

**CONTROL AND SCHEDULING FOR SYSTEMS AFFECTED BY
RANDOM PACKET LOSS**

Edwin G.W. Peters (MSc AAU, BSc AAU)

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy



**THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA**

School of Electrical Engineering and Computing

Supervision by: Prof. Minyue Fu, Dr. Damián Marelli and Prof. Daniel Quevedo

April, 2018

Statement of originality

I hereby certify that the work embodied in the thesis is my own work, conducted under normal supervision.

The thesis contains no material which has been accepted, or is being examined, for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to the final version of my thesis being made available worldwide when deposited in the University's Digital Repository, subject to the provisions of the Copyright Act 1968 and any approved embargo.

Edwin G.W. Peters

April, 2018

Abstract

Modern technologies in manufacturing, transportation, energy generation and distribution, smart buildings, smart cities and so on heavily involve large-scale, distributed networked control systems (NCSs). Nowadays, the communication between sub-systems within a NCS occurs mostly over wired networks. However, in large scale systems, the cost of installation and maintenance of these wired networks can be significant.

One method of cost reduction while also gaining flexibility, is to replace these wired networks with wireless networks. Utilizing wireless networks within NCSs allows for greater flexibility, since no new cables have to be installed when adding new sensors and/or actuators. Further, it opens up for the use of remote sensors and actuators in locations, that are prohibitive with cabled networks. These sensors and actuators can be battery powered and mounted on moving objects, such as vehicles or drones. This opens up for entirely new possibilities for control and sensing within NCSs.

However, wireless networks introduce network effects that can severely affect the performance of the sub-systems, and in some cases, lead to instability. These effects include, but are not limited to, congestion, interference, packet loss and bandwidth limitations. In this thesis, we address the controller and estimator design for NCSs that are connected with wireless networks. We show, that designs that take these network effects into account can not only achieve increased performance, but also guaranteed closed-loop stability.

The first part of this thesis considers the synthesis and analysis of controllers and estimators for networks affected by random packet loss. In particular, Chapter 3 considers controller and estimator synthesis and analysis for linear systems affected by independent and identically distributed packet loss. We establish a form of duality that extends the duality in the classical linear quadratic Gaussian result to the design for systems affected by packet loss. Chapter 4 extends the results from Chapter 3 and presents a method to synthesise controllers for networks where the packet loss model contains memory as well. This however results in a large number of controllers. To reduce the amount of controllers, we present three methods that trade-off controller complexity and control performance.

The second part of this thesis (Chapters 5 and 6) considers the control design problem for a large distributed system with a bandwidth limited wireless network. The wireless transmission protocol features a limited number of reliable transmission slots with negligible packet dropouts and a more widely available transmission period, where packet collisions and delays occur more

frequently. We propose a controller and scheduler co-design that optimally selects both the schedule on which actuators to address, and the control inputs for the sub-systems that are addressed. Simulation studies illustrate that the online optimal co-design method results in significantly improved performance over heuristic scheduling. However, the computational complexity for the online algorithms makes practical implementations of the proposed method prohibitive. To reduce the computational complexity, the design is extended by the use of a novel model predictive control (MPC) algorithm that combines approximations to infinite horizon cost functions with a short online prediction horizon. This results in improved control performance while maintaining relatively low computational complexity.

In this thesis, we show that taking networks effects, such as packet loss and bandwidth limitations, directly into account in the controller and estimator design phase leads to significant performance gains and in some cases, guaranteed closed-loop stability.

Acknowledgements

I would like to use this opportunity to express my gratitude to the people and professionals who supported me throughout my PhD studies. Their guidance, friendly advice and constructive criticism were invaluable to me during the pursuit of my goals.

First and foremost, I want to thank my supervisors, Damián Marelli, Minyue Fu and Daniel Quevedo for their constant support during my degree. Minyue always gave me the necessary freedom to conduct my research, while providing guidance when needed. I also want to thank him for offering me the opportunities to teach and tutor. I want to thank Damián for his unprecedented patience and advice, especially when solving difficult mathematical problems. Our fruitful discussions frequently went beyond, or far below, technical challenges and problems. I want to thank Daniel for inviting me to commence my PhD degree in Newcastle, and again for a visit in Germany. I further want to thank him for his guidance and ideas, especially during the first year of my studies.

I also want to use this opportunity to thank my former supervisor, Jan Østergaard. We can say that this journey started when you introduced me, and supported my visit, to Daniel during my Master degree. Further, I owe my thanks to Burak Demirel for our fruitful discussions during my short visit to Germany.

Special thanks to my many friends in Newcastle, you know who you are. I thank you for making my time in Newcastle memorable. I also thank my colleagues at the University for their support, suggestions and discussions, both technical and less technical.

I thank the University of Newcastle for the financial support that allowed me to come to Australia and pursue a PhD degree.

Last, but not least, I would like to mention my parents, Johan and Jose, my brothers Rick and Marco and my partner Sonja. Thank you for your love and support.

Notation

\triangleq	Definition
(\dots)	A sequence
$\{\dots\}$	A set
\mathbb{R}	The set of real numbers
\mathbb{R}^n	Vector with n elements taking values in the set \mathbb{R}
$\mathbb{R}^{n \times m}$	Matrix with n rows and m columns taking values in the set \mathbb{R}
\mathbb{N}	The set of natural numbers
\mathbb{Z}	The set of all integers
\mathbb{Z}_a	The set of integers $\{0, 1, \dots, a - 1\}$
$\text{card}(\mathcal{A})$	The cardinality of the set \mathcal{A}
$A > 0$ ($A < 0$)	The matrix A is positive (negative) definite
$A \geq 0$ ($A \leq 0$)	The matrix A is positive (negative) semi-definite
$A \geq B$	The matrix $A - B \geq 0$
A^T	Transpose of the matrix A
A^{-1}	The inverse of matrix A
A^\dagger	Pseudo inverse of matrix A
$\text{trace}(A)$	Trace of the square matrix A
$\sigma(A)$	Spectral radius of the matrix A
$\text{diag}\{X\}$	The matrix with X on its (block-)diagonal
I_n	The $n \times n$ matrix with ones on its diagonal
$\mathbf{0}_{n \times m}$	The $n \times m$ matrix containing zeros
a_i	The i 'th column vector in the matrix A
a_{ji}	The j 'th element in the i 'th column vector of the matrix A
$\ x\ $	Two-norm of x
$\ x\ _Q^2$	Weighted norm of x squared i.e. $x^T Q x$
$\lceil x \rceil$	Rounds x up to the nearest larger integer
$\lfloor x \rfloor$	Rounds x down to the nearest lower integer
$x \% N$	x modulo N returns the remainder after division for $\frac{x}{N}$
$\Pr\{X\}$	The probability for the event X
$\Pr\{X Y\}$	The probability for the event X knowing Y
$\Pr\{X; Z\}$	The probability for the event X which distribution is parameterized by Z

$\mathbf{E}\{X\}$	Expected value of the random variable X
$\mathbf{E}\{X Y\}$	Expected value of the random variable X knowing Y
$\mathbf{E}\{X; Z\}$	Expected value of the random variable X which distribution is parameterized by Z
$\mathbb{1}_{\mathcal{S}}(y)$	The indicator function returns one if $y \in \mathcal{S}$ or zero else
$\mathcal{U}(i, j)$	Uniform distribution in the interval $[i, j]$
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and (co-)variance Σ
$x \sim \mathcal{N}(\mu, \Sigma)$	The random variable x is Gaussian distributed
$x \sim \mathcal{U}(i, j)$	The random variable x is uniformly distributed
\otimes	The Kronecker product
$\text{vec}(A)$	The vector of the columns of A stacked i.e. $\begin{bmatrix} a_1^T & a_2^T & \dots & a_n^T \end{bmatrix}^T$
\circ	The composition of functions
$n!$	The factorial $n! = n(n-1)(n-2)\dots 1$

Abbreviations

BI	beacon interval
CA	cost averaged
CAP	contention access period
CFP	contention free period
CSMA	carrier sense multiple access
CSMA/CA	carrier sense multiple access with collision avoidance
FDMA	frequency division multiple access
flop	floating point operation
FSMC	finite state Markov chain
GA	group averaged
GTS	guaranteed time-slot
<i>i.i.d</i>	independent and identically distributed
JLS	jump linear system
LQ	linear quadratic
LQG	linear quadratic Gaussian
LQR	linear quadratic regulator
LTI	linear time-invariant
MA	Markov averaged
MJLS	Markov jump linear system
MMSE	minimum mean square error
MPC	model predictive control
MSS	mean square stability
NCS	networked control system
NP	non-deterministic polynomial-time
PAN	personal area network
RR	round robin
SDP	semi-definite program
TCP	transmission control protocol
TDMA	time division multiple access
UDP	user datagram protocol

Contents

Abstract	iii
Acknowledgements	v
Notation	vii
Abbreviations	ix
Contents	xi
1 Introduction	1
1.1 Wireless networks in NCSs	2
1.2 Outline and contributions	3
2 Background	7
2.1 Challenges in wireless networked control systems	7
2.2 Wireless networks for networked control systems	9
2.2.1 Industrial wireless networks	9
2.2.2 Packet loss modelling	10
2.3 Control and estimation over wireless networks	13
2.3.1 Control over wireless networks	13
2.3.2 Estimation over wireless networks	16
2.3.3 The separation principle	17
2.4 Summary	19
3 Duality for control and estimation for systems with i.i.d. packet losses	21
3.1 Introduction	21
3.2 Problem setting	22
3.2.1 Linear quadratic regulator	23
3.2.2 Dual offline estimator	26
3.2.3 A few notes on the separation principle	27
3.3 Stability results	28
3.3.1 Controller	28

3.3.2	Offline estimator	31
3.4	Convergence of the Riccati equation	34
3.5	Simulations	35
3.5.1	Controller	36
3.5.2	Estimator	36
3.6	Summary	37
Appendix	38
3.A	Proof of Proposition 3.2.1	38
3.B	Proof of Proposition 3.2.2	39
3.C	Proof of lemmas in Section 3.3	40
4	Control over networks with correlated packet losses	45
4.1	Introduction	45
4.2	Problem setting	47
4.3	Optimal predictive controller design	49
4.4	Sub-optimal reduced complexity controllers	52
4.4.1	Grouping the packet arrival sequences	53
4.4.2	Optimal controller for reduced Markov chain (MA)	54
4.4.3	Group averaged controller (GA)	55
4.4.4	Reducing the amount of controllers in the design (CA)	55
4.5	Simulation studies	57
4.5.1	DC servo	58
4.5.2	Mass spring system	59
4.6	Summary	60
Appendix	61
4.A	Proof of Proposition 4.3.1	61
4.B	Proof of Theorem 4.4.1	63
5	Scheduling for control over wireless networks	67
5.1	Introduction	67
5.2	Networked control setting	69
5.2.1	Network considerations	70
5.2.2	System setting	72
5.3	Stochastic control formulation	75
5.3.1	Solution to the minimization problem	76
5.3.2	Separation	77
5.3.3	Implementation	78
5.3.4	Numerical considerations	79
5.4	Simulations	80
5.4.1	Branch and bound algorithm	81
5.4.2	Numerical example	81
5.4.3	Performance of small coupled systems.	84
5.4.4	Larger system	86
5.5	Summary	87

Appendix	88
5.A Proof of Lemma 5.2.1	88
5.B Proof of Theorem 5.3.1	89
6 A periodic approach to scheduling and control co-design	93
6.1 Introduction	93
6.1.1 System setup	94
6.1.2 Periodic scheduling	95
6.1.3 A time-invariant formulation	96
6.1.4 A few notes on controllability	99
6.2 Finite horizon stochastic control	99
6.2.1 Solution to the minimization problem	100
6.2.2 Link between time-varying and LTI formulation	102
6.3 Infinite horizon approximation	103
6.3.1 Discounted cost formulation	104
6.3.2 Averaged infinite horizon	106
6.4 Implementation	108
6.4.1 Model predictive control (MPC) with final state weighting	108
6.4.2 Discounted MPC	114
6.4.3 Offline implementation	114
6.5 Simulation studies	114
6.5.1 The impact of the choice of β on the final state weighting	115
6.5.2 Comparison of the online and offline implementations	115
6.6 Summary	122
Appendix	122
6.A Proof of Theorem 6.2.1	122
6.B Proof of Lemma 6.3.1	124
6.C Proof of Theorem 6.3.2	124
6.D Proof of Lemma 6.3.3	124
6.E Proof of Theorem 6.3.4	125
6.F Proof of Theorem 6.3.5	126
6.G Proof of Theorem 6.4.1	127
6.H Proof of Lemma 6.4.2	131
6.I Proof of Theorem 6.4.3	132
7 Conclusions and future directions	135
7.1 Conclusions	135
7.2 Future work	137
7.2.1 Correlated packet dropouts	137
7.2.2 Scheduler and controller co-design	137
Bibliography	139

Introduction

Over the last half a century, significant advances have been made in digital technology, both within our homes and the industry. Particularly, the continuous advances within sensing, actuation, communication and computation have paved the way to utilize automation and control within not only the industry, but also in peoples homes and utilities.

The inputs to the actuators of control systems have evolved from mechanically or hydraulically controlled mechanisms to electronically controlled mechanisms. While the controllers that provide the actuator inputs initially were located at the physical system, they now move further away. This opened up for new methods of control, such as centralized and distributed control methods, where systems are interconnected through large-scale networks over which systems and controllers can communicate. Applications range from complex industrial plants, to the control systems in automobiles as well as the financial sector.

Since the 80's these networked control systems (NCSs) communicated over wires by utilizing protocols such as Fieldbus, HART and CAN [74]. This, however, leads to increasing costs and complexity during the installation and maintenance of the NCSs, as they become larger scale. A significant part of this cost is found in the wiring of the NCSs, which can amount to 100s or 1000s of USD per meter [2], [18], [96]. For example, the NCS in the Airbus A380 airplane, contains 530 km of wires with 40 000 connectors [39]. Additionally, cables, connectors, routers and switches in NCSs tend to wear over time. The maintenance and replacement of these component can amount to significant costs over the lifetime of the NCS [67].

In recent years, significant advances within wireless communication mediums resulted in a large scale market penetration within homes, commercial applications and wearables. The introduction of industrial wireless standards such as WirelessHART [30], ISA100 [46] and, to a certain extend, ZigBee PRO [133], introduced the possibility to adopt wireless networks within NCSs. Research of these new possibilities to utilize wireless networks within NCSs has accelerated significantly within the last decade [2], [20], [132]. However, industrial adoption of wireless networks within NCSs has yet to see a similar development. The reason for the merely slow adoption of wireless networks for the use in NCSs, is found in numerous problems and challenges that currently are not fully addressed. This includes challenges such as more varying transmission delays and robustness to interference from the environment, but also security and reduced bandwidth compared to wired counterparts. However, also new possibilities arise when using wireless networks, such as the use of battery powered wireless remote sensors and actuators. These can even be utilized to monitor

and control moving objects, such as cranes and forklifts within the NCS [1], [34]. Researchers from different fields are attempting to address these challenges with analytical frameworks for the analysis and synthesis of various aspects for utilizing wireless networks within NCSs. This amounts to dealing with disturbances and delays on the network, as well as bandwidth limitations and/or power constraints. Also, cyber-security within NCS design is receiving increasing attention [97].

In the remainder of this chapter, we clarify some of the benefits and open challenges for the use of wireless networks in NCSs. This is followed by the presentation of the thesis outline and our contributions to the field.

1.1 Wireless networks in NCSs

Systems on modern industry floors are getting more and more interconnected. Conventionally, control of the systems is performed over cabled networks through for example Fieldbus, CAN bus or HART. The expenses related to the use of wires are however significant [2], [18], [96]. With the quality of digital sensors improving while prices tend to decrease, the cabling and routing within wired NCSs amount to a greater part of the total implementation costs [2], [18]. Further, the increasing availability of low-power, wireless digital sensors and actuators [2], [18], provides a significant financial motivation to move from using wired to use wireless networking in NCS implementations. In addition, the use of relatively cheap wireless sensors provides the opportunity to measure information that was not feasible or possible to obtain using wired networks [25]. This additional information can be utilized to design more advanced and adaptive control algorithms, and to include fault detection and maintenance scheduling to reduce breakdowns and downtime. This opens up for new levels of flexibility that were not available before, such as the control of mobile autonomous vehicles or formation control of drones [1], [22], [34]. Further, the use of wireless networks in NCSs can improve the overall reliability of the NCS, since the risk of broken connectors and wires is greatly reduced [67], [96].

Wireless technology has improved significantly over recent years with the availability of low-power, low-latency protocols such as WirelessHART, ISA100 and to some extent ZigBee PRO. The former two protocols support reliable communications by the use of time division multiple access (TDMA), where users can get transmission slots at fixed time intervals. This can for example be used for periodic control and sensing. In addition to the reliable communication channel, sporadic and less time-critical communications can be performed using carrier sense multiple access with collision avoidance (CSMA/CA). The latter can also be utilized for process monitoring or control when the subsystems in a NCS have slow dynamics. For these reasons, we propose methods that perform control and estimation over wireless, rather than wired, networks.

Open challenges however limit wide adoption of wireless networks within large scale industrial NCS settings. These challenges include less predictable delays, especially in multi-hop networks, and increased risks for packet losses and collisions compared to the wired counterparts. Many researchers propose frameworks to characterize, analyze and design controllers and estimators for the use within wireless NCSs affected by packet loss and delays [13], [29], [41], [95], [113], [127], [130], [132]. While in traditional control systems sampling tends to occur periodically, NCSs do not necessarily utilize periodic sampling. This can be caused by random delays [29], or be actively implemented as a strategy to reduce network usage and preserve power on remote sensing and

actuation devices [5], [9], [38], [64]–[66], [71], [87], [103], [112], [118].

Another possible limitation is, that the bandwidth of wireless networks is significantly reduced compared to the wired counterparts. This limits the amount of information that can be received from sensors or commands that can be transmitted to actuators. Therefore, proper prioritizing and scheduling of the data may be required [9], [32], [42], [59], [92], [93], [111], [114], [120], [131].

Security concerns are another cause for the reluctance that the industry shows in the adoption of wireless networks in NCSs. While even NCSs that are connected through wired networks can be vulnerable to attacks [97], [110], it is extremely challenging to properly isolate a wireless network. Especially the well-known Stuxnet virus, that infected industrial NCSs, demonstrated the need for more research into cyber-security of NCSs [97].

We believe that wider penetration of the use of wireless networks in industrial NCSs will occur once the above issues have been addressed. The challenge is however, that reliable control and estimation over wireless NCSs requires the integration of aspects from widely different fields of research. This includes control, all layers of the network stack from the physical to application layers as well as software and network security. In this thesis, we will view this problem from a control design perspective, where the main focus is on control performance and robustness for NCSs that utilize unreliable and bandwidth limited wireless network links.

1.2 Outline and contributions

This section outlines the remainder of this thesis and highlights the novel contributions by the author.

Chapter 2: Background

This chapter presents relevant background on industrial wireless networks and the modelling of packet loss. This is followed by the state of the art of control and estimation on wireless NCSs with random packet loss. The chapter concludes with a brief review of stabilizability and separation for controller and estimator design in wireless NCSs.

Chapter 3: Duality for control and estimation for systems with i.i.d. packet losses

In this chapter, we establish a form of duality between offline estimator design and optimal control for networks that are affected by random independent and identically distributed (*i.i.d.*) packet loss. This result extends the duality that exists in the classical optimal controller and Kalman filter design to systems that are affected by random packet loss. The statistics of the network are directly taken into account in the design phase, which allows for more robustness against packet loss. The result in this chapter allows the optimal offline estimator design by solving a dual control problem. Further, we show that stochastic stability of the dual system is equivalent to stochastic stability of the original system. The chapter concludes with simulation studies to verify the designs.

This chapter is based on the results that are presented in the work:

- E. G. W. Peters, D. Marelli, M. Fu, and D. E. Quevedo, “Unified approach to controller and MMSE estimator design with intermittent communications,” in *55th IEEE conference on decision and control, CDC, Las Vegas, NV, USA, 2016*, pp. 6832–6837.

Chapter 4: Control over networks with correlated packet losses

This chapter investigates controller design for more complex correlated network models. Unlike *i.i.d* network models, these more complex models can capture memory effects on networks. Taking these models directly into account in the controller design allows for significantly improved control performance compared to *i.i.d* models. However, the optimal control laws become increasingly complex. To reduce the controller complexity, we propose three methods that trade-off controller complexity and performance. Simulation studies show, that notable performance gains can be achieved over simple *i.i.d* network models.

This chapter contains the results that are presented in the works:

- E. G. W. Peters, D. Marelli, D. E. Quevedo, and M. Fu, “Controller design for networked control systems affected by correlated packet losses,” in *IFAC world congress, Toulouse, France*, 2017,
- E. G. W. Peters, D. Marelli, D. E. Quevedo, and M. Fu, “Predictive control for networked systems affected by correlated packet loss,” *International journal of robust and nonlinear control*, 2017, Early access for Special Issue on Stochastic Predictive Control.

Chapter 5: Scheduling for control over wireless networks

In this chapter, we present a controller and scheduler co-design for a NCS that is interconnected by a shared wireless network. The network is based on the IEEE 802.15.4 protocol and is affected by random packet loss. Since the network is shared, the bandwidth that can be used for the NCS is limited. We propose a co-design method that utilizes the available bandwidth in both the more reliable and less reliable, but more accessible, parts of the transmission frame of the IEEE 802.15.4. It shows, that the optimal co-design can be split in a separate controller and scheduler phase. A model predictive control (MPC) implementation is proposed. Simulation studies that utilize the proposed design show significant performance gains over heuristic scheduling.

This chapter incorporates results from the works:

- E. G. W. Peters, D. E. Quevedo, and M. Fu, “Co-design for control and scheduling over wireless industrial control networks,” in *54th IEEE conference on decision and control, CDC, Osaka, Japan*, 2015, pp. 2459–2464,
- E. G. W. Peters, D. E. Quevedo, and M. Fu, “Controller and scheduler codesign for feedback control over IEEE 802.15.4 networks,” *IEEE transactions on control systems technology*, vol. 24, no. 6, pp. 2016–2030, 2016.

Chapter 6: A periodic approach to scheduling and control co-design

This chapter extends the results from Chapter 5. We propose a periodic framework which makes it possible to approximate the infinite horizon co-design problem. This leads to a number of methods which can greatly improve the performance of the MPC implementation that is presented in Chapter 5 while reducing the computational cost. Further, one of the proposed methods leads to closed-loop stability. Additionally, the solutions to the infinite horizon co-design problem can be

used for the design of offline periodic schedules that result in closed loop stability. The performance of the proposed algorithms is investigated by the use of simulation studies.

This chapter is based on and extends the results of the work:

- E. G. W. Peters, D. E. Quevedo, and M. Fu, “Controller and scheduler codesign for feedback control over IEEE 802.15.4 networks,” *IEEE transactions on control systems technology*, vol. 24, no. 6, pp. 2016–2030, 2016.

Other publications

The following publications are not part of this thesis, but inspired some of the presented work:

- E. Henriksson, D. E. Quevedo, E. G. W. Peters, H. Sandberg, and K. H. Johansson, “Multiple-loop self-triggered model predictive control for network scheduling and control,” *IEEE transactions on control systems technology*, vol. 23, no. 6, pp. 2167–2181, 2015,
- E. G. W. Peters, D. E. Quevedo, and J. Østergaard, “Shaped Gaussian dictionaries for quantized networked control systems with correlated dropouts,” *IEEE transactions on signal processing*, vol. 64, no. 1, pp. 203–213, 2016.

Background

Networked control systems (NCSs) are large, spatially distributed systems that utilize shared communication networks to exchange information, such as sensor measurements and commands for e.g. actuators. In the last decades, spatially distributed NCSs have gained increasing attention [13], [41]. While initially NCSs utilized wired communication channels, recent advances in wireless communication technologies have opened up for the wide-scale development and utilization of wireless communication within NCSs. The use of wireless communication within NCS architectures allows for greater flexibility and reduced wiring as well as maintenance costs. This is evident in the penetration of wireless communication within NCS applications in the industry, power grids, and the smart home. Examples on this can be found in the works [73], [75], [76] and references therein.

This section begins with a presentation of a number of challenges that arise when implementing wireless networks within NCSs. This is followed by an introduction of wireless network protocols that are designed to be used within NCSs. Going slightly more in dept, we present a number of common methods on how to model packet loss on wireless networks. We conclude this section with a brief overview of the state of the art on controller and estimator design for systems that communicate over networks affected by packet loss.

2.1 Challenges in wireless networked control systems

Wireless NCSs provide numerous advantages over wired NCSs. However, when utilizing wireless networks in the NCS, some factors need to be considered during the controller and estimator design. Figure 2.1 illustrates a NCS where the controller is connected to a single or multiple plants, respectively, through a wireless network. In the NCSs in Figure 2.1, both sensor data and control commands or inputs are transmitted over a wireless communication network between the controller and plant. While wireless communication mediums do not require wires to interconnect the devices within the NCS, certain effects that only appear in very limited amounts on wired networks become more apparent and have to be considered in the design. These effects include:

- **Interference.** Both wired and wireless networks are affected by different disturbances that affect the performance and reliability of the channel [1], [28], [34], [129], [130]. The main disturbances consist of packet collisions and interference on the channel as well as fading

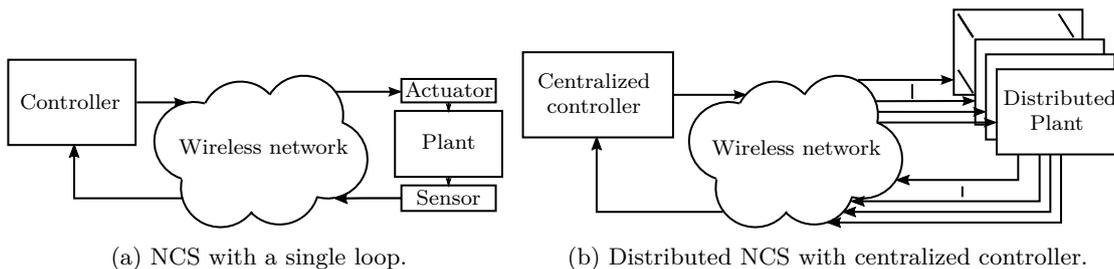


Figure 2.1: Two examples on a NCS. Figure 2.1a shows a single control loop where the controller communicates with the actuator and sensor of a plant over a wireless network. In Figure 2.1b, one centralized controller communicates with a distributed plant over a wireless network. This distributed plant has many actuators and sensors at different physical locations.

of the power of the radio wave as it propagates. This interference can be caused by other wireless networks or appliances that operate or produce noise on the channel frequencies as well as congestion when too many users transmit simultaneously. From a control design perspective, these effects can be modelled as follows:

- **Transmission delays.** While transmission delays over wired networks often could be predicted and remained fairly constant, the delays over wireless networks are much less predictable [29], [113], [129], [130]. This is caused by effects such as the transmission distance, congestion on the network, and interference from other sources, such as networks or appliances that produce noise in the frequency band which is utilized by the wireless network. Depending on the architecture of the NCS, minor delays on the network that are shorter than the sampling period can be neglected. On the other hand, long delays where packets are delayed for more than one sampling interval, can have severe impact on the performance and stability of the NCS [113]. In worst case scenario, a packet can be delayed for such a long time, that a more recent packet arrives at the plant prior to the earlier transmitted packet (commonly known as out-of-order packet arrival) [130].
- **Packet loss.** When a packet, that is transmitted does not arrive at the plant, the packet is considered lost. Packet loss can be caused by many factors, such as packet collisions in carrier sense multiple access (CSMA) protocols. These occur when multiple users transmit their data simultaneously. Further, interference on the network can have a significant impact on packet loss on a wireless network [1], [29], [34], [41], [129], [130]. Packets that are affected by a sufficiently long transmission delay or are received out-of-order are often considered as being lost [25], [130].
- **Bandwidth limitations.** Every network channel has a maximum bandwidth [127]. This bandwidth is shared among all users on the channel, since only one user can transmit or receive on a channel at any given time.¹ From a control design perspective, the following factors have to be taken into account:

¹Networks that utilize frequency division multiple access (FDMA) allow multiple users to transmit simultaneously using different channels.

