s_2(t_k)), \\ \dot{x}_{c,4}(t) &= -\frac{1}{T_2}x_{c,4}(t_k) + \frac{1}{T_2}(c_2\Delta r(t_k) - \Delta y_2(s_2(t_k))), \\ v_1(t) &= K_p^1(r_1(t_k) - y_1(s_1(t_k))) + K_i^1x_{c,1}(t_k) + K_d^1x_{c,2}(t_k),\end{aligned}$$

$$\begin{aligned}
v_2(t) &= K_p^2(r_2(t_k) - y_2(s_2(t_k))) + K_i^2 x_{c,3}(t_k) + K_d^2 x_{c,4}(t_k), \\
u_1(t) &= v_1(t_k) + K_g^1 v_2(t_k), \\
u_2(t) &= v_2(t_k) + K_g^2 v_1(t_k),
\end{aligned}$$

where  $x_{c,1}(t), x_{c,2}(t)$  are the integrator and filter states, respectively, of PID controller 1,  $x_{c,3}(t), x_{c,4}(t)$  those of controller 2. The parameters  $K_p^i, K_i^i, K_d^i, K_g^i, T_i, b_i, c_i, i = 1, 2$ , are the corresponding proportional, integral, derivative, decoupler gains, time constants, and weights, respectively.

### 5.1.2 Problem formulation

Next, we formulate the problem considered in this chapter. Considering a plant given by (5.1)–(5.2), together with PID, cascade, or decoupling control, this chapter aims to obtain an event-triggered actuation framework based on periodic event-triggered control. The event-triggering condition is given by

$$(u(t_k) - \tilde{u}(t_{k-1}))^\top \Omega (u(t_k) - \tilde{u}(t_{k-1})) > \sigma u^\top(t_k) \Omega u(t_k) + \rho \quad (5.8)$$

where  $\sigma \geq 0$  is a relative threshold,  $\Omega \in \mathbb{S}_{++}^m$  a matrix, and  $\rho > 0$  a constant, all to be determined. The event-triggered control signal is given by

$$\tilde{u}(t) = \begin{cases} u(t_k), & t \in [t_k, t_{k+1}), & \text{if (5.8) is true,} \\ \tilde{u}(t_{k-1}), & t \in [t_k, t_{k+1}), & \text{if (5.8) is false,} \end{cases} \quad (5.9)$$

with  $\tilde{u}_0 = u(t_0)$ .

**Remark 5.2.** *The positive constant  $\rho$  excludes Zeno behavior for continuous event-triggered control [181]. For periodic event-triggered control, we can derive an exponential stability condition when  $\rho = 0$ . A small  $\rho$  can reduce the event generation.*

To handle all controllers introduced above in a general form, we introduce an output feedback controller given by

$$\begin{aligned}
\dot{x}_c(t) &= A_c x_c(t_k) + \sum_{i \in \mathcal{N}} B_c^i y_i(s_i(t_k)) + \sum_{i \in \mathcal{N}} \bar{B}_c^i y_i(s_i(t_k) - h_i) \\
&\quad + B_r r(t) + \bar{B}_r r(t_k - h_0), \quad t \in [t_k, t_{k+1}), \\
u(t) &= C_c x_c(t_k) + \sum_{i \in \mathcal{N}} D_c^i y_i(s_i(t_k)) + \sum_{i \in \mathcal{N}} \bar{D}_c^i y_i(s_i(t_k) - h_i)
\end{aligned} \quad (5.10)$$

$$+ D_r r(t_k) + \bar{D}_r r(t_k - h_0), \quad (5.11)$$

where  $x_c(t) \in \mathbb{R}^{n_c}$  is the controller state. By augmenting the state  $x(t) = [x_p^\top(t), x_c^\top(t)]^\top \in \mathbb{R}^n$  with  $n \triangleq n_p + n_c$ , we have the following closed-loop system description

$$\begin{aligned} \dot{x}(t) &= Ax(t) + A_0 x(t_k) + \sum_{i \in \mathcal{N}} A_i x(s_i(t_k)) \\ &+ \sum_{i \in \mathcal{N}} \bar{A}_i x(s_i(t_k) - h_i) + B\xi(t) + B_D d(t) \\ &+ B_R r(t_k) + \bar{B}_R r(t_k - h_0) \end{aligned} \quad (5.12)$$

where

$$\begin{aligned} \xi(t) &\triangleq \tilde{u}(t) - u(t) \\ &= \tilde{u}(t_k) - u(t_k), \quad t \in [t_k, t_{k+1}), \end{aligned}$$

is the control signal error due to event-triggered actuation, and the matrices are given by

$$\begin{aligned} A &= \begin{bmatrix} A_p & 0 \\ 0 & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} 0 & B_p C_c \\ 0 & A_c \end{bmatrix}, \quad A_i = \begin{bmatrix} B_p D_c^i C_p^i & 0 \\ B_c^i C_p^i & 0 \end{bmatrix}, \\ \bar{A}_i &= \begin{bmatrix} B_p \bar{D}_c^i C_p^i & 0 \\ \bar{B}_c^i C_p^i & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_p \\ 0 \end{bmatrix}, \quad B_D = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \\ B_R &= \begin{bmatrix} B_p D_r \\ B_r \end{bmatrix}, \quad \bar{B}_R = \begin{bmatrix} B_p \bar{D}_r \\ \bar{B}_r \end{bmatrix}. \end{aligned}$$

We formulate the problem as follows: Consider the system (5.12) under PID, cascade, or decoupling control. How to design the event trigger (5.8)–(5.9) such that the closed-loop system (5.12) is exponentially stable with a given convergence rate and has setpoint tracking and disturbance rejection capabilities?

## 5.2 Main results

In this section, we present the main results of this chapter. First, we derive a stability condition of time-triggered output feedback control. We then discuss event-triggered control. Next, setpoint tracking and disturbance rejection properties are considered. We also discuss how to tune the event trigger parameters.

### 5.2.1 Stability analysis

We first derive the stability condition of the time-triggered control system given by system (5.12) with  $\xi(t) \equiv 0$ . Define the following matrices:

$$\begin{aligned}\bar{A} &\triangleq A + A_0 + \sum_{i \in \mathcal{N}} A_i, & \tilde{A} &\triangleq \bar{A} + \sum_{i \in \mathcal{N}} \bar{A}_i, & K_0 &\triangleq [0 \quad C_c], \\ K_i &= [D_c^i C_p^i \quad 0], & \bar{K}_i &\triangleq [\bar{D}_c^i C_p^i \quad 0], & \bar{K} &\triangleq K_0 + \sum_{i \in \mathcal{N}} K_i, \\ \tilde{K} &\triangleq \bar{K} + \sum_{i \in \mathcal{N}} \bar{K}_i, & K_R &= D_r, & \bar{K}_R &= \bar{D}_r.\end{aligned}$$

**Lemma 5.1.** *Consider the closed-loop system (5.12) with  $\xi(t) \equiv d(t) \equiv r(t) \equiv 0$ . Assume that there exist  $P, W_0, W_i, \bar{W}_i, Q_i, R_i \in \mathbb{S}_{++}^n, i \in \mathcal{N}$ , and  $\alpha > 0$ , such that  $\Phi = \Phi^\top = \{\Phi_{j\ell}\} < 0$  where*

$$\begin{aligned}\Phi_{11} &= P\bar{A} + \bar{A}^\top P + 2\alpha P + \sum_{i \in \mathcal{N}} [Q_i - e^{-2\alpha h_i} R_i], \\ \Phi_{1(i+1)} &= P\bar{A}_i + e^{-2\alpha h_i} R_i, \quad i = 1, \dots, N, \\ \Phi_{1(i+N+2)} &= PA_i, \quad i = 1, \dots, N, \\ \Phi_{1(i+2N+2)} &= P\bar{A}_i, \quad i = 1, \dots, N, \\ \Phi_{1(3N+2)} &= \bar{A}^\top S, \\ \Phi_{(i+1)(i+1)} &= -e^{-2\alpha h_i} (Q_i + R_i), \quad i = 1, \dots, N, \\ \Phi_{(i+1)(3N+3)} &= \bar{A}_i^\top S, \quad i = 1, \dots, N, \\ \Phi_{(i+N+2)(i+N+2)} &= -\frac{\pi^2}{4} W_i, \quad i = 0, 1, \dots, N, \\ \Phi_{(i+N+2)(3N+3)} &= A_i^\top S, \quad i = 0, 1, \dots, N, \\ \Phi_{(i+2N+2)(i+2N+2)} &= -\frac{\pi^2}{4} e^{-2\alpha h_i} \bar{W}_i, \quad i = 1, \dots, N, \\ \Phi_{(i+2N+2)(3N+3)} &= -\bar{A}_i^\top S, \quad i = 1, \dots, N, \\ \Phi_{(3N+3)(3N+3)} &= -S,\end{aligned}$$

with

$$S \triangleq h_0^2 e^{-2\alpha h_0} W_0 + \sum_{i \in \mathcal{N}} \left[ (h_0 + h_i)^2 e^{-2\alpha(h_0 + h_i)} (W_i + \bar{W}_i) + h_i^2 R_i \right],$$

and the other elements are zero matrices. Then the closed-loop system (5.12) is exponentially stable with decay rate  $\alpha$ .

*Proof.* See Appendix 5.A.  $\square$

We then have the following stability condition for the event-triggered control systems.

**Theorem 5.1.** Consider the closed-loop system (5.12) with  $d(t) \equiv r(t) \equiv 0$  and the event trigger (5.8)–(5.9) with  $\rho = 0$ . Assume that there exist  $P, W_0, W_i, \bar{W}_i, Q_i, R_i \in \mathbb{S}_{++}^n, i \in \mathcal{N}, w > 0, \sigma > 0$ , and  $\alpha > 0$ , such that

$$\Psi \triangleq \begin{bmatrix} & & & & & & PB & w\sigma\bar{K}^\top\Omega \\ & & & & & & 0 & w\sigma\bar{K}_1^\top\Omega \\ & & & & & & \vdots & \vdots \\ & & & & & & 0 & w\sigma\bar{K}_N^\top\Omega \\ & & & & & & 0 & w\sigma K_0^\top\Omega \\ & & & & & & \vdots & \vdots \\ & & & & & & 0 & w\sigma K_N^\top\Omega \\ & & & & & & 0 & w\sigma\bar{K}_1^\top\Omega \\ & & & & & & \vdots & \vdots \\ & & & & & & 0 & w\sigma\bar{K}_N^\top\Omega \\ & & & & & & SB & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -w\Omega & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & -w\sigma\Omega \end{bmatrix} < 0. \quad (5.13)$$

Then the closed-loop system (5.12) is exponentially stable with decay rate  $\alpha$ .

*Proof.* See Appendix 5.B.  $\square$

### 5.2.2 Setpoint tracking and disturbance rejection

In the previous subsection, we provided a stability condition of event-triggered control without reference signals or disturbances. In this case, the equilibrium point of the closed-loop system (5.12) is the origin. However, when there is a constant reference signal or disturbance, each element of the augmented state  $x(t)$  converges possibly to a non-zero value even if the event-triggered controller successfully stabilizes the plant. The steady-state control signal may also be non-zero, which causes a tracking error even if the controller contains

an integrator. This fact imposes modification of the event-triggering condition. Considering this, we discuss setpoint tracking and disturbance rejection of event-triggered control. We assume that the reference signal and disturbance changes are sufficiently slow compared to the plant dynamics and hence can be regarded as constants.

First, setpoint tracking is considered. We show the following lemma.

**Lemma 5.2.** *If  $\Psi < 0$ , then  $\tilde{A}$  is invertible.*

*Proof.* Let us denote

$$\zeta = \underbrace{[x^\top, \dots, x^\top]^\top}_{N+1}, 0, \dots, 0]^\top$$

where  $x \in \mathbb{R}^n$  is an arbitrary non-zero vector. Then we have

$$\zeta^\top \Psi \zeta = x^\top \left( P\tilde{A} + \tilde{A}^\top P + 2\alpha P + \sum_{i \in \mathcal{N}} [1 - e^{-2\alpha h_i} Q_i] \right) x < 0.$$

Since  $1 - e^{-2\alpha h_i} > 0$  for  $h_i > 0$ ,  $\tilde{A}$  is Hurwitz, and therefore, invertible.  $\square$

Consider the updated event-triggering condition

$$\begin{aligned} & (u(t_k) - \tilde{u}(t_{k-1}))^\top \Omega (u(t_k) - \tilde{u}(t_{k-1})) \\ & > \sigma (u(t_k) - u_c(t_k))^\top \Omega (u(t_k) - u_c(t_k)) + \rho \end{aligned} \quad (5.14)$$

where  $x_e(t_k) = -\tilde{A}^{-1}(B_R r(t_k) + \bar{B}_R r(t_k - h_0))$  and  $u_c(t_k) = \tilde{K} x_e(t_k) + \bar{K}_R r(t_k) + \bar{K} r(t_k - h_0)$ . Note that  $\tilde{A}$  is invertible when  $\Psi < 0$  (Lemma 5.2). The controller updates the signal according to

$$\tilde{u}(t) = \begin{cases} u(t_k), & t \in [t_k, t_{k+1}), & \text{if (5.14) is true,} \\ \tilde{u}(t_{k-1}), & t \in [t_k, t_{k+1}), & \text{if (5.14) is false,} \end{cases} \quad (5.15)$$

with  $\tilde{u}_0 = u(t_0)$ .

**Remark 5.3.** *The modified event trigger (5.14)–(5.15) extends the trigger (5.8)–(5.9), by including a steady-state control signal term.*

**Theorem 5.2.** *Consider the closed-loop system (5.12) with  $d(t) \equiv 0$  and the event trigger (5.14)–(5.15) with  $\rho = 0$ . Assume that there exist  $P, W_0, W_i, \bar{W}_i, Q_i, R_i \in \mathbb{S}_{++}^n, i \in \mathcal{N}, w > 0, \sigma > 0$ , and  $\alpha > 0$ , such that  $\Psi < 0$ . Then the equilibrium point  $x_e^* \triangleq -\tilde{A}^{-1}(B_R + \bar{B}_R)r$  is exponentially stable with decay rate  $\alpha$ .*

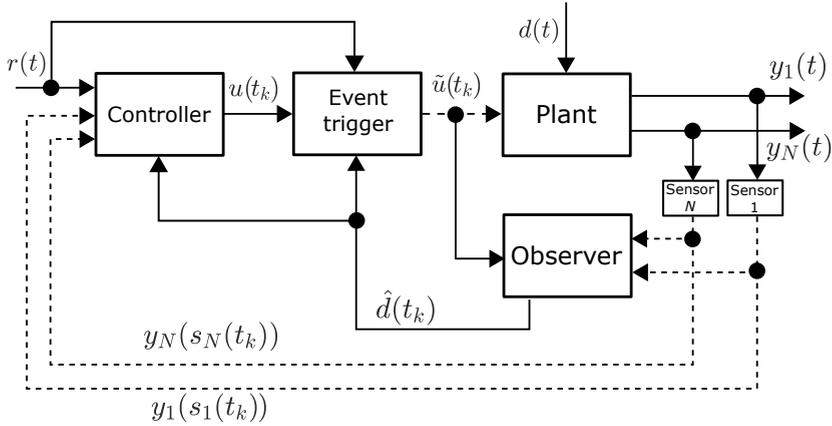


Figure 5.2: Block diagram of the event-triggered control system for setpoint tracking and disturbance rejection.

*Proof.* By Lemma 5.2, there exists an equilibrium point  $x_e^*$ . We apply a coordinate transformation  $\bar{x}(t) = x(t) - x_e^*$ . Then the system (5.12) can be written as

$$\begin{aligned} \dot{\bar{x}}(t) &= A\bar{x}(t) + A_0\bar{x}(t_k) + \sum_{i \in \mathcal{N}} A_i\bar{x}(s_i(t_k)) \\ &\quad + \sum_{i \in \mathcal{N}} \bar{A}_i\bar{x}(s_i(t_k) - h_i) + B\xi(t). \end{aligned}$$

By Theorem 5.1, this system is exponentially stable with the event-triggering condition

$$(\bar{u}(t_k) - \tilde{u}(t_{k-1}))^\top \Omega (\bar{u}(t_k) - \tilde{u}(t_{k-1})) > \sigma \bar{u}^\top(t_k) \Omega \bar{u}(t_k)$$

where  $\bar{u}(t_k) = u(t_k) - u_c(t_k)$ . This completes the proof.  $\square$

Next, we consider disturbance rejection. We introduce an observer to estimate the steady-state control signal, which cannot be obtained when there is an unknown disturbance. The block diagram of the proposed system is shown in Figure 5.2. In this system, the disturbance estimation  $\hat{d}(t_k)$  is used in the controller for feedforward compensation. To model this system, consider the augmented plant

$$\dot{x}_a(t) = A_a x_a(t) + B_a \tilde{u}(t), \quad (5.16)$$

$$y_i(t) = C_a^i x_a(t), \quad i \in \mathcal{N}, \quad (5.17)$$

where  $x_a(t) = [x_p^\top(t), d^\top]^\top \in \mathbb{R}^{n_p+n_d}$  with

$$A_a = \begin{bmatrix} A_p & B_d \\ 0 & 0 \end{bmatrix}, \quad B_a = \begin{bmatrix} B_p \\ 0 \end{bmatrix}, \quad C_a^i = \begin{bmatrix} C_p^i & 0 \end{bmatrix}.$$

For the system (5.16)–(5.17), we introduce an observer with sampled-data implementation

$$\dot{\hat{x}}_a(t) = A_a \hat{x}_a(t_k) + B_a \tilde{u}(t) + \sum_{i \in \mathcal{N}} L_i (y_i(s_i(t_k)) - L C_a \hat{x}(t_k)) \quad (5.18)$$

where  $\hat{x}_a(t) = [\hat{x}_p^\top(t), \hat{d}^\top(t)]^\top$  is the estimate of  $x_a(t)$ ,  $L_i = [L_p^{i\top}, L_d^{i\top}]^\top \in \mathbb{R}^{n_p+n_d}$ ,  $L = [L_1, \dots, L_N]$ , the observer gain, and  $C_a = [C_a^{1\top}, \dots, C_a^{N\top}]^\top$ . Introducing  $e_p(t) \triangleq x_p(t) - \hat{x}_p(t)$ ,  $e_d(t) \triangleq d - \hat{d}(t)$ , and

$$H_p^i \triangleq L_p^i C_p^i, \quad \bar{H}_p \triangleq \sum_{i \in \mathcal{N}} H_p^i, \quad H_d^i \triangleq L_d^i C_p^i, \quad \bar{H}_d \triangleq \sum_{i \in \mathcal{N}} H_d^i,$$

we have the plant state estimation error dynamics

$$\begin{aligned} \dot{e}_p(t) &= A_p x_p(t) - (A_p - \bar{H}_p) x_p(t_k) \\ &\quad - \sum_{i \in \mathcal{N}} H_p^i x_p(s_i(t_k)) + A_p e_p(t_k) + B_d e_d(t_k), \end{aligned}$$

and the disturbance estimation error dynamics

$$\dot{e}_d(t) = -\bar{H}_d x_p(t_k) - \sum_{i \in \mathcal{N}} H_p^i x_p(s_i(t_k)) + \bar{H}_d e_p(t_k).$$

This gives the controller

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t_k) + \sum_{i \in \mathcal{N}} B_c^i y_i(s_i(t_k)) + \sum_{i \in \mathcal{N}} \bar{B}_c^i y_i(s_i(t_k) - h_i) \\ &\quad + B_r r(t) + \bar{B}_r r(t_k - h_0) + B_{\hat{d}} \hat{d}(t_k), \end{aligned} \quad (5.19)$$

$$\begin{aligned} u(t) &= C_c x_c(t_k) + \sum_{i \in \mathcal{N}} D_c^i y_i(s_i(t_k)) + \sum_{i \in \mathcal{N}} \bar{D}_c^i y_i(s_i(t_k) - h_i) \\ &\quad + D_r r(t_k) + \bar{D}_r r(t_k - h_0) + D_{\hat{d}} \hat{d}(t_k). \end{aligned} \quad (5.20)$$

By augmenting the state

$$\mathbf{x}(t) \triangleq \begin{bmatrix} x_p(t) \\ x_c(t) \\ e_p(t) \\ e_d(t) \end{bmatrix} \in \mathbb{R}^{n+n_p+n_d},$$

we have the following closed-loop system description

$$\begin{aligned} \dot{\mathbf{x}}(t) = & \mathbf{A}\mathbf{x}(t) + \mathbf{A}_0\mathbf{x}(t_k) + \sum_{i \in \mathcal{N}} \mathbf{A}_i\mathbf{x}(s_i(t_k)) + \sum_{i \in \mathcal{N}} \bar{\mathbf{A}}_i\mathbf{x}(s_i(t_k) - h_i) \\ & + \mathbf{B}\xi(t) + \mathbf{B}_D d(t) + \mathbf{B}_R r(t_k) + \bar{\mathbf{B}}_R r(t_k - h_0) \end{aligned} \quad (5.21)$$

with

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} A_p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ A_p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} 0 & B_p C_c & 0 & -B_p D_{\hat{d}} \\ 0 & A_c & 0 & -B_{\hat{d}} \\ -A_p + \bar{H}_p & 0 & A_p & B_d \\ \bar{H}_d & 0 & -\bar{H}_d & 0 \end{bmatrix}, \\ \mathbf{A}_i &= \begin{bmatrix} B_p D_c^i C_p^i & 0 & 0 & 0 \\ B_c^i C_p^i & 0 & 0 & 0 \\ -H_p^i & 0 & 0 & 0 \\ -H_d^i & 0 & 0 & 0 \end{bmatrix}, \quad \bar{\mathbf{A}}_i = \begin{bmatrix} B_p \bar{D}_c^i C_p^i & 0 & 0 & 0 \\ \bar{B}_c^i C_p^i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} B_p \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B}_D = \begin{bmatrix} B_d + B_p D_{\hat{d}} \\ B_{\hat{d}} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B}_R = \begin{bmatrix} B_p D_r \\ B_r \\ 0 \\ 0 \end{bmatrix}, \\ \bar{\mathbf{B}}_R &= \begin{bmatrix} B_p \bar{D}_r \\ \bar{B}_r \\ 0 \\ 0 \end{bmatrix}. \end{aligned}$$

Similar to (5.12), let us denote

$$\begin{aligned} \bar{\mathbf{A}} &\triangleq \mathbf{A} + \mathbf{A}_0 + \sum_{i \in \mathcal{N}} \mathbf{A}_i, \quad \tilde{\mathbf{A}} \triangleq \bar{\mathbf{A}} + \sum_{i \in \mathcal{N}} \bar{\mathbf{A}}_i, \quad \mathbf{K}_0 \triangleq [0 \quad C_c \quad 0 \quad 0], \\ \mathbf{K}_i &= [D_c^i C_p^i \quad 0 \quad 0 \quad 0], \quad \bar{\mathbf{K}}_i \triangleq [\bar{D}_c^i C_p^i \quad 0 \quad 0 \quad 0], \quad \bar{\mathbf{K}} \triangleq \mathbf{K}_0 + \sum_{i \in \mathcal{N}} \mathbf{K}_i, \end{aligned}$$

$$\tilde{\mathbf{K}} \triangleq \bar{\mathbf{K}} + \sum_{i \in \mathcal{N}} \bar{\mathbf{K}}_i, \quad \mathbf{K}_R \triangleq D_r, \quad \bar{\mathbf{K}}_R \triangleq \bar{D}_r, \quad \mathbf{K}_D \triangleq D_{\hat{d}}.$$

We now consider the event-triggering condition

$$\begin{aligned} & (u(t_k) - \tilde{u}(t_{k-1}))^\top \Omega(u(t_k) - \tilde{u}(t_{k-1})) \\ & > \sigma(u(t_k) - \mathbf{u}_c(t_k))^\top \Omega(u(t_k) - \mathbf{u}_c(t_k)) + \rho \end{aligned} \quad (5.22)$$

where  $\mathbf{x}_e(t_k) \triangleq -\tilde{\mathbf{A}}^{-1}(\mathbf{B}_D \hat{d}(t_k) + \mathbf{B}_R r(t_k) + \bar{\mathbf{B}}_R r(t_k - h_0))$  and  $\mathbf{u}_c(t_k) = \tilde{\mathbf{K}} \mathbf{x}_e(t_k) + \mathbf{K}_R r(t_k) + \bar{\mathbf{K}}_R r(t_k - h_0) + \mathbf{K}_D \hat{d}(t_k)$ . Under this condition, the controller updates its signal according to

$$\tilde{u}(t) = \begin{cases} u(t_k), & t \in [t_k, t_{k+1}), \quad \text{if (5.22) is true,} \\ \tilde{u}(t_{k-1}), & t \in [t_k, t_{k+1}), \quad \text{if (5.22) is false,} \end{cases} \quad (5.23)$$

with  $\tilde{u}_0 = u(t_0)$ . This event-triggered control systems is exponentially stable, as stated next.