- **Data rate.** The bandwidth constraint leads to a maximum data rate that can be achieved on the network. This is generally the raw network data rate, which includes the packet overhead. In large distributed NCSs, this puts limits on how many devices can communicate within a given time frame [113], [127], [132].
- **Transmission interval.** Another method to reduce the bandwidth requirements is to increase the transmission interval, e.g. the time between addressing a plant in a distributed system [40], [65], [71], [103], [118], [132].

The aspects mentioned above can affect the control performance and in worst case lead to instability of the NCS. It is therefore important to take these aspects into account during the controller and estimator design phase. This thesis addresses in particular packet loss and bandwidth limitations during the controller and estimator design for NCSs. Before presenting the state of the art on controller and estimator design for NCSs that utilize wireless networks affected by packet loss, we present a number of wireless network protocols that are designed for the use within industrial wireless NCSs. This is followed by an introduction into present methods on how to formalize notions on packet loss, such that these can be taken into account during the controller and estimator design. A more detailed introduction into bandwidth constraints on wireless networks is deferred to Chapter 5.

2.2 Wireless networks for networked control systems

For the controller and estimator designs that are presented in this thesis, it is necessary to introduce some relevant fundamental aspects of wireless networks. We provide an introduction to industrial wireless network protocols, which are utilized in the NCSs that are considered for the optimal controller and scheduler co-design in Chapters 5 and 6. However, to consider packet loss and delays on the network, it is necessary to abstract the wireless network by appropriate models that can be used directly in the synthesis and analysis of NCSs.

2.2.1 Industrial wireless networks

There exist multiple wireless network protocols that are designed to be utilized for industrial applications. This includes protocols such as WirelessHART [30], ISA100 [46] and ZigBee PRO [133]. These standards are all based on the IEEE 802.15.4 standard [43], [44]. The IEEE 802.15.4 standard is designed to be utilized for low power industrial environments with real-time requirements and can operate in both a beacon and a non-beacon operated mode [43]. In this work, we however only consider the beacon operated mode.

In the beacon operated mode, a centralized personal area network (PAN) coordinator (or network coordinator) is responsible for the configuration and synchronization of the network. The synchronization is done periodically at each beacon interval (BI), which marks the beginning of a so-called *superframe*. The superframe is depicted in Figure 2.2 and is divided into an active and an inactive period. All transmissions must finish before the end of the active period of the superframe. The inactive period allows devices to go into a low power state until the next superframe commences. The active period of the superframe consists of a contention access period (CAP) and a contention free period (CFP). Both the CFP and the CAP are slotted and of a

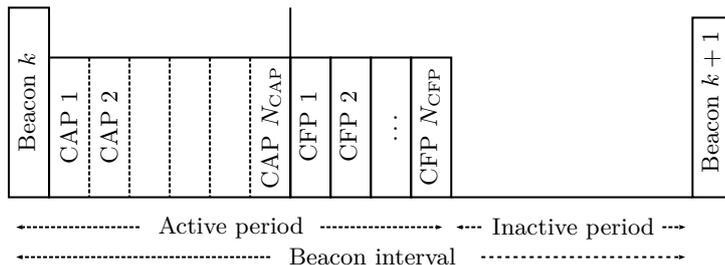


Figure 2.2: The IEEE 802.15.4 superframe.

fixed duration that is selected by the network coordinator [43]. During the CAP, any user can transmit using carrier sense multiple access with collision avoidance (CSMA/CA) with random back-off. This means, that prior to commencing a transmission, the user senses the network for ongoing transmissions. If the network is occupied, the user delays its next transmission attempt for a random amount of time. Transmissions that are delayed beyond the duration of the current superframe, will proceed in the following superframe.

Meanwhile, the CFP is designed for real-time communication. Unlike the CAP, transmissions in the CFP are allocated by the network coordinator. The IEEE 802.15.4 standard utilizes time division multiple access (TDMA)² to allocate transmission slots to users. This ensures that the probability for interference and collisions during the CFP is significantly reduced. Users can request for a transmission slot (also called a guaranteed time-slot (GTS)) in the CFP. The network coordinator assigns these GTSs on a first-come-first-served base. When a user gets a GTS granted, it can transmit during this GTS in the CFP of every superframe until the GTS gets unassigned. Both the user and network coordinator can revoke a GTS. While the number of GTSs in the CFP of each superframe is set by the network coordinator, the IEEE 802.15.4 standard only allows up to 7 GTSs to be assigned simultaneously [43]. However, the IEEE 802.15.4e extension [44] as well as WirelessHART and ISA100 allow for a larger number of GTSs in each superframe. In the IEEE 802.15.4 standard, it is optional to enable the CFP. However, the CAP is compulsory [43].

In beacon-enabled mode, the IEEE 802.15.4 standard allows for the use of acknowledgments to confirm successful transmissions in both the CAP and CFP. This means, that upon successful reception of a packet, the user transmits an acknowledgment. When acknowledgments are used, the sender of the packet knows whether the packet is successfully transmitted upon receiving the acknowledgment.

Thus, the IEEE 802.15.4 standard features two communication periods: The CFP and the CAP. The former features more reliable communications, but has limited availability. On the other hand, the latter is widely available, but features increased risk for packet delays and collisions. Next, we illustrate how these packet delays and collisions affect the systems within the NCS. This is followed by methods on how to model packet loss.

2.2.2 Packet loss modelling

Many studies consider NCS synthesis and analysis over reliable wired and wireless networks [21], [32], [40], [42], [53], [59], [92], [93], [109], [129]. However, packet loss and delays are inevitable

²The IEEE 802.15.4e standard also supports FDMA in the CFP. This is however not considered in this thesis.

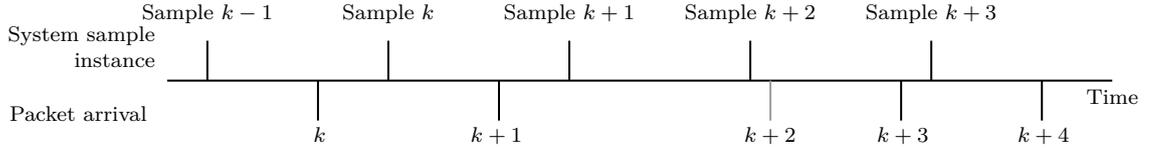


Figure 2.3: A conceptual illustration of Example 2.2.1. The periodic sampling intervals for a system is illustrated on top. At each sampling interval, the packet that has arrived (illustrated below the line) is utilized. Note that the packet that is needed for sample $k + 2$ is delayed until after sample $k + 2$. What shall the NCS do at sample $k + 2$, since no packet arrived, and at time $k + 3$, shall the NCS utilize the data from packet $k + 2$ and $k + 3$?

on implementations of wireless networks. When considering discrete-time systems, transmission delays can be neglected, provided that the maximum delay is bounded by the sampling interval of the system. It however depends largely on the sampling interval of the system whether this assumption is practical. Also, the length of the delays on wireless networks can vary significantly [113], [132]. The next example illustrates the concept of transmission delays with periodic sampling.

Example 2.2.1. *A conceptual illustration of transmission delays is provided in Figure 2.3. The periodic sampling times are illustrated above the time line. At each sampling instance, if available, the new data is applied to the actuators. For example, at sample instance k in Figure 2.3, the data that is transmitted in packet k is utilized. Likewise, at time $k + 1$, the data that is contained in packet $k + 1$ is utilized. It here does not matter when packet k and $k + 1$ arrive as long as they arrive before the sampling instance. However, as seen in Figure 2.3, sample $k + 2$ is taken before packet $k + 2$ arrives. This means, that there is no new data to be utilized. In this case, the system has to decide what data to use. Common methods in the control literature are to either use the previous data, that is sample $k + 1$, or apply 0, e.g. reset the actuators [99]. Between sample $k + 2$ and $k + 3$ both the delayed packet $k + 2$ and packet $k + 3$ arrived. Here, the system has to consider whether to utilize the older data that is contained in packet $k + 2$, or the newer data that is contained in packet $k + 3$. Generally, more recent data is preferred over the older data, and the data in packet $k + 2$ is discarded. In this setting, the packet at time $k + 2$ is considered to be lost [113]. This is often referred to as “a packet dropout occurred at time $k + 2$ ”.*

Example 2.2.1 reveals two important aspects that need to be considered in wireless NCS design:

1. What to do when delayed data arrives?
2. What to do when no data arrives?

In control system design for NCSs that are affected with packet dropouts, more recent data is preferred over older data. This means, that for example in Figure 2.3, at time $k + 3$, the data that is contained in the delayed packet $k + 2$ is discarded and the more recent data in packet $k + 3$ will be applied instead. The reason for this decision is that in NCS design, data packets tend to contain measurement data from sensors and/or inputs that are to be applied at actuators for the plant (see Figures 2.1a and 2.1b). In this case, more recent measurement or input data is preferred over delayed measurement or input data. A packet that is delayed past the sampling interval is therefore considered as being lost. This answers Item 1. The question in Item 2 will be answered later when we introduce controller design for NCSs that are affected by packet loss and delays.

2. BACKGROUND

In NCS design, packet loss is frequently modeled by using a stochastic distribution [41], [47], [61], [77]–[79], [127], [130], [132]. A successful packet transmission is indicated by a binary variable $\rho_k \in \{0, 1\}$, where $\rho_k = 1$ indicates a successful transmission at time k . The simplest packet dropout model is the independent and identically distributed (*i.i.d.*) model. This model assumes a constant *i.i.d.* probability for successful packet transmissions, which is given by

$$\Pr\{\rho_k = 1\} = \mu \quad \mu \in [0, 1]. \quad (2.1)$$

This assumption is frequently utilized in NCS design [7], [8], [53], [90], [91], [100], [104]. However, in reality, the packet arrival probability μ is affected by temporal effects and does contain memory [12], [47], [51], [128]. One way to take this into account is by making the *i.i.d.* probability in (2.1) time-varying. However, this can introduce difficulties in the NCS design and analysis. A common method to capture temporal network effects that contain memory, is the use of Markov models such as the two-state Gilbert-Elliot model which is illustrated in Figure 2.4 [33], [54], [121]. In this model, the packet arrival probability depends on a network state. In this case the probability for a successful transmission is given by

$$\Pr\{\rho_k = 1|\theta_k\},$$

where the variable $\theta \in \{\text{Good}, \text{Bad}\}$ is the network state. Here, the “Good” state represents stable network operation with a high packet arrival probability while the “Bad” state represents burst errors that are caused by network effects such as congestion and interference. This results in a lower packet arrival probability [51]. Thus,

$$\Pr\{\rho_k = 1|\theta_k = \text{Good}\} > \Pr\{\rho_k = 1|\theta_k = \text{Bad}\}.$$

The network state θ is Markov. This means that the current value of θ_k depends on the previous outcome of θ_{k-1} . This is formally stated as

$$\Pr\{\theta_k = \text{Good}|\theta_{k-1}, \theta_{k-2}, \dots, \theta_0\} = \Pr\{\theta_k = \text{Good}|\theta_{k-1}\}.$$

Associated with θ is the transition matrix

$$P_\theta = \begin{bmatrix} p_{GG} & p_{GB} \\ p_{BG} & p_{BB} \end{bmatrix},$$

where for example $p_{GB} \triangleq \Pr\{\theta_k = \text{Bad}|\theta_{k-1} = \text{Good}\}$.

The Gilbert-Elliot model in Figure 2.4 is a hidden Markov model, which means that the users can not observe the network state θ , but only the transmission outcome ρ_k . While the two-state Gilbert-Elliot model can capture network effects such as burst dropouts, the model can only capture network effects with short memory, and it still relies on that the channel quality remains stationary [12], [47], [51]. Increasing the model order of the Markov model, allows the capturing of longer memory effects and to some extent, temporal effects [51], [95], [119], [122].

Other network models utilize finite state Markov chains (FSMCs) to define the network state using recent history of packet arrivals [51], [119]. In this setting, a user can observe the network state by monitoring the packet arrivals. In the FSMC model, the probability for successful transmissions depends on a finite history of previous packet transmission outcomes, e.g.

$$\Pr\{\rho_k = 1|\rho_{k-1}, \dots, \rho_0\} = \Pr\{\rho_k = 1|\rho_{k-1}, \dots, \rho_{k-d}\} \quad d \geq 0, \quad (2.2)$$

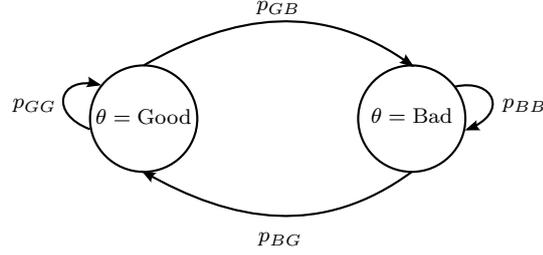


Figure 2.4: Gilbert-Elliot model with a “Good” and a “Bad” state. The arrows illustrate the transitions with their associated state transition probabilities.

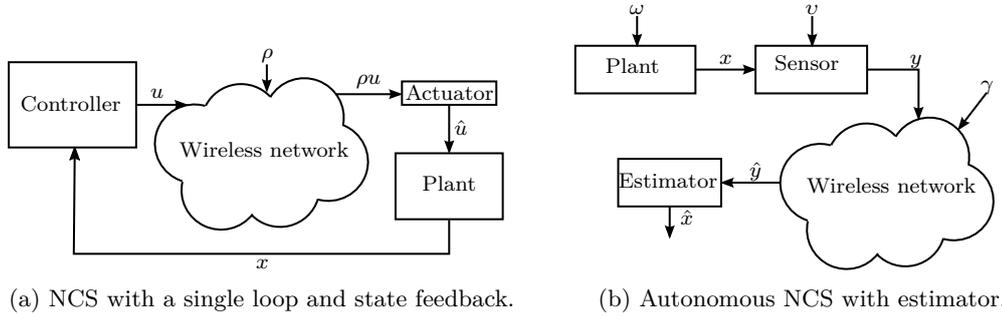


Figure 2.5: The systems that are considered in the controller design Figure 2.5a and the estimator design Figure 2.5b.

where d is the length of the relevant history. It is worth noting, that the *i.i.d* distribution in (2.1) corresponds to the model (2.2) with history length $d = 0$.

In the next section we will present the state of the art on control and estimation over wireless networks that are modelled using the *i.i.d* packet loss model presented in (2.1). Controller design for networks models that utilize the FSMC model (2.2) is studied in Chapter 4.

2.3 Control and estimation over wireless networks

This section presents the state of the art on controller and estimator design for NCSs that are affected by *i.i.d* packet dropouts.

2.3.1 Control over wireless networks

In this section, we consider a NCS as depicted in Figure 2.5a. In Figure 2.5a, the controller and actuator are connected over a wireless network that is affected by random packet dropouts. On the other hand, the state is transmitted directly over a lossless network. This is also called *state feedback*. The system dynamics are given by a discrete-time linear time-invariant (LTI) model of the form

$$x_{k+1} = Ax_k + B\hat{u}_k + D\omega_k \quad k \in \mathbb{N} \quad (2.3a)$$

$$\hat{u}_k = \rho_k u_k \quad (2.3b)$$

2. BACKGROUND

where $A \in \mathbb{R}^{m \times m}$ is the internal system model and $B \in \mathbb{R}^{m \times n}$ the input model. The vector $x_k \in \mathbb{R}^m$ is the state at time-step k and $u_k \in \mathbb{R}^n$ the input. The initial state x_0 is either known, or has a known distribution. The disturbance $\omega_k \sim \mathcal{N}(0, \Sigma_\omega)$ is white zero-mean Gaussian noise with (co-)variance Σ_ω and D is a matrix that contains the disturbance model and is of appropriate dimensions. The input u_k is transmitted over a wireless network where the binary random variable $\rho_k \in \{0, 1\}$ indicates successful packet transmissions. In this section we consider an *i.i.d* network model where the packet arrival probability is constant over time, thus, as in (2.1), the probability for a successful packet transmission is given by

$$\Pr\{\rho_k = 1\} = \mu.$$

This means, that whenever $\rho_k = 0$, no packet arrives at the controller. As posed in Item 2 in Section 2.2.2, one has to decide what the actuator in Figure 2.5a shall do. In fact, it is an open discussion in the literature, what the actuators shall do when no data is received [99]. The two most common actuator strategies are *set-to-zero* and *hold-input*. However, the work [99] concluded that none of these two actuator schemes are superior to the other. For set-to-zero, the actuators are set to their zero position upon a packet dropout. The actuator strategy in (2.3b) is a set-to-zero policy.

When using a hold-input policy, the previous actuator input is applied upon a packet dropout. This is implemented by utilizing a buffer at the actuators. In (2.3), this replaces the actuator strategy (2.3b) by

$$\hat{u}_k = \rho_k u_k + (1 - \rho_k) \hat{u}_{k-1}.$$

An alternative method to implement a hold-input strategy, is to use integrators at the actuators. In this case, the controller transmits the desired change of the value held at the actuators instead of the actual actuator value. When using integrators at the actuators, system (2.3a) can, using state-space augmentation, be restated to

$$\underbrace{\begin{bmatrix} x_{k+1} \\ b_{k+1} \end{bmatrix}}_{\xi_{k+1}} = \underbrace{\begin{bmatrix} A & B \\ \mathbf{0}_{n \times m} & I_n \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} x_k \\ b_k \end{bmatrix}}_{\xi_k} + \underbrace{\begin{bmatrix} B \\ I_n \end{bmatrix}}_{\beta} \hat{u}_k + \underbrace{\begin{bmatrix} D \\ \mathbf{0} \end{bmatrix}}_{\Gamma} \omega_k \quad (2.4)$$

$$\hat{u}_k = \rho_k u_k,$$

where the actuator input u_k is the change of actuation instead of the actuator input. It is worth noting, that system (2.4) is of the same form as the model (2.3) that utilizes a set-to-zero policy. This means, that the same controller design method can be used to design control laws for set-to-zero and hold-input actuator strategies.

More advanced actuator strategies involve packet predictive control, where the controller transmits a packet containing predicted future control inputs. These predicted control inputs are then utilized in the event of packet dropouts [58], [80], [89], or by the use of advanced packet dropout compensation methods [35], [36], [57].

Next, we will present some concepts on stability of the system (2.3). For controller design, an important question is: Is it possible to steer the state $x(k)$ of the system (2.3) from any initial position to any position in finite time? This concept is called *controllability* [4]. Consider a

deterministic system of the form

$$x_{k+1} = Ax_k + Bu_k. \quad (2.5)$$

Let

$$U_k \triangleq (u_i : i = 0, \dots, k)$$

be a sequence of control inputs. The term *controllability* then refers to the ability to steer the system (2.5) from any initial state x_0 to any final state x_k in finite time. This is formally defined as:

Definition 2.3.1 (Controllability [4, Appendix B]). *A pair (A, B) for the deterministic system (2.5) is termed controllable if, for a finite time k and any initial state $x_0 \in \mathbb{R}^m$, there exists a sequence of controls $U_k = (u_i : i = 0, \dots, k-1)$, that can drive the system (2.5) from x_0 to any state $x^* \in \mathbb{R}^m$ within k steps.*

Definition 2.3.1 can be weakened to the condition that merely the unstable part of the system matrix A has to be controllable. This is denoted *stabilizability*, and is formally defined as:

Definition 2.3.2 (Stabilizability [4]). *A pair (A, B) is stabilizable if there exists a linear transform \mathcal{T} , such that*

$$\mathcal{T}A\mathcal{T}^{-1} = \begin{bmatrix} A_c & A_{c,uc} \\ \mathbf{0} & A_{uc} \end{bmatrix} \quad \mathcal{T}B = \begin{bmatrix} B_c \\ \mathbf{0} \end{bmatrix},$$

where the pair (A_c, B_c) is controllable, and if the subspace A_{uc} has dimension larger than 0, A_{uc} is stable. That is, all eigenvalues of A_{uc} are within the unit circle.

However, note that the system (2.3) is affected by both random packet loss and a random disturbance. Due to these uncertainties, the state x_k itself is a random variable and Definitions 2.3.1 and 2.3.2 are not applicable. For stochastic systems, instead of verifying whether a sequence of controls can steer the system state to any particular value, it is investigated whether there exists a sequence of controls that ensures boundedness of the first and second moment of the state x_k . This is called mean square stability (MSS).

Definition 2.3.3 (Mean square stability (MSS)[23], [49]). *The system (2.3) is MSS if from any initial distribution on x_0 , the first moment*

$$\lim_{k \rightarrow \infty} \mathbf{E} \{ \|x_k\| \} < \infty$$

and the second moment

$$\lim_{k \rightarrow \infty} \mathbf{E} \{ \|x_k\|^2 \} < \infty.$$

Remark 2.3.1. *It is worth noting, that for Definition 2.3.3 boundedness of the second moment implies boundedness of the first moment. See [49, Theorem 3.4].*

Definition 2.3.3 leads to the following stochastic analogue to Definition 2.3.2.

Definition 2.3.4 (Stochastic stabilizability [23], [24], [49]). *The system (2.3) is stochastically stabilizable if there exists a sequence of controls $U_k = (u_i : i = 0, \dots, k-1)$, such that the system (2.3) is MSS. That is,*

$$\lim_{k \rightarrow \infty} \mathbf{E} \left\{ \|x_k\|^2 \right\} < \infty.$$

It follows directly for the NCS in (2.3), that deterministic stabilizability of the pair (A, B) is a necessary condition for stochastic stabilizability.

2.3.2 Estimation over wireless networks

Consider an autonomous discrete-time LTI system where the sensors and estimator are connected through a wireless network. This system is depicted in Figure 2.5b. This system has dynamics

$$\begin{aligned} x_{k+1} &= Ax_k + \omega_k \\ y_k &= Cx_k + v_k \\ \hat{y}_k &= \gamma_k y_k, \end{aligned}$$

where $A \in \mathbb{R}^{m \times m}$, $C \in \mathbb{R}^{q \times m}$ and the disturbances $\omega_k \sim \mathcal{N}(\mathbf{0}, \Sigma_\omega)$ and $v_k \sim \mathcal{N}(\mathbf{0}, \Sigma_v)$ are white Gaussian noise. The initial state $x_0 \sim \mathcal{N}(\mathbf{0}, P_0)$ is Gaussian distributed with covariance P_0 . \hat{y}_k is the sensor measurement that is received by the estimator. The measurements are sent over a wireless network that is affected by random packet loss. In this setting, the random variable $\gamma_k \in \{0, 1\}$ indicates successful packet transmissions. Consider the case where the network model is *i.i.d* and the packet arrival probability is constant over time, that is

$$\Pr \{\gamma_k = 1\} = \lambda.$$

The work [104] found, that when the packet dropouts are *i.i.d*, the optimal minimum mean square error (MMSE) estimator is a Kalman filter, which only performs the innovation step upon successful reception of the measurement data. Using this estimator, the state estimate

$$\hat{x}_{k+1} \triangleq \mathbf{E} \{x_{k+1} | \hat{y}_{k+1}\},$$

is given by

$$\hat{x}_{k+1} = \begin{cases} A\hat{x}_k + K_{k+1}(y_{k+1} - CA\hat{x}_k) & \text{if } \gamma_{k+1} = 1 \\ A\hat{x}_k & \text{if } \gamma_{k+1} = 0. \end{cases}$$

The estimator gain K_{k+1} is designed to minimize the estimation error covariance

$$P_{k+1} = \|\mathbf{E} \{e_{k+1}e_{k+1}^T\}\|^2,$$

where

$$e_{k+1} \triangleq x_{k+1} - \hat{x}_{k+1} \tag{2.6}$$

is the estimation error. This results in the Kalman gain

$$K_{k+1} = P_{k+1}C^T (R + CP_{k+1}C^T)^{-1} \tag{2.7}$$

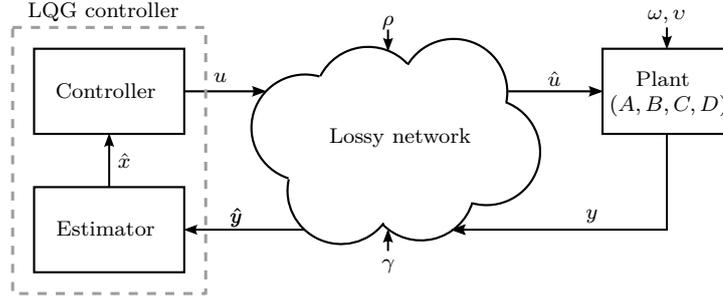


Figure 2.6: The NCS from Figure 2.1a with disturbances and packet loss parameters indicated. The controller and estimator are co-located and can share information.

and error covariance

$$P_{k+1} = \begin{cases} Q + AP_k A^T - AP_k C^T (R + CP_k C^T)^{-1} CP_k A^T & \text{if } \gamma_k = 1 \\ Q + AP_k A^T & \text{if } \gamma_k = 0, \end{cases} \quad (2.8)$$

where P_0 is the covariance of x_0 . In contrast to the traditional Kalman filter [3], where the error covariance is deterministic when P_0 is known, the error covariance (2.8) is stochastic due to the random variable γ_k . The covariance for this Kalman filter is updated online and will, contrary to the traditional Kalman filter, not converge to a steady-state value [3], [104]. However, the work [104] shows, that there exists a critical value on the packet arrival probability λ_c where for $\lambda < \lambda_c$, the error covariance becomes unbounded. It is however non-trivial to obtain the exact value of λ_c . Instead, the work [104] shows that there exist an upper and a lower bound λ_c^+ and λ_c^- , respectively. Over the years, researchers have studied methods to tighten the upper and lower bounds, and in special cases were able to make the upper and lower bounds coincide (i.e. $\lambda_c^+ = \lambda_c^-$) [68], [86], [102]. This characterization is also extended to Markovian packet dropout models [69], [94], [125].

As described above, the error covariance and Kalman gain (2.7) and (2.8) do not converge to steady-state values. Pre-computing these values requires information about the entire sample-path of γ_k . To address this, the works [27], [108] introduced jump linear system (JLS) estimators, which allows for offline computation of the Kalman gain and error covariance. Numerical simulations show, that when a sufficiently long history is considered, the performance of the JLS estimator is comparable to the time-varying Kalman filter [27].

Next, we present the controller design for systems where the controller and actuator as well as the sensor and estimator communicate over a wireless network that is affected by packet loss.

2.3.3 The separation principle

Sections 2.3.1 and 2.3.2 reviewed results on controller design using state feedback and estimator design for autonomous systems, respectively. In this section, we present the state of the art on linear quadratic Gaussian (LQG) design for NCSs that are affected by random packet loss. Figure 2.6 depicts the NCS from Figure 2.1a in more detail. Consider the system model to be

2. BACKGROUND

discrete-time LTI with dynamics

$$\begin{aligned}
 x_{k+1} &= Ax_k + B\hat{u}_k + D\omega_k \\
 y_k &= Cx_k + v_k \\
 \hat{u}_k &= \rho_k u_k \\
 \hat{y}_k &= \gamma_k y_k,
 \end{aligned} \tag{2.9}$$

where \hat{u}_k and \hat{y}_k are the control input and sensor output after being transmitted over a wireless network. The *i.i.d* random variables $\gamma_k, \rho_k \in \{0, 1\}$ indicate successful packet arrivals. Both γ_k and ρ_k are *i.i.d* where $\Pr\{\gamma_k = 1\} = \lambda$ and $\Pr\{\rho_k = 1\} = \mu$. The probabilities λ and μ remain constant over time.

The LQG controller in Figure 2.6 does not have direct access to the state x_k . Instead, it has intermittent access to the measurement \hat{y}_k that is transmitted over a network where packet loss occurs. The goal is to design an optimal controller that using measurement \hat{y}_k can compute the optimal control input u_k , such that the cost function

$$J_N = \mathbf{E} \left\{ \|x_N\|_{S_0}^2 + \sum_{k=0}^{N-1} \|x_k\|_Q^2 + \|\hat{u}_k\|_R^2 \middle| \mathcal{I}_k \right\} \tag{2.10}$$

is minimized [45], [100]. In (2.10) the set \mathcal{I}_k represents the information that is available to the LQG controller at each time-step, the positive semi-definite matrices S_0, Q and R are tuning parameters that trade-off control performance and control effort and $N \geq 0$ is the prediction horizon.

It is well-known, that for the classical case where no packet loss occurs (that is $\Pr\{\rho_k = 1\} = \Pr\{\gamma_k = 1\} = 1, \forall k$ in (2.9)), one can separately design an optimal controller for system (2.9) assuming state feedback and an optimal estimator assuming that the control input $u_k = 0$. Then, one replaces the state in the optimal feedback law $u_k = f^*(x_k)$ by the optimal MMSE estimate \hat{x}_k , such that the optimal control law becomes $u_k = f^*(\hat{x}_k)$ [4, Chapter 8]. However, when the plant and controller communicate through a wireless network that is affected by random packet loss, this procedure becomes more involved. The reason for this is, that the optimal estimator needs information on the control input \hat{u}_k that is applied at the actuators to compute the optimal estimate. When the controller and the actuators are connected over a lossy network, the control packet can be lost. In Figure 2.6, the estimator and the controller are co-located with each other. This means, that the estimator has direct access to the control input u_k prior to transmission over the network, but not necessarily to \hat{u}_k after being received by the actuator. When successful packet transmissions are acknowledged, that is, the actuator sends an acknowledgement to the controller upon successful reception of a packet, and the acknowledgements are not affected by packet dropouts, the controller has access to \hat{u}_k and \hat{y}_k . Formally, when using acknowledgements, the optimal estimator has access to the following information at each time-step:

$$\begin{aligned}
 \mathcal{I}_k &= \{\hat{y}_k, u_k, \gamma_k, \rho_k, \mathcal{I}_{k-1}\} \\
 &= \{\hat{y}_k, \hat{u}_k, \mathcal{I}_{k-1}\},
 \end{aligned}$$

with $\mathcal{I}_{-1} = \emptyset$. Many works refer to this scheme as the transmission control protocol (TCP) case [45], [100]. In this case, minimizing (2.10) results in an optimal controller that uses the feedback form

$$u_k = -L_k \hat{x}_k,$$

where the optimal MMSE estimate \hat{x}_k is of the form

$$\hat{x}_{k+1} = \begin{cases} A\hat{x}_k + \rho_k Bu_k + AK_{k+1}(y_{k+1} - CA\hat{x}_k) & \text{if } \gamma_{k+1} = 1 \\ A\hat{x}_k + \rho_k Bu_k & \text{if } \gamma_{k+1} = 0, \end{cases} \quad (2.11)$$

with the Kalman gain K_k given in (2.7). The estimation error (2.6) then becomes

$$\begin{aligned} e_{k+1} &= \begin{cases} Ax_k + \rho_k Bu_k + \omega_k - A\hat{x}_k - \rho_k Bu_k - AK_{k+1}(y_{k+1} - CA\hat{x}_k) & \text{if } \gamma_{k+1} = 1 \\ Ax_k + \rho_k Bu_k + \omega_k - A\hat{x}_k - \rho_k Bu_k & \text{if } \gamma_{k+1} = 0 \end{cases} \\ &= \begin{cases} Ae_k + \omega_k - AK_{k+1}(y_{k+1} - CA\hat{x}_k) & \text{if } \gamma_{k+1} = 1 \\ Ae_k + \omega_k & \text{if } \gamma_{k+1} = 0. \end{cases} \end{aligned}$$

Note, that the estimation error does not depend on the control input \hat{u}_k , and the error covariance is given by (2.6). In this case, the optimal estimator and optimal controller can be designed separately and thus, the separation principle holds. However, recall that the estimator (2.11) needs to know ρ_k at each time-step.

When successful transmissions from the controller to the actuator are not acknowledged, the controller does not have access to the packet transmission outcome ρ_k . In this case, the information that the LQG controller has access to is comprised of

$$\begin{aligned} \mathcal{I}_k &= \{\hat{y}_k, u_k, \gamma_k, \mathcal{I}_{k-1}\} \\ &= \{\hat{y}_k, u_k, \mathcal{I}_{k-1}\} \\ \mathcal{I}_{-1} &= \emptyset. \end{aligned}$$

This is also called the user datagram protocol (UDP) case [45], [100], [106]. For the MMSE estimator of the form (2.11), the estimation error (2.6) becomes

$$\begin{aligned} e_{k+1} &= \begin{cases} Ax_k + \rho_k Bu_k + \omega_k - A\hat{x}_k - \mathbf{E}\{\rho_k\} Bu_k - AK_{k+1}(y_{k+1} - CA\hat{x}_k) & \text{if } \gamma_{k+1} = 1 \\ Ax_k + \rho_k Bu_k + \omega_k - A\hat{x}_k - \mathbf{E}\{\rho_k\} Bu_k & \text{if } \gamma_{k+1} = 0 \end{cases} \\ &= \begin{cases} Ae_k + (\rho_k - \mathbf{E}\{\rho_k\}) Bu_k + \omega_k - AK_{k+1}(y_{k+1} - CA\hat{x}_k) & \text{if } \gamma_{k+1} = 1 \\ Ae_k + (\rho_k - \mathbf{E}\{\rho_k\}) Bu_k + \omega_k & \text{if } \gamma_{k+1} = 0. \end{cases} \end{aligned}$$

Thus, when the control inputs are not acknowledged, the estimation error depends on the control input. In this setting, the separation principle does not hold and the design becomes more involved [45], [53], [100]. This thesis is limited to estimator design where acknowledgements are utilized and successfully received with probability one.

2.4 Summary

In this chapter, we presented numerous challenges that need to be considered and addressed when using wireless networks within NCSs. These challenges include interference, which leads to packet loss and transmission delays, as well as bandwidth limitations, which lead to constraints on the data rate and the minimum achievable transmission interval. After an introduction to the IEEE 802.15.4 protocol standard, we introduced widely used methods to model packet loss on wireless networks. This is followed by a presentation of the state of the art on controller and estimator design for networks that are affected by random packet dropouts that are modelled by an *i.i.d*

2. BACKGROUND

distribution. Finally, the separation principle is presented which, under certain conditions, allows to design the optimal output feedback controller by separately designing an optimal state estimator and optimal controller. Further, we presented concepts such as controllability and stabilizability for systems that are affected by random packet loss.

Duality for control and estimation for systems with i.i.d. packet losses

In this chapter we establish a new form of duality between offline estimator and controller design for networked systems that are affected by independent and identically distributed (*i.i.d.*) packet dropouts. More specifically, we show that the optimal offline estimator can be obtained by designing an optimal controller for a dual control problem. This extends the classical duality results to networked control systems (NCSs) that are affected by random packet loss.

After a literature review in Section 3.1, we introduce the estimator and controller designs in Section 3.2. Here, the design of the optimal offline controller and estimator leads to a modified Riccati equation. In Section 3.3 we show that the existence of a solution to the modified Riccati equation is a necessary and sufficient condition for mean square stability (MSS) of the controller. We then show that the offline estimator Riccati equation is dual to the controller Riccati equation, and use this duality to show stability of the offline estimator. Section 3.3 concludes with a dual system formulation between the controller and offline estimator. This extends results from the classical linear quadratic Gaussian (LQG) design to systems that are affected by random *i.i.d.* packet dropouts. The modified Riccati equations that are obtained in Section 3.2 depend on the statistics of the communication channel between the controller and actuators or sensors and estimator. Due to this dependency, the classical detectability and controllability conditions are no longer sufficient for the existence of a solution to the modified Riccati equations. We therefore in Section 3.4 discuss, based on existing theory, sufficient conditions for the convergence of the modified Riccati equations. This chapter summarizes in Section 3.5 with numerical examples and comparisons to other controllers and estimators.

3.1 Introduction

In this chapter we study two different discrete-time systems. The first is a linear time-invariant (LTI) system where the sensor is connected to a state estimator over a wireless network. This system is depicted in Figure 3.1. In the second system, the controller is connected to the actuators over a wireless network where the controller has full knowledge of the state. This is depicted in Figure 3.2. In both systems, the wireless network is modelled to either successfully transmit the

entire packet, or to fail the transmission of the entire packet. Also the probabilities for successful transmissions are *i.i.d* and do not vary over time. Here, a successful transmission occurs with probability μ (or ϱ) and a failure occurs with probability $1 - \mu$ (or $1 - \varrho$). In the literature, there exist numerous works that study controller design for the system depicted in Figure 3.2, where *i.i.d* packet dropouts occur in the link between the controller and the actuators [11], [41], [45], [100], [105], [106], [130]. Also, the design of estimators for systems as depicted in Figure 3.1, where packet dropouts occur in the link between the sensor and estimator is well studied [88], [104], [126]. The works [50], [100] study the LQG problem with packet dropouts. In this setting, the controller and estimator design is combined. Here, works such as [100] establish a form of duality between the controller Riccati equation and the Riccati equation that bounds the performance of the online estimator. There is however no duality in the classical sense as, e.g. between the linear quadratic regulator (LQR) and Kalman filter, where one can design the Kalman filter by formulating it as a dual control problem and vice versa. In this chapter, we will establish a similar form of duality between an offline Kalman filter design and LQR design for systems that are affected by random packet dropouts. We will later clarify exactly why this duality does not exist between the online Kalman filter as presented in e.g. [98] and LQR.

Fully offline estimators will offer reduced performance compared to online estimators, such as the Kalman filter [104] or semi-offline estimators such as the jump linear system (JLS) estimator [27]. The disadvantage with an online Kalman filter is however, that the error covariance matrix has to be updated online. Since this requires matrix inversions, the computational power that is required can be undesirable for low power microcontrollers on battery powered devices for applications such as remote sensing. Further, in cases where the hardware on which the algorithms are implemented does not support floating point operations, complications arise since the matrix inversion has to be implemented in fixed point arithmetic. The JLS estimator addresses this by utilising a lookup table instead of online updating the error covariance. This results in merely minor performance detriments compared to the online Kalman filter [27]. The JLS estimator however requires that the history of the packet arrivals is tracked and that there is enough memory to store the lookup table with estimator gains. This can be avoided altogether with the offline estimator, where only one gain is applied.

Although the previous work [100] investigated the duality that exists between the Riccati equations that are used for the estimator and controller designs to establish stability for the controller and bounds for the estimator error covariance matrix, the design that we present establishes duality between the estimator error dynamics and the closed-loop control dynamics.

3.2 Problem setting

In this section, we will first introduce the estimator setting. This is, we will introduce the controller design to point out why there is no dual controller design to the online estimator design for systems that are affected by random packet dropouts. This section then concludes with the presentation of the optimal offline estimator.

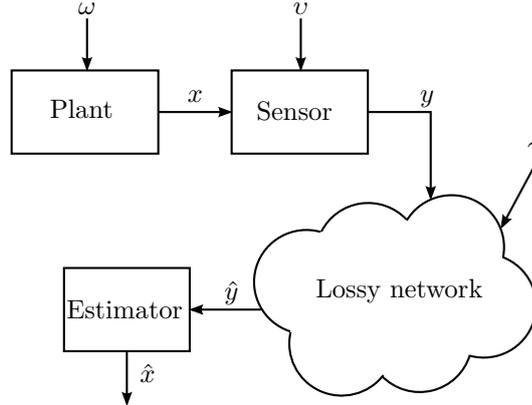


Figure 3.1: The system that is considered for the estimator design.

For the estimator we consider a system of the form

$$\begin{aligned} x_{k+1} &= Ax_k + \omega_k, \\ y_k &= \gamma_k (Cx_k + v_k), \end{aligned} \quad (3.1)$$

with $x_0 \sim \mathcal{N}(0, P_0)$, $\omega_k \sim \mathcal{N}(0, \Sigma_\omega)$, $v_k \sim \mathcal{N}(0, \Sigma_v)$ and the pair (A, C) being detectable. The pair (A, C) being detectable is equivalent to the pair (A^T, C^T) being stabilizable [4]. Also, γ_k is a stationary binary random process. Here $\gamma_k = 1$ means that the measurement y_k is successfully received by the estimator. The binary variable γ_k is *i.i.d* and $\Pr\{\gamma_k = 1\} = \mu$.

The common method to deal with packet dropouts for the estimator is to only perform the innovation step when data is received, as described in Section 2.3.2. However, note that in (2.8) in Section 2.3.2, the error covariance update is depending on the packet transmission outcome γ_k . Therefore the sequence (P_0, \dots, P_N) is unique depending on the sample path of $(\gamma_{k-q}, \dots, \gamma_k)$. Since there are 2^N different sample paths for γ_k , it is infeasible to offline compute P_k for large N . The authors of [108] address this issue by taking a finite length sample path into account when computing P_k . This results in a set of P_k 's that each depend on a particular outcome of $(\gamma_{k-q}, \dots, \gamma_{k-1})$. This allows for offline computation of P_k while providing sub-optimal performance compared to online calculations of P_k .

The above presented estimators however both rely on the available information of previous packet arrivals $(\gamma_{k-q}, \dots, \gamma_k)$. The issue with this is, that there exists no dual optimal controller to this problem. The reason for this will be pointed out later in this chapter. We will now first present the optimal controller design for the system depicted in Figure 3.2.

3.2.1 Linear quadratic regulator

For the controller we consider systems as depicted in Figure 3.2, which are of the form

$$x_{k+1} = Ax_k + \rho_k Bu_k + \omega_k, \quad (3.2)$$

with $x_0 \sim \mathcal{N}(0, \Sigma_{x_0})$, $\omega_k \sim \mathcal{N}(0, \Sigma_\omega)$ and the pair (A, B) being stabilizable. Also ρ_k is an *i.i.d* stationary binary random process, where $\rho_k = 1$ indicates that control signal u_k is successfully received by the plant. This occurs with probability $\Pr\{\rho_k = 1\} = \varrho$.

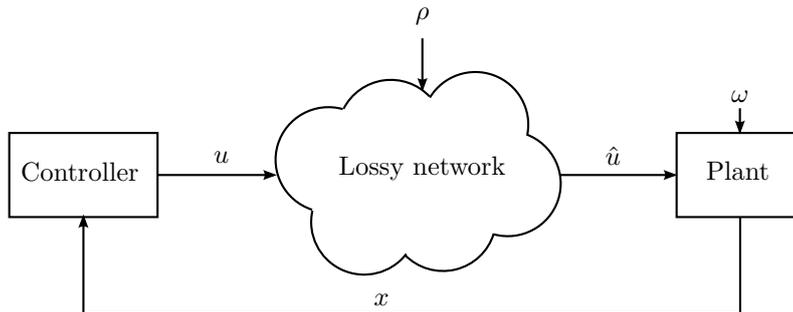


Figure 3.2: The system that is considered for the controller design.

Denote the matrix sequence

$$\mathcal{L}_{k,N} \triangleq (L_{\ell,N} : \ell = k, \dots, N-1), \quad (3.3)$$

and let $\mathcal{L}_N \triangleq \mathcal{L}_{0,N}$. Define the following control law

$$u_k = -L_{k,N}x_k, \quad k = 1, \dots, N. \quad (3.4)$$

Consider the linear quadratic (LQ) cost function

$$J(x_0, \mathcal{L}_N) = \frac{1}{N} \mathbf{E} \left\{ \sum_{k=0}^N \|x_k\|_Q^2 + \sum_{k=0}^{N-1} \|\rho_k u_k\|_R^2 \middle| x_0 \right\}, \quad (3.5)$$

where $Q \geq 0$ and $R \geq 0$ are design parameters that trade off control performance against control effort. Note, that the cost function takes the expected value of the states x_0, \dots, x_{N+1} . The controller has causal information on x and ρ . That is, at time k , the controller knows $(x_k, \dots, x_0, \rho_{k-1}, \dots, \rho_0)$. However, the controller at time k does not know the outcome of ρ_k . Since the cost (3.5) penalizes the state x and control input u into the future, it is necessary to utilize the mathematical expectation, since the future outcomes of ρ and ω in (3.2) are unknown. We want to design control laws that minimize (3.5), thus

$$\mathcal{L}_N^* \triangleq \arg \min_{\mathcal{L}_N} J(x_0, \mathcal{L}_N),$$

where $\mathcal{L}_N^* \triangleq (L_{\ell,N}^* : \ell = 0, \dots, N-1)$.

The solution of the above minimization problem is given in the following proposition.

Proposition 3.2.1. *Let Q be such that the pair $(A, Q^{\frac{1}{2}})$ is detectable and $R \geq 0$. Then, for each $k = 0, \dots, N-1$, the optimal control gain for control laws of the form (3.4) is given by ¹*

$$L_{k,N}^* = (R + B^T S_{N-k} B)^{-1} B^T S_{N-k} A, \quad (3.6)$$

with

$$\begin{aligned} S_0 &= Q, \\ S_{k+1} &= Q + A^T S_k A - \rho A^T S_k B (R + B^T S_k B)^{-1} B^T S_k A. \end{aligned} \quad (3.7)$$

¹If in (3.6) and (3.7), the matrix $R + B^T S_k B$ is singular, the inverse can be replaced by the Moore-Penrose pseudo inverse.

and $\varrho = \Pr\{\rho_k = 1\}$. This results in the optimal cost

$$J_N^*(x_0) = \frac{1}{N}x_0^T S_N x_0 + \frac{1}{N} \sum_{\ell=0}^{N-1} \text{trace}(\Sigma_\omega S_\ell).$$

Also, if the limit

$$\bar{S} \triangleq \lim_{k \rightarrow \infty} S_k \tag{3.8}$$

exists, then the asymptotic minimum cost

$$J_\infty^*(x_0) \triangleq \lim_{N \rightarrow \infty} J(x_0, \mathcal{L}_N^*) \tag{3.9}$$

is given by

$$\begin{aligned} J_\infty^*(x_0) &= J(x_0, \mathcal{L}_\infty^*) \\ &= \text{trace}(\Sigma_\omega \bar{S}), \end{aligned}$$

where the infinite sequence $\mathcal{L}_\infty^* \triangleq (L_{\ell, \infty}^* : \ell = 0, 1, \dots)$.

Proof. See Appendix 3.A. □

In view of the result in Proposition 3.2.1, whenever the limit in (3.8) exists, we define the offline control gain by

$$\bar{L} \triangleq \lim_{N \rightarrow \infty} L_{0, N-1}^*. \tag{3.10}$$

Also denote the infinite sequence of offline control gains

$$\bar{\mathcal{L}}_\infty \triangleq (\bar{L}, \bar{L}, \dots).$$

This choice results in the following state feedback law

$$u_k = -\bar{L}x_k, \tag{3.11}$$

which leads to the closed-loop system

$$x_{k+1} = (A - \rho_k B \bar{L}) x_k + \omega_k. \tag{3.12}$$

This defines our offline controller design.

Remark 3.2.1. Notice that, if we want the controller design equations (3.6) and (3.7) that are dual to those of the Kalman filter (2.7) and (2.8), we would need to replace ϱ by ρ_k in (3.7) (such replacement would have to be made together with a consistent modification of the cost function (3.5)). However, this dual design would require that the future transmission outcomes $(\rho_k, \dots, \rho_{k, N-1})$ are known at time k . This is obviously impossible because ρ_k is only causally known. It is for this reason that a dual controller to the Kalman filter is not possible. On the other hand, we can modify the estimator design, so as to obtain, instead of a Kalman filter, a design which is dual to (3.6) and (3.7). This motivates the offline estimator design which is presented in the next section.

3.2.2 Dual offline estimator

Consider the system (3.1), and the matrix sequence

$$\mathcal{K}_N \triangleq (K_{\ell,N} : \ell = 0, \dots, N-1).$$

Define the following estimator scheme

$$\tilde{x}_{k+1} = A\tilde{x}_k + \gamma_k K_{k,N} (y_k - C\tilde{x}_k), \quad k = 0, \dots, N-1, \quad (3.13)$$

with $\tilde{x}_0 \triangleq \mathbf{E}\{x_0\} = 0$. The second term on the right-hand side of (3.13) is called the innovation, which only is performed upon successfully receiving the measurement, e.g. $\gamma_k = 1$. If $\gamma_k = 0$, only the prediction $A\tilde{x}_k$ is used. The difference between the online estimator and the offline estimator is, that for the offline design, we do not assume knowledge on the history of transmission outcomes (ρ_0, \dots, ρ_k) . We therefore rely on the statistics for ρ_k . Thus, we want to design the sequence $\mathcal{K}_N \triangleq (K_{\ell,N})_0^{N-1}$, such that the estimator scheme (3.13) minimizes the following cost function

$$V(\mathcal{K}_N) = \mathbf{E} \left\{ \|x_{N-1} - \tilde{x}_{N-1}\|_2^2 \right\}, \quad (3.14)$$

which penalizes the estimation error. It is worth noting, that the offline estimator scheme (3.13) at time k uses the knowledge of γ_k to decide whether to do the innovation or not. However, the cost (3.14) does not utilize the actual knowledge on γ_k , and merely uses the expected value. This will inevitably lead to a sub-optimal performance compared to the online Kalman filter, that utilizes the full history of γ . The main practical advantage of the offline Kalman filter is, that it reduces the computational complexity significantly in implementations, allowing higher sample rates or lower computational requirements. Further, as will be illustrated next, we will obtain a dual design to the LQR design.

The sequence of optimal offline estimator gains

$$\mathcal{K}_N^* \triangleq (K_{\ell,N}^* ; \ell = 0, \dots, N-1)$$

is given by

$$\mathcal{K}_N^* \triangleq \arg \min_{\mathcal{K}_N} V(\mathcal{K}_N).$$

The solution to the above problem is given in the following proposition.

Proposition 3.2.2. *For each $k = 0, \dots, N-1$, the optimal estimator gain for estimators of the form (3.13) is given by*

$$K_{k,N}^* = AP_k C^T (\Sigma_v + CP_k C^T)^{-1}, \quad (3.15)$$

where P_0 is the initial covariance of x_0 and

$$P_{k+1} = \Sigma_w + AP_k A^T - \mu AP_k C^T (CP_k C^T + \Sigma_v)^{-1} CP_k A^T, \quad (3.16)$$

where $\mu = \mathbf{Pr}\{\gamma_k = 1\}$. Also, if the limit

$$\bar{P} \triangleq \lim_{k \rightarrow \infty} P_k \quad (3.17)$$

exists, then the asymptotic minimum cost

$$V^* \triangleq \lim_{N \rightarrow \infty} V(\mathcal{K}_N^*) \quad (3.18)$$

is given by

$$V^* = V(\mathcal{K}_\infty^*) = \text{trace}(\bar{P}), \quad (3.19)$$

where the infinite sequence $\mathcal{K}_\infty^* \triangleq (K_0^*, K_1^*, \dots)$.

Proof. See Appendix 3.B. □

In view of Proposition 3.2.2, whenever the limit in (3.17) exists, we define the offline estimator gain by

$$\bar{K} \triangleq \lim_{N \rightarrow \infty} K_{N-1, N}^*. \quad (3.20)$$

This choice then results in the following offline estimator scheme

$$\hat{x}_{k+1} = A\hat{x}_k + \gamma_k \bar{K} (y_k - C\hat{x}_k) \quad (3.21)$$

where $\hat{x}_0 \triangleq \mathbf{E}\{x_0\} = 0$. Also define the infinite sequence of offline estimator gains

$$\bar{\mathcal{K}}_\infty \triangleq (\bar{K}, \bar{K}, \dots).$$

3.2.3 A few notes on the separation principle

Consider the system

$$\begin{aligned} x_{k+1} &= Ax_k + \rho_k Bu_k + \omega_k \\ y_k &= \gamma_k (Cx_k + v_k), \end{aligned} \quad (3.22)$$

which is depicted in Figure 2.6.

It is worth noting, that in the transmission control protocol (TCP) case, i.e. where successfully received packets are acknowledged, the separation principle (Section 2.3.3) holds for system (3.22). This follows directly, since the estimator (3.13) becomes

$$\tilde{x}_{k+1} = A\tilde{x}_k + \rho_k Bu_k + \gamma_k K_{k, N} (y_k - C\tilde{x}_k).$$

Where from (3.22) $y_k = \gamma_k (Cx_k + v_k)$. Computing the estimation error in the cost (3.14) then yields

$$\begin{aligned} x_{k+1} - \tilde{x}_{k+1} &= Ax_k + \rho_k Bu_k + \omega_k - (A\tilde{x}_k + \rho_k Bu_k + \gamma_k K_{k, N} (y_k - C\tilde{x}_k)) \\ &= A(x_k - \tilde{x}_k) - \gamma_k K_{k, N} C(x_k - \tilde{x}_k) - \gamma_k K_{k, N} v_k + \omega_k. \end{aligned}$$

Note that the estimation error does not depend on the control input u_k . Therefore, the optimal offline estimator and optimal controller can be designed separately using Propositions 3.2.1 and 3.2.2. By combining the arguments from, for example, [100] and from the proof of Proposition 3.2.1, it follows that the optimal control law is then given by (3.11) where the state feedback is replaced by the state estimate, as described in Section 2.3.3.

In the user datagram protocol (UDP) case, where successful transmissions are not acknowledged, the estimate (3.13) becomes

$$\tilde{x}_{k+1} = A\tilde{x}_k + \mathbf{E}\{\rho_k\}Bu_k + \gamma_k K_{k,N}(y_k - C\tilde{x}_k).$$

Recall $\mathbf{E}\{\rho_k\} = \varrho$. This results in the estimation error

$$\begin{aligned} x_{k+1} - \tilde{x}_{k+1} &= Ax_k + \rho_k Bu_k + \omega_k - (A\tilde{x}_k + \varrho Bu_k + \gamma_k K_{k,N}(y_k - C\tilde{x}_k)) \\ &= A(x_k - \tilde{x}_k) + (\rho_k - \varrho)Bu_k - \gamma_k K_{k,N}C(x_k - \tilde{x}_k) - \gamma_k K_{k,N}\nu_k + \omega_k, \end{aligned}$$

which clearly depends on the control input u_k . The separation principle therefore does not hold for the UDP case. See for example [11], [45], [100] for more details on LQG over UDP networks.

Next, we will show stability for the controller and then establish stability for the offline estimator by formulating a dual control system.

3.3 Stability results

In this section, we show that the convergence of the Riccati equations (3.7) and (3.16) is a necessary and sufficient condition for stability of the offline estimator and the controller. We will first state this for the controller and then show that the conditions for the estimator are dual to the control part. Here, it is again important to note that both the controller and estimator depend on the channel statistics γ and ϱ , respectively. Bounds on γ and μ for when the Riccati equations converge are discussed in Section 3.4.

We consider stochastic stability in the mean square sense, which is defined in Definition 2.3.4. So to check for MSS it is sufficient to check whether $\lim_{k \rightarrow \infty} \|\mathbf{E}\{x_k x_k^T\}\|$ exists.

3.3.1 Controller

In this section we present the conditions for when the the controlled system (3.12) is MSS.

Let

$$\begin{aligned} \Psi_{k+1} &\triangleq \mathbf{E}\{x_{k+1}x_{k+1}^T\} \\ &= \mathbf{E}\left\{\left((A - \rho_k B\bar{L})x_k + \omega_k\right)\left((A - \rho_k B\bar{L})x_k + \omega_k\right)^T\right\}. \end{aligned} \quad (3.23)$$

This can be restated as

$$\Psi_{k+1} = \mathbf{E}\left\{(A - \rho_k B\bar{L})x_k x_k^T (A - \rho_k B\bar{L})^T + \omega_k \omega_k^T\right\}$$

Since ρ_k is *i.i.d.*, we can express this by the recursion

$$\begin{aligned} \Psi_{k+1} &= \mathbf{E}\left\{(A - \rho_k B\bar{L})\mathbf{E}\{x_k x_k^T\}(A - \rho_k B\bar{L})^T\right\} + \mathbf{E}\{\omega_k \omega_k^T\} \\ &= A\Psi_k A^T + \varrho(A - B\bar{L})\Psi_k(A - B\bar{L})^T + \Sigma_\omega, \end{aligned} \quad (3.24)$$

which is linear in Ψ_k .

We now present one of the main results of this section. This result states, among others, that (3.24) is bounded if and only if a solution to the Riccati equation (3.7) exists.