**Theorem 5.3.** *Consider the closed-loop system (5.21) and the event trigger (5.22)-(5.23) with  $\rho = 0$ . Assume that there exist  $\mathbf{P}, \mathbf{W}_0, \mathbf{W}_i, \mathbf{Q}_i, \mathbf{R}_i \in \mathbb{S}^{n+n_p+n_d}$ ,  $w > 0, \sigma > 0$ , and  $\alpha > 0$ , such that  $\Xi = \Xi^\top = \{\Xi_{j\ell}\} < 0$  where*

$$\begin{aligned} \Xi_{11} &= \mathbf{P}\bar{\mathbf{A}} + \bar{\mathbf{A}}^\top \mathbf{P} + 2\alpha \mathbf{P} + \sum_{i \in \mathcal{N}} \left[ \mathbf{Q}_i - e^{-2\alpha h_i} \mathbf{R}_i \right], \\ \Xi_{1(i+1)} &= \mathbf{P}\bar{\mathbf{A}}_i + e^{-2\alpha h_i} \mathbf{R}_i, \quad i = 1, \dots, N, \\ \Xi_{1(i+N+2)} &= \mathbf{P}\mathbf{A}_i, \quad i = 0, 1, \dots, N, \\ \Xi_{1(i+2N+2)} &= \mathbf{P}\bar{\mathbf{A}}_i, \quad i = 1, \dots, N, \\ \Xi_{1(3N+3)} &= \bar{\mathbf{A}}^\top \mathbf{S}, \\ \Xi_{1(3N+4)} &= \mathbf{P}\mathbf{B}, \\ \Xi_{1(3N+5)} &= w\sigma \mathbf{F}^\top \Omega, \\ \Xi_{(i+1)(i+1)} &= -e^{-2\alpha h_i} (\mathbf{Q}_i + \mathbf{R}_i), \quad i = 1, \dots, N, \\ \Xi_{(i+1)(3N+3)} &= \bar{\mathbf{A}}_i^\top \mathbf{S}, \quad i = 1, \dots, N, \\ \Xi_{(i+1)(3N+5)} &= w\sigma \bar{\mathbf{K}}_i^\top \Omega, \quad i = 1, \dots, N, \\ \Xi_{(i+N+2)(i+N+2)} &= -\frac{\pi^2}{4} \mathbf{W}_i, \quad i = 0, 1, \dots, N, \\ \Xi_{(i+N+2)(3N+3)} &= \mathbf{A}_i^\top \mathbf{S}, \quad i = 0, 1, \dots, N, \end{aligned}$$

$$\begin{aligned}
\Xi_{(N+2)(3N+5)} &= w\sigma\mathbf{F}_0^\top\Omega, \\
\Xi_{(i+N+2)(3N+5)} &= w\sigma\mathbf{K}_i^\top\Omega, \quad i = 1, \dots, N, \\
\Xi_{(i+2N+2)(i+2N+2)} &= -\frac{\pi^2}{4}e^{-2\alpha h_i}\bar{\mathbf{W}}_i, \quad i = 1, \dots, N, \\
\Xi_{(i+2N+2)(3N+3)} &= \bar{\mathbf{A}}_i^\top\mathbf{S}, \quad i = 1, \dots, N, \\
\Xi_{(i+2N+2)(3N+5)} &= w\sigma\bar{\mathbf{K}}_i^\top\Omega, \quad i = 1, \dots, N, \\
\Xi_{(3N+3)(3N+3)} &= -\mathbf{S}, \\
\Xi_{(3N+3)(3N+4)} &= \mathbf{S}\mathbf{B}, \\
\Xi_{(3N+4)(3N+4)} &= -w\Omega, \\
\Xi_{(3N+5)(3N+5)} &= -w\sigma\Omega,
\end{aligned}$$

with

$$\mathbf{S} \triangleq h_0^2 e^{-2\alpha h_0} \mathbf{W}_0 + \sum_{i \in \mathcal{N}} \left[ (h_0 + h_i)^2 e^{-2\alpha(h_0 + h_i)} (\mathbf{W}_i + \bar{\mathbf{W}}_i) + h_i^2 \mathbf{R}_i \right],$$

$\mathbf{F}_0 \triangleq \mathbf{K}_0 - \tilde{\mathbf{K}}\tilde{\mathbf{A}}^{-1}\mathbf{B}_D\mathbf{E}$ ,  $\mathbf{F} \triangleq \bar{\mathbf{K}} - \tilde{\mathbf{K}}\tilde{\mathbf{A}}^{-1}\mathbf{B}_D\mathbf{E}$  with  $\mathbf{E} \triangleq [0 \ 0 \ 0 \ I_{n_d}]$ , and the other elements are zero matrices. Then the equilibrium point  $\mathbf{x}_e^* \triangleq -\tilde{\mathbf{A}}^{-1}(\mathbf{B}_D d + (\mathbf{B}_R + \bar{\mathbf{B}}_R)r)$  is exponentially stable with decay rate  $\alpha$ .

*Proof.* We apply a coordinate transformation  $\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_e^*$ . Then the system (5.21) can be rewritten as

$$\begin{aligned}
\dot{\bar{\mathbf{x}}}(t) &= \mathbf{A}\bar{\mathbf{x}}(t) + \mathbf{A}_0\bar{\mathbf{x}}(t_k) + \sum_{i \in \mathcal{N}} \mathbf{A}_i\bar{\mathbf{x}}(s_i(t_k)) \\
&\quad + \sum_{i \in \mathcal{N}} \bar{\mathbf{A}}_i\bar{\mathbf{x}}(s_i(t_k) - h_i) + \mathbf{B}\xi(t).
\end{aligned} \tag{5.24}$$

Using

$$\begin{aligned}
\mathbf{x}_e(t_k) - \mathbf{x}_e^* &= \tilde{\mathbf{A}}^{-1}\mathbf{B}_D e_d(t_k) \\
&= \tilde{\mathbf{A}}^{-1}\mathbf{B}_D\mathbf{E}\mathbf{x}(t_k) \\
&= \tilde{\mathbf{A}}^{-1}\mathbf{B}_D\mathbf{E}\bar{\mathbf{x}}(t_k),
\end{aligned}$$

where the last equality holds since  $\mathbf{E}\mathbf{x}_e^* = 0$ , we have

$$\bar{\mathbf{u}}(t_k) - \mathbf{u}_c(t_k) = \mathbf{F}_0\bar{\mathbf{x}}(t_k) + \sum_{i \in \mathcal{N}} [\mathbf{K}_i\bar{\mathbf{x}}(s_i(t_k)) + \bar{\mathbf{K}}_i\bar{\mathbf{x}}(s_i(t_k) - h_i)].$$

Substituting this into (5.22) and following the proof of Theorem 5.1 for the system (5.24), we have that (5.21) is exponentially stable with decay rate  $\alpha$ .  $\square$

In virtue of Theorem 5.3, we can obtain the optimal threshold  $\sigma^*$  for a given convergence rate  $\alpha$ .

**Corollary 5.1.** *Given  $w > 0$  and  $\alpha > 0$ , if the SDP*

$$\sigma^* \triangleq \max \sigma \quad \text{s.t.} \quad \Xi < 0 \quad (5.25)$$

*is feasible, then the closed-loop system (5.21) with the event trigger (5.22)–(5.23) with  $\sigma = \sigma^*$  is exponentially stable with decay rate  $\alpha$ .*

**Remark 5.4.** *If only setpoint tracking is considered, the observer is not needed. The constraint  $\Xi < 0$  in (5.25) is then simplified to  $\Phi < 0$ .*

**Remark 5.5.** *To find the maximum threshold, a grid of  $w$  can be introduced and the SDP (5.25) solved for every grid point.*

## 5.3 Applications to PID, cascade, and decoupling control

In this section, we apply the theoretical results of the previous section to PID, cascade, and decoupling control. Numerical examples are also provided to illustrate how the proposed controllers and event thresholds are obtained.

### 5.3.1 PI and PID control

First, we apply the results to PI control. A PI controller is given by (5.19)–(5.20) with  $A_c = 0$ ,  $B_c^1 = -1$ ,  $\bar{B}_c^1 = 0$ ,  $B_r = 1$ ,  $\bar{B}_r = 0$ ,  $B_{\hat{d}} = 0$ ,  $C_c = K_i$ ,  $D_c^1 = -K_p$ ,  $\bar{D}_c^1 = 0$ ,  $D_r = K_p$ ,  $\bar{D}_r = 0$ , and  $D_{\hat{d}} = K_f$ , where  $K_f$  is the feedforward gain. By setting the control parameters  $K_p$ ,  $K_i$ ,  $K_f$ , sampling intervals  $h_0$ ,  $h_1$  in (5.3)–(5.4), decay rate  $\alpha$ , and observer gain  $L$ , the maximum threshold is obtained by solving SDP (5.25).

**Example 5.1.** *We first illustrate an example of PI control for a first-order linear system and compare the proposed event-triggered control (ETC) with three different strategies: event-triggered PIDPLUS control (PIDPLUS) [194], event-triggered PI control with send-on-delta triggering (SOD) [195], and time-triggered control (TTC). We assume that all the controllers are co-located at the sensor since*

Type	IAE	# events until $t = 500$
ETC	19.5	47
PIDPLUS	34.7	73
SOD	44.0	96
TTC	27.7	1667

Table 5.1: Comparison of the ETC, PIDPLUS, SOD, and TTC.

*the event-triggered PIDPLUS control needs to have the controller capability at the sensor.*

*Consider a first-order linear system*

$$\begin{aligned}\dot{x}_p(t) &= -0.7x_p(t) + \tilde{u}(t) + d(t) + w(t), \\ y(t) &= x_p(t),\end{aligned}$$

*where  $w(t) \in \mathbb{R}$  is the random process noise, which is assumed to be Gaussian with zero-mean independent and identically distributed with variance 0.25. The control parameters are given with  $K_p = 0.23$ ,  $K_i = 0.077$ , and  $K_f = -0.5$ . The sampling intervals are set with  $h_0 = h_1 = 0.3$  using the criteria in [116]. By solving SDP (5.25), we obtain  $\sigma^* = 2.72$ . The SDP can be effectively solved by YALMIP toolbox [222]. We consider the reference signal  $r(t) = 1, \forall t \geq 0$  and the disturbance  $d(t) = -0.5, \forall t \geq 250$ . The response with  $\rho = 10^{-5}$  is shown in Figure 5.3 together with PIDPLUS, SOD, and TTC.*

*To evaluate the performances, we introduce the Integral of the Absolute Error (IAE) as*

$$IAE = \int_0^{t_f} |r(t) - y_1(t)| dt \quad (5.26)$$

*with  $t_f = 500$ . The IAEs and the numbers of event generations (the average of 1000 times simulations) are shown in Table 5.1. It can be found that the ETC extremely reduces communications compared with the TTC while both controllers update each state every  $h = 0.3$  seconds. The ETC achieves smaller IAE than the PIDPLUS and SOD with less samples. This is partially because observer-based feedforward control is employed in the ETC.*

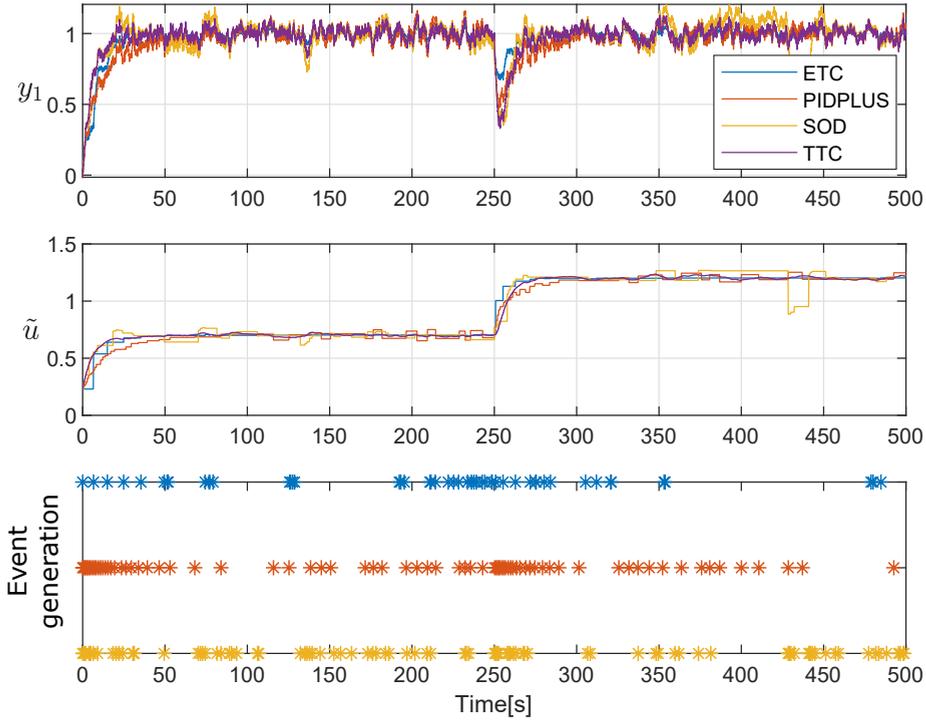


Figure 5.3: Responses of the proposed event-triggered PI control (ETC, blue), PIDPLUS [194] (red), SOD [195] (orange), and TTC (purple). The top plot describes the outputs, the middle control signals, and the bottom event generations.

	PID	PID+FF
$\sigma^*$	0.19	0.40

Table 5.2: Obtained thresholds for the PID (including PI-D and I-PD) control with and without FF.

Next, we consider PID control. Setting PID control parameters  $K_p$ ,  $K_i$ ,  $K_d$ ,  $T_1$ ,  $K_f$ , and sampling intervals  $h_0$ ,  $h_1$  in (5.5)–(5.7), a PID controller can be written as the form (5.19)–(5.20) with

$$\begin{aligned}
 A_c &= \begin{bmatrix} 0 & 0 \\ 0 & -1/T_1 \end{bmatrix}, \quad B_c^1 = \begin{bmatrix} -1 \\ -1/T_1 h_1 \end{bmatrix}, \quad \bar{B}_c^1 = \begin{bmatrix} 0 \\ 1/T_1 h_1 \end{bmatrix}, \\
 B_r &= \begin{bmatrix} 1 \\ c/T_1 h_0 \end{bmatrix}, \quad \bar{B}_r = \begin{bmatrix} 0 \\ -c/T_1 h_0 \end{bmatrix}, \quad B_{\hat{d}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad C_c = [K_i \quad K_d], \\
 D_c^1 &= -K_p, \quad \bar{D}_c^1 = 0, \quad D_r = bK_p, \quad \bar{D}_r = 0, \quad D_{\hat{d}} = K_f.
 \end{aligned}$$

Note that PID controllers have variations based on settings of weight parameters  $b$  and  $c$ . We indicate the case  $b = c = 1$  by PID,  $b = 1, c = 0$  by PI-D, and  $b = c = 0$  by I-PD. Setting decay rate  $\alpha$  and observer gain  $L$ , the maximum threshold  $\sigma^*$  is obtained by solving SDP (5.25).

**Example 5.2.** Consider a linear system

$$\begin{aligned}
 \dot{x}_p(t) &= \begin{bmatrix} -1 & 1 \\ 0 & -10 \end{bmatrix} x_p(t) + \begin{bmatrix} 0 \\ 4 \end{bmatrix} (\tilde{u}(t) + d(t)), \\
 y(t) &= [2.5 \quad 0] x_p(t),
 \end{aligned}$$

and PID, PI-D, and I-PD controllers with the parameters:  $K_p = 1.63$ ,  $K_i = 2.71$ ,  $K_d = 0.075$ , and  $T_1 = 0.052$ . The control parameters are obtained by applying MATLAB function `pidtune`. By solving SDPs (5.25) with intervals  $h_0 = h_1 = 0.01$  and  $\alpha = 0.3$  for the two cases: PID both with and without feedforward controller (FF,  $K_f = -0.5$ ), we find the event thresholds as in Table 5.2.

We consider the reference signal  $r(t) = 1, \forall t \geq 0$  and disturbance  $d(t) = -2, \forall t \geq 10$ . Numerical results with  $\rho = 10^{-5}$  for ETC with and without FF, compared to TTC, are shown in Figures 5.4–5.6 and Table 5.3.

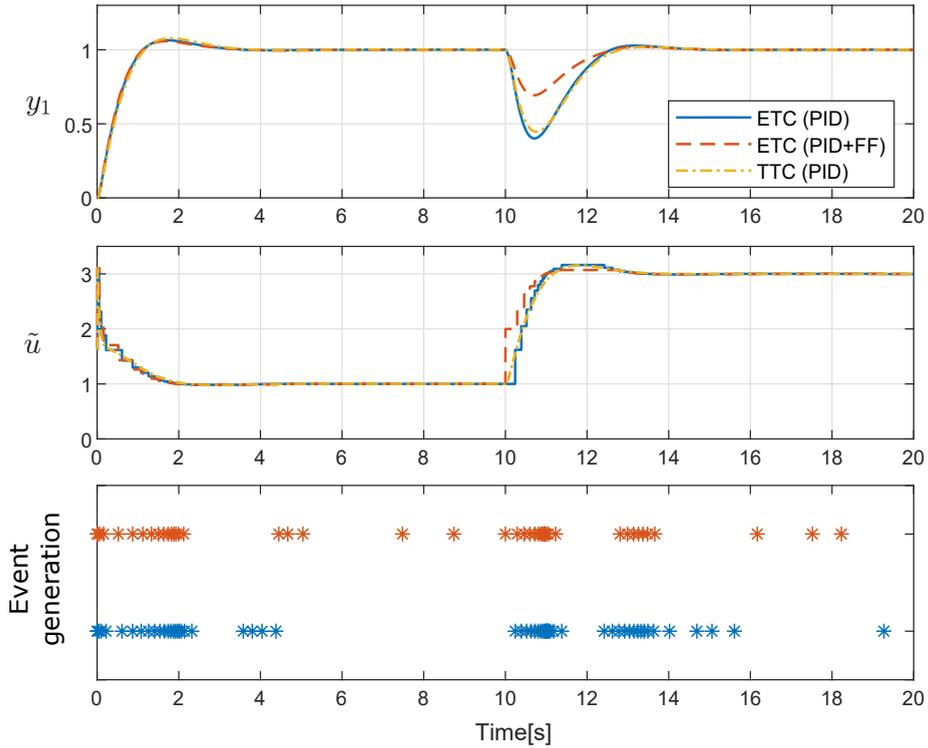


Figure 5.4: Responses of the event-triggered PID control (ETC without FF: blue, with FF: red) and the time-triggered control (TTC: orange). In each figure, three plots show the outputs, control signals, and event generations of the ETCs from the top.

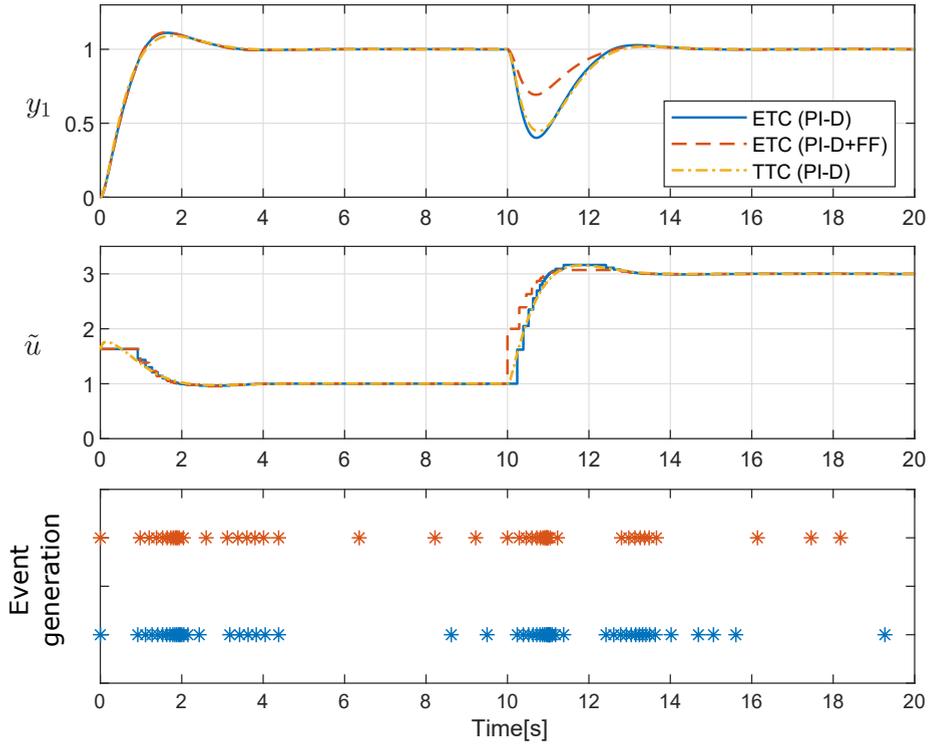


Figure 5.5: Responses of the event-triggered PI-D control (ETC without FF: blue, with FF: red) and the time-triggered control (TTC: orange). In each figure, three plots show the outputs, control signals, and event generations of the ETCs from the top.

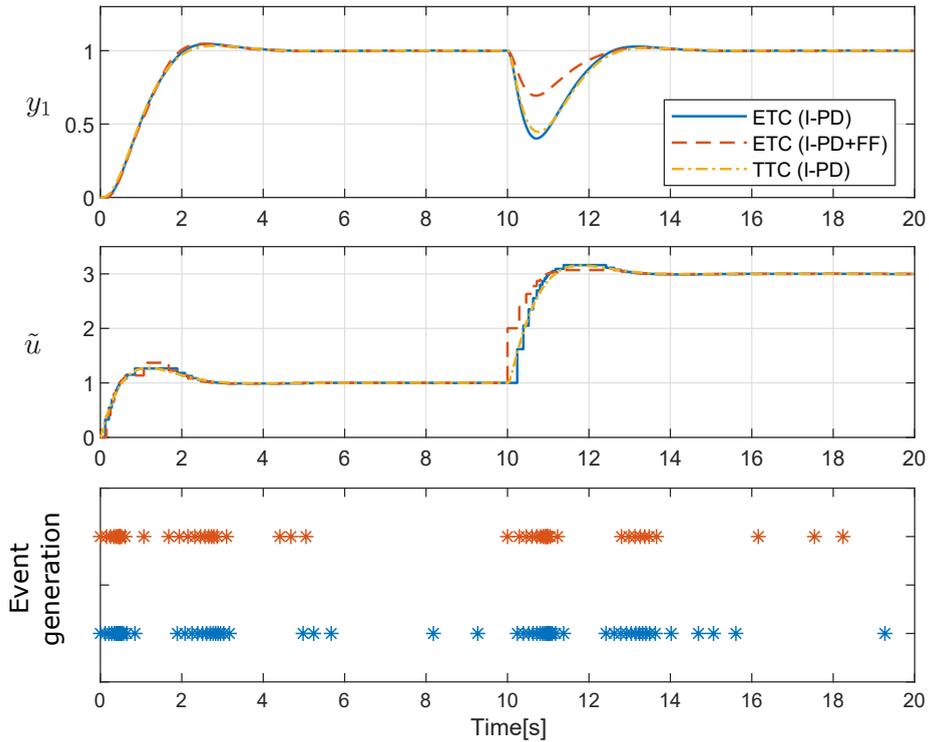


Figure 5.6: Responses of the event-triggered I-PD control (ETC without FF: blue, with FF: red) and the time-triggered control (TTC: orange). In each figure, three plots show the outputs, control signals, and event generations of the ETCs from the top.

Type	$T_r$	$M$	$T_s$	IAE	# events untill $T_s$
ETC (PID)	0.51	6.4	2.80	1.34	21
ETC (PID+FF)	0.49	5.8	2.73	0.94	16
TTC (PID)	0.53	7.7	3.04	1.37	299
ETC (PI-D)	0.59	11.1	2.98	1.46	18
ETC (PI-D+FF)	0.59	11.4	2.96	1.07	14
TTC (PI-D)	0.56	9.1	3.08	1.40	303
ETC (I-PD)	1.15	4.5	3.54	1.90	31
ETC (I-PD+FF)	1.18	4.6	3.41	1.50	25
TTC (I-PD)	1.14	3.3	3.54	1.85	340

Table 5.3: Comparison of the ETCs and TTCs.

As performance metrics, we introduce the rise time  $T_r$ <sup>1</sup>, overshoot  $M$ <sup>2</sup>, settling time  $T_s$ <sup>3</sup>, and the IAE (5.26) with  $t_f = 20$ . It can be found from Figures 5.4–5.6 that the ETCs compensate for the step disturbances as we showed in Theorem 5.3. Thanks to feedforward controllers, the effects of the disturbances are reduced more efficiently. The IAEs of the ETCs with FF are smaller than those of the ETCs without FF and TTCs as in Table 5.3. The response of each ETC (blue solid line) are similar to the TTC (orange dash-dotted line) in Figures 5.4–5.6. We can see that the event-triggered controllers can track their setpoints as well as the time-triggered controllers without performance degradation. In fact, the quantities such as rise time, overshoot, and settling time for the ETCs and TTCs are almost same (Table. 5.3). We also indicate the number of event generations until  $t = T_s$ . It can be seen that event generations are extremely reduced compared to the TTCs while keeping the control performance.

<sup>1</sup>Time that the step response  $y_1(t)$  takes to rise from 10% to 90%

<sup>2</sup>Percentage overshoot compared to the setpoint

<sup>3</sup>Time that error  $|r - y(t)|$  to fall within 2% of the setpoint

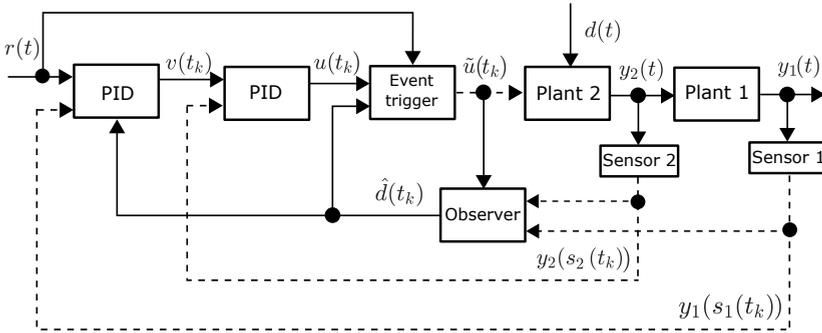


Figure 5.7: Event-triggered cascade control with an observer.

### 5.3.2 Cascade control

With given control and sampling parameters, a cascade controller (5.19)–(5.20) is given by

$$\begin{aligned}
 A_c &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1/T_1 & 0 & 0 \\ K_1^1 & K_d^1 & 0 & 0 \\ 0 & 0 & 0 & -1/T_2 \end{bmatrix}, & B_c^1 &= \begin{bmatrix} -1 \\ -1/T_1 h_1 \\ -K_p^1 \\ 0 \end{bmatrix}, \\
 B_c^2 &= \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1/T_2 h_2 \end{bmatrix}, & \bar{B}_c^1 &= \begin{bmatrix} 0 \\ 1/T_1 h_1 \\ 0 \\ 0 \end{bmatrix}, & \bar{B}_c^2 &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/T_2 h_2 \end{bmatrix}, \\
 B_r &= \begin{bmatrix} 1 \\ c/T_1 h_0 \\ bK_p^1 \\ 0 \end{bmatrix}, & \bar{B}_r &= \begin{bmatrix} 0 \\ -c/T_1 h_0 \\ 0 \\ 0 \end{bmatrix}, & B_{\hat{d}} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_f \end{bmatrix}, \\
 C_c &= [K_p^2 K_1^1 \quad K_p^2 K_d^1 \quad K_1^2 \quad K_d^2], & D_c^1 &= -K_p^1 K_p^2, & D_c^2 &= -K_p^2, \\
 \bar{D}_c^1 &= 0, & \bar{D}_c^2 &= 0, & D_r &= bK_p^1 K_p^2, & \bar{D}_r &= 0, & D_{\hat{d}} &= K_p^2 K_f.
 \end{aligned}$$

The block diagram of event-triggered cascade control with an observer is shown in Figure 5.7.

**Example 5.3.** Consider the system illustrated by Figure 5.7, where plant 1 is

	CAS	CAS+FF	PI	PI+FF
$\sigma^*$	0.79	0.80	0.07	0.18

Table 5.4: Obtained thresholds for the cascade and PI control.

given by

$$\dot{x}_{p,1}(t) = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1.2 & 1 \\ 0 & 0 & -1.5 \end{bmatrix} x_{p,1}(t) + \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} y_2(t),$$

$$y_1(t) = [2.5 \ 0 \ 0] x_{p,1}(t),$$

and plant 2 by

$$\dot{x}_{p,2}(t) = -2x_{p,2}(t) + 2(\tilde{u}(t) + d(t)),$$

$$y_2(t) = 1.5x_{p,2}(t).$$

We apply event-triggered cascade control (CAS) with a PID-PI pair:  $K_p^1 = 0.18$ ,  $K_i^1 = 0.0791$ ,  $K_d^1 = 0.0439$ ,  $T_1 = 0.0175$  for the outer controller and  $K_p^2 = 0.244$ ,  $K_i^2 = 1.82$  for the inner controller, with and without feedforward control  $K_f = -0.5$ . We compare this to PI control with  $K_p = 0.109$ ,  $K_i = 0.0488$ . Solving SDPs (5.25) with intervals  $h_0 = 0.02$ ,  $h_1 = 0.04$ ,  $h_2 = 0.02$ , and  $\alpha = 0.1$ , the event thresholds are obtained as summarized in Table 5.4.

We consider the disturbance signal  $d(t) = 0.05, \forall t \geq 0$ . Numerical results with  $\rho = 10^{-8}$  are shown in Figure 5.8 and Table 5.5. It can be found from Figure 5.8 that the proposed event-triggered cascade controller compensates for the disturbance. Compared to the event-triggered PI control, the effect of the disturbance is significantly reduced. The maximum value of output  $y_1(t)$  is about 0.05 (without FF), while that of the PI control is more than 0.25. The IAE of the cascade control are around 0.3, while that of the PI control are around 1.0. We can also see that the disturbances are effectively compensated by the feedforward control in both cascade and PI controllers. We indicate the number of event generations of each event-triggered controller until  $t = 20$  in Table 5.5. It can be seen that event generations are extremely reduced compared to the time-triggered cascade control with slight performance degradation, which can be further reduced by introducing feedforward control.

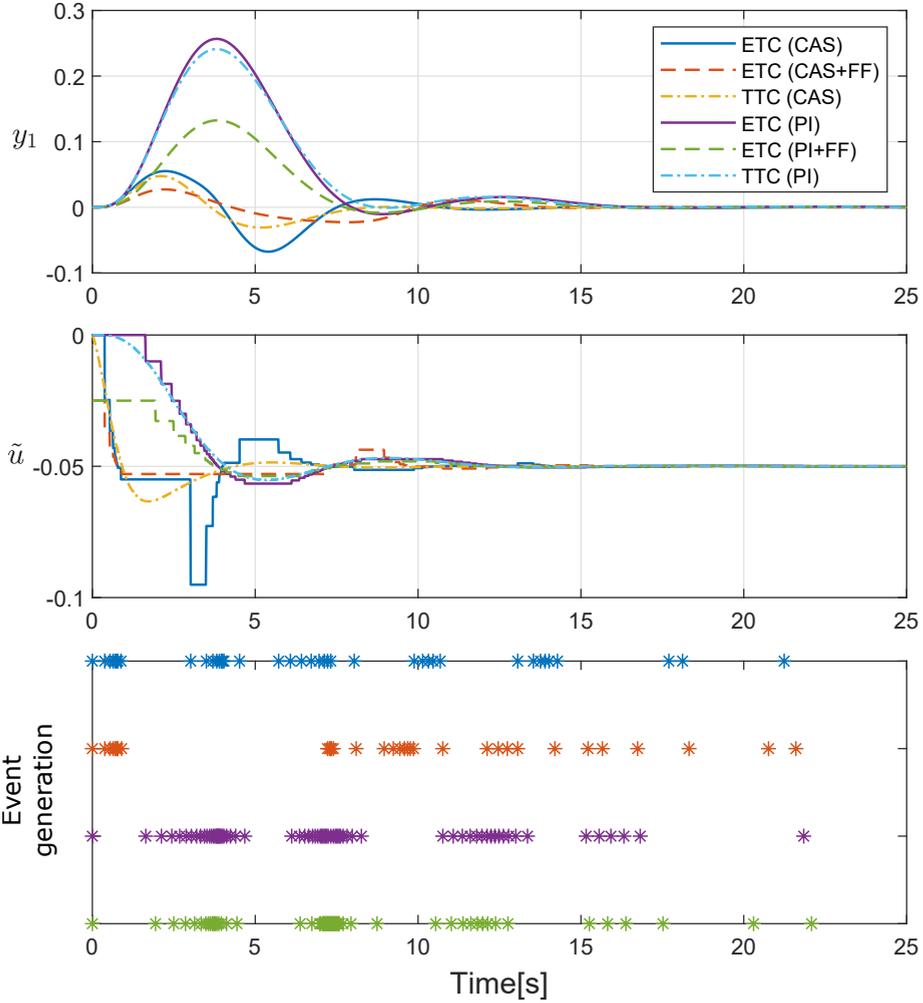


Figure 5.8: Responses of the event-triggered cascade control (ETC (CAS)), time-triggered cascade control (TTC (CAS)), event-triggered PI control (ETC (PI)), and time-triggered PI control (TTC (PI)). The top plot describes the outputs, the middle control signals, and the bottom the event generations of ETCs.

Type	$y_{1,\max}$	IAE	# events until $t = 20$
ETC (CAS)	0.055	0.296	41
ETC (CAS+FF)	0.027	0.176	32
TTC (CAS)	0.048	0.177	1000
ETC (PI)	0.257	1.058	73
ETC (PI+FF)	0.132	0.547	65
TTC (PI)	0.241	1.025	1000

Table 5.5: Numerical results of the event-triggered and time-triggered cascade/PI control.

### 5.3.3 Decoupling control

A decoupling controller can be written by the form of (5.19)–(5.20) with

$$\begin{aligned}
 A_c &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1/T_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/T_2 \end{bmatrix}, \quad B_c^1 = \begin{bmatrix} -1 \\ -c_1/T_1 h_1 \\ 0 \\ 0 \end{bmatrix}, \\
 B_c^2 &= \begin{bmatrix} 0 \\ 0 \\ -1 \\ -c_2/T_2 h_2 \end{bmatrix}, \quad \bar{B}_c^1 = \begin{bmatrix} 0 \\ c_1/T_1 h_1 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{B}_c^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ c_2/T_2 h_2 \end{bmatrix}, \\
 B_r &= \begin{bmatrix} 1 & 0 \\ c_1/T_1 h_0 & 0 \\ 0 & 1 \\ 0 & c_2/T_2 h_0 \end{bmatrix}, \quad \bar{B}_r = \begin{bmatrix} 0 & 0 \\ -c_1/T_1 & 0 \\ 0 & 0 \\ 0 & -c_2/T_2 \end{bmatrix}, \\
 C_c &= \begin{bmatrix} K_i^1 & K_d^1 & K_g^1 K_i^2 & K_g^1 K_d^2 \\ K_g^2 K_i^1 & K_g^2 K_d^1 & K_i^2 & K_d^2 \end{bmatrix}, \quad D_c^1 = \begin{bmatrix} -K_p^1 \\ -K_g^2 K_p^1 \end{bmatrix}, \\
 D_c^2 &= \begin{bmatrix} -K_g^1 K_p^2 \\ -K_p^2 \end{bmatrix}, \quad \bar{D}_c^1 = \bar{D}_c^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad D_r = \begin{bmatrix} b_1 K_p^1 & b_1 K_g^1 K_p^2 \\ b_2 K_g^2 K_p^1 & b_2 K_p^2 \end{bmatrix}, \\
 \bar{D}_r &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_d = \begin{bmatrix} K_f^1 & K_g^1 K_f^2 \\ K_g^2 K_f^1 & K_f^2 \end{bmatrix},
 \end{aligned}$$

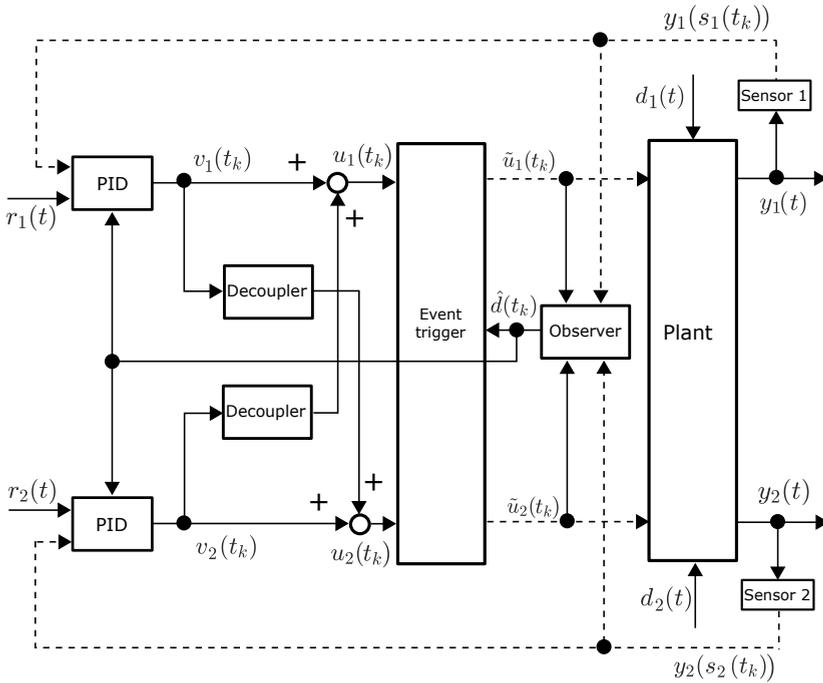


Figure 5.9: Event-triggered decoupling control with an observer.

where  $K_f^1$  and  $K_f^2$  are the feedforward gains of controllers 1 and 2, respectively.

The block diagram of event-triggered decoupling control with an observer is shown in Figure 5.9.

**Example 5.4.** Consider the distillation column [25], which state-space formulation is given by

$$\dot{x}_p(t) = \begin{bmatrix} -0.0599 & 0 & 0 & 0 \\ 0 & -0.0476 & 0 & 0 \\ 0 & 0 & -0.0917 & 0 \\ 0 & 0 & 0 & -0.0694 \end{bmatrix} x_p(t)$$

	Decoup.	PI
$\sigma^*$	0.097	0.14

Table 5.6: Obtained thresholds for the decoupling control and PI control.

Type	$y_{1 \max}$	IAE ( $y_1$ )
ETC (Decoup.)	0.17	5.56
TTC (Decoup.)	0.15	4.96
ETC (PI)	0.49	13.07
TTC (PI)	0.44	12.26

Table 5.7: Numerical results of the event-triggered and time-triggered decoupling/PI control (disturbance rejection).

$$\begin{aligned}
 & + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tilde{u}(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} d(t), \\
 y_1(t) & = [0.767 \quad 0.90 \quad 0 \quad 0] x_p(t), \\
 y_2(t) & = [0 \quad 0 \quad 0.605 \quad 1.347] x_p(t).
 \end{aligned}$$

We introduce decoupling control consisting of two PI controllers  $K_p^1 = 0.0537$ ,  $K_i^1 = 0.0112$ ,  $K_p^2 = 0.0814$ ,  $K_i^2 = 0.0146$ ,  $K_f^1 = K_f^2 = 0$ , and decouplers  $[K_g^1, K_g^2] = [-1.0336, -0.2381]$ . We compare this to non-decoupled PI control, i.e.,  $[K_g^1, K_g^2] = [0, 0]$ . By solving SDPs (5.25) with intervals  $h_0 = h_1 = h_2 = 0.5$ , and  $\alpha = 0.025$ , event thresholds are obtained (Table 5.6).

We consider the step reference signal  $r(t) = [0, 1]^\top, \forall t \geq 0$ . Numerical results with  $\rho = 10^{-6}$  are shown in Figure 5.10, Tables 5.7, and 5.8. It can be found from Figure 5.10 and Table 5.7 that the proposed event-triggered decoupling controller compensates well for the disturbance. The number of events until the settling time of  $y_2(t)$  is extremely reduced compared to the time-triggered decoupling control with slight performance degradation as shown in Table 5.8.

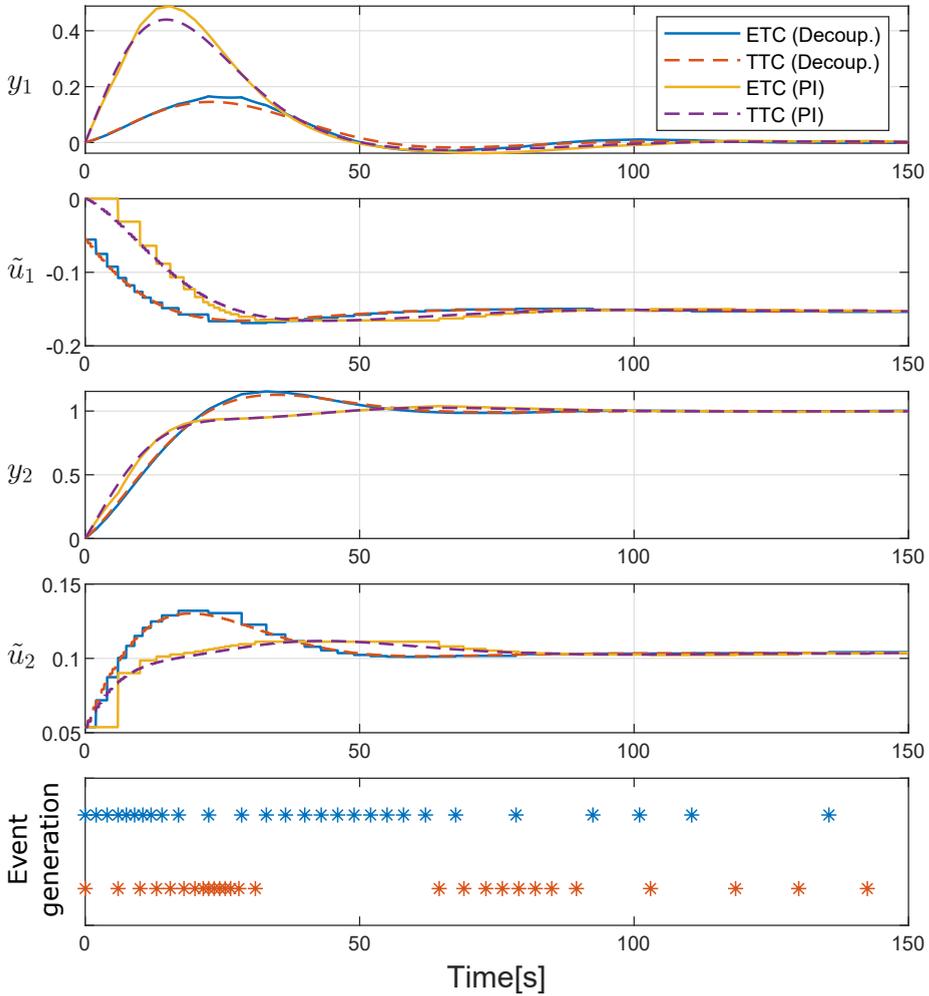


Figure 5.10: Responses of the event-triggered decoupling control (ETC (Decoup.)), time-triggered decoupling control (TTC (Decoup.)), event-triggered PI control (ETC (PI)), and time-triggered PI control (TTC (PI)). From the top, each plot describes the outputs  $y_1(t)$  and control signal  $\tilde{u}_1(t)$  of one control loop, outputs  $y_2(t)$  and control signal  $\tilde{u}_2(t)$  of another, and the event generations of ETCs.

Type	$T_r$	$M$	$T_s$	IAE ( $y_2$ )	# events until $T_s$
ETC (Decoup.)	12.67	15.4	54.66	14.03	19
TTC (Decoup.)	12.46	12.7	57.42	13.55	115
ETC (PI)	9.89	3.9	80.56	11.06	20
TTC (PI)	9.43	2.5	74.38	10.54	149

Table 5.8: Numerical results of the event-triggered and time-triggered decoupling/PI control (setpoint tracking).

## 5.4 Summary

This chapter studied periodic event-triggered PID, cascade, and decoupling control. The controllers updated their commands when the values went beyond given thresholds. We formulated an output feedback control system as a general form of the three systems, and derived an exponential stability condition. Furthermore, it was shown that the proposed controller has a capability of setpoint tracking and disturbance rejection. Event threshold tuning was also proposed. We then applied the framework to PI, PID, cascade, and decoupling control. The numerical examples showed that the proposed controllers reduced the communication load while maintaining the control performance.

### 5.A Proof of Lemma 5.1

Before presenting the proof, we introduce the following lemma.

**Lemma 5.3.** ([126]) *Let  $z : [a, b] \rightarrow \mathbb{R}^n$  be an absolutely continuous function with a square integrable first order derivative such that  $z(a) = 0$  or  $z(b) = 0$ . Then for any  $\alpha > 0$  and  $W \in \mathbb{S}_{++}^n$ , the following inequality holds:*

$$\int_a^b e^{2\alpha t} z^\top(t) W z(t) dt \leq e^{2|\alpha|(b-a)} \frac{4(b-a)^2}{\pi^2} \int_a^b e^{2\alpha t} \dot{z}^\top(t) W \dot{z}(t) dt.$$

Now, we derive the stability condition of the system (5.12) with  $\xi(t) \equiv 0$ .

Consider the functional

$$V = V_0 + V_{W_0} + \sum_{i \in \mathcal{N}} V_{W_i} + \sum_{i \in \mathcal{N}} V_{\bar{W}_i} + \sum_{i \in \mathcal{N}} V_{R_i} \quad (5.27)$$

where

$$\begin{aligned} V_0 &\triangleq x(t)^\top P x(t), \\ V_{W_0} &\triangleq h_0^2 e^{2\alpha h_0} \int_{t_k}^t e^{-2\alpha(t-s)} \dot{x}(s)^\top W_0 \dot{x}(s) ds \\ &\quad - \frac{\pi^2}{4} \int_{t_k}^t e^{-2\alpha(t-s)} \delta_0^\top(s) W_0 \delta_0(s) ds, \\ V_{W_i} &\triangleq (h_0 + h_i)^2 e^{2\alpha(h_0+h_i)} \int_{s_i(t_k)}^t e^{-2\alpha(t-s)} \dot{x}(s)^\top W_i \dot{x}(s) ds \\ &\quad - \frac{\pi^2}{4} \int_{s_i(t_k)}^t e^{-2\alpha(t-s)} \delta_i^\top(s) W_i \delta_i(s) ds, \\ V_{\bar{W}_i} &\triangleq (h_0 + h_i)^2 e^{2\alpha(h_0+h_i)} \int_{s_i(t_k)-h_i}^t e^{-2\alpha(t-s)} \dot{x}(s)^\top \bar{W}_i \dot{x}(s) ds \\ &\quad - \frac{\pi^2}{4} \int_{s_i(t_k)-h_i}^t e^{-2\alpha(t-s)} \bar{\delta}_i^\top(s) \bar{W}_i \bar{\delta}_i(s) ds, \\ V_{R_i} &\triangleq \int_{t-h_i}^t e^{-2\alpha(t-s)} x^\top(s) R_i x(s) ds \\ &\quad + h_i \int_{-h_i}^0 \int_{t+\theta}^t e^{-2\alpha(t-s)} \dot{x}^\top(s) R_i \dot{x}(s) ds d\theta, \end{aligned}$$

with  $\delta_0(t) \triangleq x(t_k) - x(t)$ ,  $\delta_i(t) \triangleq x(s_i(t_k)) - x(t)$  and  $\bar{\delta}_i(t) \triangleq x(s_i(t_k) - h_i) - x(t)$ . Using Lemma 5.3 and  $t - t_k \leq h_0$ , we have  $V_{W_0} \geq 0$ . In addition, we have  $V_{W_i} \geq 0$  and  $V_{\bar{W}_i} \geq 0$  as

$$t - s_i(t_k) = t - t_k + (t_k - s_i(t_k)) \leq h_0 + h_i.$$

We take the derivatives of each term

$$\begin{aligned} \dot{V}_0 + 2\alpha V_0 &= x^\top(t) (P\bar{A} + \bar{A}^\top + 2\alpha P) x(t) \\ &\quad + 2 \sum_{i \in \mathcal{N}} x^\top(t) P A_i x(t - h_i) + 2 \sum_{i=0}^N x^\top(t) P A_i \delta(t) \end{aligned}$$

$$\begin{aligned}
& + 2 \sum_{i \in \mathcal{N}} x^\top(t) P \bar{A}_i \bar{\delta}(t), \\
\dot{V}_{W_0} + 2\alpha V_{W_0} & = h_0^2 e^{2\alpha h_0} \dot{x}^\top(t) W_0 \dot{x}(t) - \frac{\pi^2}{4} \bar{\delta}_0^\top(t) W_0 \bar{\delta}_0(t), \\
\dot{V}_{W_i} + 2\alpha V_{W_i} & = (h_0 + h_i)^2 e^{2\alpha(h_0+h_i)} \dot{x}^\top(t) W_i \dot{x}(t) - \frac{\pi^2}{4} \bar{\delta}_i^\top(t) W_i \bar{\delta}_i(t), \\
\dot{V}_{\bar{W}_i} + 2\alpha V_{\bar{W}_i} & = (h_0 + h_i)^2 e^{2\alpha(h_0+h_i)} \dot{x}^\top(t) \bar{W}_i \dot{x}(t) \\
& \quad - \frac{\pi^2}{4} e^{-2\alpha h_i} \bar{\delta}_i^\top(t) \bar{W}_i \bar{\delta}_i(t),
\end{aligned}$$

and, by Jensen's inequality [223],

$$\begin{aligned}
\dot{V}_{R_i} + 2\alpha V_{R_i} & \leq x^\top(t) (Q_i - e^{-2\alpha h_i} R_i) x(t) + 2e^{-2\alpha h_i} x^\top(t) R_i x(t - h_i) \\
& \quad - e^{-2\alpha h_i} x^\top(t - h_i) (Q_i + R_i) x(t - h_i) \\
& \quad + h_i^2 \dot{x}^\top(t) R_i \dot{x}(t).
\end{aligned}$$

By Schur complements, we have that  $\dot{V} + 2\alpha V < 0$  if  $\Phi < 0$ .