Theorem 3.3.1. *Assume Σ_ω is bounded and the pair (A, B) stabilizable. Then, the following conditions are equivalent:*

1. *The second moment of the state using the offline control law (3.11) is bounded as $k \rightarrow \infty$ i.e.*

$$\lim_{N \rightarrow \infty} \|\Psi_N\| < \infty.$$

2. *The Riccati equation (3.7) is bounded as $N \rightarrow \infty$ i.e.*

$$\lim_{N \rightarrow \infty} \|S_N\| < \infty.$$

3. *All eigenvalues of $\sqrt{1 - \varrho}A$ are strictly within the unit circle.*

4. *The optimal solution to the infinite horizon cost (3.9), is given by*

$$J_\infty^*(x_0) = J_\infty(x_0, \bar{\mathcal{L}}_\infty).$$

Theorem 3.3.1 says that the existence of a solution to (3.8) is necessary and sufficient for MSS and links this to the eigenvalues of the system matrix A and the statistics of the packet dropouts. Further, the control law given by (3.11) minimizes the infinite horizon cost function. The proof of Theorem 3.3.1 is presented in Section 3.3.1.1.

3.3.1.1 Proof of Theorem 3.3.1

To prove the result in Theorem 3.3.1, we first establish that the Riccati equation (3.7) can be expressed as a map that depends on the controller gain (3.6). We then show, by adopting a result from [49], that if (3.16) converges, then the same Riccati map using the offline controller gain (3.10) converges to the same solution. We then establish a link between the map for the offline controller and (3.23). The last step involves showing that the Riccati map converges if and only if (3.23) converges. Finally Theorem 3.3.1 is proved by combining these results.

Next we will present the proof for Theorem 3.3.1. However, we first need a number of results. While lemmas are presented in the following, their proofs have been deferred to Appendix 3.C.

Define for a known matrix L ,

$$H_L \triangleq A - BL.$$

Note that H_L is the deterministic closed-loop response using control law L . Define the map

$$g_L(X) \triangleq (1 - \varrho)AXA^T + \varrho H_L X H_L^T + \Sigma_\omega,$$

which for a known L is linear in X . Then, it follows from (3.24) that

$$\Psi_{k+1} = g_{\bar{L}}(\Psi_k). \tag{3.25}$$

In the next lemma, we show that also the Riccati equation (3.7) can be expressed using a linear map.

Lemma 3.3.2. For every $k \in \mathbb{N}$,

$$S_{k+1} = f_{L_{N-k-1,N}^*}(S_k),$$

with $S_0 = Q$, $L_{k,N}^*$ defined by (3.6) and

$$f_L(X) \triangleq (1 - \varrho) A^T X A + \varrho H_L^T X H_L + Q + \varrho L^T R L. \quad (3.26)$$

For a known sequence of gains \mathcal{L}_N as defined in (3.3), denote the composition

$$f_{\mathcal{L}_N}^N(X) = f_{L_{0,N}} \circ f_{\mathcal{L}_{1,N}}^{N-1}(X),$$

with $f_{\mathcal{L}_{N,N}}^0(X) \triangleq X$. When for all $L_k \in \mathcal{L}_N$ it holds true that $L_k = L_0$, $k = 0, \dots, N-1$, the above can be stated as

$$f_{L_0}^N(X) \triangleq f_{L_0} \circ f_{L_0}^{N-1}(X). \quad (3.27)$$

Now note that by Lemma 3.3.2, the Riccati equation (3.7) can be expressed as

$$S_N = f_{\mathcal{L}_N^*}^N(Q),$$

where \mathcal{L}_N^* is given in Proposition 3.2.1. Compositions of the map $g_L(X)$ can be written similarly. This means that we can state (3.25) as

$$\Psi_{k+1} = g_L^k(\Psi_0). \quad (3.28)$$

In the following, we denote

$$\begin{aligned} L^* &\triangleq \arg \min_L \text{trace}(f_L(X)) \\ \mathcal{L}_N^* &\triangleq \arg \min_{\mathcal{L}_N} \text{trace}(f_{\mathcal{L}_N}^N(X)), \end{aligned}$$

such that

$$\begin{aligned} \text{trace}(f_{L^*}(X)) &= \min_L \text{trace}(f_L(X)) \\ \text{trace}(f_{\mathcal{L}_N^*}^N(X)) &= \min_{\mathcal{L}_N} \text{trace}(f_{\mathcal{L}_N}^N(X)). \end{aligned}$$

The following lemma is adopted from [49, Theorem 5.2] and is needed for the proof of Theorem 3.3.1. The lemma states that if there exists a solution to the Riccati equation (3.7) as $k \rightarrow \infty$ then, applying the offline control gain (3.10) in (3.27) will result in the same solution.

Lemma 3.3.3. The following holds true

$$\lim_{k \rightarrow \infty} S_k = \lim_{k \rightarrow \infty} f_L^k(Q).$$

For later reference, we introduce the following two lemmas. The first states that, for a given L , the recursions induced by the map f_L are stable if and only if those induced by g_L are stable. The second states necessary and sufficient conditions for the maps $f_L(X)$ and $g_L(X)$ to be stable.

Lemma 3.3.4. *For any $X > 0$ and L , the following holds true*

$$\lim_{k \rightarrow \infty} \|f_L^k(X)\| < \infty \iff \lim_{k \rightarrow \infty} \|g_L^k(X)\| < \infty.$$

Lemma 3.3.5. *For all X , it holds true that*

$$\begin{aligned} \lim_{N \rightarrow \infty} \|f_L^N(X)\| < \infty &\iff \sigma\left(\sqrt{(1-\varrho)A}\right) < 1 \\ \lim_{N \rightarrow \infty} \|g_L^N(X)\| < \infty &\iff \sigma\left(\sqrt{(1-\varrho)A}\right) < 1 \end{aligned}$$

where $\sigma(A)$ is the spectral radius of A .

We are now ready to present the proof of Theorem 3.3.1.

Proof of Theorem 3.3.1. By (3.28) we have that

$$\lim_{N \rightarrow \infty} \|\Psi_N\| < \infty \iff \lim_{N \rightarrow \infty} \|g_L^N(\Psi_0)\| < \infty. \quad (3.29)$$

By Lemma 3.3.5 we have that

$$\lim_{N \rightarrow \infty} \|g_L^N(\Psi_0)\| < \infty \iff \lim_{N \rightarrow \infty} \|g_L^N(X)\| < \infty \quad \forall X \quad (3.30)$$

if and only if all eigenvalues of $\sqrt{1-\varrho}A$ are strictly within the unit circle. By Lemma 3.3.4 it follows that

$$\lim_{N \rightarrow \infty} \|g_L^N(X)\| < \infty \iff \lim_{N \rightarrow \infty} \|f_L^N(X)\| < \infty \quad \forall X. \quad (3.31)$$

Since this holds for all X , it also holds for Q . Thus, from Lemma 3.3.3 we have that

$$\lim_{N \rightarrow \infty} \|f_L^N(Q) < \infty\| \iff \lim_{N \rightarrow \infty} \|S_N\| < \infty. \quad (3.32)$$

Equivalence between Items 1 and 2 then follows by combining (3.29) to (3.32). The necessary and sufficient condition in Item 3 follows directly from Lemma 3.3.5. Then, combining the results in Theorem 3.3.1 and Lemmas 3.3.3 and 3.3.5 and the fact that $\bar{S} \triangleq \lim_{k \rightarrow \infty} S_k$ establishes the equivalence between Items 1 to 3 and Item 4. □

3.3.2 Offline estimator

For the estimator, we show that the convergence of (3.16) is necessary and sufficient for MSS of the offline estimator. This is done by reformulating the estimator design as a dual controller design problem. Also, we reformulate the offline estimator MSS condition as a dual controller MSS condition.

First, define the estimation error of the offline estimator (3.21) by

$$\begin{aligned} e_k &\triangleq x_k - \hat{x}_k \\ &= Ax_{k-1} + \omega_{k-1} - \left((A - \gamma_{k-1}\bar{K}C) \hat{x}_{k-1} + \gamma_{k-1}\bar{K}y_{k-1} \right). \end{aligned}$$

Controller	A	B	Q	R	ϱ	ρ_k	$L_{N-k,N}$	S_k	\bar{L}	\bar{S}	x_k	ω_k
Estimator	A^T	C^T	Σ_ω	Σ_v	μ	γ_k	$K_{k,N}^T$	P_k	\bar{K}^T	\bar{P}	e_k	$\omega_k + \gamma_k \bar{K} v_k$

Table 3.1: Substitutions to formulate the estimator as a dual controller or vice versa.

By (3.1) we have that $y_k = \gamma_k (Cx_k + v_k)$. Inserting this in the above, collecting the terms and noting that $\gamma_k^2 = \gamma_k$ results in the recursion

$$\begin{aligned} e_k &= (A - \gamma_{k-1} \bar{K} C) (x_{k-1} - \hat{x}_{k-1}) + \omega_{k-1} - \gamma_{k-1} \bar{K} v_{k-1} \\ &= (A - \gamma_{k-1} \bar{K} C) e_{k-1} + \omega_{k-1} - \gamma_{k-1} \bar{K} v_{k-1}. \end{aligned} \quad (3.33)$$

The error covariance of the offline estimator is given by

$$\Upsilon_k \triangleq \mathbf{E} \{e_k e_k^T\}. \quad (3.34)$$

The following result shows that the boundedness of the solution to the estimator Riccati equation (3.16) as $k \rightarrow \infty$ is necessary and sufficient for MSS of the offline estimator (3.21). Further, it links the existence of the solution to (3.16) with the eigenvalues of the system matrix A and the optimal solution to the infinite horizon cost function (3.19).

Theorem 3.3.6. *Consider P_k from (3.16) and the pair (C, A) detectable. Then, the following conditions are equivalent:*

1. *The second moment of the prediction error of the offline estimator (3.21) is bounded as $k \rightarrow \infty$ i.e.*

$$\lim_{k \rightarrow \infty} \|\Upsilon_k\| < \infty.$$

2. *The estimator Riccati equation (3.16) is bounded as $k \rightarrow \infty$ i.e.*

$$\lim_{k \rightarrow \infty} \|P_k\| < \infty.$$

3. *All eigenvalues of $\sqrt{1 - \mu}A$ are strictly within the unit circle.*
4. *The offline estimator scheme (3.21) minimizes the cost (3.18) i.e.*

$$V^* = V(\bar{K}_\infty).$$

Theorem 3.3.6 shows that the offline estimator is MSS if and only if there exists a solution to the estimator Riccati equation (3.16). It also shows, that the offline estimator minimizes the infinite horizon cost (3.19). The proof of Theorem 3.3.6 is deferred to Section 3.3.2.2.

3.3.2.1 Duality

Before showing the proof of Theorem 3.3.6, we establish a form of duality between the offline estimator and the controller. This allows one to formulate the estimator into a dual controller and vice-versa. The dual formulation can be obtained using the mappings that are presented in

Table 3.1. For example, using the mappings from Table 3.1, the controller Riccati equation (3.7) becomes the estimator Riccati equation (3.16) and the control gain (3.6) becomes the estimator gain (3.15). Thus, the optimal offline estimator can be obtained by formulating the estimator design problem into a dual controller design and compute the control Riccati equation (3.7) and control gain (3.6). The obtained control gain (3.6) can then be mapped to the offline estimator gain (3.15) using the maps in Table 3.1. Similarly, the optimal controller can be obtained by formulating a dual estimator problem using the mappings in Table 3.1.

Also, an interesting observation is, that the estimator error dynamics from (3.33),

$$e_k = (A - \gamma_{k-1}\bar{K}C) e_{k-1} + \omega_{k-1} - \gamma_{k-1}\bar{K}v_{k-1}$$

can, using the mappings in Table 3.1, be reformulated to the dual closed-loop dynamics of the controller. This results in

$$\begin{aligned} x_k &= (A^T - \rho_{k-1}\bar{L}^T B^T) x_{k-1} + \omega_{k-1} \\ &= (A - \rho_{k-1}B\bar{L})^T x_{k-1} + \omega_{k-1}, \end{aligned}$$

which is on the same form as (3.12).

3.3.2.2 Proof of Theorem 3.3.6

To prove the result that is presented in Theorem 3.3.6, we first show that the estimator Riccati equation (3.16) can be stated as a map that depends on the optimal estimator gain (3.15). We then show that also the error covariance of the offline estimator (3.34) can be stated using the same Riccati map using the offline estimator gain (3.20). We then, using the substitutions from Table 3.1 establish duality between the estimator and controller Riccati equations. The proof of Theorem 3.3.6 is then done by combining these results.

First, define the map

$$h_K(P) \triangleq (1 - \mu)APA^T + \mu(A - KC)P(A - KC)^T + \Sigma_\omega + \mu K\Sigma_v K^T. \quad (3.35)$$

The relation to the estimation Riccati equation and the map (3.35) is stated in the following lemma.

Lemma 3.3.7. *The optimal error covariance for the estimator is given by*

$$P_{k+1} = h_{K_{k,N}^*}(P_k), \quad (3.36)$$

where $K_{k,N}^*$ is defined in (3.15). Also,

$$\Upsilon_{k+1} = h_{\bar{K}}(\Upsilon_k). \quad (3.37)$$

Proof. The proof is found in Appendix 3.C. □

Corollary 3.3.8. *When using the substitutions from Table 3.1 in (3.26), it follows directly that*

$$h_{\bar{K}}^N(P_k) = f_{\bar{K}^T}^N(P_k) \quad \forall N, k.$$

Corollary 3.3.8 establishes duality between the offline estimator map $h_{\bar{K}}$ and the controller map $f_{\bar{L}}$ for the dual control system.

We are now ready to state the proof for Theorem 3.3.6. This proof utilizes the duality between the estimator and controller.

Proof of Theorem 3.3.6. By Lemma 3.3.7 we have

$$\begin{aligned}\Upsilon_k &= h_{\bar{K}}(\Upsilon_{k-1}) \\ &= h_{\bar{K}}^k(\Upsilon_0)\end{aligned}$$

for any $\Upsilon_0 > 0$. Thus,

$$\lim_{k \rightarrow \infty} \|\Upsilon_k\| < \infty \iff \lim_{k \rightarrow \infty} \|h_{\bar{K}}^k(\Upsilon_0)\| < \infty. \quad (3.38)$$

By Corollary 3.3.8 it follows directly that

$$\lim_{k \rightarrow \infty} \|h_{\bar{K}}^k(\Upsilon_0)\| < \infty \iff \lim_{k \rightarrow \infty} \|f_{\bar{L}}^k(\Upsilon_0)\| < \infty. \quad (3.39)$$

Then, by Lemma 3.3.4

$$\lim_{k \rightarrow \infty} \|f_{\bar{L}}^k(\Upsilon_0)\| < \infty \iff \lim_{k \rightarrow \infty} \|g_{\bar{L}}^k(\Upsilon_0)\| < \infty. \quad (3.40)$$

Since this holds for any $\Upsilon_0 > 0$ it must also hold for $\Upsilon_0 = P_0$. It then follows by Theorem 3.3.1 that

$$\lim_{k \rightarrow \infty} \|g_{\bar{L}}^k(P_0)\| < \infty \iff \lim_{k \rightarrow \infty} \|S_k\| < \infty. \quad (3.41)$$

Finally, using the substitutions in Table 3.1 we have that

$$S_k = P_k. \quad (3.42)$$

The equivalence between Items 1 and 2 then follows by combining (3.38) to (3.41) and substituting (3.42). By the duality that was established in Corollary 3.3.8, the equivalence in Items 3 and 4 and their link to Items 1 and 2 follows directly by Lemmas 3.3.2 and 3.3.4. \square

3.4 Convergence of the Riccati equation

In Section 3.3 we showed that the boundedness of the Riccati equations (3.7) and (3.16) is necessarily and sufficient for MSS of the closed-loop control system or estimator. While we for the ease of exposure discuss the results for the estimator, they hold true for the controller due to the duality. Item 3 in Theorem 3.3.6 states that convergence of the Riccati equation requires that

$$\left| \sigma \left(\sqrt{(1 - \mu) A} \right) \right| < 1.$$

What would be interesting here, is to find the bound on μ for which a solution to (3.16) exists.

It is worth noting that the form of the Riccati equations (3.7) and (3.16) are similar to the expression that the authors of [104] use to determine an upper bound for the performance of their online Kalman filter. The results on these bounds can therefore be applied directly on (3.7) and

(3.16). The authors of [104] found that if the unstable poles can be canceled such that $A - \bar{K}C = 0$, the lower bound on μ for which a solution to (3.16) exists, is given by

$$\underline{\mu} = 1 - \frac{1}{\sigma(A)^2}. \quad (3.43)$$

When multiple unstable poles are present and C is not invertible, it is more difficult to find bounds on μ . The work [104] presented a numerical method to find the bound on μ by solving a semi-definite program (SDP).

The more recent work [102] stated an analytical expression to find $\underline{\mu}$. The idea is to split the unstable subspace of A and C into partitions. Each partition is defined by a triple (A, C, \mathcal{I}_n) , where we for each $\mathcal{I}_n = \{i_1, \dots, i_\ell\}$, where $1 \leq i_1 < i_2 < \dots < i_\ell \leq n$, have the partitioned matrices $A(\mathcal{I}_n)$ and $C(\mathcal{I}_n)$. Here, the matrix $A(\mathcal{I}_n)$ is constructed by selecting a block diagonal part of A by selecting the elements $A(i, j)$ where $i, j \in \mathcal{I}_n$. Also, $C(\mathcal{I}_n) = [C_{i_1}, \dots, C_{i_\ell}]$. In total it is, for a $n \times n$ unstable subspace of A , possible to construct $2^n - 1$ different partitions. It is then possible to find a bound on μ for each of these partitions. The bound on μ for the system (3.1) is then found by the maximum $\underline{\mu}$ of the sub systems. This is given by the expression

$$\underline{\mu} = \max_{\mathcal{I}_n} \left(1 - \frac{1}{r(\mathcal{I}_n) \sqrt{|\bar{A}_u(\mathcal{I}_n)|^2}} \right), \quad (3.44)$$

where

$$r(\mathcal{I}_n) \triangleq \text{rank} \{ \bar{C}_u(\mathcal{I}_n) \}.$$

Note that with only one unstable eigenvalue in A , (3.43) equals (3.44).

3.5 Simulations

In this section, we illustrate the presented concepts in a numerical example where we separately design a controller and estimator. We design the estimator by formulating the estimation problem into a dual control problem. We also compare the performance of the offline estimator to the online Kalman filter and the JLS pseudo-offline design [108]. We consider the following system dynamics

$$A = \begin{bmatrix} 1.2 & 1 & 0 \\ 0 & 0.9 & 1 \\ 0 & 0 & 0.6 \end{bmatrix}$$

with the eigenvalues given on the diagonal,

$$B = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

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

Notice that the pairs (A, B) and (A^T, C^T) are both fully controllable. Further let $\Sigma_\omega = I_3$, where I_n is the $n \times n$ identity matrix, and $\Sigma_v = 1$. Packet dropouts are acknowledged, such that the estimator at time k knows both ρ_k and γ_k .

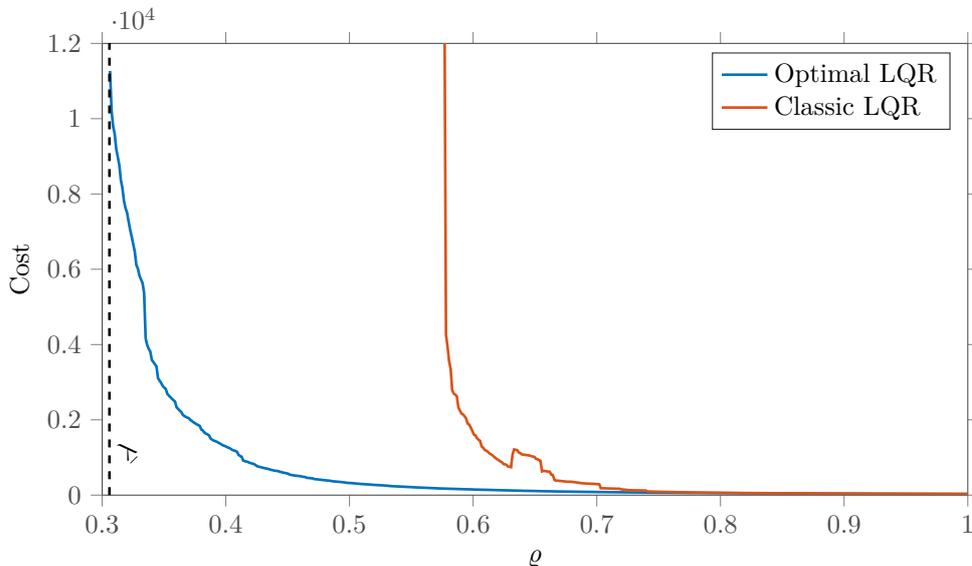


Figure 3.3: The cost $x_k^T Q x_k + u_k^T R u_k$ for the controller for different ρ . The lower bound $\underline{\rho}$, calculated using (3.43), is plotted with the dashed line. The classical LQR that does not take packet dropouts into account is shown for comparison

3.5.1 Controller

We first design the controller. Since there is only one unstable pole in A , we can use (3.43) to find the bound $\underline{\rho} = 0.3059$. This means that it is not possible to find a stabilizing controller if $\rho < 0.3059$. The performance of the controller for different ρ is shown in Figure 3.3. The results are obtained by averaging over 1000 simulations each of 1000 time-steps.. As expected, the optimal controller from Proposition 3.2.1 and the classical LQR perform identical for $\rho = 1$. However, when ρ decreases, the performance of the classical LQR degrades significantly. The classical LQR can no longer maintain stability for $\rho < 0.55$. On the other hand, the optimal LQR that takes the packet dropouts into account is able to maintain stability until ρ reaches the lower bound $\underline{\rho}$. The performance of the optimal LQR that takes the packet dropouts into account is significantly better than the classical LQR. Even at $\rho = 0.58$, where the classical LQR reaches a cost larger than 10000, the cost of the proposed design is merely 206. Also observe how the cost diverges as ρ approaches the lower bound $\underline{\rho}$.

As an example of the controller design. Consider the success probability $\rho = 0.5$, and the parameters $Q = I_3$ and $R = 0.1$. This results in the control gain

$$\bar{L} = \begin{bmatrix} 0.3422 & 0.9728 & 1.3638 \end{bmatrix},$$

which gives closed-loop eigenvalues located at $0.6677 \pm 0.045j$ and 0.0007 .

3.5.2 Estimator

We design the estimator by reformulating it as a dual control problem, as explained in Section 3.3.2.1. It is straightforward to see that in this case $\underline{\mu} = \underline{\rho} = 0.3059$.

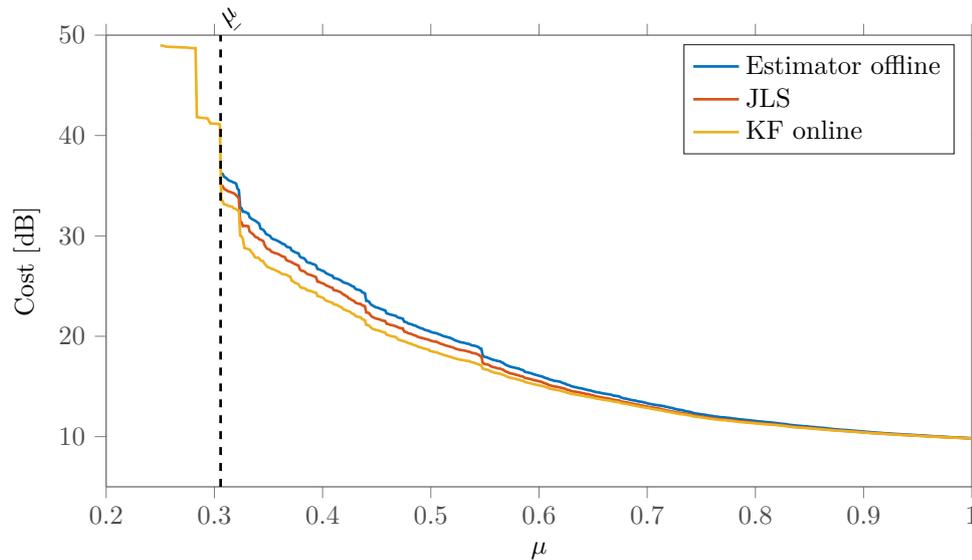


Figure 3.4: The cost $e_k^T e_k$ in dB for the different estimators for varying μ . The dashed horizontal line illustrates the lower bound $\underline{\mu}$, calculated using (3.43).

The performance of the offline estimator for different values of μ is shown in Figure 3.4. Here $\underline{\mu}$ is marked with the dashed line. The performance for the online Kalman filter [104] and for JLS with a loss horizon of length 2 [108] are plotted as well for comparison. Note that the JLS with horizon 1 would result in exactly the same performance as the offline estimator. It is worth noting that, while for small values of μ the Kalman filter and JLS estimator perform significantly better than the offline estimator, the performance difference decreases significantly as μ increases.

As an example for the estimator design. Consider a success probability of $\mu = 0.5$. This results in the estimator gain

$$\bar{K} = \begin{bmatrix} 1.3468 & 0.1622 & 0.007 \end{bmatrix}^T,$$

which results in closed-loop eigenvalues at $0.6693 \pm 0.0959j$ and 0.0075.

3.6 Summary

We established a new link between the design of an offline estimator and a LQR controller for systems affected by packet loss that is modelled by *i.i.d* distributions. We have designed an estimator and a controller, both being offline in the sense of minimizing a cost function based solely on system and channel statistics, and therefore minimizing the amount of online computations. We have pointed out the duality of these two designs which permits the estimator design for a given system to be performed by formulating a dual system, and then design a controller for the latter. We have therefore extended the classical duality that exists between a LQR and a Kalman filter to the case of systems that are affected by random packet loss.