## 5.B Proof of Theorem 5.1

First, note that by the event-triggering condition (5.8), for some  $w \geq 0$ , we have

$$w \sigma u^\top(t) \Omega u(t) - w \xi^\top(t) \Omega \xi(t) \geq 0. \quad (5.28)$$

Introducing the functional (5.27) gives

$$\dot{V} + 2\alpha V \leq \dot{V} + 2\alpha V + w \sigma u^\top(t_k) \Omega u(t_k) - w \xi^\top(t) \Omega \xi(t).$$

Substituting  $u(t_k) = K_0 x(t_k) + K_i x(s_i(t_k)) + \bar{K}_i x(s_i(t_k) - h_i)$  into this, and applying Schur complements, we have  $\dot{V} + 2\alpha V < 0$  if  $\Psi < 0$ .



## Chapter 6

# Event-triggered Actuation for Time-delay Systems

Process plants usually have time delays due to physical movements of material or energy. Time delays may deteriorate control performance or even destabilize systems. The Smith predictor is widely used in process control applications to compensate for large time delays by predicting the plant output using a simple plant model. This chapter investigates event-triggered PI control for time-delay systems as a way to reduce communication load. The plant is given by continuous-time linear systems with parametric uncertainties. We consider sampled-data PI control with the Smith predictor.

The main contributions of this chapter are as follows:

- We derive an exponential stability condition of time-triggered PI control systems with parametric uncertainties (Theorem 6.1).
- We introduce periodic event-triggered PI control for time-delay systems. An exponential stability condition is derived (Theorem 6.2).
- We propose an event threshold tuning procedure under a given stability margin (Corollary 6.1). The optimal threshold is obtained by solving a semi-definite programming (SDP) problem.
- We provide numerical examples to illustrate that our proposed controller reduces the communication load without performance degradation compared to conventional PI control.

The remainder of the chapter is organized as follows. Section 6.1 describes the PI control system with the Smith predictor. An exponential stability condition is derived. In Section 6.2, we propose event-triggered PI control and derive an exponential stability condition. We provide numerical examples in Section 6.3. The conclusion is presented in Section 6.4.

## 6.1 Time-triggered PI control for time-delay systems

In this chapter, we consider a continuous-time linear plant with a process time-delay. The plant is controlled by a PI controller with the Smith predictor (Figure 6.1). In this section, we introduce the plant with uncertain parameters, the predictor, and the PI controller. An exponential stability condition for the closed-loop system is derived.

### 6.1.1 System model

Consider a plant with a constant process time delay given by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u(t - \eta), \quad (6.1)$$

$$y(t) = C_p x_p(t), \quad (6.2)$$

where  $x_p(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}$ ,  $y(t) \in \mathbb{R}$ ,  $\eta > 0$ , are the state, control signal, output, and a constant process time delay, respectively. We assume that the sensor samples and transmits its measurement every  $h$  time interval. Let  $t_k, k \in \mathbb{N}_0$ , be the time of transmission  $k$  of the sensor, i.e.,  $t_{k+1} - t_k = h$  for all  $t \geq 0$ . A sampled-data implementation of a predictor, which updates its state every  $h$  time interval, is given by

$$\dot{\hat{x}}_p(t) = \hat{A}_p \hat{x}_p(t_k) + \hat{B}_p u(t), \quad t \in [t_k, t_{k+1}), \quad (6.3)$$

$$\hat{y}(t) = C_p \hat{x}_p(t_k), \quad (6.4)$$

where  $\hat{x}_p(t) \in \mathbb{R}^n$  and  $\hat{y}(t) \in \mathbb{R}$  are the predictions of the plant state and the output. A PI controller is given by

$$\dot{x}_c(t) = r - e(t_k) - \hat{y}(t_k), \quad t \in [t_k, t_{k+1}), \quad (6.5)$$

$$u(t) = K_i x_c(t_k) + K_p (r - e(t_k) - \hat{y}(t_k)), \quad (6.6)$$

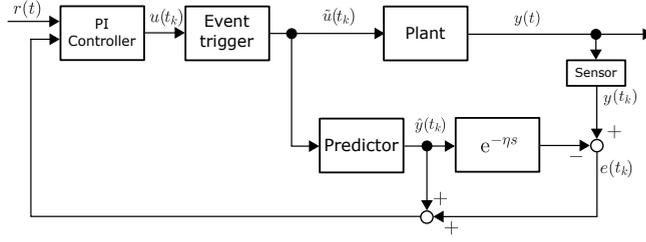


Figure 6.1: Event-triggered PI control with the Smith predictor. The event trigger is introduced in Section 6.2.

where  $x_c(t) \in \mathbb{R}$  is the controller state,  $r \in \mathbb{R}$  the constant reference signal,  $e(t) \triangleq y(t) - \hat{y}(t - \eta)$  the prediction error.

We make the following assumptions on the uncertainty of the plant.

**Assumption 6.1.** *The system matrix  $A_p$  and the vector  $B_p$  reside in the uncertain polytopes*

$$A_p = \sum_{i \in \mathcal{N}} \lambda_i A_p^{(i)}, \quad B_p = \sum_{i \in \mathcal{N}} \mu_i B_p^{(i)},$$

where  $A^{(i)}$  and  $B^{(i)}$ ,  $i \in \mathcal{N} \triangleq \{1, \dots, N\}$  are the vertex matrices and vectors, respectively, and  $\lambda_i, \mu_i \in [0, 1]$ , are constants with  $\sum_{i \in \mathcal{N}} \lambda_i = 1$  and  $\sum_{i \in \mathcal{N}} \mu_i = 1$ .

**Assumption 6.2.** *The system (6.1)–(6.2) with the uncertain polytopes is  $(A_p^{(i)}, B_p^{(i)})$  controllable and  $(C_p, A_p^{(i)})$  observable for all  $i \in \mathcal{N}$ .*

By augmenting the state  $x(t) \triangleq [x_p^\top(t + \eta), \hat{x}_p^\top(t), x_c^\top(t)]^\top \in \mathbb{R}^{2n+1}$ , we have the following closed-loop system description

$$\dot{x}(t) = Ax(t) + A_1x(t_k) + A_2x(t_k - \eta) + B_Rr, \quad t \in [t_k, t_{k+1}), \quad (6.7)$$

with

$$A = \begin{bmatrix} A_p & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & -B_p K_p C_p & B_p K_p \\ 0 & \hat{A}_p - \hat{B}_p K_p C_p & \hat{B}_p K_p \\ 0 & -C_p & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -B_p K_p C_p & B_p K_p C_p & 0 \\ -\hat{B}_p K_p C_p & \hat{B}_p K_p C_p & 0 \\ -C_p & C_p & 0 \end{bmatrix}, \quad B_R = \begin{bmatrix} B_p K_p \\ \hat{B}_p K_p \\ 1 \end{bmatrix}.$$

**Remark 6.1.** Suppose that Assumption 6.1 holds. Then the matrices  $A, A_1, A_2,$  and  $B_R$  reside in the uncertain polytope

$$\begin{aligned} A &= \sum_{i \in \mathcal{N}} \lambda_i A^{(i)}, \quad 0 \leq \lambda_i \leq 1, \quad \sum_{i \in \mathcal{N}} \lambda_i = 1, \\ A_1 &= \sum_{i \in \mathcal{N}} \mu_i A_1^{(i)}, \quad 0 \leq \mu_i \leq 1, \quad \sum_{i \in \mathcal{N}} \mu_i = 1, \\ A_2 &= \sum_{i \in \mathcal{N}} \mu_i A_2^{(i)}, \quad 0 \leq \mu_i \leq 1, \quad \sum_{i \in \mathcal{N}} \mu_i = 1, \\ B_R &= \sum_{i \in \mathcal{N}} \mu_i B_R^{(i)}, \quad 0 \leq \mu_i \leq 1, \quad \sum_{i \in \mathcal{N}} \mu_i = 1, \end{aligned}$$

where

$$\begin{aligned} A^{(i)} &= \begin{bmatrix} A_p^{(i)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_1^{(i)} = \begin{bmatrix} 0 & -B_p^{(i)} K_p C_p & B_p^{(i)} K_p \\ 0 & \hat{A}_p - \hat{B}_p K_p C_p & \hat{B}_p K_p \\ 0 & -C_p & 0 \end{bmatrix}, \\ A_2^{(i)} &= \begin{bmatrix} -B_p^{(i)} K_p C_p & B_p^{(i)} K_p C_p & 0 \\ -\hat{B}_p K_p C_p & \hat{B}_p K_p C_p & 0 \\ -C_p & C_p & 0 \end{bmatrix}, \quad B_R^{(i)} = \begin{bmatrix} B_p^{(i)} K_p \\ \hat{B}_p K_p \\ 1 \end{bmatrix}. \end{aligned}$$

### 6.1.2 Stability of time-triggered PI control

We derive the stability condition of the closed-loop system (6.7).

**Theorem 6.1.** Consider the closed-loop system (6.7). Suppose that Assumption 6.1 holds. Given  $K_p, K_p \in \mathbb{R}$ , and  $\alpha > 0$ , assume that there exist  $P, R_1, R_2, W_1, W_2 \in \mathbb{S}_{++}^{2n+1}$ , such that

$$\Phi^{(i)} = \begin{bmatrix} \Phi_{11}^{(i)} & \Phi_{12}^{(i)} & P A_1^{(i)} & P A_2^{(i)} & (A^{(i)} + A_1^{(i)})^\top Q \\ * & \Phi_{22} & 0 & 0 & A_2^{(i)\top} Q \\ * & * & -\frac{\pi^2}{4} W_1 & 0 & A_1^{(i)\top} Q \\ * & * & * & -\frac{\pi^2}{4} e^{-2\alpha\eta} W_2 & A_2^{(i)\top} Q \\ * & * & * & * & -Q \end{bmatrix} < 0, \quad (6.8)$$

for all  $i \in \mathcal{N}$ , where

$$\Phi_{11}^{(i)} \triangleq P(A^{(i)} + A_1^{(i)}) + (A^{(i)} + A_1^{(i)})^\top P + 2\alpha P + R_1 - e^{-2\alpha\eta} R_2,$$

$$\begin{aligned}\Phi_{12}^{(i)} &\triangleq PA_2^{(i)} + e^{-2\alpha\eta}R_2, \\ \Phi_{22} &\triangleq -e^{-2\alpha\eta}(R_1 + R_2), \\ Q &\triangleq \eta^2R_2 + h^2e^{2\alpha h}(W_1 + e^{-2\alpha\eta}W_2).\end{aligned}$$

Then the closed-loop system (6.7) with  $r = 0$  is exponentially stable with decay rate  $\alpha$ .

*Proof.* See Appendix 6.A. □

## 6.2 Event-triggered control of time-delay systems

In this section, we introduce event-triggered control. We derive an exponential stability condition and propose a way to tune the event threshold with given control parameters.

### 6.2.1 System model of event-triggered PI control

Consider the system

$$\dot{x}_p(t) = A_px_p(t) + B_p\tilde{u}(t - \eta)$$

where  $\tilde{u}(t)$  is the event-triggered control signal. We assume that  $\tilde{u}(t)$  is updated by checking the event-triggering condition

$$(u(t_k) - \tilde{u}(t_{k-1}))^2 > \sigma u^2(t_k) + \rho \quad (6.9)$$

at every sampling time  $t_k$ ,  $k \in \mathbb{N}_0$ , where  $\sigma \geq 0$  is a relative threshold and  $\rho$  a constant. The event-triggered control signal is given by

$$\tilde{u}(t) = \begin{cases} u(t_k), & t \in [t_k, t_{k+1}), \quad \text{if (6.9) is true,} \\ \tilde{u}(t_{k-1}), & t \in [t_k, t_{k+1}), \quad \text{if (6.9) is false,} \end{cases}$$

with  $\tilde{u}_0 = u(t_0)$ . Define the control signal error as

$$\begin{aligned}\xi(t) &\triangleq \tilde{u}(t) - u(t) \\ &= \tilde{u}(t_k) - u(t_k), \quad t \in [t_k, t_{k+1}).\end{aligned}$$

Then the closed-loop system is given by

$$\dot{x}(t) = Ax(t) + A_1x(t_k) + A_2x(t_k - \eta) + B\xi(t) + B_Rr \quad (6.10)$$

where

$$B = \begin{bmatrix} B_p \\ \hat{B}_p \\ 0 \end{bmatrix}.$$

**Remark 6.2.** Suppose that Assumption 6.1 holds. Then  $B$  resides in the uncertain polytope

$$B = \sum_{i=1}^N \mu_i B^{(i)}, \quad 0 \leq \mu_i \leq 1, \quad \sum_{i=1}^N \mu_i = 1,$$

where

$$B^{(i)} = \begin{bmatrix} B_p^{(i)} \\ \hat{B}_p \\ 0 \end{bmatrix}.$$

## 6.2.2 Stability of event-triggered PI control

We have the following stability condition.

**Theorem 6.2.** Consider the closed-loop system (6.10) with  $\rho = 0$ . Suppose that Assumption 6.1 holds. Given  $K_p, K_p \in \mathbb{R}$ , and  $\alpha > 0$ , assume that there exist  $P, R_1, R_2, W_1, W_2 \in \mathbb{S}_{++}^{2n+1}$ ,  $w > 0$ , and  $\sigma > 0$ , such that

$$\Psi^{(i)} = \left[ \begin{array}{ccccc|cc} & & & & & PB^{(i)} & w\sigma K_1^\top \\ & & & & & 0 & w\sigma K_2^\top \\ & & & & & 0 & w\sigma K_1^\top \\ & & & & & 0 & w\sigma K_2^\top \\ & & & & & QB^{(i)} & 0 \\ \hline * & * & * & * & * & -w & 0 \\ * & * & * & * & * & 0 & -w\sigma \end{array} \right] < 0, \quad (6.11)$$

for all  $i \in \mathcal{N}$ , where

$$K_1 = [0 \quad -K_p C_p \quad K_p], \quad K_2 = [-K_p C_p \quad K_p C_p \quad 0].$$

Then the closed-loop system (6.10) with the event-triggering condition (6.9) with  $r = 0$  is exponentially stable with decay rate  $\alpha$ .

*Proof.* See Appendix 6.B. □

**Remark 6.3.** For  $r \neq 0$ , we need to apply a coordinate transformation  $\tilde{x}(t) = x(t) - x_e$  where  $x_e^* = -(A + A_1 + A_2)^{-1} B_R r$  is the equilibrium point. Note that  $A + A_1 + A_2$  is invertible when a continuous controller (i.e.,  $h = 0$ ) stabilizes the system and therefore  $A + A_1 + A_2$  is Hurwitz (Lemma 5.2). The event-triggering condition (6.9) is replaced by

$$(u(t_k) - \tilde{u}(t_{k-1}))^2 > \sigma(u(t_k) - u_e)^2 + \rho$$

where  $u_e = (K_1 + K_2)x_e$  is the steady-state control signal. Theorem 6.2 can be applied if we know the exact model  $\hat{A}_p = A_p$  and  $\hat{B}_p = B_p$ . Otherwise, we use the prediction of  $A, A_1, A_2, B_R$  denoted as  $\hat{A}, \hat{A}_1, \hat{A}_2, \hat{B}_R$ . The prediction matrices are given by replacing  $A_p, B_p$  in  $A, A_1, A_2$ , and  $B_R$  by  $\hat{A}_p, \hat{B}_p$ . The event-triggering condition (6.9) is replaced by

$$(u(t_k) - \tilde{u}(t_{k-1}))^2 > \sigma(u(t_k) - \hat{u}_e)^2 + \rho$$

where  $\hat{u}_e = -(K_1 + K_2)(\hat{A} + \hat{A}_1 + \hat{A}_2)^{-1} \hat{B}_R r$  is the prediction of steady-state control signal. In this case, the prediction error  $e_u \triangleq \hat{u}_e - u_e$  leads to the steady-state tracking error. The observer introduced in Chapter 5 can be used to avoid the error.

Using (6.11), we can tune the event threshold  $\sigma$  to give a minimum communication load satisfying a given stability margin  $\alpha$ .

**Corollary 6.1.** Suppose that Assumption 6.1 holds. Given  $K_p, K_p \in \mathbb{R}, w > 0$ , and  $\alpha > 0$ , if the SDP:

$$\sigma^* \triangleq \max \sigma \quad \text{s.t.} \quad \Psi^{(i)} < 0, \quad i \in \mathcal{N}, \quad (6.12)$$

is feasible, then the closed-loop system (6.10) under the event-triggering condition (6.9) with  $\sigma = \sigma^*$  is exponentially stable with decay rate  $\alpha$ .

## 6.3 Numerical examples

This section provides numerical examples to illustrate our theoretical results. Consider a first-order linear system

$$\dot{x}_p(t) = ax_p(t) + b\tilde{u}(t - \eta), \quad (6.13)$$

$$y(t) = x_p(t), \quad (6.14)$$

	Comm. until $t = 50$	Comm. Reduction	IAE
ETS ( $\eta = 1$ )	285	43.0%	3.39
TTS ( $\eta = 1$ )	500	0%	3.52
TTPI ( $\eta = 1$ )	500	0%	4.52
ETS ( $\eta = 3$ )	316	36.8%	5.37
TTS ( $\eta = 3$ )	500	0%	5.43
TTPI ( $\eta = 3$ )	500	0%	$+\infty$
Open-loop	-	-	18.94

Table 6.1: Number of communications, their reductions, and the IAE for the strategies: ETS, TTS, and TTPI, with  $\eta = 1$  and  $\eta = 3$ .

where the system parameters take their values in  $a \in [-0.055, -0.045]$  and  $b \in [0.45, 0.55]$ . We use a plant model

$$\dot{\hat{x}}_p(t) = -0.05\hat{x}_p(t) + 0.5\tilde{u}(t - \eta).$$

The linear matrix inequality (6.8) guarantees the exponential stability of the time-triggered PI control system (6.13)–(6.14) with the control parameters  $K_p = 0.816$ ,  $K_i = 0.293$ , sampling interval  $h = 0.2$ , decay rate  $\alpha = 0.04$  for time delays  $\eta \leq 3.4$ .

### Initial response

We first see the responses of two different time delays  $\eta = 1$  and  $\eta = 3$  with the reference  $r = 0$  and initial state  $x(t) = [1, 0, 0]$ ,  $t \in [-3, 0]$ . By solving SDPs (6.12), we obtain the event thresholds  $\sigma^* = 0.245$  and  $\sigma^* = 0.014$  for  $\eta = 1$  and  $\eta = 3$ , respectively. The SDP can be solved effectively by YALMIP toolbox [222]. To evaluate the system performance, we use the Integral of the Absolute Error (IAE)

$$\text{IAE} = \int_0^{+\infty} |r - y(t)| dt.$$

The results for three strategies: the event-triggered PI control with the Smith predictor (ETS), time-triggered PI control with the Smith predictor (TTS), and time-triggered PI control (TTPI), together with the open-loop

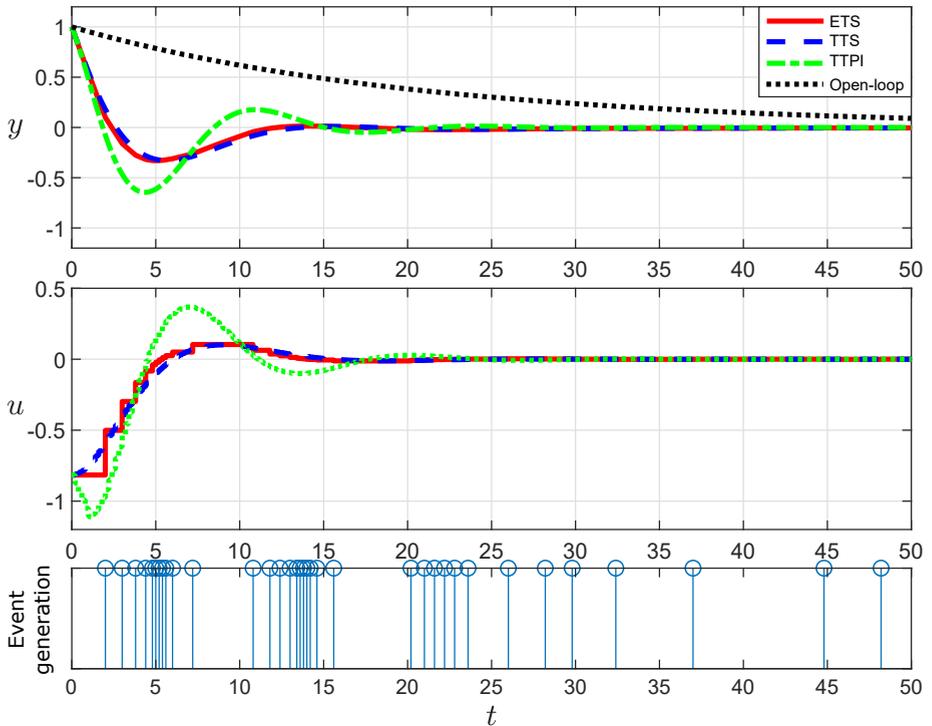


Figure 6.2: Responses to the initial state  $x(t) = [1, 0, 0]$ ,  $t \in [-1, 0]$  (top:  $y(t)$ , middle:  $u(t)$ ) of the four strategies with time delay  $\eta = 1$ : the event-triggered PI control with the Smith predictor (ETS: red solid line), time-triggered PI control with the Smith predictor (TTS: blue dashed line), time-triggered PI control without the Smith predictor (TTPI: green dash-dot line), and open-loop system (black dot line). The bottom plot shows the event generations of the ETS at the controller.

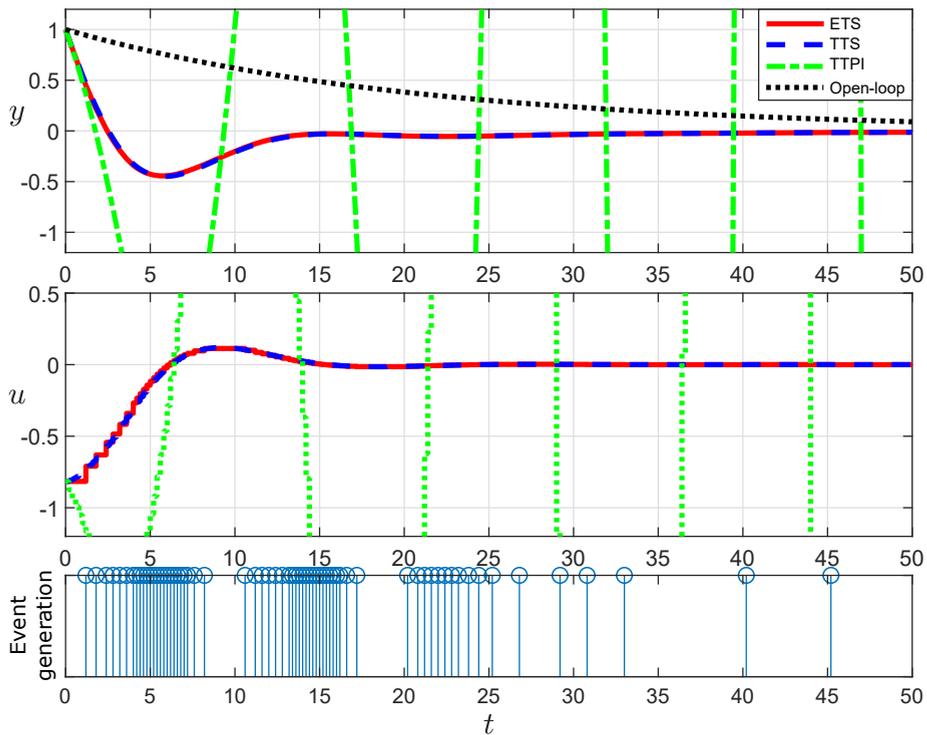


Figure 6.3: Responses to the initial state  $x(t) = [1, 0, 0]$ ,  $t \in [-3, 0]$  (top:  $y(t)$ , middle:  $u(t)$ ) of the four strategies with time delay  $\eta = 3$ . The bottom plot shows the event generations of the ETS at the controller.

system are summarized in Table 6.1. The responses for  $\eta = 1$  and  $\eta = 3$  with  $\rho = 0$  are shown in Figure 6.2 and Figure 6.3, respectively, where we assume that the unknown system parameters are given by  $a = -0.048$  and  $b = 0.52$ . It can be found that the ETS and TTS successfully compensate for the time-delays and the outputs converge to the origin. However, the TTPI is more oscillative in Figure 6.2 ( $\eta = 1$ ) and does not stabilize the system in Figure 6.3 ( $\eta = 3$ ). In fact, the IAEs for the ETS and the TTS are close as in Table 6.1, while those for the TTPI are larger or diverges. The third plots in Figure 6.2 and Figure 6.3 show the time instances of the control command updates. We can see, as well as Table 6.1, that the communications between the controller and actuator are performed only 35 times and 66 times until  $t = 50$ . Including the communications between the sensor and controller, the ETS reduces the communications by 43.0% and 36.8% compared to the TTS.

### Setpoint tracking

Next, we show the responses with  $r = 1$ . The results are shown in Figure 6.4. In setpoint tracking, we need to apply a coordinate transformation  $\tilde{x}(t) = x(t) - \hat{x}_e$  where  $\hat{x}_e = -(\hat{A} + \hat{A}_1 + \hat{A}_2)^{-1} \hat{B}_R r$ . In Figure 6.4, the ETS has a similar response as the TTS with only 30 samplings until  $t = 50$ , even though there remains a small oscillation due to inexact  $\hat{u}_e$ .

## 6.4 Summary

In this chapter, we studied periodic event-triggered actuation applied to time-delay systems. We considered PI controllers, which updated its control signals when the values went beyond a given threshold. A state-space formulation of the Smith predictor was introduced to compensate for time delays. We derived the exponential stability condition under the assumption that the system parameters resided in uncertain polytopes. Based on this result, we proposed the event threshold tuning. Numerical examples showed that our proposed controller reduces the communication load with slight performance degradation.

### 6.A Proof of Theorem 6.1

Consider the functional

$$V = V_0 + V_R + V_{W_1} + V_{W_2} \quad (6.15)$$

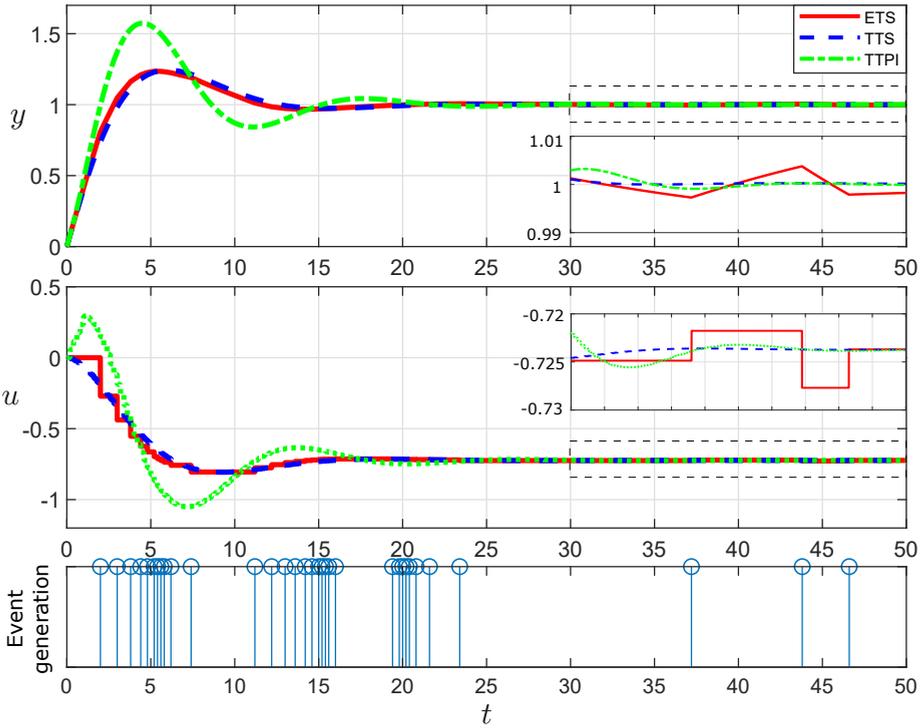


Figure 6.4: Responses to the setpoint tracking  $r(t) = 1, \forall t \geq 0$  of the three strategies with time delay  $\eta = 1$ . The bottom plot shows the event generations of the ETS at the controller.

where

$$\begin{aligned}
V_0 &\triangleq x^\top(t)Px(t), \\
V_R &\triangleq \int_{t-\eta}^t e^{2\alpha(s-t)}x^\top(s)R_1x(s)ds \\
&\quad + \eta \int_{-\eta}^0 \int_{t+\theta}^t e^{2\alpha(s-t)}\dot{x}^\top(s)R_2\dot{x}(s)dsd\theta, \\
V_{W_1} &\triangleq h^2e^{2\alpha h} \int_{t_k}^t \dot{x}^\top(s)W_1\dot{x}(s)ds \\
&\quad - \frac{\pi^2}{4} \int_{t_k}^t e^{2\alpha(s-t)}[x(s) - x(t_k)]^\top W_1[x(s) - x(t_k)]ds, \\
V_{W_2} &\triangleq h^2e^{2\alpha h} \int_{t_k-\eta}^t \dot{x}^\top(s)W_2\dot{x}(s)ds \\
&\quad - \frac{\pi^2}{4} \int_{t_k-\eta}^{t-\eta} e^{2\alpha(s-t)}[x(s) - x(t_k - \eta)]^\top W_2[x(s) - x(t_k - \eta)]ds.
\end{aligned}$$

Using Lemma 5.3 and  $t - t_k \leq h$ , we have  $V_{W_1} \geq 0$  and  $V_{W_2} \geq 0$ . We take the derivatives of each term

$$\begin{aligned}
&\dot{V}_0 + 2\alpha V_0 \\
&= x^\top(t)P\dot{x}(t) + \dot{x}^\top(t)Px(t) + 2\alpha x^\top(t)Px(t) \\
&= x^\top(t)(P(A + A_1) + P(A + A_1)^\top + 2\alpha P)x(t) + x^\top(t)PA_2x(t - \eta) \\
&\quad + x^\top(t)PA_1\delta_1(t) + x^\top(t)PA_2\delta_2(t) + x^\top(t - \eta)A_2^\top Px(t) \\
&\quad + \delta_1^\top(t)A_1^\top Px(t) + \delta_2^\top(t)A_2^\top Px(t),
\end{aligned}$$

and

$$\begin{aligned}
\dot{V}_{W_1} + 2\alpha V_{W_1} &= h^2e^{2\alpha h}\dot{x}^\top(t)W_1\dot{x}(t) - \frac{\pi^2}{4}\delta_1^\top(t)W_1\delta_1(t), \\
\dot{V}_{W_2} + 2\alpha V_{W_2} &= h^2e^{2\alpha h}\dot{x}^\top(t)W_2\dot{x}(t) - \frac{\pi^2}{4}e^{-2\alpha\eta}\delta_2^\top(t)W_2\delta_2(t),
\end{aligned}$$

where  $\delta_1(t) \triangleq x(t_k) - x(t)$  and  $\delta_2(t) \triangleq x(t_k - \eta) - x(t - \eta)$ . For  $V_R$ , by Jensen's inequality [72], we have

$$\dot{V}_R + 2\alpha V_R$$

$$\begin{aligned}
&= x^\top(t)R_1x(t) - e^{-2\alpha\eta}x^\top(t-\eta)R_1x(t-\eta) + \eta^2\dot{x}^\top(t)R_2\dot{x}(t) \\
&\quad - \eta \int_{t-\eta}^t e^{-2\alpha(t-s)}\dot{x}^\top(s)R_2\dot{x}(s)ds \\
&\leq x^\top(t)R_1x(t) - e^{-2\alpha\eta}x^\top(t-\eta)R_1x(t-\eta) + \eta^2\dot{x}^\top(t)R_2\dot{x}(t) \\
&\quad - e^{-2\alpha\eta} \int_{t-\eta}^t \dot{x}^\top(s)dsR_2 \int_{t-\eta}^t \dot{x}(s)ds \\
&= x^\top(t)(R_1 - e^{-2\alpha\eta}R_2)x(t) + e^{-2\alpha\eta}x^\top(t)R_2x(t-\eta) \\
&\quad + e^{-2\alpha\eta}x^\top(t-\eta)R_2x(t) + \eta^2\dot{x}^\top(t)R_2\dot{x}(t) \\
&\quad - e^{-2\alpha\eta}x^\top(t-\eta)(R_1 + R_2)x(t-\eta).
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\dot{V} + 2\alpha V \leq \phi^\top &\left( \begin{array}{cccc} \bar{\Phi}_{11} & \bar{\Phi}_{12} & \bar{\Phi}_{13} & \bar{\Phi}_{14} \\ * & \bar{\Phi}_{22} & \bar{\Phi}_{23} & \bar{\Phi}_{24} \\ * & * & \bar{\Phi}_{33} & \bar{\Phi}_{34} \\ * & * & * & \bar{\Phi}_{44} \end{array} \right. \\
&\quad \left. + \begin{array}{c} \bar{\Phi}_{15} \\ \bar{\Phi}_{25} \\ \bar{\Phi}_{35} \\ \bar{\Phi}_{45} \end{array} \bar{\Phi}_{55}^{-1} \begin{array}{cccc} \bar{\Phi}_{51} & \bar{\Phi}_{52} & \bar{\Phi}_{53} & \bar{\Phi}_{54} \end{array} \right) \phi < 0
\end{aligned}$$

where  $\phi \triangleq [x^\top(t), x^\top(t-\eta), \delta_1^\top(t), \delta_2^\top(t)]^\top$  and

$$\begin{aligned}
\bar{\Phi}_{11} &= P(A + A_1) + (A + A_1)^\top P + 2\alpha P + R_1 - e^{-2\alpha\eta}R_2, \\
\bar{\Phi}_{12} &= PA_2 + e^{-2\alpha\eta}R_2, \quad \bar{\Phi}_{13} = PA_1, \quad \bar{\Phi}_{14} = PA_2, \\
\bar{\Phi}_{15} &= (A + A_1)^\top Q, \quad \bar{\Phi}_{22} = -e^{-2\alpha\eta}(R_1 + R_2), \\
\bar{\Phi}_{23} &= 0, \quad \bar{\Phi}_{24} = 0, \quad \bar{\Phi}_{25} = A_2^\top Q, \\
\bar{\Phi}_{33} &= -\frac{\pi^2}{4}W_1, \quad \bar{\Phi}_{34} = 0, \quad \bar{\Phi}_{35} = A_1^\top Q, \\
\bar{\Phi}_{44} &= -\frac{\pi^2}{4}e^{-2\alpha\eta}W_2, \quad \bar{\Phi}_{45} = A_2^\top Q, \quad \bar{\Phi}_{55} = -Q.
\end{aligned}$$

The proof completes by Schur complements and since  $\bar{\Phi} = \{\bar{\Phi}_{\ell m}\}$ ,  $\ell, m = 1, \dots, 5$ , is affine in  $A$ ,  $A_1$ , and  $A_2$ .

## 6.B Proof of Theorem 6.2

First, note that by the event-triggering condition (6.9), for some  $w \geq 0$ , we have  $w\sigma u^2(t_k) - w\xi^2(t) \geq 0$ . Introducing the functional (6.15) gives

$$\begin{aligned} & \dot{V} + 2\alpha V \\ & \leq \phi^\top \bar{\Phi}_{1:4} \phi + x^\top(t) PB \xi(t) + \xi^\top(t) B^\top P x(t) + \dot{x}^\top(t) Q \dot{x}(t) \\ & \quad + w\sigma u^2(t_k) - w\xi^2(t) \\ & = \psi^\top \left[ \begin{array}{ccccc|c} & & & & & PB \\ & & & & & 0 \\ & & \bar{\Phi}_{1:4} & & & 0 \\ & & & & & 0 \\ & & & & & 0 \\ \hline \bar{B}^\top \bar{P} & 0 & 0 & 0 & 0 & -w \end{array} \right] \psi + \dot{x}^\top(t) Q \dot{x}(t) + w\sigma u^2(t_k), \end{aligned}$$

where  $\psi = [x^\top(t), x^\top(t - \eta), \delta_1^\top(t), \delta_2^\top(t), \xi^\top(t)]^\top$  and  $\bar{\Phi}_{1:4}$  is the submatrix obtained by omitting the 5-th row and column vectors from  $\bar{\Phi}$ . Substituting  $u(t_k) = K_1 x(t_k) + K_2 x(t_k - \eta)$  and applying Schur complements, we have that  $\dot{V} + 2\alpha V < 0$  if

$$\bar{\Psi} \triangleq \left[ \begin{array}{ccccc|cc} & & & & & PB & w\sigma K_1^\top \\ & & & & & 0 & w\sigma K_2^\top \\ & & & & & 0 & w\sigma K_1^\top \\ & & \bar{\Phi} & & & 0 & w\sigma K_2^\top \\ & & & & & QB & 0 \\ \hline * & * & * & * & * & -w & 0 \\ * & * & * & * & * & 0 & -w\sigma \end{array} \right] < 0.$$

The proof completes since  $\bar{\Psi}$  is affine in  $A$ ,  $A_1$ ,  $A_2$ , and  $B$ .



## Chapter 7

# Event-triggered Controller Switching

Deploying a new sensor can improve closed-loop control performance of process control systems. Wireless sensors can be deployed easily since they do not require cabling. In the previous two chapters, we considered fully wireless control and developed event-triggered actuation for multi-loop process control and time-delay systems. In contrast, this chapter investigates partially wireless control. We propose a way to introduce new wireless sensors to an operating hard-wired control system to improve its control performance. In particular, we consider the scenario that i) a hard-wired feedback control is regulating a plant with a potential disturbance, and ii) we introduce feedforward or cascade control by deploying a wireless sensor to mitigate the effect of the disturbance.

The usage of wireless sensors should be minimized since they usually have no reliable energy sources. The idea to tackle this problem is to activate wireless sensors only when a disturbance appears. Sensor activation switches a controller to a feedforward or cascade control mode from a PI control mode. In this chapter, we propose a controller switching framework utilizing event-triggered sampling.

The main contributions of this chapter is as follows:

- We introduce output feedback control that is a general form of feedforward and cascade control. Stability conditions are derived using the Lyapunov–Krasovskii functional (Lemma 7.1).

- We propose a controller switching framework as a switched output feedback control system. The stability conditions for the proposed control system are derived (Theorem 7.1).
- Applications to cascade and feedforward control are studied. It is shown in numerical examples that our framework reduces communication between sensors and a controller while providing the disturbance rejection capability.

The remainder of the chapter is organized as follows. Section 7.1 describes the system considered. We introduce output feedback control and derive stability conditions. In Section 7.2, we propose a controller switching framework. Section 7.3 discusses the applications of this framework to cascade and feedforward control. We provide numerical examples in Section 7.4. The conclusion is presented in Section 7.5.

## 7.1 Output feedback control with multiple sensors

In this section, we introduce a continuous-time linear system monitored by multiple sensors and controlled by an output feedback controller. Stability conditions are derived under bounded sampling intervals of each sensor.

### 7.1.1 System model

Consider a plant given by

$$\dot{x}_p(t) = A_p x_p(t) + B_p u(t) + B_d d(t), \quad (7.1)$$

$$y_i(t) = C_{p,i} x_p(t), \quad i \in \mathcal{N} \triangleq \{1, \dots, N\}, \quad (7.2)$$

where  $x_p(t) \in \mathbb{R}^{n_p}$ ,  $u(t) \in \mathbb{R}^m$ ,  $d(t) \in \mathbb{R}^{n_d}$  and  $y_i(t) \in \mathbb{R}^{q_i}$  are the state, control signal, disturbance, and measurement by sensor  $i \in \mathcal{N}$ , respectively. The matrices  $A_p$ ,  $B_p$ ,  $B_d$ , and  $C_{p,i}$ ,  $i \in \mathcal{N}$  are real matrices of appropriate dimensions. We consider an output feedback controller

$$\dot{x}_c(t) = A_c x_c(t) + \sum_{\mathcal{N}} B_{c,i} y_i(s_i(t)) + B_r r(t), \quad (7.3a)$$

$$u(t) = C_c x_c(t) + \sum_{\mathcal{N}} D_{c,i} y_i(s_i(t)) + D_r r(t), \quad (7.3b)$$

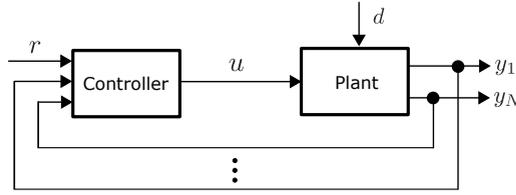


Figure 7.1: Output feedback control system with  $N$  sampled-data measurements.

that employs sampled-data measurements, where  $x_c(t) \in \mathbb{R}^{n_c}$  is the controller state,  $r(t) \in \mathbb{R}^{n_r}$  the reference signal, and  $s_i(t) \in \mathbb{R}$  the latest time instance at time  $t$  when sensor  $i$  transmitted its measurement. The matrices  $A_c$ ,  $B_{c,i}$ ,  $B_r$ ,  $C_c$ ,  $D_{c,i}$ ,  $D_r$ ,  $i \in \mathcal{N}$ , are real matrices of appropriate dimensions. The block diagram of the system considered is depicted in Figure 7.1.

By augmenting the state  $x(t) = [x_p^\top(t), x_c^\top(t)]^\top \in \mathbb{R}^{n_p+n_c}$ , we have the following time-delay closed-loop system description

$$\dot{x}(t) = Ax(t) + \sum_{\mathcal{N}} A_i x(t - \tau_i(t)) + B_D d(t) + B_R r(t), \quad (7.4)$$

$$y_i(t) = C_i x(t), \quad i \in \mathcal{N}, \quad (7.5)$$

where

$$A = \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix}, \quad A_i = \begin{bmatrix} B_p D_{c,i} C_{p,i} & 0 \\ B_{c,i} C_{p,i} & 0 \end{bmatrix},$$

$$B_D = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \quad B_R = \begin{bmatrix} B_p D_r \\ B_r \end{bmatrix},$$

$$C_i = [C_{p,i} \quad 0], \quad i \in \mathcal{N},$$

and  $\tau_i(t) = t - s_i(k)$  with  $\dot{\tau}_i(t) = 1$  for all  $i \in \mathcal{N}$  are time delays due to sampling.

### 7.1.2 Stability analysis

We derive stability conditions for the closed-loop system (7.4) with bounded sampling intervals. We assume that any sampling satisfies  $t - s_i(t) \leq h_i, \forall t > 0$ . For simplicity, consider the two-sensors case, i.e.,  $N = 2$ . We have the following lemma.

**Lemma 7.1.** *Assume that there exist  $P, U_1, U_2 \in \mathbb{S}_{++}^n$ , some matrices  $P_2, P_3$ , and a constant  $\alpha > 0$  such that the linear matrix inequalities (LMIs)*

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \star & \Phi_{22} + h_1 U_1 + h_2 U_2 \end{bmatrix} < 0, \quad (7.6)$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & -h_1 P_2^\top A_1 \\ \star & \Phi_{22} + h_2 U_2 & -h_1 P_3^\top A_1 \\ \star & \star & -h_1 U_1 e^{-2\alpha h_1} \end{bmatrix} < 0, \quad (7.7)$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & -h_2 P_2^\top A_2 \\ \star & \Phi_{22} + h_1 U_1 & -h_2 P_3^\top A_2 \\ \star & \star & -h_2 U_2 e^{-2\alpha h_2} \end{bmatrix} < 0, \quad (7.8)$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & -h_1 P_2^\top A_1 & -h_2 P_2^\top A_2 \\ \star & \Phi_{22} & -h_1 P_3^\top A_1 & -h_2 P_3^\top A_2 \\ \star & \star & -h_1 U_1 e^{-2\alpha h_1} & 0 \\ \star & \star & 0 & -h_2 U_2 e^{-2\alpha h_2} \end{bmatrix} < 0, \quad (7.9)$$

where

$$\Phi_{11} = P_2^\top (A + A_1 + A_2) + (A + A_1 + A_2)^\top P_2 + 2\alpha P,$$

$$\Phi_{12} = P - P_2^\top + (A + A_1 + A_2)^\top P_3,$$

$$\Phi_{22} = -P_3 - P_3^\top,$$

are feasible. Then the closed-loop system (7.4) with  $d(t) \equiv 0$  and  $r(t) \equiv 0$  is exponentially stable with decay rate  $\alpha > 0$  for all sampling instants less than or equal to  $h_1$  for sensor 1 and  $h_2$  for sensor 2.

*Proof.* See Appendix 7.A. □

**Remark 7.1.** *Lemma 7.1 can be extended to  $N \geq 3$ . In this case,  $2^N$  LMIs will appear. Our assumption that  $N = 2$  is reasonable as many process control loops consist of at most two sensors such as feedforward control and cascade control.*

## 7.2 Event-triggered controller switching

The main idea of this chapter is to activate a sensor to improve the transient response only when the output fluctuates. In this section, we propose a controller switching framework, which activates another sensor only when its measurement gap from the last sampling goes beyond a given threshold.