Appendix

3.A Proof of Proposition 3.2.1

Denote the optimal cost $J_N^*(x_0) \triangleq J(x_0, \mathcal{L}_N^*)$. Note that with controls of the form $u_k = -L_{k,N}x_k$, the closed-loop system becomes $x_{k+1} = (A - \rho_k B L_{k,N})x_k + \omega_k$. Then, for any $1 < n \leq N$,

$$J_N^*(x_0) = \min_{(L_{0,N}, \dots, L_{N-1,N})} \frac{1}{N} \mathbf{E} \left\{ \sum_{k=0}^{N-(n-1)} \|x_k\|_Q^2 + \|\rho_k L_{k,N} x_k\|_R^2 + n J_n^*(x_{N-n}) \middle| x_0 \right\},$$

where

$$\begin{aligned} J_0^*(x_N) &= \mathbf{E} \left\{ \|x_N\|_Q^2 \middle| x_N \right\} \\ J_1^*(x_{N-1}) &= \min_{L_{N-1,N}} \mathbf{E} \left\{ \|x_{N-1}\|_Q^2 + \|\rho_{N-1} L_{N-1,N} x_{N-1}\|_R^2 + J_0^*(x_N) \middle| x_{N-1} \right\} \end{aligned}$$

and for $2 \leq k \leq N$

$$J_k^*(x_{N-k}) = \frac{1}{k} \min_{L_{k,N}} \mathbf{E} \left\{ \|x_k\|_Q^2 + \|\rho_k L_{k,N} x_k\|_R^2 + (k-1) J_{k-1}^*(x_{N-k+1}) \middle| x_k \right\}.$$

The argument then proceeds by induction. In step 0, we obtain

$$J_0^*(x_N) = \mathbf{E} \left\{ \|x_N\|_Q^2 \middle| x_N \right\} = x_N^T Q x_N.$$

Then, at step $n-1$, where $2 \leq n < N$, we assume that there exist S_{n-1} and Y_{n-1} such that

$$J_{n-1}^*(x_{N-(n-1)}) = \frac{1}{n-1} \left(x_{N-(n-1)}^T S_{n-1} x_{N-(n-1)} + Y_{n-1} \right),$$

where $S_0 = Q$ and $Y_0 = 0$. Then, at step n we have

$$\begin{aligned} J_n(x_{N-n}, (L_{N-n,N}, L_{N-n+1,N}^*, \dots, L_{N-1,N}^*)) &= \\ &= \frac{1}{n} \mathbf{E} \left\{ \|x_{N-n}\|_Q^2 + \|\rho_{N-n} L_{N-n,N} x_{N-n}\|_R^2 + (n-1) J_{n-1}^*(x_{N-n+1}) \middle| x_{N-n} \right\} \\ &= \frac{1}{n} \mathbf{E} \left\{ x_{N-n}^T Q x_{N-n} + \rho_{N-n}^2 x_{N-n}^T L_{N-n,N}^T R L_{N-n,N} x_{N-n} \right. \\ &\quad \left. + (n-1) J_{n-1}^*((A - \rho_{N-n} B L_{N-n,N}) x_{N-n} + \omega_{N-n}) \middle| x_{N-n} \right\} \\ &= \frac{1}{n} x_{N-n}^T \left(\varrho L_{N-n,N}^T (R + B^T S_{n-1} B) L_{N-n,N} - \varrho L_{N-n,N}^T B^T S_{n-1} A + \varrho A^T S_{n-1} B L_{N-n,N} \right. \\ &\quad \left. + Q + A^T S_{n-1} A \right) x_{N-n} + \frac{1}{n} \left(\text{trace}(\Sigma_\omega S_{n-1}) + Y_{n-1} \right). \end{aligned}$$

Let $M \triangleq \varrho (R + B^T S_{n-1} B)$ and add and subtract $x_{N-n}^T (\varrho^2 A^T S_{n-1} B M^{-1} B^T S_{n-1} A) x_{N-n}$ to the above. Rewriting this results in

$$\begin{aligned} J_n(x_{N-n}, (L_{N-n,N}, L_{N-n+1,N}^*, \dots, L_{N-1,N}^*)) &= \\ &= \frac{1}{n} x_{N-n}^T \left(Q + A^T S_{n-1} A - \varrho^2 A^T S_{n-1} B M^{-1} B^T S_{n-1} A \right. \\ &\quad \left. + \left(M^{\frac{1}{2}} L_{N-n,N} - \varrho M^{-\frac{1}{2}} B^T S_{n-1} A \right)^T \left(M^{\frac{1}{2}} L_{N-n,N} - \varrho M^{-\frac{1}{2}} B^T S_{n-1} A \right) \right) x_{N-n} \\ &\quad + \frac{1}{n} \left(\text{trace}(\Sigma_\omega S_{n-1}) + Y_{n-1} \right). \end{aligned}$$

Clearly, the above is minimized when

$$\left(M^{\frac{1}{2}} L_{N-n,N} - \varrho M^{-\frac{1}{2}} B^T S_{n-1} A \right) = 0.$$

Solving this for $L_{N-n,N}$ results in

$$\begin{aligned} L_{N-n,N}^* &= \varrho M^{-1} B^T S_{n-1} A \\ &= (R + B^T S_{n-1} B)^{-1} B^T S_{n-1} A. \end{aligned}$$

Then, the minimum cost is given by

$$\begin{aligned} J_n^*(x_{N-n}) &= \frac{1}{n} x_{N-n}^T \left(Q + A^T (S_{n-1} - \varrho S_{n-1} B (R + B^T S_{n-1} B)^{-1} B^T S_{n-1}) A \right) x_{N-n} \\ &\quad + \frac{1}{n+1} \left(\text{trace}(\Sigma_\omega S_{n-1}) + Y_{n-1} \right) \\ &= \frac{1}{n} \left(x_{N-n}^T S_n x_{N-n} + Y_n \right), \end{aligned}$$

where S_n is given in (3.7) and

$$Y_n = \sum_{\ell=0}^{n-1} \text{trace}(\Sigma_\omega S_\ell).$$

Proceeding as in the above, one obtains the optimal cost

$$J_N^*(x_0) = \frac{1}{N} x_0^T S_N x_0 + \frac{1}{N} \sum_{\ell=0}^{N-1} \text{trace}(\Sigma_\omega S_\ell).$$

When letting $N \rightarrow \infty$, the cost becomes

$$\begin{aligned} J_\infty^* &= \lim_{N \rightarrow \infty} J_N^*(x_1) \\ &= \lim_{N \rightarrow \infty} \left(\frac{1}{N} x_1^T S_N x_1 + \frac{1}{N} \sum_{\ell=0}^{N-1} \text{trace}(\Sigma_\omega S_\ell) \right), \end{aligned} \tag{3.A.1}$$

where, provided that $\lim_{N \rightarrow \infty} S_N$ exists, the first term in (3.A.1) goes to zero. This results in

$$\begin{aligned} J_\infty^* &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{\ell=0}^{\infty} \text{trace}(\Sigma_\omega S_\ell) \\ &= \text{trace}(\Sigma_\omega \bar{S}) \end{aligned}$$

where \bar{S} is the asymptotic solution to (3.8). □

3.B Proof of Proposition 3.2.2

Define the estimation error at time k by

$$\tilde{e}_k \triangleq x_k - \tilde{x}_k.$$

Then, the cost (3.14) at time $k = 0$ is given by

$$E(K_0) = \mathbf{E} \left\{ \|\tilde{e}_0\|_2^2 \right\} = \|\mathbf{E} \{\tilde{e}_0\}\|_2^2 = \text{trace}(\mathbf{E} \{\tilde{e}_0 \tilde{e}_0^T\}).$$

Now since $\hat{x}_0 \triangleq \mathbf{E}\{x_0\} = 0$, we have that

$$E^* = \text{trace}(\mathbf{E}\{x_0 x_0^T\}) = \text{trace}(P).$$

Define

$$P_k \triangleq \mathbf{E}\{\tilde{e}_k \tilde{e}_k^T\},$$

where

$$\begin{aligned} P_{k+1} &= \mathbf{E}\{\tilde{e}_{k+1} \tilde{e}_{k+1}^T\} \\ &= \mathbf{E}\left\{\left((A - \gamma_k K_k C) \tilde{e}_k + \omega_k - \gamma_k K_k v_k\right) \left((A - \gamma_k K_k C) \tilde{e}_k + \omega_k - \gamma_k K_k v_k\right)^T\right\} \\ &= \mathbf{E}\left\{(A - \gamma_k K_k C) \tilde{e}_k \tilde{e}_k^T (A^T - \gamma_k C^T K_k^T) + \omega_k \omega_k^T + \gamma_k^2 K_k v_k v_k^T K_k^T\right\} \\ &= \underbrace{AP_k A^T + \Sigma_\omega}_W + K_k \underbrace{\mu (CP_k C^T + \Sigma_v)}_M K_k^T - \mu AP_k C^T K_k^T - \mu K_k C P_k A^T \\ &= W + K_k M K_k^T - \mu AP_k C^T K_k^T - \mu K_k C P_k A^T. \end{aligned}$$

By adding and subtracting $\mu^2 AP_k C^T M^{-1} CP_k A^T$ on the right hand side of the last equation, we obtain

$$\begin{aligned} P_{k+1} &= W + \left(K_k M^{\frac{1}{2}} - \mu AP_k C^T M^{-\frac{1}{2}}\right) \left(K_k M^{\frac{1}{2}} - \mu AP_k C^T M^{-\frac{1}{2}}\right)^T \\ &\quad - \mu^2 AP_k C^T M^{-1} CP_k A^T. \end{aligned} \quad (3.B.1)$$

Here $\text{trace}(P_{k+1})$ is minimized with respect to K_k when the term

$$0 = \left(K_k M^{\frac{1}{2}} - \mu AP_k C^T M^{-\frac{1}{2}}\right) \left(K_k M^{\frac{1}{2}} - \mu AP_k C^T M^{-\frac{1}{2}}\right)^T.$$

This occurs only if

$$\begin{aligned} 0 &= K_k M^{\frac{1}{2}} - \mu AP_k C^T M^{-\frac{1}{2}} \\ K_k &= \mu AP_k C^T M^{-1} \\ &= AP_k C^T (CP_k C^T + \Sigma_v)^{-1}, \end{aligned}$$

which is (3.15), and (3.16) follows directly from (3.B.1).

Now, since P_{k+1} is a function of P_k , we have that if $\lim_{k \rightarrow \infty} P_k < \infty$, then the stationary cost given by

$$\begin{aligned} V(\mathcal{K}_\infty^*) &= \lim_{k \rightarrow \infty} \text{trace}(P_k) \\ &= \text{trace}(\bar{P}), \end{aligned}$$

where \bar{P} is the asymptotic solution to (3.17) □

3.C Proof of lemmas in Section 3.3

Proof of Lemma 3.3.2. From (3.7), we have

$$S_{k+1} = Q + (1 - \varrho) A^T S_k A + \varrho M, \quad (3.C.1)$$

where

$$\begin{aligned} M &= A^T S_k A - A^T S_k B (R + B^T S_k B)^{-1} B^T S_k A \\ &= A^T S_k A - A^T S_k B (R + B^T S_k B)^{-1} (R + B^T S_k B) (R + B^T S_k B)^{-1} B^T S_k A. \end{aligned} \quad (3.C.2)$$

To simplify notation in this proof, define

$$L_{S_k} \triangleq L_{N-k-1, N}^* \quad k = 0, \dots, N-1, \quad (3.C.3)$$

where $L_{k, N}$ is the control gain (3.6). Substituting (3.C.3) into (3.C.2) results in

$$M = A^T S_k A - L_{S_k}^T (R + B^T S_k B) L_{S_k}. \quad (3.C.4)$$

Also with $H_{L_{S_k}} \triangleq A - B L_{S_k}$ we have that

$$A^T S_k A = H_{L_{S_k}}^T S_k H_{L_{S_k}} - L_{S_k}^T B^T S_k B L_{S_k} + A^T S_k B L_{S_k} + L_{S_k}^T B^T S_k A \quad (3.C.5)$$

and

$$A^T S_k B L_{S_k} = L_{S_k}^T (R + B^T S_k B) L_{S_k}. \quad (3.C.6)$$

Inserting (3.C.5) and (3.C.6) into (3.C.4) yields

$$M = H_{L_{S_k}}^T S_k H_{L_{S_k}} + L_{S_k}^T R L_{S_k}.$$

Replacing the above into (3.C.1) leads to

$$\begin{aligned} S_{k+1} &= (1 - \varrho) A^T S_k A + \varrho H_{L_{S_k}}^T S_k H_{L_{S_k}} + Q + \varrho L_{S_k}^T R L_{S_k} \\ &= f_{L_{S_k}}(S_k), \end{aligned}$$

and the result follows by substituting (3.C.3) in the above. \square

For the next result, define for any $X \geq 0$

$$\begin{aligned} L^* &\triangleq \arg \min_L \text{trace}(f_L(X)) \\ \mathcal{L}_N^* &\triangleq \arg \min_{\mathcal{L}_N} \text{trace}(f_{\mathcal{L}_N}^N(X)) \end{aligned}$$

such that

$$\begin{aligned} \text{trace}(f_{L^*}(X)) &= \min_L \text{trace}(f_L(X)) \\ \text{trace}(f_{\mathcal{L}_N^*}^N(X)) &= \min_{\mathcal{L}_N} \text{trace}(f_{\mathcal{L}_N}^N(X)). \end{aligned}$$

It follows directly from (3.C.3) and Proposition 3.2.1 that

$$\begin{aligned} f_{L^*}(S_k) &= f_{L_{S_k}}(S_k) \\ &= f_{L_{N-k-1, N}^*}(S_k) \\ f_{\mathcal{L}_N^*}^N(Q) &= S_N. \end{aligned}$$

We can now present the next result. This results shows that $f_L(X)$ in (3.26) is monotonic, and is needed later.

Lemma 3.C.1. For $0 \leq X \leq Y$, it holds true that $0 \leq f_{L^*}(X) \leq f_{L^*}(Y)$, and for all N , it holds true that $0 \leq f_{\mathcal{L}_N^*}^N(X) \leq f_{\mathcal{L}_N^*}^N(Y)$.

Proof. Denote $L_X \triangleq \arg \min_L \text{trace}(f_L(X))$. Then, it follows from Proposition 3.2.1 and Lemma 3.3.2 that

$$f_{L^*}(X) = f_{L_X}(X) \leq f_L(X) \quad \forall L.$$

Also notice that, for all L

$$f_L(X) \leq f_L(Y).$$

Putting this together yields

$$f_{L^*}(X) = f_{L_X}(X) \leq f_{L_Y}(X) \leq f_{L_Y}(Y) = f_{L^*}(Y).$$

It then follows immediately that this holds true for all N . \square

Proof of Lemma 3.3.3. By (3.26) it is easy to verify that for $0 \leq Q$

$$Q \leq f_{L^*}(Q),$$

and therefore by Lemma 3.C.1 also for all N

$$Q \leq f_{\mathcal{L}_{N-1}^*}^{N-1}(Q) \leq f_{\mathcal{L}_N^*}^N(Q). \quad (3.C.7)$$

Also, since

$$\bar{S} \triangleq \lim_{k \rightarrow \infty} S_k = \lim_{N \rightarrow \infty} f_{\mathcal{L}_N^*}^N(Q), \quad (3.C.8)$$

we have that $Q \leq \bar{S}$. It is easy to verify that for all N , $\bar{S} = f_{\bar{L}}^N(\bar{S})$ where \bar{L} is defined in (3.10). Then, by (3.C.7) and Lemma 3.C.1 we have that for all N ,

$$f_{\mathcal{L}_N^*}^N(Q) \leq f_{\bar{L}}^N(Q) \leq f_{\bar{L}}^N(\bar{S}). \quad (3.C.9)$$

This must therefore also hold on the limit. Combining (3.C.8) and (3.C.9) and taking the limits yields

$$\bar{S} = \lim_{N \rightarrow \infty} f_{\mathcal{L}_N^*}^N(Q) \leq \lim_{N \rightarrow \infty} f_{\bar{L}}^N(Q) \leq \lim_{N \rightarrow \infty} f_{\bar{L}}^N(\bar{S}) = \bar{S},$$

and the result follows. \square

Proof of Lemma 3.3.4. Using properties of the Kronecker product, we can state

$$\begin{aligned} \text{vec}(f_L(X)) &= F_L \text{vec}(X) + \text{vec}(K_L) \\ \text{vec}(g_L(X)) &= G_L \text{vec}(X) + \text{vec}(M), \end{aligned} \quad (3.C.10)$$

where $K_L \triangleq Q + \varrho L^T R L$, $M \geq 0$ and $\text{vec}(\cdot)$ denotes the operation of converting a matrix into a vector by stacking its columns. The matrices

$$\begin{aligned} F_L &\triangleq (1 - \varrho)(A^T \otimes A^T) + \varrho(H_L^T \otimes H_L^T) \\ G_L &\triangleq (1 - \varrho)(A \otimes A) + \varrho(H_L \otimes H_L). \end{aligned} \quad (3.C.11)$$

Also,

$$\begin{aligned}\text{vec}(f_L^N(X)) &= F_L^N \text{vec}(f_L^{N-1}(X)) + \text{vec}(K_L) \\ &= F_L^N \text{vec}(X) + \sum_{n=0}^{N-1} F_L^n \text{vec}(K_L).\end{aligned}$$

As $N \rightarrow \infty$, the above is bounded if and only if F_L^N tends to zero. The result then follows since

$$F_L = G_L^T.$$

□

Proof of Lemma 3.3.5. For all N and all $X \geq 0$ we have from (3.C.10) that

$$\text{vec}(f_L^N(X)) = F_L^N \text{vec}(X) + \sum_{n=0}^{N-1} F_L^n \text{vec}(K_L).$$

Taking the limit yields

$$\lim_{N \rightarrow \infty} \text{vec}(f_L^N(X)) = \lim_{N \rightarrow \infty} F_L^N \text{vec}(X) + \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} F_L^n \text{vec}(K_L).$$

Here the first part on the right hand side goes to zero if and only if $\lim_{N \rightarrow \infty} F_L^N = 0$ and the second part on the right hand side converges to $\text{vec}(K_L)$ if and only if $\lim_{N \rightarrow \infty} F_L^N = 0$. This requires all eigenvalues of F_L to be within the unit circle. By the definition of F_L in (3.C.11), this is equivalent to all eigenvalues of $\sqrt{(1-\varrho)}A$ being inside the unit circle. The result for g_L^N follows by Lemma 3.3.4. □

Proof of Lemma 3.3.7. Using the substitutions from Table 3.1 in (3.7), the control Riccati equation becomes the estimator Riccati equation (3.16) and the map f (defined in (3.26)) turns into h . Equation (3.36) then follows directly from Lemma 3.3.2.

To show (3.37), observe that

$$\begin{aligned}\Upsilon_{k+1} &= \mathbf{E}\{e_{k+1}e_{k+1}^T\} \\ &= \mathbf{E}\left\{\left((A - \gamma_k \bar{K}C)e_k + \omega_k - \gamma_k \bar{K}v_k\right)\left((A - \gamma_k \bar{K}C)e_k + \omega_k - \gamma_k \bar{K}v_k\right)^T\right\} \\ &= \mathbf{E}\left\{\left(A - \gamma_k \bar{K}C\right)e_k e_k^T \left(A - \gamma_k \bar{K}C\right)^T + \omega_k \omega_k^T + \gamma_k \bar{K}v_k v_k^T \bar{K}^T\right\} \\ &= \mathbf{E}\left\{\left(1 - \gamma_k\right)Ae_k e_k^T A^T + \gamma_k \left(A - \bar{K}C\right)e_k e_k^T \left(A - \bar{K}C\right)^T + \omega_k \omega_k^T + \gamma_k \bar{K}v_k v_k^T \bar{K}^T\right\} \\ &= (1 - \mu)A\mathbf{E}\{e_k e_k^T\}A^T + \mu \left(A - \bar{K}C\right)\mathbf{E}\{e_k e_k^T\}\left(A - \bar{K}C\right)^T + \mathbf{E}\{\omega_k \omega_k^T\} \\ &\quad + \mu \bar{K}\mathbf{E}\{v_k v_k^T\}\bar{K}^T \\ &= (1 - \mu)A\Upsilon_k A^T + \mu \left(A - \bar{K}C\right)\Upsilon_k \left(A - \bar{K}C\right)^T + \Sigma_\omega + \mu \bar{K}\Sigma_v \bar{K}^T \\ &= h_{\bar{K}}(\Upsilon_k),\end{aligned}$$

which is exactly what is stated in (3.37). □

Control over networks with correlated packet losses

In this chapter, we consider the controller design for networked control systems (NCSs) that feature more complex network models than in Chapter 3. The limitation of the independent and identically distributed (*i.i.d*) network model that we considered in Chapter 3, is that it can not capture network effects that contain memory, such as fading, congestion, interference, etc. For this reason, we consider network models that can be modelled using correlated distributions. By taking these correlated network models into account directly in the controller design allows for an optimal controller that is tailored to the network. As we will show, the optimal controller design results in controllers that depend on the packet arrival sequences. This means, that for every possible packet arrival sequence, there will be a (unique) controller. The framework that we present in this chapter can be utilized for the controller design for any network that can be modelled using conditional distributions. However, if the conditional distributions depend on a long packet arrival sequence, the proposed method will result in a large number of controllers. This can be impractical on small embedded systems with limited storage space or limited computational resources. We therefore present three methods that will reduce the amount of controllers while trading off performance compared to the optimal design.

This chapter is structured as follows: In Section 4.1 we present the motivation for the controller design for the correlated network models. This includes a review of relevant existing work on both controller design and network modeling. In Section 4.2, we present the system and network models, where we then present the optimal controller design in Section 4.3. Next, we present three methods to reduce the controller complexity in Section 4.4. In Section 4.5, the performance of the presented controllers is benchmarked against a traditional linear quadratic regulator (LQR) controller and the controller design from Section 3.2.1 that only considers *i.i.d* network models. Section 4.6 summarizes the results presented in this chapter.

4.1 Introduction

In Chapter 3 we showed that, compared to traditional LQR, significant performance gains can be achieved when the network model is directly taken into account in the controller design. However,

since the network model is taken into account, the performance of the controller is highly dependent on how accurately the network model represents the real network that will be used in the NCS. Traditionally, as done in Chapter 3 and explained in Section 2.2.2, *i.i.d* models are used. These models are however very limited in representing realistic networks, since the *i.i.d* model of the packet dropouts assume that the network contains no memory. That is, no burst dropouts, congestion or fading effects are captured. Further, moving objects within a NCS can result in significant variations of the channel quality, which can not be captured by *i.i.d* models. [1], [34]. To address this issue, Gilbert and Elliot [33], [54] introduced the first-order Gilbert-Elliot Markov model that is described in Section 2.2.2. The Gilbert-Elliot model is also considered for controller design, see for example [21], [70], [80], [109], [129]. However, the two-state Gilbert-Elliot model still relies on that channel quality does not vary dramatically [12], [51]. This can be addressed to some extent by increasing the order of the Markov model [51], [95], [119], [122]. As explained in Section 2.2.2, the network state in Gilbert-Elliot Markov models can generally not be observed by the users of the network. The work [51] presents a different approach. Here the network is divided in two states: one where no packet loss occurs, and one where packet loss occurs. In [51], the “bad” state of the network is modelled using a fourth-order finite state Markov chain (FSMC) model. Unlike the Gilbert-Elliot model, in the FSMC model, the network state depends on a finite history of recent packet arrivals. This means that users can observe the network state by monitoring packet transmission outcomes. Higher order packet loss models tend to provide a more accurate fit for wireless networks [51], [95], [119], [122]. For this reason, we propose a controller design for systems where the link between the controller and actuator can be modelled by any conditional distribution. This allows the proposed framework to be used for any order Markov models. It also follows, that the proposed framework generalizes the controller design for the Gilbert-Elliot model and the *i.i.d* model that is presented in Chapter 3. Further, the proposed controller design can be used in conjunction with a traditional LQR to provide an optimal controller for the network model that is presented in [51]. In this case, one would use the LQR for the network state with no packet loss, and switch to the proposed controller as soon as packet dropouts occur.

The optimal controller design is, as in Chapter 3, done by minimizing a quadratic cost function that penalizes future states and control inputs. In the design, the future system states are predicted using the models of both the system and the network, as well as the current state and the history of packet arrivals. This results in a control law that is a function of the state and packet arrival history. A key issue is that, at the time when a control input is computed, the controller does not know whether this input will actually reach the actuator. Such a design problem can fit into the Markov jump linear system framework by assuming that the packet transmission at the current time will be successful [19], [23], [31], [70], [115]. The difference in our approach is, that we take advantage of the structure of the Markov transition matrix, which results in simpler expressions. Designing the control law in this manner leads to controllers that depend on the system state and on the history of packet dropouts. This results in a controller for every possible observation of packet dropouts, which results in high complexity control laws (meaning: many controllers). The complexity increases exponentially for longer packet dropout histories. It also requires significant amounts of memory to be stored in a lookup table e.g. in microcontrollers. At this point one might ask, whether it is necessary to take the entire packet arrival history into account when designing the control law. To tackle this question, we investigate sub-optimal reduced complexity

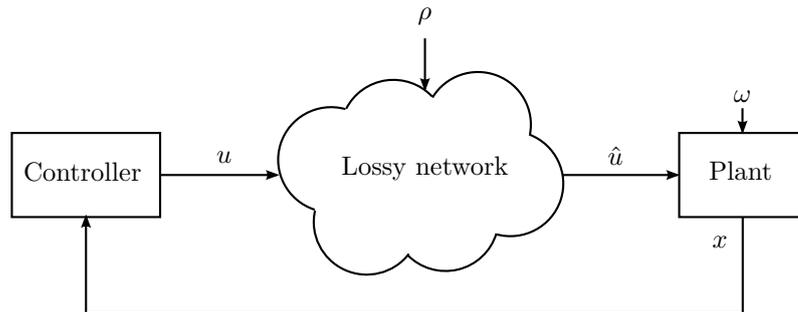


Figure 4.1: The system where the controller and the actuators are connected through a link that is affected by packet dropouts.

control laws, that feature fewer controllers, and illustrate the loss of performance compared to a full complexity control law through simulation studies. While there exists an extensive literature on performance and complexity trade-offs in terms of minimizing a cost function for estimation over networks with correlated packet dropouts [27], [108], the controller design has rarely been considered in this setting. One main difference between the estimator and controller design is that the network is part of a closed-loop system. This makes the controller design more difficult.

4.2 Problem setting

We consider a linear time-invariant system where the controller is connected to the actuators through a network that is affected by correlated packet losses. The system is illustrated in Figure 4.1 and has dynamics

$$\begin{aligned} x_{k+1} &= Ax_k + B\hat{u}_k + \omega_k, \\ \hat{u}_k &= \rho_k u_k \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $\omega_k \sim \mathcal{N}(0, \Sigma_\omega)$ is zero-mean white Gaussian noise with covariance Σ_ω . The binary variable $\rho_k \in \{0, 1\}$ indicates whether the packet at time k is successfully received at the actuator.

The packet arrivals for the model in (4.1) are random and correlated. This means that the probability for the packet at time k to arrive, depends on the history of previously transmitted packets. In this work, we consider the case where the relevant length of the previous history is of a given length. More formally, we can state this as

$$\Pr \{\rho_k = 1 | \rho_{k-1}, \dots, \rho_{k-\infty}\} = \Pr \{\rho_k = 1 | \rho_{k-1}, \dots, \rho_{k-d}\},$$

where $0 \leq d < \infty$ is the length of the relevant history. In the present setting, the controller always receives acknowledgements on successful packet transmissions, and therefore causally knows the exact packet arrival history. Note that the current model generalizes the *i.i.d* model that we used in Chapter 3, since the *i.i.d* model corresponds to a history of length 0. To simplify the notation, we accumulate the history of packet dropouts in the variable $\Theta_k \in \Xi = \{1, \dots, 2^d\}$ to denote the binary sequence $(\rho_{k-1}, \dots, \rho_{k-d})$, where $\Theta_k = i$ refers to a certain realization of $(\rho_{k-1}, \dots, \rho_{k-d})$.

To obtain this, we assign the outcomes of ρ_k to values of Θ_k by

$$\Theta_k \triangleq 1 + \sum_{\ell=1}^d \rho_{k-\ell} 2^{d-\ell}. \quad (4.2)$$

This means that for example, the packet arrival sequence $(1, 0, 0, 1)$ gets assigned to

$$\Theta = 1 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 10.$$

Note that in (4.2), every possible binary sequence $(\rho_{k-1}, \dots, \rho_{k-d})$ gets assigned to a unique value of Θ . Also, note that

$$\rho_{k-1} = \begin{cases} 0, & \text{if } \Theta_k \leq r \\ 1, & \text{if } \Theta_k > r, \end{cases} \quad (4.3)$$

where $r = 2^{d-1}$. Here we find that Θ_k is first-order Markov, since

$$\begin{aligned} \Pr \{ \Theta_{k+1} = j | \Theta_k = i, \Theta_{k-1}, \dots, \Theta_0 \} \\ &= \Pr \{ \Theta_{k+1} = j | \Theta_k = i \} \\ &\triangleq p_{i,j}. \end{aligned}$$

Also, denote $\Pr \{ \Theta_k = i \} \triangleq \pi_i$.

If for a known packet arrival sequence $\Theta_k = i$ the packet arrival $\rho_k = 1$, then the value of $\Theta_{k+1} = \phi(\Theta_k)$. Here the function $\phi: \Xi \rightarrow \{r+1, \dots, d\}$ and is given by

$$\phi(i) = \left\lceil \frac{i}{2} \right\rceil + 2^{d-1} \in \{r+1, \dots, 2^d\}. \quad (4.4)$$

Likewise when at time k the packet arrival history is given by $\Theta_k = i$ and the packet transmission failed ($\rho_k = 0$), $\Theta_{k+1} = \bar{\phi}(i)$, where $\bar{\phi}: \Xi \rightarrow \{1, \dots, r\}$ and is given by

$$\bar{\phi}(i) = \left\lfloor \frac{i}{2} \right\rfloor \in \{1, \dots, r\}. \quad (4.5)$$

It hereby follows that

$$\Pr \{ \rho_k = 1 | \Theta_k = i \} = \Pr \{ \Theta_{k+1} = \phi(i) | \Theta_k = i \} \triangleq p_{i,\phi(i)},$$

and

$$\Pr \{ \rho_k = 0 | \Theta_k = i \} = \Pr \{ \Theta_{k+1} = \bar{\phi}(i) | \Theta_k = i \} \triangleq p_{i,\bar{\phi}(i)} = 1 - p_{i,\phi(i)}.$$

Example 4.2.1. Consider the case of a packet dropout distribution with $d = 2$. This means that the probability for a successful transmission at time k depends on the outcomes of ρ_{k-1} and ρ_{k-2} . This is illustrated in Figure 4.2. The Markov states $\Theta \in \Xi$, where $\Xi = \{1, 2, 3, 4\}$ are assigned using (4.2). The states are illustrated in gray in Figure 4.2. This can be put into a transition matrix as

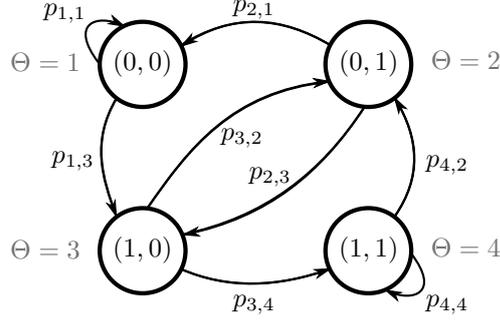


Figure 4.2: Illustration of the assignment of Θ to packet arrival sequences (ρ_{k-1}, ρ_{k-2}) by (4.2). Here the sequence $(1, 0)$ means that $\rho_{k-1} = 1$ and $\rho_{k-2} = 0$. The probability $p_{2,1} = \Pr\{\Theta_{k+1} = 1 | \Theta_k = 2\} = \Pr\{\rho_k = 0 | \rho_{k-1} = 0, \rho_{k-2} = 1\}$.

$$P = \begin{array}{c} \begin{array}{cc} \overbrace{\Theta = 1 \quad \Theta = 2}^{\rho_k=0} & \overbrace{\Theta = 3 \quad \Theta = 4}^{\rho_k=1} \\ \Theta = 1 & \Theta = 2 & \Theta = 3 & \Theta = 4 \end{array} \\ \left[\begin{array}{cccc} p_{1,1} & 0 & p_{1,3} & 0 \\ p_{2,1} & 0 & p_{2,3} & 0 \\ 0 & p_{3,2} & 0 & p_{3,4} \\ 0 & p_{4,2} & 0 & p_{4,4} \end{array} \right] \begin{array}{l} \Theta = 1 \\ \Theta = 2 \\ \Theta = 3 \\ \Theta = 4 \end{array} \end{array}$$

where $p_{1,3} = \Pr\{\rho_k = 1 | \rho_{k-1} = 0, \rho_{k-2} = 0\}$. Here $\phi(1) = 3$ and $\bar{\phi}(1) = 1$, thus $p_{1,\phi(i)} = p_{1,3}$. \square

4.3 Optimal predictive controller design

We utilize predictive control to design a control law that minimizes a receding horizon cost function which penalizes predicted future states and controls, as is done in LQR design. The system (4.1) is however affected by random packet dropouts and noise, which are only known for the past, and not the current and future times. We therefore formulate the cost function using the expectation operator, where the expectation is taken over the packet dropout process ρ_k and the process noise ω_k . When minimizing the cost function using dynamic programming, we obtain feedback control policies as functions of the state and packet arrival history. The controller has at time k access to the state x_k and the Markov state Θ_k . We, thus, obtain controllers of the form

$$u_k = f_k(\Theta_k, x_k), \quad k = 0, 1, \dots, N-1.$$

We want to design the control law to minimize the receding horizon cost function

$$J_N(U_{0,N}, x_0, \Theta_0) = \mathbf{E} \left\{ \sum_{k=0}^{N-1} \|x_k\|_Q^2 + \rho_k \|u_k\|_R^2 + \|x_N\|_{S_{N,\Theta_N}}^2 \middle| x_0, \Theta_0 \right\}, \quad (4.6)$$

where $Q \geq 0$, $R \geq 0$ and $S_{N,i} \geq 0$, $\forall i \in \Xi$ are design parameters that trade-off control performance vs. control effort, the sequence $U_{k,N} \triangleq (u_\ell: \ell = k, \dots, N-1)$ and the prediction horizon $N \in \mathbb{N}$.

Remark 4.3.1. *The problem that we presented here can be fitted into the general Markov jump linear system framework, see e.g. [23]. The difference between the framework presented in [23] and the current problem setting is, that we, in the current framework, at time k do not have the outcome*

of ρ_k available. However, with the framework that is presented in [23], the same control laws as presented in Proposition 4.3.1 can be obtained by assuming that at time k the packet transmission will always be successful.

In the rest of this section, the result that we present will differ from [23] on two points: 1. In the presented setting, it is a consequence of the problem formulation that the optimal strategy is to always design the control packet to assume that the current transmission is successful. 2. We take advantage of the special sparse structure of the transition matrix P , which results in simpler expressions, making it computationally more efficient.

Next, we present the optimal controller design.

Proposition 4.3.1. *The optimal control at each time-step is given by the linear function*

$$u_k^* = L_{k, \Theta_k}^* x_k \quad k = 0, \dots, N-1, \quad (4.7)$$

where

$$L_{k,i}^* = - (R + B^T S_{k+1, \phi(i)} B)^{-1} B^T S_{k+1, \phi(i)} A \quad \forall i \in \Xi, \quad (4.8)$$

with

$$S_{k,i} = Q + A^T \left(S_{k+1, \bar{\phi}(i)} p_{i, \bar{\phi}(i)} + S_{k+1, \phi(i)} p_{i, \phi(i)} \right) A - p_{i, \phi(i)} A^T S_{k+1, \phi(i)} B (R + B^T S_{k+1, \phi(i)} B)^{-1} B^T S_{k+1, \phi(i)} A, \quad (4.9)$$

and $\phi(i)$ and $\bar{\phi}(i)$ defined in (4.4) and (4.5). Here, $p_{i, \phi(i)} = \Pr \{ \Theta_{k+1} = \phi(i) | \Theta_k = i \}$ and $S_{N,i}$ is given for all $i \in \Xi$. This results in the optimal cost

$$J_N^*(x_0, \Theta_0) = x_0^T S_{0, \Theta_0} x_0 + c_{0, \Theta_0}, \quad (4.10)$$

where

$$c_{k,i} = \sum_{j \in \{ \phi(i), \bar{\phi}(i) \}} (\text{trace}(\Sigma_\omega S_{k+1, j}) + c_{k+1, j}) p_{i, j} \quad (4.11)$$

with $c_{N,i} = 0$ for all i .

Proof. The proof is given in Appendix 4.A. □

It is worth to note from (4.8) that the control law at time k only depends on Θ_k for the computation of S_{k+1} , which in turn depends on Θ_{k+1} . Now, since the transition probabilities to go from Θ_k to Θ_{k+1} are not used in (4.8), the value of $L_{k,i}$ will not depend on ρ_{k-d} . Therefore, only 2^{d-1} out of the 2^d control laws are unique and need to be stored in a lookup table.

Remark 4.3.2. *Many network models such as the Gilbert Elliot model [33], [54] (also referred to as the pq model) use a Markov chain with length $d = 1$. It is worth noting that in this case using Proposition 4.3.1 only one unique controller gain is obtained. This means that in this case, the controller does not have to observe the history of the packet dropouts to find the appropriate control gain.*

The results in Proposition 4.3.1 can be vectorized for a relevant history of length d by defining the matrix

$$\mathcal{V}_{i,j} \triangleq \begin{bmatrix} I_n \sqrt{p^{(i+1),(j+1)}} & \mathbf{0}_n & \cdots & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n \\ \mathbf{0}_n & I_n \sqrt{p^{(i+3),(j+2)}} & \cdots & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_n & \mathbf{0}_n & \cdots & I_n \sqrt{p^{(i+2^d-1),(j+r)}} \end{bmatrix} \in \mathbb{R}^{(2^d-1)n \times rn}$$

and

$$\mathcal{W}_{0,j} \triangleq \begin{bmatrix} \mathcal{V}_{0,j} \\ \mathbf{0}_{n,rn} \end{bmatrix} \in \mathbb{R}^{2^d n \times rn} \quad \mathcal{W}_{1,j} \triangleq \begin{bmatrix} \mathbf{0}_{n,rn} \\ \mathcal{V}_{1,j} \end{bmatrix} \in \mathbb{R}^{2^d n \times rn}.$$

Now define the following matrices describing the successful and lost transmissions by

$$\begin{aligned} \mathcal{W} &\triangleq \begin{bmatrix} \mathbf{0}_{2^d n \times rn} & \mathcal{W}_{0,r} & \mathbf{0}_{2^d n \times rn} & \mathcal{W}_{1,r} \end{bmatrix} \in \mathbb{R}^{2^d n \times 2^{d+1} n} \\ \bar{\mathcal{W}} &\triangleq \begin{bmatrix} \mathcal{W}_{0,0} & \mathbf{0}_{2^d n \times rn} & \mathcal{W}_{1,0} & \mathbf{0}_{2^d n \times rn} \end{bmatrix} \in \mathbb{R}^{2^d n \times 2^{d+1} n}, \end{aligned}$$

respectively. Define $\bar{Q} \triangleq \text{diag}\{Q\}$, $\bar{A} \triangleq \text{diag}\{A\}$, $\bar{B} \triangleq \text{diag}\{B\}$. Also let

$$\begin{aligned} \bar{R} &\triangleq \text{diag}\{[Rp_{1,\phi(1)}, Rp_{2,\phi(2)}, \dots, Rp_{d,\phi(d)}]\} \\ \bar{S}_k &\triangleq \text{diag}\{[S_{k,1}, S_{k,2}, \dots, S_{k,2^d}]\}, \end{aligned}$$

and

$$\begin{aligned} \Gamma_k &\triangleq \mathcal{W} \hat{S}_k \mathcal{W}^T \\ \bar{\Gamma}_k &\triangleq \bar{\mathcal{W}} \hat{S}_k \bar{\mathcal{W}}^T, \end{aligned}$$

where

$$\hat{S}_k \triangleq I_2 \otimes \bar{S}_k.$$

Then, (4.9) can be expressed as

$$\bar{S}_k = \bar{Q} + \bar{A}^T (\Gamma_{k+1} + \bar{\Gamma}_{k+1}) \bar{A} - \bar{A}^T \Gamma_{k+1} \bar{B} (\bar{R} + \bar{B}^T \Gamma_{k+1} \bar{B})^{-1} \bar{B}^T \Gamma_{k+1} \bar{A}.$$

The augmented control gains (4.8) can then be computed by

$$\mathcal{L}_k^* = (\bar{R} + \bar{B}^T \Gamma_{k+1} \bar{B})^{-1} \bar{B}^T \Gamma_{k+1} \bar{A},$$

where \mathcal{L}^* is a block diagonal matrix with the control gains $L_{k,i}^*$, $i \in \Xi$ on the diagonal. The control gain $L_{k,i}^*$ can be extracted as the i 'th $n \times m$ diagonal block of \mathcal{L}_k^* .

To implement the predictive controller design from Proposition 4.3.1, one would use a lookup table that contains the control gains, and then at each time-step use the history that is contained in Θ_k to select the corresponding control gain. This requires enough memory to store 2^{d-1} control gains. Here, one question that arises is, whether it is necessary to take the full packet dropout history into account in the controller design to guarantee a reasonable control performance. Omitting part of the history will result in fewer control laws, and thereby a smaller lookup table. In the next section, we present sub-optimal controller designs that reduce the amount of unique control laws at the cost of performance.

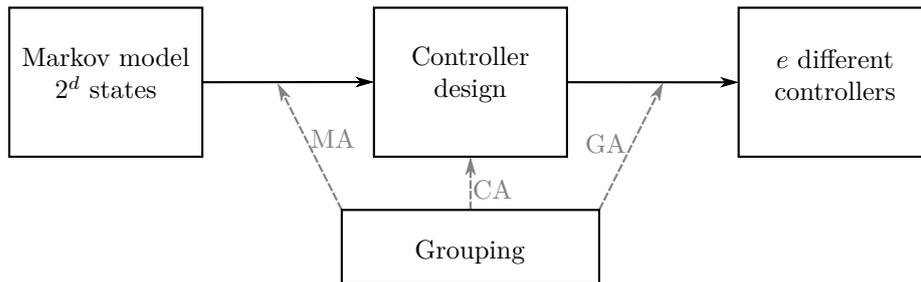


Figure 4.3: An illustration of where to do the grouping in the controller design. Here one can reduce the number of Markov states prior to the controller design (MA), reduce the number of control laws after the controller design (GA) or, reduce the number of control laws while doing the controller design (CA).

4.4 Sub-optimal reduced complexity controllers

When designing the optimal control law for the system (4.1) affected by correlated packet losses we, as described in Section 4.3, obtain a controller for every possible sequence of packet arrivals. For a relevant history of length d , this results in 2^{d-1} unique controllers. Thus, the amount of controllers will grow exponentially when a longer relevant history is considered. In the remainder of this chapter, we study methods to reduce the number of controllers. This however, will inevitably result in some degradation of the control performance. We therefore investigate different methods that reduce the number of controllers while maintaining satisfactory control performance.

We propose to reduce the amount of controllers by designing a controller for a group of possible packet arrival sequences. For example, consider a history length $d = 3$, such that $\Theta \in \Xi$ and the set $\Xi = \{1, 2, \dots, 8\}$. Designing the control law using Proposition 4.3.1 would result in 4 controllers. In this section we present methods that instead allows one to design only 2 or 1 controllers for the set Ξ . This is done by grouping multiple states together and then design a controller for each group. This however leads to two questions: 1. How to group the Markov states that govern the different packet arrival sequences, and 2. what controller to use for each of these groups?

Thus, this problem breaks down to a grouping problem and a controller design problem. First, we present a method on how to group the packet arrival sequences. Then, we present three possible methods for the controller design. The three controller design methods differ in where in the design process the grouping, and thereby reduction (or quantization) is done. This is illustrated graphically in Figure 4.3. The three methods are:

- **Markov averaged (MA)** The grouping is done for the Markov chain. Here we reduce or aggregate the Markov chain prior to the controller design. The controller design is thus done for the reduced Markov chain, which therefore will lead to a reduced number of controllers.
- **Group averaged (GA)** The grouping is done after the controller design. The controller design is thus done for the full Markov chain, and afterwards the number of controllers is reduced.
- **Cost averaged (CA)** In this method the controller design is done for the full Markov chain and the reduction of the number of controllers is enforced in the design.

In the MA design method, the grouping is done when reducing the Markov chain prior to the controller design. This means that the optimal controller is designed for a less accurate network model, which will result in performance loss. In the GA controller design, one first obtains the full complexity optimal controllers for the full Markov chain. To reduce the amount of controllers, the desired amount of controllers, are found by reducing the optimal controllers. It is worth noting that in this case, the quantization (or reduction) is done at the end of the design process. This means that the sub-optimality is first introduced at the end of the design process. Both of these methods utilize the controller design that is presented in Proposition 4.3.1, and are relatively simple to implement. Finally in the CA design method, the quantization is enforced when the controller design is performed. Here the sub-optimality is taken into account in the cost function directly. The control laws that are obtained here should therefore result in a lower cost, and therefore also better performance, than the controllers that are obtained using the MA and GA methods. However, the CA controller design is more complex.

Next in Section 4.4.1, we present a method on how to group the packet arrival sequences. This grouping method is applied in all three reduced complexity controller designs. The difference between the controller design methods is, as mentioned before, where in the design process the grouping function is applied. After presenting the grouping method, we present the MA, GA and CA designs in Sections 4.4.2 to 4.4.4.

4.4.1 Grouping the packet arrival sequences

In this section we discuss how to group the packet arrival sequences, such that we then can design a controller for each of these groups. In Section 4.2 we assigned the packet arrival sequences to values $\Theta \in \Xi$ using (4.2). We define a surjective function $g : \Xi \rightarrow \hat{\Xi}$ which maps the individual packet arrival sequences to a group, for which we then design a controller for. Since we aim at reducing the number of controllers, $\text{card}(\hat{\Xi}) \leq \text{card}(\Xi)$, with $\text{card}(\cdot)$ denoting the cardinality, or number of elements in the set. Let $\hat{\Theta} \in \hat{\Xi}$ indicate the number of the group and associate with $\hat{\Theta} = v$ the subset $\hat{\Xi}_v$, which is defined as

$$\hat{\Xi}_v \triangleq \{i \in \Xi : g(i) = v\}, \quad (4.12)$$

to indicate which Θ get mapped to $\hat{\Theta} = v$. Note that $\hat{\Xi}_v$ are disjoint sets ($\hat{\Xi}_v \cap \hat{\Xi}_j = \emptyset$ when $v \neq j$).

The design of the grouping function is a well-known problem [26], [101]. The only method on how to find the optimal grouping function is by performing a combinatorial search among all possible grouping functions, which is a non-deterministic polynomial-time (NP)-hard problem. To avoid this, we instead propose to design the grouping function g , such that the packet arrival sequences are grouped by their most recent packet arrival history. The intuition behind this is, that the most recent history has the largest impact on the outcomes of future packet dropouts. Thus, we group the packet arrival sequences such that the most recent packet arrival history $(\rho_{k-1}, \rho_{k-2}, \dots, \rho_{d-q})$, where $0 \leq q \leq d$, is identical for all $\Theta \in \hat{\Xi}_v$. This results in a total of 2^{d-q} groups. The grouping function g is in this case given by

$$g(\Theta) = \left\lfloor \frac{\Theta}{2^q} \right\rfloor, \quad (4.13)$$

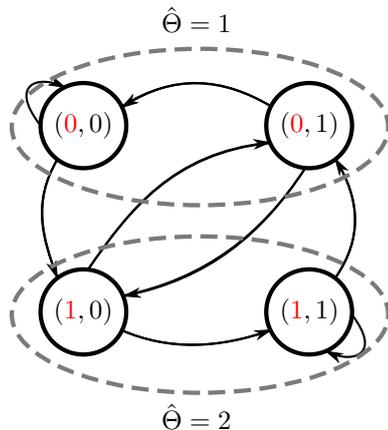


Figure 4.4: An example of the grouping of the packet arrival sequences. Here $q = 1$ and the relevant history in each group (encircled by the dashed line) is marked in red.

and let $\hat{\Theta} = g(\Theta)$. We then design control laws for each group, such that these become

$$\hat{u}_k = \hat{f}_k(\hat{\Theta}_k, x_k), \quad (4.14)$$

where the variable $\hat{\Theta}_k \in \hat{\Xi}$ with $\hat{\Xi} = \{1, \dots, 2^{d-q}\}$, and each $\hat{\Theta}_k = i$ is linked to a given packet arrival sequence of the most recent $d - q$ packet arrivals.

Example 4.4.1. Consider the case from Example 4.2.1. Then, with $q = 1$, we group the packet arrival sequences by the most recent transmission outcome. Thus, $\hat{\Theta}$ takes values in $\hat{\Xi} = \{1, 2\}$. This is illustrated in Figure 4.4. Using (4.12) and (4.13) we obtain that $\hat{\Xi}_1 = (1, 2)$ and $\hat{\Xi}_2 = (3, 4)$. \square

4.4.2 Optimal controller for reduced Markov chain (MA)

In the MA controller design, we first reduce the order of the Markov chain and then design an optimal controller for this reduced Markov chain. We use the grouping function g from (4.13) to obtain the groups of packet arrival sequences as explained in Section 4.4.1. We then compute the transition probabilities between the groups $\hat{\Theta} \in \hat{\Xi}$ from the transition probabilities of the full packet arrival sequences. This is called aggregation of the Markov chain, and is a well-studied topic for efficient steady-state analysis of large Markov chains (see [101] and the references therein). We hereby obtain a reduced order Markov chain with transition matrix $\hat{P} \in \mathbb{R}^{2^{d-q} \times 2^{d-q}}$. The aggregation is done as follows: For a given state space and a choice of grouping of the reduced state space, the aggregate steady-state probabilities are for each group computed by [101]

$$\bar{\pi}_v = \Pr\{\hat{\Theta} = v\} = \sum_{j \in \hat{\Xi}_v} \pi_j, \quad \forall v \in \hat{\Xi},$$

and the aggregate transition probabilities are given by

$$\hat{p}_{v,\ell} = \Pr\{\hat{\Theta}_k = \ell \mid \hat{\Theta}_{k-1} = v\} = \frac{\sum_{j \in \hat{\Xi}_v} \sum_{i \in \hat{\Xi}_\ell} \pi_i P_{i,j}}{\sum_{j \in \hat{\Xi}_v} \pi_j}.$$

This forms the aggregated (or reduced) transition matrix \hat{P} .

Afterwards, the optimal control law is designed using Proposition 4.3.1 for the reduced Markov transition matrix. This results in a reduced number of controllers. Since we reduce, or quantize, the Markov transition matrix prior to the controller design, we obtain an optimal control law for a Markov model that is representative, but not completely describing the actual network model that is affecting the system to be controlled.

4.4.3 Group averaged controller (GA)

The other simple approach that we present for the reduced complexity controller is the GA controller. With this method, we first design the optimal controllers for the entire Markov chain in the state-space Ξ using Proposition 4.3.1. Then, the controllers are grouped using the grouping function g from (4.13) in Section 4.4.1, such that there is one controller for every group $\hat{\Theta}$ of packet arrival sequences.

The reduced controllers will be of the form

$$\hat{u}_k = \hat{L}_{k,v} x_k \qquad g(\Theta_k) = v,$$

where the control gains are given by

$$\begin{aligned} \hat{L}_{k,v} &= \mathbf{E} \left\{ L_k \mid \hat{\Theta}_k = v \right\}, & \forall v \in \hat{\Xi}. \\ &= \sum_{j \in \hat{\Xi}_i} L_{k,j} \frac{\pi_j}{\hat{\pi}_v}, \end{aligned}$$

with

$$\hat{\pi}_v \triangleq \mathbf{Pr} \left\{ \hat{\Theta}_k = v \right\} = \sum_{j \in \hat{\Xi}_v} \pi_j.$$

4.4.4 Reducing the amount of controllers in the design (CA)

For the CA controller design, we force the control gain to be equal for all possible packet arrival sequences that are contained in a group $\hat{\Theta}$. The grouping is done as described in Section 4.4.1. The control law is thus designed by taking the expected value over the full history of $(\rho_{k-1}, \dots, \rho_{k-d})$ in the cost function and then setting the controller for different outcomes of Θ to be identical. The difference to the MA design is, that in the MA design, the Markov chain is reduced prior to the controller design. The sub-optimality is therefore introduced into the design process before the controllers design. On the contrary, the CA controller takes the full Markov chain into account while minimizing the cost. The sub-optimality is then introduced directly when minimizing the cost function by forcing some of the controllers to be equal.

More formally, we aim at forcing the grouping of the packet arrival sequences through in the cost function that we utilize to design the controller. This is done by designing a control law of the form

$$\hat{u}_{k,i} = \hat{f}_k(g(i), x_k), \qquad \forall i \in \hat{\Xi} \tag{4.15}$$

to minimize the finite horizon cost function

$$\begin{aligned} \bar{J}_N(U_{0,N}, x_0) &= \mathbf{E} \{J_N(U_{0,N}, x_0, \Theta_0) | x_0\} \\ &= \mathbf{E} \left\{ \sum_{k=0}^{N-1} \|x_k\|_Q^2 + \rho_k \|u_k\|_R^2 + \|x_N\|_{S_{N,\Theta_N}}^2 \middle| x_0 \right\}. \end{aligned} \quad (4.16)$$

Since g maps Ξ onto $\hat{\Xi}$, the controller $\hat{f}_k(g(i), x_k)$ is identical for all $i \in \Xi$ that $g(i)$ maps to the same $\hat{\Theta} \in \hat{\Xi}$. We therefore obtain the sequence of reduced controllers $\bar{U}_{0,N} = (\bar{u}_0, \dots, \bar{u}_{N-1})$ with $\bar{u}_k = (\hat{u}_{k,1}, \dots, \hat{u}_{k,2^d-q})$. The controller design is thus done by evaluating the optimization problem

$$\begin{aligned} \min_{U_{0,N}} \quad & \bar{J}_N(U_{0,N}, x_0) \\ \text{subject to } & u_{k,i} = \hat{f}_k(g(i), x_k) \quad i \in \Xi \\ & k = 0, \dots, N-1. \end{aligned} \quad (4.17)$$

Also note that the control law (4.15) is only allowed to depend on x_k and through $g(\Theta_k)$ on $\hat{\Theta}_k$, and thereby not on previously applied controls. This allows us to obtain closed form solutions to the problem (4.17). Note however, that these control laws not necessarily are optimal, as described in [116], [117], [123]. The control laws that are presented in [116], [117] can, however, not be computed using closed form expressions, and an iterative algorithm is required. We therefore instead focus on sub-optimal control laws that allow us to obtain closed form expressions. This leads to the following result.

Theorem 4.4.1. *The controls of the form (4.15) for (4.17) are given by*

$$\hat{u}_k^* = \bar{L}_{k,v}^* x_k, \quad g(\Theta_k) = v \quad (4.18)$$

where the control gain for each $v \in \hat{\Xi}$ is given by

$$\bar{L}_{k,v}^* = -(\bar{R}_v + B^T \bar{S}_{k+1,v} B)^{-1} B^T \bar{S}_{k+1,v} A. \quad (4.19)$$

For all $i \in \Xi$

$$\begin{aligned} S_{k,i} &= Q + A^T S_{k+1, \bar{\phi}(i)} A p_{i, \bar{\phi}(i)} + \left(\bar{L}_{k, g(i)}^* \right)^T R \bar{L}_{k, g(i)}^* p_{i, \phi(i)} \\ &\quad + \left(A + B \bar{L}_{k, g(i)}^* \right)^T S_{k+1, \phi(i)} \left(A + B \bar{L}_{k, g(i)}^* \right) p_{i, \phi(i)}, \end{aligned} \quad (4.20)$$

where $S_{N,i} = S_N$ and

$$\begin{aligned} \bar{S}_{k,v} &= \sum_{i \in \hat{\Xi}_v} S_{k, \phi(i)} p_{i, \phi(i)} \pi_i \\ \bar{R}_v &= R \sum_{i \in \hat{\Xi}_v} p_{i, \phi(i)} \pi_i. \end{aligned} \quad (4.21)$$

This results in the cost

$$\bar{J}_N^*(x_0) = \sum_{i \in \Xi} x_0^T S_{0,i} x_0 \pi_i + c_{0,i} \pi_i, \quad (4.22)$$

where

$$c_{k,i} = \text{trace} \left(\Sigma_\omega \left(S_{k+1, \phi(i)} p_{i, \phi(i)} + S_{k+1, \bar{\phi}(i)} p_{i, \bar{\phi}(i)} \right) \right) + c_{k+1, \phi(i)} p_{i, \phi(i)} + c_{k+1, \bar{\phi}(i)} p_{i, \bar{\phi}(i)}, \quad (4.23)$$

with $c_{N,i} = 0$, $i \in \Xi$.

$(\rho_{k-1}, \dots, \rho_{k-4})$	Θ	$p_{i,\bar{\phi}(i)}$ and $\Pr\{\rho_k = 0\}$	$p_{i,\phi(i)}$ and $\Pr\{\rho_k = 1\}$
0000	1	$p_{1,1} = 0.2$	$p_{1,9} = 0.8$
0001	2	$p_{2,1} = 0.65$	$p_{2,9} = 0.35$
0010	3	$p_{3,2} = 0.78$	$p_{3,10} = 0.22$
0011	4	$p_{4,2} = 0.2$	$p_{4,10} = 0.8$
0100	5	$p_{5,3} = 0.85$	$p_{5,11} = 0.15$
0101	6	$p_{6,3} = 0.3$	$p_{6,11} = 0.7$
0110	7	$p_{7,4} = 0.75$	$p_{7,12} = 0.25$
0111	8	$p_{8,4} = 0.2$	$p_{8,12} = 0.8$
1000	9	$p_{9,5} = 0.6$	$p_{9,13} = 0.4$
1001	10	$p_{10,5} = 0.8$	$p_{10,13} = 0.2$
1010	11	$p_{11,6} = 0.5$	$p_{11,14} = 0.5$
1011	12	$p_{12,6} = 0.3$	$p_{12,14} = 0.7$
1100	13	$p_{13,7} = 0.5$	$p_{13,15} = 0.5$
1101	14	$p_{14,7} = 0.1$	$p_{14,15} = 0.9$
1110	15	$p_{15,8} = 0.4$	$p_{15,16} = 0.6$
1111	16	$p_{16,8} = 0.7$	$p_{16,16} = 0.3$

Table 4.1: The non-zero Markov transition probabilities that are used for the simulations.