Let us define two controllers, one of which computes the control signal using one sensor output  $y_1(t)$ , and another controller uses two outputs  $y_1(t)$  and  $y_2(t)$ . Consider the following two controllers

$$\dot{x}_c^1(t) = A_c^1 x_c^1(t) + B_{c,1}^1 y_1(s_1(t)) + K_b^1 \phi^1(t) + B_r^1 r, \quad (7.10a)$$

$$u^1(t) = C_c^1 x_c^1(t) + D_{c,1}^1 y_1(s_1(t)) + D_r^1 r, \quad (7.10b)$$

and

$$\begin{aligned} \dot{x}_c^2(t) &= A_c^2 x_c^2(t) + B_{c,1}^2 y_1(s_1(t)) + B_{c,2}^2 y_2(s_2(t)) \\ &\quad + K_b^1 \phi^2(t) + B_r^2 r, \end{aligned} \quad (7.11a)$$

$$u^2(t) = C_c^2 x_c^2(t) + D_{c,1}^2 y_1(s_1(t)) + D_{c,2}^2 y_2(s_2(t)) + D_r^2 r(t), \quad (7.11b)$$

where  $x_c^1(t) \in \mathbb{R}^{n_c^1}$ ,  $x_c^2(t) \in \mathbb{R}^{n_c^2}$  are the controller states and  $\phi^i(t) \triangleq u(t) - u^i(t)$  the control signal error. Here,  $u(t)$  is the actual control signal to the actuator, which is defined by

$$u(t) = \begin{cases} u^1(t), & \text{if } \sigma(t) = 1, \\ u^2(t), & \text{if } \sigma(t) = 2, \end{cases} \quad (7.12)$$

where  $\sigma : \mathbb{R} \rightarrow \{1, 2\}$  is the controller index function. It takes  $\sigma(t) = 1$  when controller 1 is activated, and  $\sigma(t) = 2$  when controller 2. The block diagram of this switching controller is illustrated in Figure 7.2. The terms  $K_b^i \phi^i(t)$  are called bumpless transfer introduced to reduce the effect of controller switching [221]. Let us note that controller (7.10) uses only the measurement from sensor 1, while controller (7.11) uses both sensors 1 and 2. Augmented by  $\mathbf{x}^\top(t) = [x_p^\top(t), x_c^1{}^\top(t), x_c^2{}^\top(t)]^\top \in \mathbb{R}^{n_p+n_c^1+n_c^2}$ , we obtain the following hybrid system description

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A^{\sigma(t)} \mathbf{x}(t) + A_1^{\sigma(t)} \mathbf{x}(s_1(t)) + A_2^{\sigma(t)} \mathbf{x}(s_2(t)) \\ &\quad + B_D^{\sigma(t)} d(t) + B_R^{\sigma(t)} r(t), \quad \sigma(t) \in \{1, 2\}, \end{aligned} \quad (7.13)$$

$$y_1(t) = C_1 \mathbf{x}(t), \quad (7.14)$$

$$y_2(t) = C_2 \mathbf{x}(t), \quad (7.15)$$

where

$$A^1 = \begin{bmatrix} A_p & B_p C_c^1 & 0 \\ 0 & A_c^1 & 0 \\ 0 & K_b^2 C_c^1 & A_c^2 - K_b^2 C_c^2 \end{bmatrix},$$

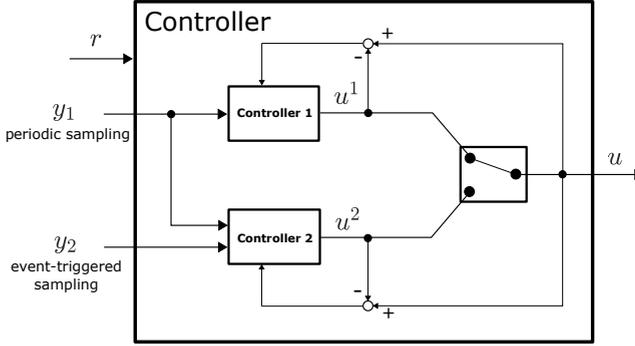


Figure 7.2: Block diagram of an event-triggered switching controller. Controller 1 computes control signal  $u^1$  and controller 2 computes  $u^2$ . The control signal to the plant  $u$  is chosen based on the switching rule.

$$\begin{aligned}
 A_1^1 &= \begin{bmatrix} B_p D_{c,1}^1 C_{p,1} & 0 & 0 \\ B_{c,1}^1 C_{p,1} & 0 & 0 \\ B_{c,1}^2 C_{p,2} + K_b^2 (D_{c,1}^1 - D_{c,1}^2) C_{p,1} & 0 & 0 \end{bmatrix}, \\
 A_2^1 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_{c,2}^2 C_{p,2} - K_b^2 D_{c,2}^2 C_{p,2} & 0 & 0 \end{bmatrix}, \\
 B_D^1 &= \begin{bmatrix} B_d \\ 0 \\ 0 \end{bmatrix}, \quad B_R^1 = \begin{bmatrix} B_p D_r^1 \\ B_r^1 \\ B_r^2 + K_b^2 (D_r^1 - D_r^2) \end{bmatrix}, \\
 A^2 &= \begin{bmatrix} A_p & 0 & B_p C_c^2 \\ 0 & A_c^1 - K_b^1 C_c^1 & K_b^1 C_c^2 \\ 0 & 0 & A_c^2 \end{bmatrix}, \\
 A_1^2 &= \begin{bmatrix} B_p D_{c,1}^2 C_{p,1} & 0 & 0 \\ B_{c,1}^1 C_{p,2} + K_b^1 (D_{c,1}^2 - D_{c,1}^1) C_{p,1} & 0 & 0 \\ B_{c,1}^2 C_{p,1} & 0 & 0 \end{bmatrix}, \\
 A_2^2 &= \begin{bmatrix} B_p D_{c,2}^2 C_{p,1} & 0 & 0 \\ K_b^1 D_{c,2}^2 D_{c,2}^2 C_{p,1} & 0 & 0 \\ B_{c,2}^2 C_{p,2} & 0 & 0 \end{bmatrix},
 \end{aligned}$$

$$B_D^2 = \begin{bmatrix} B_d \\ 0 \\ 0 \end{bmatrix}, \quad B_R^2 = \begin{bmatrix} B_p D_r^2 \\ B_r^1 + K_b^1 (D_r^2 - D_r^1) \\ B_r^2 \end{bmatrix},$$

$$C_1 = [C_{p,1} \ 0 \ 0], \quad C_2 = [C_{p,2} \ 0 \ 0].$$

We define controller switching consisting of three modes  $\mathcal{Q} \triangleq \{q_1, q_2, q_3\}$ , see Figure 7.3. The initial states of the switching is assumed to be  $q(0) = q_1$ ,  $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^{n_p+n_c^1+n_c^2}$ ,  $s_1(0) = s_2(0) = 0$ , where  $q: \mathbb{R} \rightarrow \mathcal{Q}$  is the mode index function. Each mode is characterized by which controller is activated and when the sensor measurements are transmitted.

- In mode  $q_1$ , controller 1 is used, and only sensor 1 transmits the measurement with every  $h_1$  interval to the controller:

$$q_1 : \begin{cases} \sigma(t) = 1, \\ s_1(t) = \lfloor t/h_1 \rfloor h_1, \\ s_2(t) = 0. \end{cases}$$

- In mode  $q_2$ , controller 2 is used. Sensor 1 continues to transmit the measurement with every  $h_1$  interval to the controller, but sensor 2 transmits through event-triggered sampling:

$$q_2 : \begin{cases} \sigma(t) = 2, \\ s_1(t) = \lfloor t/h_1 \rfloor h_1, \\ s_2(t) = \min_{t'} \{t' : (\|y_2(t) - y_2(t')\| \geq \delta \\ \wedge t - t' \geq h_{\min}) \vee t - t' = h_2\}, \end{cases}$$

where the minimum inter-sampling time  $h_{\min} < h_2$  is introduced to avoid Zeno behavior.

- In mode  $q_3$ , controller 1 is used and only sensor 1 transmits the measurement with every  $h_1$  interval to the controller:

$$q_3 : \begin{cases} \sigma(t) = 1, \\ s_1(t) = \lfloor t/h_1 \rfloor h_1, \\ s_2(t) = s_2(t'), \end{cases}$$

where  $t' = \max_t \{t : q(t) = q_2\}$ .

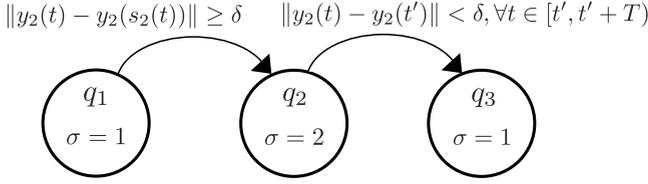


Figure 7.3: Mode transition diagram among  $q_1$ ,  $q_2$ , and  $q_3$ .

Mode transition from mode  $q_1$  to  $q_2$  occurs when  $\|y_2(t) - y_2(s_2(t))\| \geq \delta$ , and from  $q_2$  to  $q_3$  occurs at  $t = t_1 + T$  when  $\|y_2(t) - y_2(t_1)\| < \delta, \forall t \in [t_1, t_1 + T)$ , for some  $t_1$ .

Now, we make the following assumption.

**Assumption 7.1.** *There exist  $P^j, U_1^j, U_2^j \in \mathbb{S}_{++}^n$ , and some matrices  $P_2^j, P_3^j$  for  $j = 1, 2$ , such that the LMIs (7.6)–(7.9) hold in which the matrices are replaced by  $P = P^j, U_1 = U_1^j, U_2 = U_2^j, P_2 = P_2^j, P_3 = P_3^j$ , and*

$$\begin{aligned}\Phi_{11} &= P_2^\top (A^j + A_1^j + A_2^j) + (A^j + A_1^j + A_2^j)^\top P_2 + 2\alpha P, \\ \Phi_{12} &= P - P_2^\top + (A^j + A_1^j + A_2^j)^\top P_3, \\ \Phi_{22} &= -P_3 - P_3^\top.\end{aligned}$$

Assumption 7.1 guarantees that both controllers (7.10) and (7.11) without switching stabilize the plant (7.1)–(7.2). The following theorem summarizes that the proposed switching framework yields a stable closed-loop system.

**Theorem 7.1.** *Suppose Assumption 7.1 holds. The event-triggered switching control system defined by the plant (7.1)–(7.2) and the controllers (7.10)–(7.12) is asymptotically stable with  $d(t) \equiv 0$  and  $r(t) \equiv 0$ .*

*Proof.* First, note that the state  $\mathbf{x}(t)$  converges to the origin if the system stays in mode  $q_1$  for all  $t \geq 0$ . We show that, in mode  $q_2$ , there exists time instance  $t_1$  such that  $\|y_2(t) - y_2(t_1)\| < \delta$  for  $t > t_1$ . Due to Assumption 7.1, the system with controller 2 is asymptotically stable. Thus, there exists a time instance  $t'_1$  such that  $y_2(t)$  never leaves the  $\delta/2$ -neighbourhood of the origin for  $t > t'_1$ . Taking  $t_1 > t'_1$  as the first sensor 2 sampling time after  $t'_1$ , then, for  $t > t_1$ , we have  $\|y_2(t) - y_2(t_1)\| < \delta$ . This guarantees that the system goes to mode  $q_3$  after  $t = t_1 + T$ . The proof completes since in mode  $q_3$ , the system with controller 1 is also asymptotically stable.  $\square$

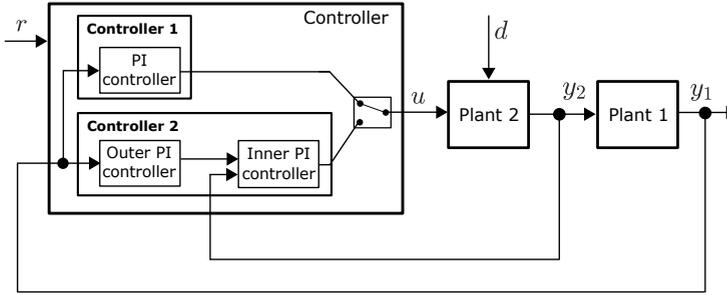


Figure 7.4: Event-triggered cascade control with controller switching.

## 7.3 Applications to cascade and feedforward control

In this section, we apply the controller switching to cascade and feedforward control. Our idea is to use a PI controller when disturbances are not present and to activate a cascade controller or a feedforward controller when they are believed to be present.

### 7.3.1 Cascade control

In cascade control, the outer controller computes its control signal for the inner controller. The inner controller then sends its control signal to the actuator. Corresponding to the controller switching framework, the cascade control is used in mode  $q_2$ , while PI control is activated in mode  $q_1$  and  $q_3$ .

The block diagram of the event-triggered cascade control with controller switching is shown in Figure 7.4. Plants 1 and 2 are given by

$$\begin{aligned}\dot{x}_{p1}(t) &= A_{p1}x_{p1}(t) + B_{p1}y_2(t), \\ \dot{x}_{p2}(t) &= A_{p2}x_{p2}(t) + B_{p2}u(t) + B_{d2}d(t),\end{aligned}$$

with

$$y_1(t) = C_{p1}x_{p1}(t), \quad y_2(t) = C_{p2}x_{p2}(t),$$

where  $x_{p1}(t)$  and  $x_{p2}(t)$  are the states of plants 1 and 2, respectively. Thus, we have

$$A_p = \begin{bmatrix} A_{p1} & B_{p1}C_{p2} \\ 0 & A_{p2} \end{bmatrix}, \quad B_p = \begin{bmatrix} 0 \\ B_{p2} \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 \\ B_{d2} \end{bmatrix},$$

$$C_p^1 = [C_{p1} \ 0], \quad C_p^2 = [0 \ C_{p2}].$$

Consider PI control for both outer and inner controllers. Denote  $x_{c1}(t)$  and  $u_1(t)$  as the outer controller state and its control signal, respectively. Then we have

$$\begin{aligned} \dot{x}_{c1}(t) &= r(t) - y_1(s_1(t)), \\ u_1(t) &= K_{i1}^2 x_{c1}(t) + K_{p1}^2 (r(t) - y_1(s_1(t))). \end{aligned}$$

In the same way, we describe the inner controller as

$$\begin{aligned} \dot{x}_{c2}(t) &= u_1(t) - y_2(s_2(t)), \\ u^2(t) &= K_{i2}^2 x_{c2}(t) + K_{p2}^2 (u_1(t) - y_2(s_2(t))), \end{aligned}$$

where  $x_{c2}(t)$  is the secondary controller state. Introducing an augmented controller state  $x_c^{\top}(t) = [x_{c1}^{\top}(t) \ x_{c2}^{\top}(t)]^{\top}$ , we obtain a cascade controller as (7.11) with

$$\begin{aligned} A_c^2 &= \begin{bmatrix} 0 & 0 \\ K_{i1}^2 & 0 \end{bmatrix}, \quad B_{c,1}^2 = \begin{bmatrix} -1 \\ -K_{p1}^2 \end{bmatrix}, \quad B_{c,2}^2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \\ B_r^2 &= \begin{bmatrix} 1 \\ K_{p1}^2 \end{bmatrix}, \quad C_c^2 = [K_{p2}^2 K_{i1}^2 \quad K_{i2}^2], \\ D_{c,1}^2 &= -K_{p2}^2 K_{p1}^2, \quad D_{c,2}^2 = -K_{p2}^2, \quad D_r^2 = K_{p2}^2 K_{p1}^2. \end{aligned}$$

In the same way, we obtain a PI controller as (7.10) with

$$\begin{aligned} A_c^1 &= 0, \quad B_{c,1}^1 = -1, \quad B_r^1 = 1, \\ C_c^1 &= K_i^1, \quad D_{c,1}^1 = -K_p^1, \quad D_r^1 = K_p^1. \end{aligned} \quad (7.16)$$

### 7.3.2 Feedforward control

The block diagram of the event-triggered feedforward control with controller switching is shown in Figure 7.5. Plant 2 can be an uncontrolled stable plant, a closed-loop system, or an independent controller located in a different place. The plants are given by

$$\begin{aligned} \dot{x}_{p1}(t) &= A_{p1} x_{p1}(t) + B_{p1} u(t) + B_{d1} w(t), \\ \dot{x}_{p2}(t) &= A_{p2} x_{p2}(t) + B_{d2} d(t), \end{aligned}$$

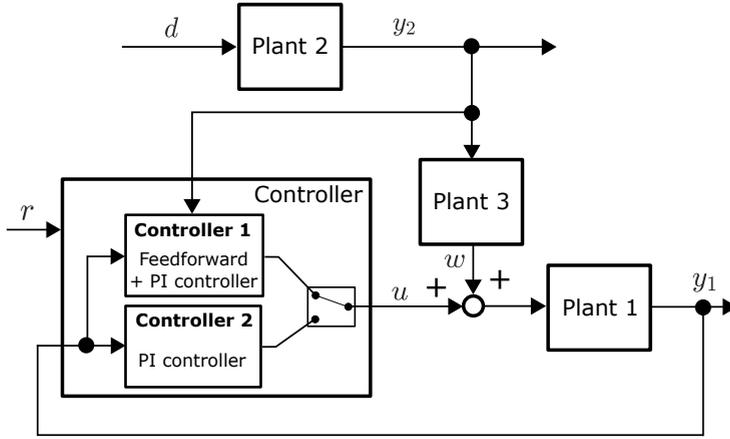


Figure 7.5: Event-triggered feedforward control with controller switching.

$$\dot{x}_{p3}(t) = A_{p3}x_{p3}(t) + B_{d3}y_2(t),$$

with

$$\begin{aligned} y_1(t) &= C_{p1}x_{p1}(t), & y_2(t) &= C_{p2}x_{p2}(t), \\ w(t) &= C_{p3}x_{p3}(t). \end{aligned}$$

The feedforward controller used in mode  $q_2$  is described as (7.11) with

$$\begin{aligned} A_c^2 &= 0, & B_{c,1}^2 &= -1, & B_{c,2}^2 &= 0, & B_r^2 &= 1, \\ C_c^2 &= K_f^2, & D_{c,1}^2 &= -K_p^2, & D_{c,2}^2 &= K_f^2, & D_r^2 &= K_p^2, \end{aligned}$$

where  $K_f^2$  is a feedforward gain. In mode  $q_1$  and  $q_3$ , the controller is given by (7.10) with (7.16).

## 7.4 Numerical examples

In this section, we provide numerical examples of the proposed framework applied to cascade and feedforward control.

### 7.4.1 Event-triggered cascade control

We first illustrate cascade control, where the plant is given by

$$\begin{aligned}\dot{x}_{p1}(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -3 \end{bmatrix} x_{p1}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} y_2(t), \\ \dot{x}_{p2}(t) &= -2x_{p2}(t) + u(t) + d(t), \\ y_1(t) &= [10 \ 0 \ 0] x_{p1}(t), \quad y_2(t) = 3x_{p2}(t).\end{aligned}$$

Lemma 7.1 guarantees that both controllers with the parameters  $K_p^1 = 0.0119$ ,  $K_i^1 = 0.0140$ ,  $K_b^1 = 50$ ,  $K_{p1}^2 = 0.015$ ,  $K_{i1}^2 = 0.0209$ ,  $K_{p2}^2 = 0.244$ ,  $K_{i2}^2 = 1.8209$ , and  $K_b^{2T} = [5, 5]$  stabilize the system with  $h_1 = 1.5$ ,  $h_2 = 0.3$ . We introduce the proposed event-triggered controller switching with  $\delta = 0.2$  and  $T = 3$ .

Figure 7.6 shows the response to the external disturbance  $d(t) = 5, \forall t \geq 5$ . It can be found that the disturbance activates sensor 2, and the controller is switched to the cascade control. In mode  $q_2$ , sensor 2 takes frequent samplings at the beginning. After several periodic samplings, sensor 2 is deactivated and the mode is switched to  $q_3$  (Figure 7.6: bottom).

Figure 7.7 compares outputs  $y_1(t)$  for three cases: the proposed event-triggered cascade control with controller switching (ET cascade control, red solid line), cascade control with constant sampling rates with  $h_1 = 1.5$ ,  $h_2 = 0.3$  (TT cascade control), and PI control with  $h_1 = 1.5$  (TT PI control, green dot line). Apparently, the cascade controllers dramatically reduce the effect of the disturbance. Furthermore, since the proposed control suspends sensor 2 samplings after the mode is switched, fewer samplings are needed than the cascade control with constant sampling rates. The proposed control takes 41 samples only in  $q_2$ , while the control with constant sampling rates takes 117 samplings until  $t = 35$  and the total samplings will constantly increase.

### 7.4.2 Event-triggered feedforward control

We next show a numerical example of the proposed event-triggered feedforward control. The plant is given by

$$\dot{x}_{p1}(t) = \begin{bmatrix} -5 & 0 \\ 1 & 0 \end{bmatrix} x_{p1}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w(t),$$

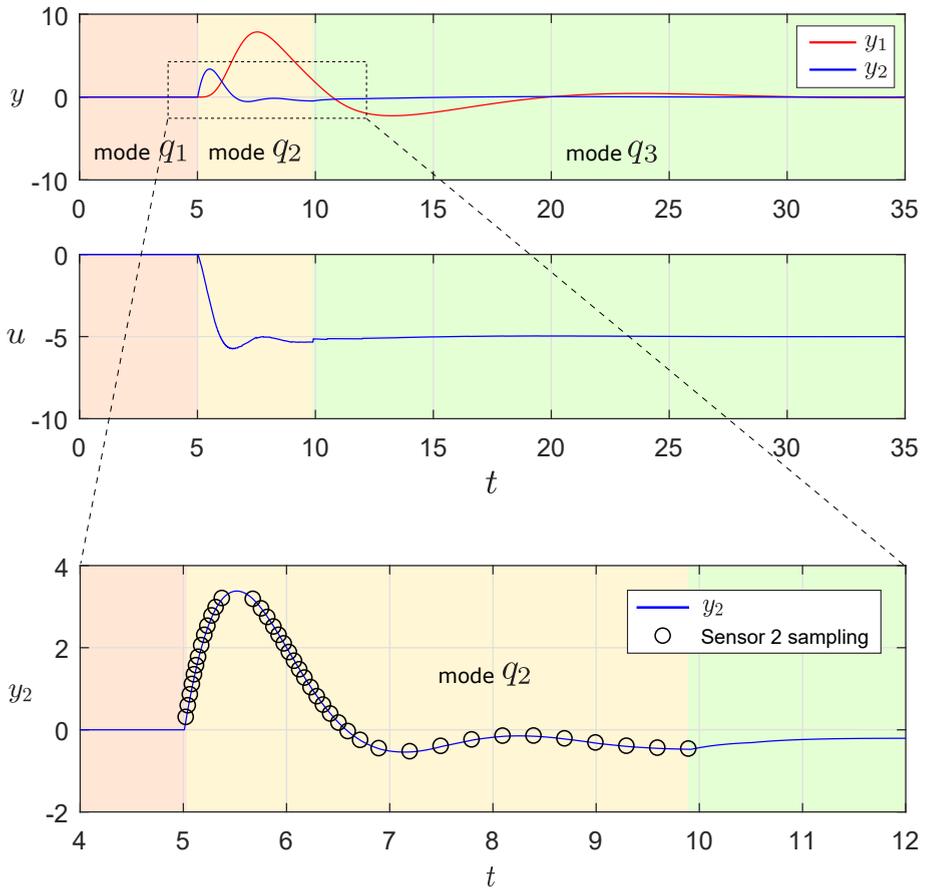


Figure 7.6: Response of event-triggered cascade control with controller switching (red: mode  $q_1$ , yellow: mode  $q_2$ , and green: mode  $q_3$ ).

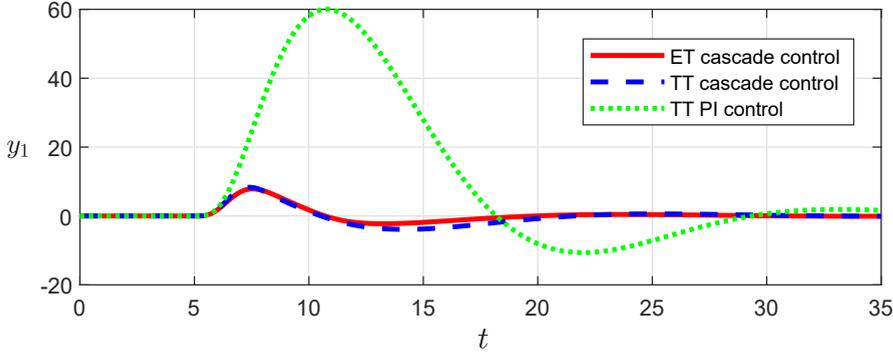


Figure 7.7: Outputs for the three cases: the proposed event-triggered cascade control with controller switching (ET cascade control, red solid line), cascade control with constant sampling rates (TT cascade control, blue dashed line), and PI control (TT PI control, green dot line).

$$\begin{aligned}\dot{x}_{p2}(t) &= -x_{p2}(t) + d(t), & \dot{x}_{p3}(t) &= -5x_{p2}(t) + y_2(t), \\ y_1(t) &= \begin{bmatrix} 0 & 10 \end{bmatrix} x_{p1}(t), & y_2(t) &= 3x_{p2}(t), & w(t) &= x_{p3}(t).\end{aligned}$$

Lemma 7.1 guarantees that both controllers with the parameters  $K_p^1 = 0.85$ ,  $K_i^1 = 0.0241$ ,  $K_b^1 = 50$ ,  $K_p^2 = 0.325$ ,  $K_{i1}^2 = 0.288$ ,  $K_f^2 = -0.1$ , and  $K_b^2 = 50$  stabilize the system with  $h_1 = 0.5$ ,  $h_2 = 2.5$ . We introduce the proposed event-triggered controller switching with  $\delta = 0.1$  and  $T = 12.5$ .

Figure 7.8 shows the response to the reference signal  $r(t) = 1, \forall t \geq 0$ , and the external disturbance  $d(t) = 0.1, \forall t \geq 15$ . The reference signal does not activate sensor 2 and the mode stays mode  $q_1$ . The disturbance occurs at  $t = 15$ , which results in mode switching. In mode  $q_2$ , the feedforward controller takes the corrective action based on sensor 2 measurements as shown in the bottom plot of Figure 7.8. Sensor 2 takes frequent samples until around  $t = 18$ , then is deactivated at  $t = 30$  after several periodic samplings.

Figure 7.9 compares outputs for three cases: the proposed event-triggered feedforward control with controller switching (ET FF + PI control, red solid line), feedforward control with constant sampling rates with  $h_1 = 0.8$ ,  $h_2 = 2.5$  (TT FF + PI control, blue dashed line), and PI control with  $h_1 = 0.8$  (TT PI control, green dot line). The proposed controller realizes the same step response as the PI control which has a smaller overshoot than the feedforward

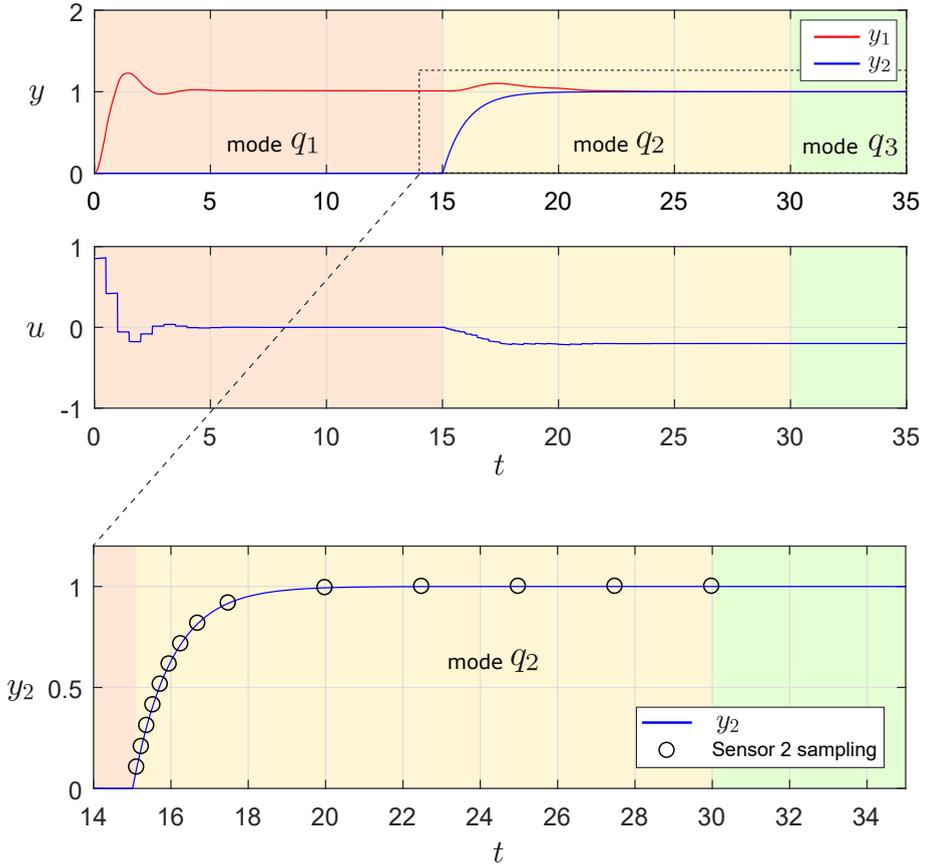


Figure 7.8: Response of event-triggered feedforward control with controller switching (red: mode  $q_1$ , yellow: mode  $q_2$ , and green: mode  $q_3$ ).