Proof. The proof of Theorem 4.4.1 is presented in Appendix 4.B. □

4.5 Simulation studies

We compare the performance of the predictive controller designs proposed in Sections 4.3 and 4.4 using Monte Carlo simulation studies. The controllers are implemented as model predictive control (MPC), that is, we set $x_k = L_{0,\Theta_k} x_k$ for all k . The performance of the proposed controller designs is demonstrated for both a DC servo and a mass spring system with three masses. In both systems, the network that connects the controller and the actuator is modelled by a $d = 4$ 'th-order Markov chain. This results in 16 possible packet arrival sequences, and thereby 8 unique controllers. The conditional packet arrival and loss probabilities for the network model are given in Table 4.1. Both simulations consider a system of the form (4.1) with $\Sigma_\omega = I_m$. The results are obtained by averaging 500 simulations each of length 50 000. The costs (4.6) and (4.16) are computed with horizon length $N = 500$ to obtain a steady-state solution. A standard LQR controller that does not take the network conditions into account is used to benchmark. Simulations of the controller design for *i.i.d* packet dropouts that we presented in Section 3.2.1 are also added for comparison. Here the *i.i.d* probability for a transmission to be successful is computed out of the steady-state probabilities of the Markov chain presented in Table 4.1. This results in a probability of $\lambda = 0.5308$ for successful transmission. The performance is measured using the empirical cost

$$\text{cost} = \frac{1}{50\,000} \sum_{k=1}^{50\,000} x_k^T Q x_k + \rho_k u_k^T R u_k.$$

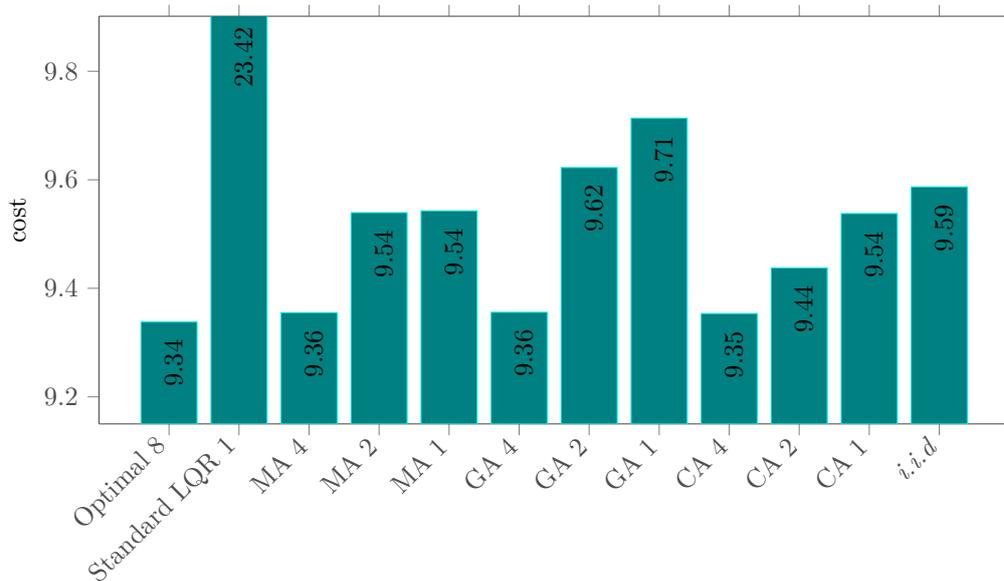


Figure 4.5: Averaged results for simulations of a DC servo. The optimal control law is designed using Proposition 4.3.1 and the MA, GA and CA control laws are designed as described in Sections 4.4.2 to 4.4.4. The reference *i.i.d.* controller is from Section 3.2.1. The number trialing the name indicates the amount of unique controllers.

4.5.1 DC servo

We consider a DC servo with states $x = [\dot{\theta} \ \theta]^T$ with θ being the angular position and $\dot{\theta}$ the angular velocity. The continuous time model is given by

$$A = \begin{bmatrix} \frac{1}{\tau} & 0 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} \frac{K}{\tau} \\ 0 \end{bmatrix}.$$

We select the time constant $\tau = 5$ and gain $K = 50$. The system has been discretized using `c2d` in MATLAB R2017a with sampling interval 1 s. This results in the discrete dynamics

$$A = \begin{bmatrix} 0.819 & 0 \\ 0.906 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 9.063 \\ 4.683 \end{bmatrix},$$

with open loop eigenvalues located at 1 and 0.819.

The simulation results in Figure 4.5 show that the optimal stochastic controller design (Optimal 8) presented in Proposition 4.3.1 significantly outperforms LQR, which results in a cost of 23.4. Compared to the *i.i.d.* controller design from Section 3.2.1, the optimal controller results in a 2.6% performance increase.

When reducing the number of controllers to 4 controllers, the performance is only slightly reduced compared to the optimal design. However, when only two controllers are desired, the CA controller (from Theorem 4.4.1) results in the lowest performance reduction, followed by the MA design (Section 4.4.2). Interestingly, the performance of the MA design is identical when one or two controllers are used. This matches the performance of the CA design with one controller. Both the CA and MA controllers outperform the *i.i.d.* controller. The GA design, presented in Section 4.4.3, does not lose much performance when four controllers are desired. Reducing the

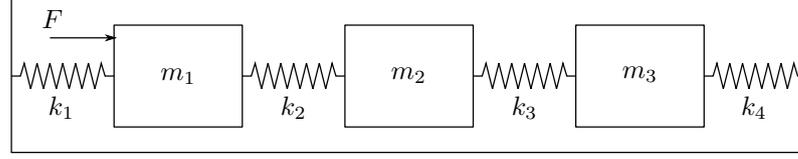


Figure 4.6: An illustration of the model with three masses connected by springs that is considered in the simulation in Section 4.5.2. The actuator exerts force F on the first mass.

number of controllers to one or two in the GA design, however results in lower performance than the *i.i.d* design.

4.5.2 Mass spring system

In this simulation we consider a the system in Figure 4.6 with three masses on a surface connected by springs to each other and side walls. There is an actuator exerting a force to the first mass. The actuator is equipped with an integrator, such that the previous control input is held at the actuator. The masses are denoted by m and are all equal ($m = m_1 = m_2 = m_3$). The spring coefficients are denoted by k and are also all equal ($k = k_1 = k_2 = k_3 = k_4$). The continuous time dynamics of this system are given by

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -2k_m & k_m & 0 & 0 & 0 & 0 & 1 \\ k_m & -2k_m & k_m & 0 & 0 & 0 & 0 \\ 0 & k_m & -2k_m & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m} \\ 0 \\ 0 \\ \frac{1}{m} \end{bmatrix},$$

where $k_m \triangleq \frac{k}{m}$. We set $m = 1$ and $k = 0.323$. This gives open loop eigenvalues at $\pm 1.05j$, $\pm 0.804j$, $\pm 0.435j$ and one at 0 due to the integrator. Discretizing the system using `c2d` in MATLAB R2017a with sampling interval 1 s results in the discrete-time dynamics

$$A = \begin{bmatrix} 0.698 & 0.145 & 0.00407 & 0.897 & 0.0504 & 0.00083 & 0.474 \\ 0.145 & 0.702 & 0.145 & 0.0504 & 0.897 & 0.0504 & 0.0129 \\ 0.00407 & 0.145 & 0.698 & 0.00083 & 0.0504 & 0.897 & 0.00014 \\ -0.563 & 0.257 & 0.0158 & 0.698 & 0.145 & 0.00407 & 0.897 \\ 0.257 & -0.547 & 0.257 & 0.145 & 0.702 & 0.145 & 0.0504 \\ 0.0158 & 0.257 & -0.563 & 0.00407 & 0.145 & 0.698 & 0.00083 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.474 \\ 0.0129 \\ 0.00014 \\ 0.897 \\ 0.0504 \\ 0.00083 \\ 1 \end{bmatrix}.$$

The discrete-time system has eigenvalues at $0.498 \pm 0.867j$, $0.694 \pm 0.72j$, $0.907 \pm 0.421j$ and 1.

The simulation results are shown in Figure 4.7. The standard LQR did not manage to maintain stability and is therefore omitted. The optimal controller that takes the full history into account (Optimal 8) performs 18.2% better than the *i.i.d* controller from Section 3.2.1. When the number of controllers is decreased, the control performance is reduced. Similar to the simulations for the DC servo, the CA controller outperformed the MA and GA controllers. When only one control law

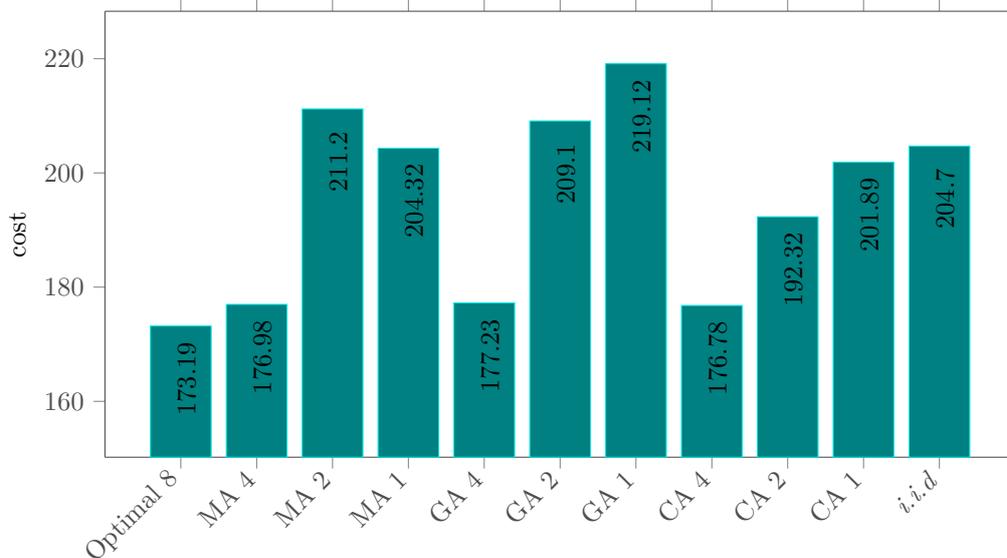


Figure 4.7: Averaged results for the simulations of a mass spring system with three masses. The optimal control law designed using Proposition 4.3.1 and the MA, GA and CA control laws are designed as described in Sections 4.4.2 to 4.4.4. The reference *i.i.d.* controller is from Section 3.2.1. The number trailing the name indicates the amount of unique controllers.

is desired, the MA and CA controllers outperform the *i.i.d.* controller from Section 3.2.1 by a small margin. It is however noteworthy that with MA with only one controller, the performance is better than MA with two controllers. This is believed to be due to the combination of a sub-optimal selection of the grouping function and a sub-optimality that is introduced when the Markov chain is reduced. The CA method outperforms the other methods, however with only one controller, the performance difference between the CA and MA is small.

4.6 Summary

In this chapter, we studied stochastic predictive control for systems that are affected by correlated packet dropouts between the controller and actuators. The probability for a successful transmission depends on a history of packet transmission outcomes. We obtained feedback policies that take into account the packet arrival history by selecting representative gains from a lookup table. Further, we presented a vectorized formulation of the problem, which makes the numerical solutions easier to obtain. Simulation studies show that for the given network, the proposed optimal controller outperforms standard linear quadratic regulator controller design by a large margin for stable systems. As expected, the controller also offers an increased performance compared to the *i.i.d.* controller design from Section 3.2.1. When simulating the larger system with the given network, the standard LQR controller is unable to maintain stability. On the other hand, the proposed optimal controller that takes the full packet arrival history into account and the *i.i.d.* controller from Chapter 3 maintain stability. The proposed controller outperforms the *i.i.d.* by a significant margin in simulation studies. However, the disadvantage of the proposed controller compared to the *i.i.d.* controller is, that the implementation and design is significantly more complex. The design

requires iterating a coupled Riccati equation (4.9) and (4.20), where the number of equations grows exponentially when longer packet arrival histories are taken into account. These Riccati equations can, however, be solved offline. This will result in a large lookup table that contains the controllers. An implementation requires some multiplexing in the controller and enough memory to store the lookup table.

In the second part of this chapter, we asked the question, whether it is necessary to have such a large lookup table to achieve adequate control performance. To answer that, we proposed three sub-optimal designs that trade-off the complexity and control performance. The difference between these methods is where, during the design process, the reduction of the amount of controllers is performed. For the given network model, all three of these methods show merely a minor performance degradation when the number of controllers is reduced, which allows for a good trade-off between performance and controller complexity. Of the three algorithms, the best control performance for reduced complexity controller design were obtained using the cost averaged (CA) controller, which enforces the reduction of the number of controllers through the cost. It is also noteworthy that while the design procedure of the Markov averaged (MA) controller is simple, there is only little performance loss compared to the CA controller. Another interesting observation is, that when only one controller is desired, the MA and CA reduced controller designs outperform the *i.i.d* by a small margin. The reason for this is, that the MA controller is designed using a Markov chain with two states, which contains more information about the network than the *i.i.d* model. For the CA controller the performance gain compared to the *i.i.d*, as well as the MA and GA approach, is achieved since the entire Markov chain is directly taken into account within the controller design. This results in a single controller that minimizes a cost function that takes the entire Markov chain into account.

Appendix

4.A Proof of Proposition 4.3.1

The cost function (4.6) is minimized using dynamic programming [15]. Define the optimal cost

$$J_N^*(x_0, \Theta_0) \triangleq \min_{U_{0,N-1}} J_N(U_{0,N-1}, x_0, \Theta_0).$$

Note that the cost (4.6) at any stage n can be written as

$$J_N^*(x_0, \Theta_0) = \min_{(u_0, \dots, u_n)} \mathbf{E} \left\{ \sum_{k=0}^n \|x_k\|_Q^2 + \rho_k \|u_k\|_R^2 + J_{N-(n+1)}^*(Ax_n + \rho_n Bu_n + \omega_n, \Theta_{n+1}) \middle| \Theta_0, x_0 \right\}.$$

The argument proceeds by induction. At stage N we have the cost

$$J_0^*(x_N, \Theta_N) = \mathbf{E} \{ x_N^T S_N x_N | \Theta_N, x_N \} = x_N^T S_{N,i} x_N.$$

Now at stage $k+1$ we assume there for each $\Theta_{k+1} \in \Xi$ exists an $S_{k+1, \Theta_{k+1}}$ and $c_{k+1, \Theta_{k+1}}$, such that

$$J_{N-(k+1)}^*(x_{k+1}, \Theta_{k+1}) = x_{k+1}^T S_{k+1, \Theta_{k+1}} x_{k+1} + c_{k+1, \Theta_{k+1}}.$$

Then at stage k we have

$$J_{N-k}^*(x_k, \Theta_k) = \min_{u_k} \mathbf{E} \left\{ \|x_k\|_Q^2 + \rho_k \|u_k\|_R^2 + J_{N-(k+1)}^*(Ax_k + \rho_k B u_k + \omega_k, \Theta_{k+1}) \mid \Theta_k, x_k \right\}.$$

Using (4.1) and writing this out yields

$$J_{N-k}^*(x_k, \Theta_k) = \min_{u_k} \mathbf{E} \left\{ x_k^T (Q + A^T S_{k+1, \Theta_{k+1}} A) x_k + \rho_k u_k^T (R + B^T S_{k+1, \Theta_{k+1}} B) u_k \right. \\ \left. + 2\rho_k u_k^T B^T S_{k+1, \Theta_{k+1}} A x_k + \omega_k^T S_{k+1, \Theta_{k+1}} \omega_k + c_{k+1, \Theta_{k+1}} \mid \Theta_k, x_k \right\}$$

where we used the fact that $\rho_k^2 = \rho_k$ and that ω_k is white. Computing the expectation and recalling that ρ_k is only non-zero when $\Theta_{k+1} > r$, as shown in (4.3), yields

$$J_{N-k}^*(x_k, \Theta_k) = \min_{u_k} \sum_{j \in \Xi} \left(x_k^T (Q + A^T S_{k+1, j} A) x_k + \mathbf{1}_{\{r+1, \dots, d\}}(j) u_k^T (R + B^T S_{k+1, j} B) u_k \right. \\ \left. + 2 \mathbf{1}_{\{r+1, \dots, d\}}(j) u_k^T B^T S_{k+1, j} A x_k + \text{trace}(\Sigma_\omega S_{k+1, j}) + c_{k+1, j} \right) \mathbf{Pr} \{ \Theta_{k+1} = j \mid \Theta_k \},$$

where the indicator function $\mathbf{1}_{\mathcal{S}}(i) = 1$ if $i \in \mathcal{S}$. By (4.4) and (4.5) we have that $\mathbf{Pr} \{ \Theta_{k+1} = j \mid \Theta_k \}$ is only non-zero when $j = \phi(\Theta_k)$ and $j = \bar{\phi}(\Theta_k)$. Utilizing this results in

$$J_{N-k}^*(x_k, \Theta_k) = \min_{u_k} \sum_{j \in \{\phi(\Theta_k), \bar{\phi}(\Theta_k)\}} \left(x_k^T (Q + A^T S_{k+1, j} A) x_k + \text{trace}(\Sigma_\omega S_{k+1, j}) + c_{k+1, j} \right. \\ \left. + \mathbf{1}_{\{r+1, \dots, d\}}(j) u_k^T (R + B^T S_{k+1, j} B) u_k \right. \\ \left. + 2 \mathbf{1}_{\{r+1, \dots, d\}}(j) u_k^T B^T S_{k+1, j} A x_k \right) \mathbf{Pr} \{ \Theta_{k+1} = j \mid \Theta_k \}.$$

By defining $\bar{S}_{k+1, i} \triangleq \sum_{j \in \{\phi(i), \bar{\phi}(i)\}} S_{k+1, j} p_{i, j}$, the above can be written as

$$J_{N-k}^*(x_k, \Theta_k) = \min_{u_k} x_k^T (Q + A^T \bar{S}_{k+1, \Theta_k} A) x_k \\ + \mathbf{Pr} \{ \Theta_{k+1} = \phi(\Theta_k) \mid \Theta_k \} \left(u_k^T (R + B^T S_{k+1, \phi(\Theta_k)} B) u_k + 2u_k^T B^T S_{k+1, \phi(\Theta_k)} A x_k \right) \\ + \text{trace}(\Sigma_\omega \bar{S}_{k+1, \Theta_k}) + \sum_{j \in \{\phi(\Theta_k), \bar{\phi}(\Theta_k)\}} c_{k+1, j} \mathbf{Pr} \{ \Theta_{k+1} = j \mid \Theta_k \}. \quad (4.A.1)$$

The u_k that minimizes (4.A.1) is found by taking the gradient of (4.A.1) with respect to u_k and setting it equal to 0. This yields

$$0 = 2 \mathbf{Pr} \{ \Theta_{k+1} = \phi(\Theta_k) \mid \Theta_k \} (R + B^T S_{k+1, \phi(\Theta_k)} B) u_k \\ + 2 \mathbf{Pr} \{ \Theta_{k+1} = \phi(\Theta_k) \mid \Theta_k \} B^T S_{k+1, \phi(\Theta_k)} A x_k.$$

Here the terms $\mathbf{Pr} \{ \Theta_{k+1} = \phi(\Theta_k) \mid \Theta_k \}$ cancel out. Isolating u_k results in (4.7) and (4.8).

Substituting the optimal control (4.7) into (4.A.1) results in the optimal cost at stage k

$$J_{N-k}^*(x_k, \Theta_k) = x_k^T S_{k, \Theta_k} x_k + c_{k, \Theta_k}$$

where S_{k, Θ_k} is as in (4.9) and c_{k, Θ_k} is given in (4.11). Proceeding as in the above then leads to (4.10). \square

4.B Proof of Theorem 4.4.1

This proof is organized as follows: We first show that the cost (4.16) can be expressed as a sum of the costs given the history $\hat{\Theta}_k$ for all $\hat{\Theta}_k \in \hat{\Xi}$. We then show that each of these costs can be stated as a function of the cost at the current stage plus the cost at the future stages. Finally, we show by induction that the optimal control laws of the form (4.15) are given by (4.19).

To simplify the presentation of the proof, let

$$V_k \triangleq \|x_k\|_Q^2 + \rho_k \|\hat{u}_{k,v}\|_R^2.$$

Then, the cost (4.16) at stage $N - k$ can be stated as

$$\begin{aligned} \bar{J}_{N-k}(x_k, \bar{U}_{k,N}) &= \mathbf{E} \left\{ \sum_{\ell=k}^{N-1} \|x_\ell\|_Q^2 + \rho_\ell \|\hat{u}_{\ell,v}\|_R^2 + \|x_N\|_{S_0}^2 \middle| x_k \right\} \\ &= \mathbf{E} \left\{ \sum_{\ell=k}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \middle| x_k \right\} \\ &= \sum_{v \in \hat{\Xi}} \mathbf{E} \left\{ \sum_{\ell=k}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \middle| x_k, \hat{\Theta}_k = v \right\} \mathbf{Pr} \left\{ \hat{\Theta}_k = v \right\} \\ &= \sum_{v \in \hat{\Xi}} \mathbf{E} \left\{ J_{N-k}(x_k, \bar{U}_{k,N}, \Theta_k) \middle| x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v, \end{aligned} \quad (4.B.1)$$

where $\hat{\pi}_v \triangleq \mathbf{Pr} \left\{ \hat{\Theta}_k = v \right\} = \sum_{i \in \hat{\Xi}_v} \pi_i$. Define the cost (4.B.1) conditioned on $\hat{\Theta}_k = v$ by

$$F_{N-k}(x_k, \bar{U}_{k,N}, v) \triangleq \mathbf{E} \left\{ J_{N-k}(x_k, \bar{U}_{k,N}, \Theta_k) \middle| x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v. \quad (4.B.2)$$

This allows us to state (4.B.1) as

$$\bar{J}_{N-k}(x_k, \bar{U}_{k,N}) = \sum_{v \in \hat{\Xi}} F_{N-k}(x_k, (\hat{u}_{k,v}, \bar{U}_{k+1,N}), v). \quad (4.B.3)$$

Next, we show that the partial cost (4.B.2) can be expressed as a function of the cost at stage $N - k$ plus the cost at stage $N - (k + 1)$.

Lemma 4.B.1. *With control law (4.15), the partial costs (4.B.2) can at each stage $N - k$ be expressed as*

$$F_{N-k}(x_k, (\hat{u}_{k,v}, \bar{u}_{k+1,N}), v) = \mathbf{E} \left\{ \mathcal{L}_{N-k}(x_k, (\hat{u}_{k,v}, \bar{u}_{k+1,N}), \Theta_k) \middle| x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v,$$

where

$$\begin{aligned} \mathcal{L}_{N-k}(x_k, (\hat{u}_{k,v}, \bar{U}_{k+1,N}), \Theta_k) &\triangleq x_k^T Q x_k + \rho_k \hat{u}_{k,v}^T R \hat{u}_{k,v} \\ &\quad + J_{N-k-1}(Ax_k + \rho_k B \hat{u}_k + \omega_k, \bar{U}_{k+1,N}, \Theta_{k+1}) \end{aligned}$$

and J_{N-k-1} is given in (4.6).

Proof. For each $F_{N-k}(x_k, (\hat{u}_{k,v}, \bar{U}_{k+1,N}), v)$ we have

$$\begin{aligned} F_{N-k}(x_k, (\hat{u}_{k,v}, \bar{U}_{k+1,N}), v) &= \mathbf{E} \left\{ \sum_{\ell=k}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \middle| x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v \\ &= \mathbf{E} \left\{ V_k \middle| x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v + \underbrace{\mathbf{E} \left\{ \sum_{\ell=k+1}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \middle| x_k, \hat{\Theta}_k = v \right\}}_{C_k} \hat{\pi}_v, \end{aligned} \quad (4.B.4)$$

where computing the second expectation on the right-hand-side yields

$$\begin{aligned} C_k &= \mathbf{E} \left\{ \mathbf{E} \left\{ \sum_{\ell=k+1}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \middle| x_{k+1}, \Theta_{k+1} \right\} \middle| x_k, \hat{\Theta}_k = v \right\} \\ &= \int \mathbf{E} \left\{ \sum_{\ell=k+1}^{N-1} V_\ell + \|x_N\|_{S_0}^2 \middle| x_{k+1}, \Theta_{k+1} \right\} \mathbf{Pr} \left\{ x_{k+1}, \Theta_{k+1} \middle| x_k, \hat{\Theta}_k = v \right\} dx_{k+1} d\Theta_{k+1} \\ &= \int J_{N-k-1}(x_{k+1}, \bar{U}_{k+1,N}, \Theta_{k+1}) \mathbf{Pr} \left\{ x_{k+1}, \Theta_{k+1} \middle| x_k, \hat{\Theta}_k = v \right\} dx_{k+1} d\Theta_{k+1} \\ &= \mathbf{E} \left\{ J_{N-k-1}(x_{k+1}, \bar{U}_{k+1,N}, \Theta_{k+1}) \middle| x_k, \hat{\Theta}_k = v \right\}. \end{aligned} \quad (4.B.5)$$

The result then follows by substituting (4.B.5) back into (4.B.4). \square

Now we can proof Theorem 4.4.1 using an induction argument. At stage 0, we have the cost

$$\bar{J}_0^*(x_N) = \mathbf{E} \left\{ x_N^T S_N x_N \middle| x_N \right\} = \sum_{i \in \Xi} \bar{J}_{0,i}^*(x_N) \pi_i,$$

with $\bar{J}_{0,i}^*(x_N)$ defined in (4.B.1). Then, at stage $N-k-1$, consider the optimal cost with control laws of the form (4.14) to be

$$\bar{J}_{N-k-1}^*(x_{k+1}) = \sum_{i \in \Xi} \bar{J}_{N-k-1,i}^*(x_{k+1}) \pi_i,$$

where

$$\begin{aligned} \bar{J}_{N-k-1,i}^*(x_{k+1}) &\triangleq \bar{J}_{N-k-1,i}(x_{k+1}, \bar{U}_{k+1,N}^*) & i \in \Xi \\ &= x_{k+1}^T S_{k+1,i} x_{k+1} + c_{k+1,i}. \end{aligned}$$

Then, at stage $N-k$, we have from (4.B.3) and Lemma 4.B.1 for $\hat{\Theta}_k = v$

$$\begin{aligned} F_{N-k}^*(x_k, v) &= \min_{\hat{u}_{k,v}} \mathbf{E} \left\{ \mathcal{L}_{N-k, \Theta_k}(x_k, \hat{u}_{k,v}) \middle| x_k, \hat{\Theta}_k = v \right\} \hat{\pi}_v \\ &= \min_{\hat{u}_{k,v}} \sum_{i \in \Xi} \mathbf{E} \left\{ \mathcal{L}_{N-k, \Theta_k}(x_k, \hat{u}_{k,v}) \middle| x_k, \Theta_k = i \right\} \mathbf{Pr} \left\{ \Theta_k = i \middle| \hat{\Theta}_k = v \right\} \hat{\pi}_v \\ &\stackrel{(a)}{=} \min_{\hat{u}_{k,v}} \sum_{i \in \hat{\Xi}_v} \underbrace{\mathbf{E} \left\{ \mathcal{L}_{N-k, \Theta_k}(x_k, \hat{u}_{k,v}) \middle| x_k, \Theta_k = i \right\}}_{D_i} \pi_i, \end{aligned} \quad (4.B.6)$$

where we in (a) used Bayes' law. Now

$$\begin{aligned} D_i &= \mathbf{E} \left\{ \mathbf{E} \left\{ \mathcal{L}_{N-k, \Theta_k}(x_k, \hat{u}_{k,v}) \middle| x_k, \Theta_{k+1}, \Theta_k \right\} \middle| x_k, \Theta_k = i \right\} \\ &= \sum_{j \in \Xi} \mathbf{E} \left\{ \mathcal{L}_{N-k, i}(x_k, \hat{u}_{k,v}) \middle| x_k, \Theta_{k+1} = j \right\} p_{i,j} \\ &\stackrel{(a)}{=} \sum_{j \in \{\phi(i), \phi(i)\}} \mathcal{M}_{N-k, j}(x_k, \hat{u}_{k,v}) p_{i,j}, \end{aligned} \quad (4.B.7)$$

where (a) follows by (4.4) and (4.5) and

$$\begin{aligned} \mathcal{M}_{N-k,j}(x_k, \hat{u}_{k,v}) &\triangleq \mathbf{E} \{ \mathcal{L}_{N-k,i}(x_k, \hat{u}_{k,v}) | x_k, \Theta_{k+1} = j \} \\ &= x_k^T (Q + A^T S_{k+1,j} A) x_k + \text{trace}(\Sigma_\omega S_{k+1,j}) + c_{k+1,j} \\ &\quad + \mathbf{1}_{(r+1, \dots, d)}(j) \hat{u}_{k,v}^T (R + B^T S_{k+1,j} B) \hat{u}_{k,v} + 2 \mathbf{1}_{(r+1, \dots, d)}(j) \hat{u}_{k,v}^T B^T S_{k+1,j} A x_k. \end{aligned}$$