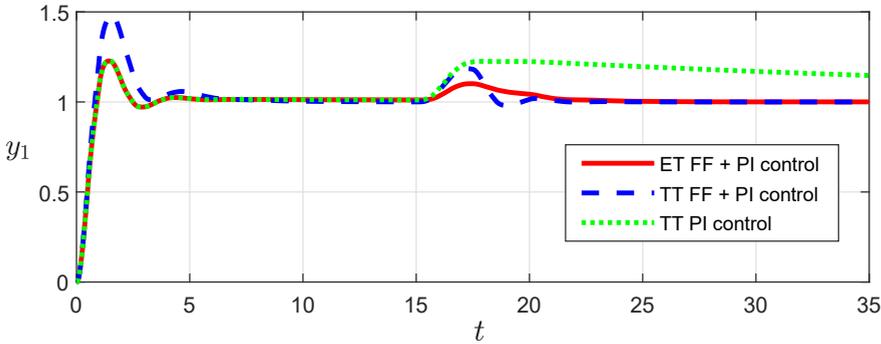


Figure 7.9: Outputs for the three cases: the proposed event-triggered feedforward control with controller switching (ET FF + PI control, red solid line), feedforward control with constant sampling rates (TT FF + PI control, blue dashed line), and PI control (TT PI control, green dot line).

control. It achieves better disturbance rejection compared to the PI control since the controller is switched to the one with a large integral gain and with a feedforward controller. It has even better performance than the feedforward control with constant sampling rates since the event-triggered samplings can rapidly react to the disturbance.

## 7.5 Summary

In this chapter, we proposed an event-triggered controller switching framework for plants monitored by multiple sensors. We considered a scenario that a controller regulated a plant with a hard-wired sensor, and another wireless sensor was deployed to improve closed-loop control performance. The wireless sensor was activated in an event-triggered fashion to attenuate the effects of external disturbances. We derived the stability conditions of the proposed systems. It was shown that this framework could be applied to cascade and feedforward control. Numerical examples showed that the framework could mitigate the effect of disturbances with fewer samplings compared to controllers with constant sampling and without switching.

## 7.A Proof of Lemma 7.1

The proof follows [124]. Denote  $x_t(\theta) = x(t + \theta)$ , then consider the Lyapunov–Krasovskii functional

$$\begin{aligned} V_s(t, x(t), \dot{x}_t) &= V(t) \\ &= x^\top(t)Px(t) + V_{U_1}(t, \dot{x}_t) + V_{U_2}(t, \dot{x}_t) \end{aligned}$$

where

$$V_{U_i}(t, \dot{x}_t) = (h_i - \tau_i(t)) \int_{s_i(t)}^t e^{2\alpha(s-t)} \dot{x}^\top(s) U_i \dot{x}(s) ds$$

for  $i = 1, 2$ . Since

$$\begin{aligned} &\dot{V}_{U_i} + 2\alpha V_{U_i} \\ &= - \int_{s_i(t)}^t e^{2\alpha(s-t)} \dot{x}^\top(s) U_i \dot{x}(s) ds + (h_i - \tau_i(t)) \dot{x}^\top(t) U_i \dot{x}(t) \\ &\leq -e^{2\alpha h_i} \int_{s_i(t)}^t \dot{x}^\top(s) U_i \dot{x}(s) ds + (h_i - \tau_i(t)) \dot{x}^\top(t) U_i \dot{x}(t), \end{aligned}$$

we have

$$\begin{aligned} &\dot{V}(t) + 2\alpha V(t) \\ &\leq 2\dot{x}^\top(t)Px(t) + 2\alpha x^\top(t)Px(t) \\ &\quad - \sum_{i=1}^2 \left[ e^{2\alpha h_i} \int_{t-\tau_i(t)}^t \dot{x}^\top(s) U_i \dot{x}(s) ds - (h_i - \tau_i(t)) \dot{x}^\top(t) U_i \dot{x}(t) \right]. \quad (7.17) \end{aligned}$$

Defining

$$v_i = \frac{1}{\tau_i(t)} \int_{t-\tau_i(t)}^t \dot{x}(s) ds,$$

we apply Jensen's inequality

$$\int_{t-\tau_i(t)}^t \dot{x}^\top(s) U_i \dot{x}(s) ds \geq \tau_i(t) v_i^\top U_i v_i$$

and the descriptor method [224], where the right-hand side of the description

$$0 = 2[x^\top(t)P_2^\top + \dot{x}^\top(t)P_3^\top]$$

$$\times [(A + A_1 + A_2)x(t) - \tau_1(t)A_1v_1 - \tau_2(t)A_2v_2 - \dot{x}(t)]$$

with some matrices  $P_2, P_3 \in \mathbb{R}^{n \times n}$  is added into (7.17). Thus, by setting  $\eta(t) = [x^\top, \dot{x}^\top(t), v_1^\top, v_2^\top]$ , we have

$$\frac{d}{dt}V(t) + 2\alpha V(t) \leq \eta^\top(t)\Psi\eta(t) \leq 0$$

if

$$\Psi(\tau_1, \tau_2) = \begin{bmatrix} \Phi_{11} & \Phi_{12} & -\tau_1 P_2^\top A_1 & -\tau_2 P_2^\top A_2 \\ * & \Phi_{22} + \Phi'_{22} & -\tau_1 P_3^\top A_1 & -\tau_2 P_3^\top A_2 \\ * & * & -\tau_1 U_1 e^{-2\alpha h_1} & 0 \\ * & * & 0 & -\tau_2 U_2 e^{-2\alpha h_2} \end{bmatrix} < 0 \quad (7.18)$$

where  $\Phi'_{22} = (h_1 - \tau_1(t))U_1 + (h_2 - \tau_2(t))U_2$ . The LMI (7.18) holds if (7.6)–(7.9) are satisfied since

$$\begin{aligned} \eta^\top \Psi(\tau_1, \tau_2) \eta &= \frac{h_1 - \tau_1}{h_1} \left( \frac{h_2 - \tau_2}{h_2} \eta^\top \Psi(0, 0) \eta + \frac{\tau_2}{h_2} \eta^\top \Psi(0, h_2) \eta \right) \\ &\quad + \frac{\tau_1}{h_1} \left( \frac{h_2 - \tau_2}{h_2} \eta^\top \Psi(h_1, 0) \eta + \frac{\tau_2}{h_2} \eta^\top \Psi(h_1, h_2) \eta \right). \end{aligned}$$

## Chapter 8

# Conclusions and Future Research

This chapter summarizes the main results of the thesis and discusses possible directions for future research.

### 8.1 Conclusions

In this thesis, we focused on two essential problems for wireless process control: the design of a multi-hop network scheduler (Part I) and event-triggered process control algorithms (Part II).

#### **Multi-hop network scheduler**

In Chapter 3, we proposed a multi-hop network scheduler for monitoring systems. We formulated an optimization problem minimizing an infinite-time averaged estimation error covariance taking into account sensor energy consumption. By exploiting the necessary conditions for network scheduling optimality, we showed that the problem could be divided into two subproblems: tree planning and sensor selection. The tree planning subproblem gives optimal routes from sensors to the remote estimator under a given set of scheduled sensors. It was shown that a solution could be derived efficiently. The sensor selection subproblem offers an optimal sensor selection at every time instance. By transforming the problem to a Markov decision process, we derived a condition for the existence of a periodic optimal schedule. We proposed

two algorithms for suboptimal schedules. It was demonstrated that the proposed algorithms are effective through three numerical examples.

In Chapter 4, we proposed a co-design framework for sensor scheduling, routing, and control over a multi-hop wireless sensor and actuator network. We considered a sensor–actuator network where the controllers were co-located with the corresponding actuators. In this framework, we showed that optimal schedules, routes, and controllers could be obtained separately. The optimal schedules were given by covariance-based threshold policies, routes by solving the minimum-cost path problem between each sensor and actuator, and controllers by the standard linear quadratic Gaussian controller design problem. We also provided algorithms for sensors and actuators to switch routes and schedules locally in case of network link outage. The applicability of the theoretical results and the proposed algorithms were illustrated in a numerical example.

### **Event-triggered process control**

In Chapter 5, we studied periodic event-triggered actuation applied to PID, cascade, and decoupling control. The controllers updated its command when its value went beyond a given threshold. We formulated an event-triggered output feedback control system with delayed samplings, and its exponential stability was derived in the form of linear matrix inequalities (LMIs). It was shown that the proposed controller was capable of setpoint tracking and disturbance rejection. Event threshold tuning was also proposed. We applied the framework to PI, PID, cascade, and decoupling control. The numerical examples showed that the proposed controllers reduced the communication load while maintaining control performance.

In Chapter 6, we investigated periodic event-triggered actuation applied to time-delay systems. We considered a PI controller updating its control command when it went beyond a given threshold. The Smith predictor was introduced to compensate for time delays. Stability conditions in the form of LMIs under the assumption that the system parameters resided in uncertain polytopes were derived. Based on this result, an event threshold tuning procedure was proposed. Numerical examples showed that the controller reduced the communication load with only a slight performance degradation.

In Chapter 7, we proposed an event-triggered controller switching framework for plants monitored by multiple sensors. We studied output feedback control systems where one sensor was activated in an event-triggered fash-

ion to attenuate the effects of an external disturbance. We derived stability conditions of the proposed closed-loop systems. It was shown that this framework could be applied to cascade and feedforward control. Numerical examples showed that the proposed algorithm mitigated disturbances with fewer samplings than controllers with constant sampling and without the proposed switching.

## 8.2 Future research directions

There are several interesting research directions based on the work presented in this thesis. We summarize some possible extensions in this section.

### **Multi-hop scheduling under limited number of timeslots**

Our framework in Part I assumed that we could use as many timeslots as we need in every single superframe. However, the number of timeslots in a single superframe is limited in existing protocols such as WirelessHART and ISA100.11a. We are planning to consider this case, in which sensor data may not reach an estimator or a controller in a single superframe, in the future. An optimal network schedule, as the one derived in Chapter 3, does not then necessarily form a tree graph. Moreover, we cannot obtain optimal routes and schedules independently for each control loop, as was done in Chapter 4.

### **Network reconfiguration under detailed channel models**

Optimal schedules and routes are not fixed as the wireless environment in process plants may change significantly over time. In Chapter 4, we proposed algorithms to switch schedules and routes when a link is unavailable. The algorithms can be improved by introducing channel variation models to make our framework tolerant to network environment change. For example, a semi-Markov chain model can be used [115].

### **Applications of event-triggered control**

As an extension of the work in Part II, it is interesting to consider other control architectures commonly used in process control systems, for example, split-range control and override control [25]. Other types of predictors can also be

considered. In [225], the authors propose a sequence of subpredictors to stabilize a plant with a long time delay. Event-triggered control with such predictors could be developed. Uncertain time-varying delays can also be considered by applying the results in [226]. Anti-windup compensation could be extended to other control architectures [201, 227].

# Bibliography

- [1] A. Ahlén, J. Åkerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, “Toward wireless control in industrial process automation: A case study at a paper mill”, *IEEE Control Systems Magazine*, vol. 39, no. 5, pp. 36–57, 2019.
- [2] T. Samad, P. McLaughlin, and J. Lu, “System architecture for process automation: Review and trends”, *Journal of Process Control*, vol. 17, no. 3, pp. 191–201, 2007.
- [3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions”, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.
- [4] A. J. Isaksson, I. Harjunoski, and G. Sand, “The impact of digitalization on the future of control and operations”, *Computers & Chemical Engineering*, vol. 114, pp. 122–129, 2017.
- [5] H. Forbes, “ExxonMobil’s quest for the future of process automation”, *ARC Insights*, ARC Advisory Group, 2016.
- [6] J. Åkerberg, M. Gidlund, and M. Björkman, “Future research challenges in wireless sensor and actuator networks targeting industrial automation”, in *Proceedings of IEEE International Conference on Industrial Informatics*, 2011, pp. 410–415.
- [7] Emerson Automation Solutions, *Emerson reaches one billion hours of wireless operations*, 2012. [Online]. Available: <https://www.automation.com/en-us/articles/2012-2/emerson-reaches-one-billion-hours-of-wireless-oper>.
- [8] HART Communication Foundation, *WirelessHart Overview*, 2014.

- [9] International Society of Automation, *Wireless systems for industrial automation: Process control and related applications, ISA100.11a-2009*, 2009.
- [10] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, “Wireless network design for control systems: A survey”, *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 978–1013, 2018.
- [11] A. Willig, “Recent and emerging topics in wireless industrial communications: A selection”, *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, 2008.
- [12] T. S. Rappaport and C. D. McGillem, “UHF fading in factories”, *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 1, pp. 40–48, 1989.
- [13] H. Hashemi, “The indoor radio propagation channel”, *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, 1993.
- [14] M. Eriksson and T. Olofsson, “On long-term statistical dependences in channel gains for fixed wireless links in factories”, *IEEE Transactions on Communications*, vol. 64, no. 7, pp. 3078–3091, 2016.
- [15] C. Fischione, *An introduction to wireless sensor networks*. KTH Royal Institute of Technology, 2014.
- [16] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0”, *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.
- [17] Q. Wang and J. Jiang, “Comparative examination on architecture and protocol of industrial wireless sensor network standards”, *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2197–2219, 2016.
- [18] K. Yu, Z. Pang, M. Gidlund, J. Åkerberg, and M. Björkman, “RE-ALFLOW: Reliable real-time flooding-based routing protocol for industrial wireless sensor networks”, *International Journal of Distributed Sensor Networks*, vol. 10, no. 7, pp. 1–17, 2014.
- [19] T. Lennvall, J. Åkerberg, E. Hansen, and K. Yu, “A new wireless sensor network TDMA timing synchronization protocol”, in *Proceedings of IEEE International Conference on Industrial Informatics*, 2016, pp. 606–611.

- [20] P. Agrawal, A. Ahlén, T. Olofsson, and M. Gidlund, “Long term channel characterization for energy efficient transmission in industrial environments.”, *IEEE Transactions on Communications*, vol. 62, no. 8, pp. 3004–3014, 2014.
- [21] T. Olofsson, A. Ahlén, and M. Gidlund, “Modeling of the fading statistics of wireless sensor network channels in industrial environments”, *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3021–3034, 2016.
- [22] S. Bennett, “A brief history of automatic control”, *IEEE Control Systems Magazine*, vol. 16, no. 3, pp. 17–25, 1996.
- [23] I. Verhappen and A. Pereira, *Foundation Fieldbus*. ISA, 2008.
- [24] PROFIBUS & PROFINET International, *PROFIBUS Standard*. [Online]. Available: <https://www.profibus.com/>.
- [25] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, *Process Dynamics and Control*. John Wiley & Sons, 2010.
- [26] R. K. Wood and M. W. Berry, “Terminal composition control of a binary distillation column”, *Chemical Engineering Science*, vol. 28, no. 9, pp. 1707–1717, 1973.
- [27] O. J. M. Smith, “A controller to overcome dead time”, *ISA Journal*, vol. 6, pp. 28–33, 1959.
- [28] K. J. Åström, C. C. Hang, and B. C. Lim, “A new smith predictor for controlling a process with an integrator and long dead-time”, *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 343–345, 1994.
- [29] S. Majhi and D. P. Atherton, “Modified smith predictor and controller for processes with time delay”, *IEE Proceedings - Control Theory and Applications*, vol. 146, no. 5, pp. 359–366, 1999.
- [30] R. Sanz, P. García, and P. Albertos, “A generalized smith predictor for unstable time-delay SISO systems”, *ISA Transactions*, vol. 72, pp. 197–204, 2018.
- [31] A. Ingimundarson and T. Hägglund, “Robust tuning procedures of dead-time compensating controllers”, *Control Engineering Practice*, vol. 9, no. 11, pp. 1195–1208, 2001.
- [32] J. Walrand and P. P. Varaiya, *High-performance communication networks*. Morgan Kaufmann, 2000.

- [33] IEEE 802.15.4, *IEEE 802.15.4 Standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (WPANs)*, 2006. [Online]. Available: <http://www.ieee802.org/15/pub/TG4.html>.
- [34] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, Y. Selén, and J. Sköld, “5G wireless access: Requirements and realization”, *IEEE Communications Magazine*, vol. 52, no. 12, pp. 42–47, 2014.
- [35] M. Gidlund, T. Lennvall, and J. Åkerberg, “Will 5G become yet another wireless technology for industrial automation?”, in *Proceedings of IEEE International Conference on Industrial Technology*, 2017, pp. 1319–1324.
- [36] G. Fodor and S. Sorrentino, “D2D communications?—What part will it play in 5G?”, *Ericsson Blog*, vol. 6, 2014. [Online]. Available: <https://www.ericsson.com/research-blog/device-device-communications/>.
- [37] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, “Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges”, *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.
- [38] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, “Network slicing to enable scalability and flexibility in 5G mobile networks”, *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, 2017.
- [39] S. Petersen and S. Carlsen, “WirelessHART versus ISA100. 11a: The format war hits the factory floor”, *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, 2011.
- [40] A. A. Kumar S, K. Øvsthus, and L. M. Kristensen, “An industrial perspective on wireless sensor networks—A survey of requirements, protocols, and challenges”, *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.
- [41] D. Chen, M. Nixon, and A. Mok, *WirelessHART: Real-Time Mesh Network for Industrial Automation*. Springer, 2010.
- [42] M. Nobre, I. Silva, and L. A. Guedes, “Routing and scheduling algorithms for wirelessHART networks: A survey”, *Sensors*, vol. 15, no. 5, pp. 9703–9740, 2015.

- [43] M. Nixon, D. Chen, T. Blevins, and A. K. Mok, "Meeting control performance over a wireless mesh network", in *Proceedings of IEEE International Conference on Automation Science and Engineering*, 2008, pp. 540–547.
- [44] F. Li, Z. Zhang, Z. Jia, and L. Ju, "Superframe scheduling for data aggregation in WirelessHart networks", in *Proceedings of IEEE International Conference on High Performance Computing and Communications*, 2015, pp. 1540–1545.
- [45] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks", *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1151, 2005.
- [46] P. Neumann, "Communication in industrial automation—What is going on?", *Control Engineering Practice*, vol. 15, no. 11, pp. 1332–1347, 2007.
- [47] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches.", *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [48] J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 29–47, 2007.
- [49] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems", *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, 2016.
- [50] R. Candell, M. Kashef, Y. Liu, K. B. Lee, and S. Fofou, "Industrial wireless systems guidelines: Practical considerations and deployment life cycle", *IEEE Industrial Electronics Magazine*, vol. 12, no. 4, pp. 6–17, 2018.
- [51] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey", *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [52] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey", *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [53] *Zigbee specification*, 2006. [Online]. Available: <https://www.zigbee.org/download/standards-zigbee-specification/>.

- [54] N. Aakvaag, M. Mathiesen, and G. Thonet, "Timing and power issues in wireless sensor networks-an industrial test case", in *Proceedings of International Conference on Parallel Processing*, 2005, pp. 419–426.
- [55] J. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks", *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.
- [56] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks", in *Proceedings of IEEE International Workshop on Quality of Service*, 2006, pp. 83–92.
- [57] A. E. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in static wireless networks", *IEEE Transactions on Communications*, vol. 55, no. 11, pp. 2185–2192, 2007.
- [58] P. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincent, "Breath: An adaptive protocol for industrial control applications using wireless sensor networks", *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 821–838, 2011.
- [59] M. Sha, D. Gunatilaka, C. Wu, and C. Lu, "Empirical study and enhancements of industrial wireless sensor-actuator network protocols", *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 696–704, 2017.
- [60] V. Pimentel and B. G. Nickerson, "A safety function response time model for wireless industrial control", in *Proceedings of Annual Conference of the IEEE Industrial Electronics Society*, 2014, pp. 3878–3884.
- [61] J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "Comparative study of energy-efficient sampling approaches for wireless control networks", *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 416–424, 2010.
- [62] K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson, "Scheduling smart home appliances using mixed integer linear programming", in *Proceedings of IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 5144–5149.
- [63] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey", *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

- [64] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks", *IEEE Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [65] J. Dou, Z. Guo, J. Cao, and G. Zhang, "Data aggregation rate assumption for wireless sensors", in *Proceedings of IEEE International Symposium on Radio-Frequency Integration Technology*, 2007, pp. 278–281.
- [66] S. K. Singh, P. Kumar, and J. P. Singh, "A survey on successors of LEACH protocol", *IEEE Access*, vol. 5, pp. 4298–4328, 2017.
- [67] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems", *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.
- [68] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
- [69] W. P.M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nešić, "Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance", *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781–1796, 2010.
- [70] M. C. F. Donkers, W. P.M. H. Heemels, N. Van de Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach", *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2101–2115, 2011.
- [71] M. C. F. Donkers, W. P.M. H. Heemels, D. Bernardini, A. Bemporad, and V. Shneer, "Stability analysis of stochastic networked control systems", *Automatica*, vol. 48, no. 5, pp. 917–925, 2012.
- [72] E. Fridman, *Introduction to time-delay systems: Analysis and control*. Springer, 2014.
- [73] K. Liu, A. Selivanov, and E. Fridman, "Survey on time-delay approach to networked control", *Annual Reviews in Control*, vol. 48, pp. 57–79, 2019.
- [74] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

- [75] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations", *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.
- [76] L. Shi, M. Epstein, and R. M. Murray, "Kalman filtering over a packet-dropping network: A probabilistic perspective", *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 594–604, 2010.
- [77] Y. Xu and J. P. Hespanha, "Estimation under uncontrolled and controlled communications in networked control systems", in *Proceedings of IEEE Conference on Decision and Control and European Control Conference*, 2005, pp. 842–847.
- [78] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet drop", *IEEE Transactions on Automatic Control*, vol. 53, no. 5, p. 1311, 2008.
- [79] C. O. Savage and B. F. La Scala, "Optimal scheduling of scalar Gauss–Markov systems with a terminal cost function", *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1100–1105, 2009.
- [80] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage", *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.
- [81] Y. Mo, R. Ambrosino, and B. Sinopoli, "Sensor selection strategies for state estimation in energy constrained wireless sensor networks", *Automatica*, vol. 47, no. 7, pp. 1330–1338, 2011.
- [82] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems", *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.
- [83] W. Zhang, M. P. Vitus, J. Hu, A. Abate, and C. J. Tomlin, "On the optimal solutions of the infinite-horizon linear sensor scheduling problem", in *Proceedings of IEEE Conference on Decision and Control*, 2010, pp. 396–401.
- [84] Y. Mo, E. Garone, and B. Sinopoli, "On infinite-horizon sensor scheduling", *Systems & Control Letters*, vol. 67, pp. 65–70, 2014.
- [85] L. Zhao, W. Zhang, J. Hu, A. Abate, and C. J. Tomlin, "On the optimal solutions of the infinite-horizon linear sensor scheduling problem", *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2825–2830, 2014.

- [86] S. T. Jawaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems", *Automatica*, vol. 61, pp. 282–288, 2015.
- [87] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms", *Automatica*, vol. 78, pp. 202–210, 2017.
- [88] L. Shi, P. Cheng, and J. Chen, "Optimal periodic sensor scheduling with limited resources", *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2190–2195, 2011.
- [89] L. Shi and H. Zhang, "Scheduling two Gauss–Markov systems: An optimal solution for remote state estimation under bandwidth constraint", *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 2038–2042, 2012.
- [90] D. Shi and T. Chen, "Approximate optimal periodic scheduling of multiple sensors with constraints", *Automatica*, vol. 49, no. 4, pp. 993–1000, 2013.
- [91] D. Han, J. Wu, H. Zhang, and L. Shi, "Optimal sensor scheduling for multiple linear dynamical systems", *Automatica*, vol. 75, pp. 260–270, 2017.
- [92] S. Wu, X. Ren, S. Dey, and L. Shi, "Optimal scheduling of multiple sensors over shared channels with packet transmission constraint", *Automatica*, vol. 96, pp. 22–31, 2018.
- [93] S. Trimpe and R. D’Andrea, "Event-based state estimation with variance-based triggering", *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3266–3281, 2014.
- [94] A. S. Leong, S. Dey, and D. E. Quevedo, "Sensor scheduling in variance based event triggered estimation with packet drops", *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1880–1895, 2017.
- [95] J. Sijs and M. Lazar, "Event based state estimation with time synchronous updates", *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2650–2655, 2012.
- [96] J. Wu, Q. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality", *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.

- [97] L. He, J. Chen, and Y. Qi, "Event-based state estimation: Optimal algorithm with generalized closed skew normal distribution", *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 321–328, 2018.
- [98] J. Wu, K. H. Johansson, and L. Shi, "A stochastic online sensor scheduler for remote state estimation with time-out condition", *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3110–3116, 2014.
- [99] D. Han, Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, and L. Shi, "Stochastic event-triggered sensor schedule for remote state estimation", *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2661–2675, 2015.
- [100] S. Weerakkody, Y. Mo, B. Sinopoli, D. Han, and L. Shi, "Multi-sensor scheduling for state estimation with event-based, stochastic triggers", *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2695–2701, 2016.
- [101] M. Xia, V. Gupta, and P. J. Antsaklis, "Networked state estimation over a shared communication medium", *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1729–1741, 2017.
- [102] D. Shi, T. Chen, and M. Darouach, "Event-based state estimation of linear dynamic systems with unknown exogenous inputs", *Automatica*, vol. 69, pp. 275–288, 2016.
- [103] A. Molin, H. Esen, and K. H. Johansson, "Scheduling networked state estimators based on value of information", *Automatica*, vol. 110, 2019.
- [104] A. S. Leong, A. Ramaswamy, D. E. Quevedo, H. Karl, and L. Shi, "Deep reinforcement learning for wireless sensor scheduling in cyber-physical systems", *Automatica*, vol. 113, 2020.
- [105] S. Wu, X. Ren, Q.-S. Jia, K. H. Johansson, and L. Shi, "Learning optimal scheduling policy for remote state estimation under uncertain channel condition", *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 579–591, 2019.
- [106] A. S. Leong, S. Dey, G. N. Nair, and P. Sharma, "Power allocation for outage minimization in state estimation over fading channels", *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3382–3397, 2011.
- [107] A. S. Leong and S. Dey, "Power allocation for error covariance minimization in Kalman filtering over packet dropping links", in *Proceedings of IEEE Conference on Decision and Control*, 2012, pp. 3335–3340.

- [108] D. E. Quevedo, A. Ahlén, A. S. Leong, and S. Dey, “On kalman filtering over fading wireless channels with controlled transmission powers”, *Automatica*, vol. 48, no. 7, pp. 1306–1316, 2012.
- [109] X. Ren, J. Wu, K. H. Johansson, G. Shi, and L. Shi, “Infinite horizon optimal transmission power control for remote state estimation over fading channels”, *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 85–100, 2018.
- [110] T. Soleymani, S. Zoppi, M. Vilgelm, S. Hirche, W. Kellerer, and J. S. Baras, “Covariance-based transmission power control for estimation over wireless sensor networks”, in *Proceedings of European Control Conference*, 2018.
- [111] A. Nayyar, T. Başar, D. Teneketzis, and V. V. Veeravalli, “Optimal strategies for communication and remote estimation with an energy harvesting sensor”, *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2246–2260, 2013.
- [112] M. Nourian, A. S. Leong, and S. Dey, “Optimal energy allocation for kalman filtering over packet dropping links with imperfect acknowledgments and energy harvesting constraints”, *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2128–2143, 2014.
- [113] S. Knorn, S. Dey, A. Ahlén, and D. E. Quevedo, “Optimal energy allocation in multi sensor estimation over wireless channels using energy harvesting and sharing”, *IEEE Transactions on Automatic Control*, 2019.
- [114] D. E. Quevedo, A. Ahlén, and K. H. Johansson, “State estimation over sensor networks with correlated wireless fading channels”, *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 581–593, 2013.
- [115] A. S. Leong, D. E. Quevedo, A. Ahlén, and K. H. Johansson, “On network topology reconfiguration for remote state estimation.”, *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3842–3856, 2016.
- [116] K. J. Åström and B. Wittenmark, *Computer-controlled Systems: Theory and Design*. Prentice-Hall, 2013.
- [117] D. Nešić and A. R. Teel, “Input-output stability properties of networked control systems”, *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1650–1667, 2004.

- [118] M. Tabbara, D. Nešić, and A. R. Teel, “Stability of wireless and wireline networked control systems”, *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1615–1630, 2007.
- [119] P. Naghshtabrizi, J. P. Hespanha, and A. R. Teel, “Exponential stability of impulsive systems with application to uncertain sampled-data systems”, *Systems & Control Letters*, vol. 57, no. 5, pp. 378–385, 2008.
- [120] D. Nešić and D. Liberzon, “A unified framework for design and analysis of networked and quantized control systems”, *IEEE Transactions on Automatic control*, vol. 54, no. 4, pp. 732–747, 2009.
- [121] G. C. Walsh, H. Ye, and L. G. Bushnell, “Stability analysis of networked control systems”, *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.
- [122] M. H. Mamduhi and S. Hirche, “Try-once-discard scheduling for stochastic networked control systems”, *International Journal of Control*, vol. 92, no. 11, pp. 2532–2546, 2019.
- [123] E. Fridman, A. Seuret, and J.-P. Richard, “Robust sampled-data stabilization of linear systems: An input delay approach”, *Automatica*, vol. 40, no. 8, pp. 1441–1446, 2004.
- [124] E. Fridman, “A refined input delay approach to sampled-data control”, *Automatica*, vol. 46, no. 2, pp. 421–427, 2010.
- [125] K. Liu and E. Fridman, “Wirtinger’s inequality and Lyapunov-based sampled-data stabilization”, *Automatica*, vol. 48, no. 1, pp. 102–108, 2012.
- [126] A. Selivanov and E. Fridman, “Observer-based input-to-state stabilization of networked control systems with large uncertain delays”, *Automatica*, vol. 74, pp. 63–70, 2016.
- [127] K. Liu, E. Fridman, and L. Hetel, “Stability and  $L_2$ -gain analysis of networked control systems under Round-Robin scheduling: A time-delay approach”, *Systems & Control Letters*, vol. 61, no. 5, pp. 666–675, 2012.
- [128] K. Liu and E. Fridman, “Networked-based stabilization via discontinuous Lyapunov functionals”, *International Journal of Robust and Nonlinear Control*, vol. 22, no. 4, pp. 420–436, 2012.

- [129] K. Liu, E. Fridman, and L. Hetel, “Networked control systems in the presence of scheduling protocols and communication delays”, *SIAM Journal on Control and Optimization*, vol. 53, no. 4, pp. 1768–1788, 2015.
- [130] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control”, in *Proceedings of IEEE Conference on Decision and Control*, 2012, pp. 3270–3285.
- [131] K. Liu, V. Suplin, and E. Fridman, “Stability of linear systems with general sawtooth delay”, *IMA Journal of Mathematical Control and Information*, vol. 27, no. 4, pp. 419–436, 2010.
- [132] V. Gupta, B. Hassibi, and R. M. Murray, “Optimal LQG control across packet-dropping links”, *Systems & Control Letters*, vol. 56, no. 6, pp. 439–446, 2007.
- [133] V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi, “Data transmission over networks for estimation and control”, *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1807–1819, 2009.
- [134] V. Gupta and N. C. Martins, “On stability in the presence of analog erasure channel between the controller and the actuator”, *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 175–179, 2010.
- [135] D. Hristu-Varsakelis and L. Zhang, “LQG control of networked control systems with access constraints and delays”, *International Journal of Control*, vol. 81, no. 8, pp. 1266–1280, 2008.
- [136] G. N. Nair and R. J. Evans, “Stabilizability of stochastic linear systems with finite feedback data rates”, *SIAM Journal on Control and Optimization*, vol. 43, no. 2, pp. 413–436, 2004.
- [137] S. Tatikonda and S. Mitter, “Control under communication constraints”, *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1056–1068, 2004.
- [138] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, “Feedback control under data rate constraints: An overview”, *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.
- [139] T. Tanaka, P. M. Esfahani, and S. K. Mitter, “LQG control with minimum directed information: Semidefinite programming approach”, *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 37–52, 2018.

- [140] V. Causevic, P. U. Abara, and S. Hirche, "Information-constrained optimal control of distributed systems with power constraints", in *Proceedings of European Control Conference*, 2018, pp. 1505–1510.
- [141] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems", *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 470–474, 2013.
- [142] B. Demirel, V. Gupta, D. E. Quevedo, and M. Johansson, "On the trade-off between communication and control cost in event-triggered dead-beat control", *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2973–2980, 2017.
- [143] A. S. Leong, D. E. Quevedo, T. Tanaka, S. Dey, and A. Ahlén, "Event-based transmission scheduling and LQG control over a packet dropping link", in *Proceedings of IFAC World Congress*, 2017, pp. 8945–8950.
- [144] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "LQG control and sensing co-design", *IEEE Transactions on Automatic Control*, 2020.
- [145] M. Pezzutto, F. Tramarin, S. Dey, and L. Schenato, "Adaptive transmission rate for LQG control over Wi-Fi: A cross-layer approach", *Automatica*, vol. 119, 2020.
- [146] D. Maity, M. H. Mamduhi, S. Hirche, K. H. Johansson, and J. S. Baras, "Optimal LQG control under delay-dependent costly information", *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 102–107, 2019.
- [147] A. Molin and S. Hirche, "Price-based adaptive scheduling in multi-loop control systems with resource constraints", *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3282–3295, 2014.
- [148] C. Ramesh, H. Sandberg, and K. H. Johansson, "Design of state-based schedulers for a network of control loops", *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 1962–1975, 2013.
- [149] M. H. Mamduhi, A. Molin, D. Tolić, and S. Hirche, "Error-dependent data scheduling in resource-aware multi-loop networked control systems", *Automatica*, vol. 81, pp. 209–216, 2017.
- [150] S. M. Asghari, Y. Ouyang, and A. Nayyar, "Optimal local and remote controllers with unreliable uplink channels", *IEEE Transactions on Automatic Control*, vol. 64, no. 5, pp. 1816–1831, 2019.

- [151] M. H. Mamduhi, D. Maity, J. S. Baras, and K. H. Johansson, "A cross-layer optimal co-design of control and networking in time-sensitive cyber-physical systems", *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 917–922, 2020.
- [152] B. Demirel, A. Ramaswamy, D. E. Quevedo, and H. Karl, "DeepCAS: A deep reinforcement learning algorithm for control-aware scheduling", *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 737–742, 2018.
- [153] A. Redder, A. Ramaswamy, and D. E. Quevedo, "Deep reinforcement learning for scheduling in large-scale networked control systems", in *Proceedings of IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2019.
- [154] R. Alur, A. d’Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, "Compositional modeling and analysis of multi-hop control networks", *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2345–2357, 2011.
- [155] B. Demirel, Z. Zou, P. Soldati, and M. Johansson, "Modular design of jointly optimal controllers and forwarding policies for wireless control", *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3252–3265, 2014.
- [156] M. Klügel, M. H. Mamduhi, O. Ayan, M. Vilgelm, K. H. Johansson, S. Hirche, and W. Kellerer, "Joint cross-layer optimization in real-time networked control systems", *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1903–1915, 2020.
- [157] F. Smarra, A. D’Innocenzo, and M. D. Di Benedetto, "Optimal co-design of control, scheduling and routing in multi-hop control networks", in *Proceedings of IEEE Conference on Decision and Control*, 2012, pp. 1960–1965.
- [158] A. D’Innocenzo, M. D. Di Benedetto, and E. Serra, "Fault tolerant control of multi-hop control networks", *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1377–1389, 2013.
- [159] A. D’Innocenzo, F. Smarra, and M. D. Di Benedetto, "Resilient stabilization of multi-hop control networks subject to malicious attacks", *Automatica*, vol. 71, pp. 1–9, 2016.

- [160] F. Smarra, A. D’Innocenzo, and M. D. Di Benedetto, “Approximation methods for optimal network coding in a multi-hop control network with packet losses”, in *Proceedings of European Control Conference*, 2015, pp. 1962–1967.
- [161] D. E. Quevedo, K. H. Johansson, A. Ahlén, and I. Jurado, “Adaptive controller placement for wireless sensor–actuator networks with erasure channels”, *Automatica*, vol. 49, no. 11, pp. 3458–3466, 2013.
- [162] J. Araújo, Y. Ariba, P. Park, H. Sandberg, and K. H. Johansson, “Control over a hybrid MAC wireless network”, in *Proceedings of IEEE International Conference on Smart Grid Communications*, 2010, pp. 197–202.
- [163] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, “System architectures, protocols and algorithms for aperiodic wireless control systems”, *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 175–184, 2014.
- [164] B. Wu, M. D. Lemmon, and H. Lin, “Formal methods for stability analysis of networked control systems with IEEE 802.15. 4 protocol”, *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1635–1645, 2018.
- [165] C. Ramesh, H. Sandberg, and K. H. Johansson, “Multiple access with attention-based tournaments for monitoring over wireless networks”, in *Proceedings of European Control Conference*, 2009, pp. 4302–4307.
- [166] —, “Steady state performance analysis of multiple state-based schedulers with CSMA”, in *Proceedings of IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 4729–4734.
- [167] —, “Performance analysis of a network of event-based systems”, *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3568–3573, 2016.
- [168] A. I. Maass, D. Nešić, R. Postoyan, and P. M. Dower, “ $\mathcal{L}_p$  stability of networked control systems implemented on WirelessHART”, *Automatica*, vol. 109, 2019.
- [169] A. I. Maass and D. Nešić, “Stabilization of non-linear networked control systems closed over a lossy WirelessHART network”, *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 996–1001, 2019.

- [170] A. I. Maass, D. Nešić, R. Postoyan, and P. M. Dower, “Observer design for non-linear networked control systems with persistently exciting protocols”, *IEEE Transactions on Automatic Control*, vol. 65, no. 7, pp. 2992–3006, 2019.
- [171] G. D. Di Girolamo and A. D’Innocenzo, “Codesign of controller, routing and scheduling in WirelessHART networked control systems”, *International Journal of Robust and Nonlinear Control*, vol. 29, no. 7, pp. 2171–2187, 2019.
- [172] D. Baumann, F. Mager, R. Jacob, L. Thiele, M. Zimmerling, and S. Trimpe, “Fast feedback control over multi-hop wireless networks with mode changes and stability guarantees”, *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 2, pp. 1–32, 2019.
- [173] D. Baumann, F. Mager, M. Zimmerling, and S. Trimpe, “Control-guided communication: Efficient resource arbitration and allocation in multi-hop wireless control systems”, *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 127–132, 2020.
- [174] K. J. Åström and B. Bernhardsson, “Comparison of periodic and event based sampling for first-order stochastic systems”, in *Proceedings of IFAC World Congress*, vol. 11, 1999, pp. 301–306.
- [175] K.-E. Årzén, “A simple event-based PID controller”, in *Proceedings of IFAC World Congress*, vol. 18, 1999, pp. 423–428.
- [176] M. Rabi and K. H. Johansson, “Event-triggered strategies for industrial control over wireless networks”, in *Proceedings of Wireless Internet Conference*, 2008, pp. 1–7.
- [177] A. Cervin and K. J. Aström, “On limit cycles in event-based control systems”, in *Proceedings of IEEE Conference on Decision and Control*, 2007, pp. 3190–3195.
- [178] W. P.M. H. Heemels, J. H. Sandee, and P. P. J. van den Bosch, “Analysis of event-driven controllers for linear systems”, *International Journal of Control*, vol. 81, no. 4, pp. 571–590, 2008.
- [179] J. Lunze and D. Lehmann, “A state-feedback approach to event-based control”, *Automatica*, vol. 46, no. 1, pp. 211–215, 2010.
- [180] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks”, *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

- [181] M. C. F. Donkers and W. P.M. H. Heemels, “Output-based event-triggered control with guaranteed  $\mathcal{L}_\infty$ -gain and improved and decentralized event-triggering”, *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1362–1376, 2012.
- [182] A. Anta and P. Tabuada, “To sample or not to sample: Self-triggered control for nonlinear systems”, *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.
- [183] W. P.M. H. Heemels and M. C. F. Donkers, “Model-based periodic event-triggered control for linear systems”, *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.
- [184] W. P.M. H. Heemels, M. C. F. Donkers, and A. R. Teel, “Periodic event-triggered control for linear systems”, *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, 2012.
- [185] W. Wu, S. Reimann, D. Görges, and S. Liu, “Suboptimal event-triggered control for time-delayed linear systems”, *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1386–1391, 2015.
- [186] A. Selivanov and E. Fridman, “Sampled-data implementation of derivative-dependent control using artificial delays”, *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3594–3600, 2018.
- [187] —, “Event-triggered  $H_\infty$  control: A switching approach”, *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3221–3226, 2016.
- [188] D. Lehmann and J. Lunze, “Event-based output-feedback control”, in *Proceedings of Mediterranean Conference Control and Automation*, 2011, pp. 982–987.
- [189] E. Garcia and P. J. Antsaklis, “Event-triggered output feedback stabilization of networked systems with external disturbance”, in *Proceedings of IEEE Conference on Decision and Control*, 2014, pp. 3566–3571.
- [190] S. Tarbouriech, A. Seuret, J. M. Gomes da Silva Jr., and D. Sbarbaro, “Observer-based event-triggered control co-design for linear systems”, *IET Control Theory & Applications*, vol. 10, no. 18, pp. 2466–2473, 2016.
- [191] X. Wang, H. Yu, and F. Hao, “Observer-based disturbance rejection for linear systems by aperiodical sampling control”, *IET Control Theory & Applications*, vol. 11, no. 10, pp. 1561–1570, 2017.

- [192] L. G. Moreira, J. M. Gomes da Silva Jr., S. Tarbouriech, and A. Seuret, "Observer-based event-triggered control for systems with slope-restricted nonlinearities", *International Journal of Robust and Nonlinear Control*, vol. 30, no. 17, pp. 7409–7428, 2020.
- [193] S. Durand and N. Marchand, "Further results on event-based PID controller", in *Proceedings of European Control Conference*, 2009, pp. 1979–1984.
- [194] U. Tiberi, J. Araújo, and K. H. Johansson, "On event-based PI control of first-order processes", in *Proceedings of IFAC Conference on Advances in PID Control*, 2012, pp. 448–453.
- [195] G. A. Kiener, D. Lehmann, and K. H. Johansson, "Actuator saturation and anti-windup compensation in event-triggered control", *Discrete Event Dynamic Systems*, vol. 24, no. 2, pp. 173–197, 2014.
- [196] J. Song, A. K. Mok, D. Chen, M. Nixon, T. Blevins, and W. Wojsznis, "Improving PID control with unreliable communications", in *Proceedings of ISA EXPO Technical Conference*, 2006, pp. 17–19.
- [197] O. Kaltiokallio, L. M. Eriksson, and M. Bocca, "On the performance of the PIDPLUS controller in wireless control systems", in *Proceedings of Mediterranean Conference Control and Automation*, 2010, pp. 707–714.
- [198] T. Blevins, M. Nixon, and W. Wojsznis, "PID control using wireless measurements", in *Proceedings of American Control Conference*, 2014, pp. 790–795.
- [199] D. Lehmann, G. A. Kiener, and K. H. Johansson, "Event-triggered PI control: Saturating actuators and anti-windup compensation", in *Proceedings of IEEE Conference on Decision and Control*, 2012, pp. 6566–6571.
- [200] D. Lehmann and K. H. Johansson, "Event-triggered PI control subject to actuator saturation", in *Proceedings of IFAC Conference on Advances in PID Control*, 2012, pp. 430–435.
- [201] L. G. Moreira, L. B. Groff, J. M. Gomes da Silva Jr., and S. Tarbouriech, "PI event-triggered control under saturating actuators", *International Journal of Control*, vol. 92, no. 7, pp. 1634–1644, 2019.
- [202] A. Seuret, C. Prieur, S. Tarbouriech, and L. Zaccarian, "LQ-based event-triggered controller co-design for saturated linear systems", *Automatica*, vol. 74, pp. 47–54, 2016.

- [203] D. Liu and G.-H. Yang, “Event-triggered control for linear systems with actuator saturation and disturbances”, *IET Control Theory & Applications*, vol. 11, no. 9, pp. 1351–1359, 2017.
- [204] S. Tarbouriech, G. Garcia, J. M. Gomes da Silva Jr., and I. Queinnec, *Stability and Stabilization of Linear Systems with Saturating Actuators*. Springer, 2011.
- [205] U. Tiberi, C.-F. Lindberg, and A. J. Isaksson, “Dead-band self-triggered PI control for processes with dead-time”, in *Proceedings of IFAC Conference on Advances in PID Control*, 2012, pp. 442–447.
- [206] J. M. Gomes da Silva Jr., W. F. Lages, and D. Sbarbaro, “Event-triggered PI control design.”, in *Proceedings of IFAC World Congress*, 2014, pp. 6947–6952.
- [207] S. Reimann, D. H. Van, S. Al-Areqi, and S. Liu, “Stability analysis and PI control synthesis under event-triggered communication”, in *Proceedings of European Control Conference*, 2015, pp. 2174–2179.
- [208] T. Blevins, D. Chen, M. Nixon, and W. Wojsznis, *Wireless Control Foundation: Continuous and Discrete Control for the Process Industry*. International Society of Automation, 2015.
- [209] D. Lehmann and J. Lunze, “Extension and experimental evaluation of an event-based state-feedback approach”, *Control Engineering Practice*, vol. 19, no. 2, pp. 101–112, 2011.
- [210] M. Beschi, S. Dormido, J. Sanchez, and A. Visioli, “Characterization of symmetric send-on-delta PI controllers”, *Journal of Process Control*, vol. 22, no. 10, pp. 1930–1945, 2012.
- [211] S. Reimann, W. Wu, and S. Liu, “PI control and scheduling design for embedded control systems”, in *Proceedings of IFAC World Congress*, 2014, pp. 11 111–11 116.
- [212] T. Norgren, J. Styrud, A. J. Isaksson, J. Åkerberg, and T. Lindh, “Industrial evaluation of process control using non-periodic sampling”, in *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, 2012, pp. 1–8.
- [213] C. F. Lindberg and A. J. Isaksson, “Comparison of different sampling schemes for wireless control subject to packet losses”, in *Proceedings of International Conference on Event-Based Control, Communication and Signal Processing*, 2015, pp. 1–8.

- [214] A. Cervin, “LQG-optimal PI and PID control as benchmarks for event-based control”, in *Proceedings of International Conference on Event-Based Control, Communication and Signal Processing*, 2016, pp. 1–8.
- [215] A. Cervin and M. T. Andr en, “LQG-optimal versus simple event-based PID controllers”, in *Proceedings of American Control Conference*, 2020, pp. 3678–3684.
- [216] L. I. Sennott, *Stochastic Dynamic Programming and the Control of Queueing Systems*. John Wiley & Sons, 2009.
- [217] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volume I*. Athena Scientific, 2017.
- [218] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2005.
- [219] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Springer Science & Business Media, 2003, vol. 24.
- [220] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.
- [221] K. J.  str m and T. H gglund, *Advanced PID Control*. International Society of Automation, 2006.
- [222] J. L fberg, “YALMIP: A toolbox for modeling and optimization in matlab”, in *Proceedings of IEEE International Symposium on Computer-Aided Control System Design*, 2004, pp. 284–289.
- [223] E. Fridman, “Tutorial on Lyapunov-based methods for time-delay systems”, *European Journal of Control*, vol. 20, no. 6, pp. 271–283, 2014.
- [224] —, “New Lyapunov–Krasovskii functionals for stability of linear retarded and neutral type systems”, *Systems & Control Letters*, vol. 43, no. 4, pp. 309–319, 2001.
- [225] M. Najafi, S. Hosseinnia, F. Sheikholeslam, and M. Karimadini, “Closed-loop control of dead time systems via sequential sub-predictors”, *International Journal of Control*, vol. 86, no. 4, pp. 599–609, 2013.
- [226] V. L chapp , E. Moulay, and F. Plestan, “Prediction-based control for LTI systems with uncertain time-varying delays and partial state knowledge”, *International Journal of Control*, vol. 91, no. 6, pp. 1403–1414, 2018.

- [227] T. Iwaki, J. Wu, and K. H. Johansson, “Event-triggered feedforward control subject to actuator saturation for disturbance compensation”, in *Proceedings of European Control Conference*, 2018, pp. 501–506.