Substituting (4.B.7) back into (4.B.6) yields

$$F_{N-k}^*(x_k, v) = \sum_{i \in \hat{\Xi}_v} \mathcal{M}_{N-k, \bar{\phi}(i)}(x_k, 0) p_{i, \bar{\phi}(i)} \pi_i + \min_{\hat{u}_{k,v}} \mathcal{M}_{N-k, \phi(i)}(x_k, \hat{u}_{k,v}) p_{i, \phi(i)} \pi_i.$$

Writing out $\mathcal{M}_{N-k, \bar{\phi}(i)}$ and $\mathcal{M}_{N-k, \phi(i)}$ yields

$$\begin{aligned} F_{N-k}^*(x_k, v) &= \sum_{i \in \hat{\Xi}_v} \sum_{j \in \{ \bar{\phi}(i), \phi(i) \}} \left[x_{N-1}^T (Q + A^T S_{k+1,j} A) x_{N-1} + \text{trace}(\Sigma_\omega S_{k,j}) + c_{k+1,j} \right] p_{i,j} \pi_i \\ &\quad + \min_{\hat{u}_{k,v}} \left[\sum_{i \in \hat{\Xi}_v} \left(\hat{u}_{k,v}^T (R + B^T S_{k+1, \phi(i)} B) \hat{u}_{k,v} + 2 \hat{u}_{k,v}^T B^T S_{k, \phi(i)} A x_k \right) p_{i, \phi(i)} \pi_i \right]. \quad (4.B.8) \end{aligned}$$

Here the last part can, using (4.21), be rewritten to

$$\min_{\hat{u}_{k,v}} \left[\hat{u}_{k,v}^T (\bar{R}_v + B^T \bar{S}_{k+1,v} B) \hat{u}_{k,v} + 2 \hat{u}_{k,v}^T B^T \bar{S}_{k+1,v} A x_k \right].$$

Taking the gradient of (4.B.8) with respect to $\hat{u}_{k,v}$ and setting this equal to 0 then results in (4.18) and (4.19). Inserting the result into D_i in (4.B.7) and rewriting this yields

$$\begin{aligned} D_i &= x_k^T S_{k,i} x_k + \sum_{j \in \{ \phi(i), \bar{\phi}(i) \}} (\text{trace}(\Sigma_\omega S_{k,j}) + c_{k+1,j}) p_{i,j} \\ &= x_k^T S_{k,i} x_k + c_{k,i} \\ &= \bar{J}_{N-k,i}^*(x_k), \quad (4.B.9) \end{aligned}$$

where $S_{k,i}$ is given in (4.20) and $c_{k,i}$ in (4.23). Combining (4.B.3), (4.B.6) and (4.B.9) then results in the cost (4.22). \square

Scheduling for control over wireless networks

In this chapter, we consider a large discrete-time distributed linear time-invariant (LTI) system where a central controller is connected to many actuators through a wireless network. This network is affected by packet dropouts and bandwidth limitations. Due to these bandwidth limitations, we can not address all actuators at every sampling interval. Hence, a scheduler is needed to select at each time-step which actuator to address. In addition, the controller has to select the control input that is transmitted to the addressed actuators. In this chapter, we present a co-design for an optimal controller and scheduler, such that at every time-step an optimal schedule and optimal control law are obtained. We present a method to co-design the controller/scheduler by minimizing a finite horizon linear quadratic (LQ) cost function. This results in an algorithm which provides online scheduling and control laws that can be implemented using model predictive control (MPC). Simulation studies show that significant performance gains can be achieved over heuristic round robin (RR) scheduling. The ideas that we present in this chapter are extended in Chapter 6.

This chapter is structured as follows: After the introduction and literature review, the networked control system (NCS) problem formulation is presented in Section 5.2. The controller and scheduler co-design is presented in Section 5.3. In Section 5.4, the performance of the co-design is compared to simple RR heuristics by the use of Monte Carlo simulations. A summary concludes this chapter in Section 5.5.

5.1 Introduction

The emerging development of wireless networks in industrial environments with possibilities for real-time communications has significantly accelerated the research area of networked estimation and control in recent years. Many applications already utilize NCSs, such as robotics, load control on power grids and the smart home [73], [75], [76].

In this section, we consider the control of large scale NCSs that are connected through wireless networks for which bandwidth limitations apply. If the NCS is sufficiently large, not all nodes of the system can be addressed at any time and some form of scheduling has to be utilized. Works such as [5], [7], [9], [38], [64]–[66], [71], [87], [103], [112], [118] utilize scheduling with the aim to lower the sample rate of NCSs. This is done by utilizing open- or closed-loop scheduling policies (self- or event-triggered control, respectively) for the controller and scheduler co-design. The main

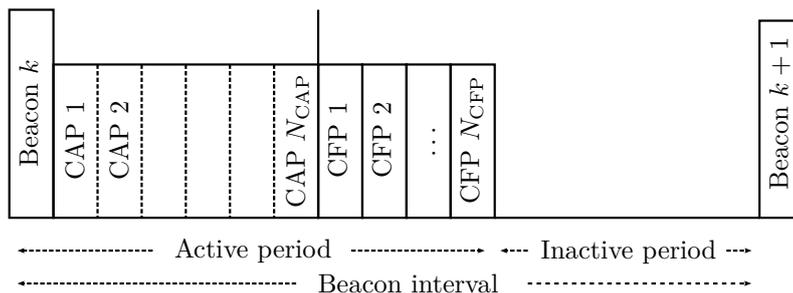


Figure 5.1: The IEEE 802.15.4 superframe structure.

motivation in these aforementioned studies is to reduce network utilization. While this can lead to lower power consumption of the nodes on the network, the network will be under-utilized. Instead, the NCS control and scheduling problem can be seen as a resource allocation problem; how can a finite resource (the network) be utilized optimally to satisfy control goals, such as the minimization of a cost-function? The works [42], [59], [60], [72], [92], [93], [111], [120], [131] address this problem in a deterministic setting by proposing a co-design with a combined scheduler and controller. This results in co-design laws on what actuators to address and what to send to the addressed actuators. However, as discussed in Section 2.2, wireless networks are subject to random effects such as disturbances, congestion and interference. This reduces the link quality and can result in packet delays and/or losses. As shown in Chapters 3 and 4, it is important that these effects are taken into account in the controller design [7], [9], [114].

In this chapter, we present a novel method that permits optimal utilization of the IEEE 802.15.4 superframe which is described in Section 2.2.1. Multiple widely implemented industry standards, such as WirelessHART and ISA100 are based on the IEEE 802.15.4 MAC layer. The IEEE 802.15.4 superframe (see Figure 5.1) contains both a more reliable contention free period (CFP) and a less reliable contention access period (CAP). For the control of a NCS, the CFP is preferred due to its increased reliability. The CFP utilizes time division multiple access (TDMA) to schedule transmissions, where a so-called network coordinator assigns slots in the CFP on a first-come-first-served base. A limitation is however, that the IEEE 802.15.4 standard only allows 7 slots to be assigned in the CFP of each superframe¹. When this is not sufficient, the less reliable CAP can be utilized. Transmissions in the CAP utilize a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. This however leads to increased chances for collisions and delays within the CAP. However, the CSMA/CA mechanism in the CAP allows for more flexibility, since transmissions in the CAP are not assigned by the network coordinator.

We consider a large distributed NCS that generally has more actuators than the number of available slots in the CFP in a superframe. The network is shared with other users that compete for bandwidth, which further reduces the amount of slots in the CFP that are accessible. This means, that one has to decide which actuators to address in the CFP, which ones in the CAP and, in case not all actuators can be addressed during the CFP and CAP, which ones not to address. Also, we need to select what input to send to the actuators that are addressed. The current chapter investigates optimal scheduling and control co-design over IEEE 802.15.4 networks where we take the advantage of both the CAP and CFP periods. We implement the algorithm using MPC ideas.

¹It is worth noting that the IEEE 802.15.4e extension allows more than 7 in the CFP [44].

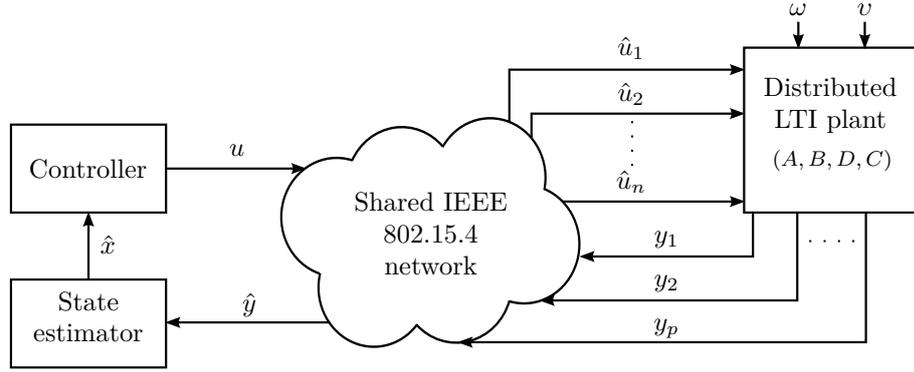


Figure 5.2: The control system with many actuator and sensor nodes.

The performance of the proposed co-design algorithm is compared to RR heuristics by the use of simulation studies.

Notation: In this chapter, the time indexing is done in brackets $x(k)$ instead of subscripts (x_k), which was used in previous chapters. We reserve the subscripted notation x_i to be the i 'th element in a vector x , or a_i the i 'th row in the matrix A .

5.2 Networked control setting

We consider a spatially distributed large control system with many sensors and actuators as illustrated in Figure 5.2. The sensors and actuators are placed at different physical locations and are not directly connected to each other. In Figure 5.2, \hat{y} and \hat{u} are the measurement and control signals, respectively, that are received after being transmitted over a shared wireless network. The measurements are transmitted to a state estimator that is co-located at the controller. The estimator produces the state estimate \hat{x} , that is transmitted directly to the controller. The control input u is transmitted to the actuators over a wireless network. The actuators are in general not interconnected, so each actuator has to be addressed individually. In this work, we use the optional acknowledgments from the IEEE 802.15.4 standard to confirm successful transmissions to the actuators. Thus, the controller knows the transmission outcomes of u . Since the estimator is co-located at the controller, the transmission outcomes of u are also available to the estimator.

The distributed systems all have their individual dynamics and share a synchronized sampling interval. The dynamics of the individual system are combined in the augmented NCS plant model

$$\begin{aligned} x(k+1) &= Ax(k) + B\hat{u}(k) + F\omega(k) \\ y(k) &= Cx(k) + \nu(k), \end{aligned} \tag{5.1}$$

where $x(k) \in \mathbb{R}^m$, $\hat{u}(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$ and A, B, F, C are matrices of appropriate dimensions. The plant disturbance $\omega(k) \sim \mathcal{N}(\mathbf{0}_d, \Sigma_\omega)$ and measurement disturbance $\nu(k) \sim \mathcal{N}(\mathbf{0}_p, \Sigma_\nu)$ are zero-mean Gaussian with covariances Σ_ω and Σ_ν , respectively. The control input that is received by the actuators is given by

$$\hat{u}(k) = \begin{bmatrix} \hat{u}_1(k) & \cdots & \hat{u}_n(k) \end{bmatrix}^T,$$

where the binary vector $\gamma(k) \in \mathbb{Z}_2^n$ and each

$$\hat{u}_j(k) = \begin{cases} u_j(k) & \text{if } \gamma_j(k) = 1 \\ 0 & \text{if } \gamma_j(k) = 0 \end{cases} \quad 1 \leq j \leq n, \quad (5.2)$$

with $\gamma_j(k) = 1$ indicating a successful transmission. The measurement vector that is received by the estimator is given by

$$\hat{y}(k) = [\hat{y}_1 \quad \cdots \quad \hat{y}_p]^T,$$

where each

$$\hat{y}_j(k) = \begin{cases} y_j(k) & \text{if } \psi_j(k) = 1 \\ 0 & \text{if } \psi_j(k) = 0 \end{cases} \quad 1 \leq j \leq p.$$

Definition 5.2.1 (Pseudo inverse). *In this section, we define the pseudo inverse for a symmetric block-diagonal matrix $P \in \mathbb{R}^{m \times m}$ with $\text{rank}\{P\} \leq m$ as*

$$P^\dagger \triangleq \bar{T}(P + T)^{-1}\bar{T},$$

where T and \bar{T} are block diagonal and satisfy

$$T = T^T \quad (5.3)$$

$$TPT = \mathbf{0} \quad (5.4)$$

$$\text{rank}\{P + T\} = m \quad (5.5)$$

$$\bar{T} = \bar{T}^T \quad (5.6)$$

$$\bar{T}\bar{T} = \bar{T} \quad (5.7)$$

$$\bar{T}P = P\bar{T} = P \quad (5.8)$$

$$\bar{T}T = T\bar{T} = \mathbf{0}. \quad (5.9)$$

Lemma 5.2.1. *The pseudo inverse in Definition 5.2.1 is unique and satisfies the following conditions:*

$$MM^\dagger M = M \quad (5.10)$$

$$M^\dagger MM^\dagger = M^\dagger \quad (5.11)$$

$$(MM^\dagger)^T = MM^\dagger \quad (5.12)$$

$$(M^\dagger M)^T = M^\dagger M. \quad (5.13)$$

Proof. The proof is shown in Appendix 5.A. □

5.2.1 Network considerations

The system (5.1) is connected through a wireless network that is shared with other users. The network is based upon the IEEE 802.15.4 topology, which is explained in greater detail in Section 2.2. In the current setting, we utilize two temporally separated superframes:

- one sensing superframe, in which sensor measurements are transmitted to the estimator, and

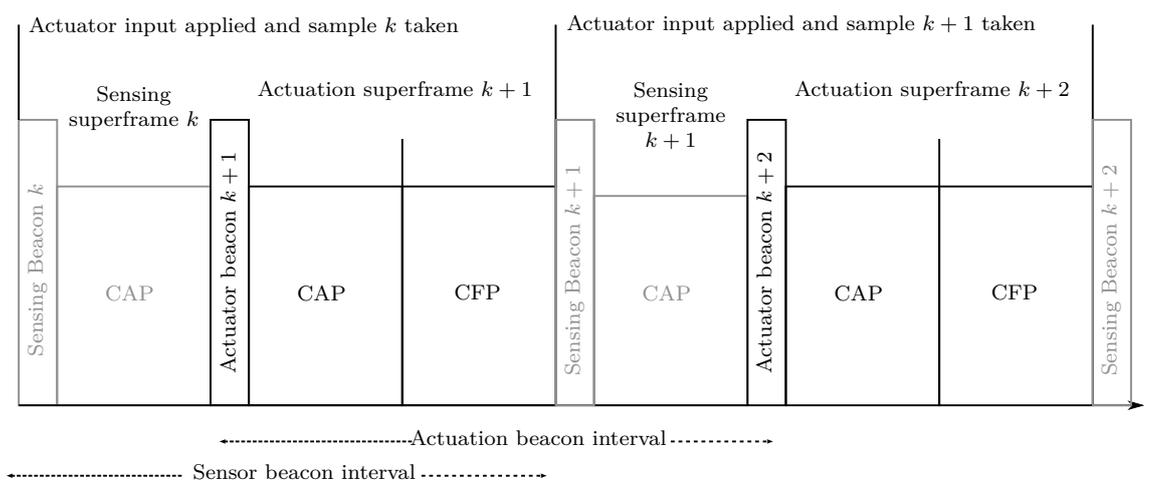


Figure 5.3: Superframe structure for the networked control setup. Here the sensor measurements at time-step k are transmitted in the sensing superframe, which finishes right before the actuation superframe commences. The state estimator then predicts the state $\hat{x}(k+1|k)$ which the controller uses to compute the control input $u(k+1)$ and schedule. The estimation superframe does not feature a CFP

- one actuation superframe, in which control signals are transmitted to the actuators.

This superframe structure is illustrated in Figure 5.3. It is worth noting, that the sensing superframe for the ease of exposition does not feature a CFP.

In this chapter, we focus on the co-design of the controller. Therefore, the sensor measurements are for simplicity all transmitted during the CAP of the sensing superframe in Figure 5.2. In the current setting, the sensor measurements are still affected by packet loss. Scheduling of the sensor measurements for the estimation can be considered a dual problem to the scheduling and co-design of the actuators.

Since we aim at maximizing the control performance for the system (5.1) over a finite resource (the number of transmissions that can fit in a superframe), the ideal strategy would be to utilize the entire superframe for the controller design. That is, 7 slots in the CFP and as many transmissions in the CAP as can fit in the duration of the CAP. This would give the optimal control performance provided that no other users utilize the network. However, since the network is shared with other users that also utilize bandwidth, the aforementioned strategy would result in network overload. We therefore, for fairness on the network, restrict the controller to only utilize a maximum number of N_{CFP} and N_{CAP} slots in the CFP and CAP, respectively, in each superframe. This is illustrated in Figure 5.1.

We synchronize the sampling interval of the augmented system (5.1) to the beacon interval of the sensing superframe. This is illustrated in Figure 5.3. This means, that at the beacon interval of the sensing superframe, all controls in the vector $\hat{u}(k)$ are applied to the actuators, and the measurement $y(k)$ is taken and transmitted. Between two sensing superframe beacons, the following actions are performed:

1. At time k , controls $\hat{u}(k)$ are applied at the actuators and measurement $y(k)$ is sampled and transmitted in the sensing superframe.

2. The estimator receives $\hat{y}(k)$ and computes state prediction $\hat{x}(k+1)$.
3. The controller computes the optimal schedule and control inputs $u(k+1)$
4. The schedule is applied and the controls are transmitted during the actuation superframe

It is worth noting that in the above, the controller uses the prediction of the state $x(k+1)$, $\hat{x}(k+1)$ to compute the optimal schedule and controls. The reason for this is, that the control inputs will first be applied at the actuators at time $k+1$. We describe in more detail how this is implemented in Section 5.2.2. By synchronizing the sampling interval to the beacon frame of the superframe, we can ignore inter-transmission times and delays within the superframe.

Remark 5.2.1. *It is also possible to transmit both the sensor measurements and control inputs in the same superframe. In this case, the control inputs are not applied at the actuators before the sensors are sampled. This means, that the controller at time k needs to transmit control inputs that are applied when sample $k+1$ is taken. However, the controller is only receiving sample k while it is transmitting the controls that are applied at sample $k+1$. Thus, the controller has to compute the control inputs that are applied at sample $k+1$ using the state prediction that is based on the measurement that is taken at time $k-1$.*

In the current setting, we utilize the following assumption for delayed transmissions.

Assumption 5.2.1. *We consider transmissions in a CAP or CFP that are delayed until the next superframe as being lost. These occur with fixed packet dropout probabilities $1-p_{CAP}$ and $1-p_{CFP}$, respectively, where $p_{CFP} > p_{CAP}$. For simplicity we consider the packet loss rates to be constants and known.*

While Assumption 5.2.1 might seem restrictive, it is worth noting, that the controller at the next superframe has new data available. This data is used to compute an updated schedule and control laws. It is therefore preferable to transmit these new control inputs instead of attempting to retransmit the old control inputs.

Remark 5.2.2. *The beginning of the superframe is indicated by a beacon frame, that the network coordinator transmits. If a user does not receive this beacon frame it shall, according to the IEEE 802.15.4 standard [43], not attempt to transmit data in that superframe. The probability that a user does not receive a beacon frame can be incorporated in the packet loss probabilities $1-p_{CAP}$ and $1-p_{CFP}$.*

5.2.2 System setting

With the network described, we now present a more detailed description of the system (5.1) and actuator strategies (5.2). First recall, that the sensor measurements are transmitted over the network in the CAP. We need a state estimator to obtain an estimate of the state that then is used in the controller and scheduler co-design. We utilize an estimator that takes packet dropouts into account to perform the state estimation. This estimator is based on the Kalman filter that is

described in Section 2.3.2, and is presented in [100], [105]. This leads to the optimal estimator

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + B\hat{u}(k) + K(k)(\hat{y}(k) - C(\psi(k))\hat{x}(k)) \\ P(k+1) &= AP(k)A^T + \Sigma_\omega - K(k)C(\psi(k))P(k)A^T,\end{aligned}\quad (5.14)$$

where the matrix

$$C(\psi(k)) \triangleq \begin{bmatrix} \psi_1(k)C_1 \\ \vdots \\ \psi_p(k)C_p \end{bmatrix},$$

with $\Pr\{\psi_i = 1\} = p_{\text{est}}$. The Kalman gain is given by

$$K(k) = P(k)C^T(\psi(k))\left(R + C(\psi(k))P_{k+1}C(\psi(k))^T\right)^\dagger,$$

where \dagger denotes the matrix pseudo inverse defined in Definition 5.2.1.

Since successful packet transmissions are acknowledged (the transmission control protocol (TCP) case), the estimation error becomes

$$\begin{aligned}x(k+1) - \hat{x}(k+1) &= Ax(k) + B\hat{u}(k) + F\omega(k) - (A\hat{x}(k) + B\hat{u}(k) + K(k)(\hat{y}(k) - C(\psi(k))\hat{x}(k))) \\ &= A(x(k) - \hat{x}(k)) - K(k)C(\psi(k))(x(k) - \hat{x}(k)) - K(k)\hat{v}(k) + F\omega(k),\end{aligned}\quad (5.15)$$

where

$$\hat{v}_i(k) = \begin{cases} \nu_i(k) & \text{if } \psi_i(k) = 1 \\ 0 & \text{if } \psi_i(k) = 0 \end{cases} \quad 1 \leq i \leq p.$$

It is worth noting that (5.15) does not depend on the control input $\hat{u}(k)$. This means that the optimal estimator can be designed separately from the controller.

In the following, we consider the controller and scheduler co-design for the state feedback case. That is, the controller/scheduler has full access to the state $x(k)$. In Section 5.3.2 we briefly discuss controller and scheduler co-design for the case where the controller does not have access to the state $x(k)$ and instead receives intermittent observations $y(k)$.

As already mentioned in Section 5.2.1, the actuators are to be controlled over an IEEE 802.15.4 compatible network, which can only support a limited amount of data during the CAP and CFP. We are interested in a situation where the combined amount of available slots in the CAP and CFP in each superframe is less than the amount of actuator nodes in the system. This means, that we have to schedule which actuators to address in every actuation superframe. Recall further from Assumption 5.2.1, that a packet which at time k is delayed until a future superframe, is considered as being lost. The probabilities for these delays to occur are independent and identically distributed (*i.i.d.*) and are included in the packet dropout probabilities p_{CAP} or p_{CFP} .

We let the length n vector $S(k)$ be the schedule at time k , given by

$$S(k) \triangleq \begin{bmatrix} s_1(k) & s_2(k) & \dots & s_n(k) \end{bmatrix}^T \in \mathbb{S}, \quad (5.16)$$

where each entry, $s_j(k)$, indicates whether actuator j is addressed during the CAP, CFP or is not addressed. To indicate in which period actuator j is addressed, we set the entry $s_j(k)$ equal to

the probability of a successful transmission during the period in the superframe. Thus, the set \mathbb{S} is defined as

$$\mathbb{S} \triangleq \left\{ S = \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \middle| s_i \in \{0, p_{\text{CAP}}, p_{\text{CFP}}\} \right\}, \quad (5.17)$$

where p_{CAP} and p_{CFP} are the (known or estimated) probabilities for a successful transmission in the CAP and CFP, respectively. Denote the set of admissible schedules by $\mathbb{S}_a \subset \mathbb{S}$. An admissible schedule satisfies that, N_{CAP} elements are p_{CAP} and N_{CFP} elements are p_{CFP} , while the remaining elements are zero. The set \mathbb{S}_a is formally stated as

$$\mathbb{S}_a \triangleq \left\{ S \in \mathbb{S} \middle| \sum_{i=1}^n \mathbb{1}_{p_{\text{CAP}}}(s_i) = N_{\text{CAP}} \wedge \sum_{j=1}^n \mathbb{1}_{p_{\text{CFP}}}(s_j) = N_{\text{CFP}} \right\}, \quad (5.18)$$

where the indicator function $\mathbb{1}_a(b) = 1$ if $a = b$ and zero otherwise.

A length N scheduling sequence is then defined by

$$\mathcal{S}_{k,N} \triangleq (S(\ell) \in \mathbb{S} : \ell = k, \dots, N-1),$$

while a length N admissible scheduling sequence is defined by

$$\mathcal{S}_{k,N} \triangleq (S(\ell) \in \mathbb{S}_a : \ell = k, \dots, N-1) \in \Xi_N,$$

where Ξ_N is the set of all admissible scheduling sequences. Further, let $\mathcal{S}_N \triangleq \mathcal{S}_{0,N}$.

The vector $\gamma(k) \in \mathbb{Z}_2^n$ contains all transmission outcomes for the control inputs that are transmitted in superframe k . Recall from (5.2) that $\gamma_j(k) = 1$ if the transmission to actuator j was successful and $\gamma_j(k) = 0$ otherwise. Thus the distribution of $\gamma_j(k)$ is given by

$$\Pr \{\gamma_j(k) = 1; S(k)\} = s_j(k), \quad j \in \{1, 2, \dots, n\},$$

where $\Pr \{A; B\}$ means that the distribution of the random variable A is parameterized by B .

As described in Section 2.3, there are various possibilities of what the actuator should do in case it either is not addressed at time k , or the transmission of the packet containing the control signal failed. However, for the ease of exposition, in what follows we adopt the set-to-zero strategy from (5.2).

Define

$$B(\gamma(k)) \triangleq B \text{diag} \{\gamma(k)\}, \quad (5.19)$$

where $\text{diag} \{\gamma(k)\}$ creates a $n \times n$ matrix with the entries of $\gamma(k)$ on its diagonal. Using this, we combine (5.1) and (5.2) to obtain the NCS model

$$x(k+1) = Ax(k) + B(\gamma(k))u(k) + F\omega(k). \quad (5.20)$$

Example 5.2.1. In (5.1), consider the input matrix

$$B = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix} \in \mathbb{R}^{m \times 4},$$

and the schedule at time k given by

$$S(k) = \begin{bmatrix} p_{CFP} & p_{CAP} & p_{CAP} & 0 \end{bmatrix}^T.$$

If the transmission outcomes at time k are

$$\gamma(k) = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T,$$

then,

$$B(\gamma(k)) = B \operatorname{diag} \left\{ \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \right\} = \begin{bmatrix} b_1 & b_2 & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

This means that the first and second actuators will receive their control input. \square

Within the current setup, scheduling amounts to designing the schedule $S(k)$ in (5.16). Since $p_{CFP} > p_{CAP}$ one would ideally like to assign every actuator in the CFP in each superframe. However, this can only be done if the number of actuators does not exceed the number of available slots in the CFP (N_{CFP}). Otherwise, the CAP needs to be used as well. Further, if the number of actuators exceeds the total amount of available slots in the CFP and CAP ($N_{CFP} + N_{CAP}$), some actuators will not be addressed at all in that superframe. This raises the question: Which actuators to address in which slot in a superframe, which ones to omit and what data should be sent?

To solve this scheduling and controller co-design problem, ideally one would like to utilize the scheduling sequence and control policies that minimize an infinite horizon LQ cost function. However, it is in this case not feasible to evaluate an infinite horizon cost function since this results in an infinitely long scheduling sequence and associated control policy. The matrix $B(\gamma(k))$ is further stochastic and depends on $S(k)$ for each k . Due to these uncertainties in the system, there might not exist a solution to the infinite horizon cost function, see e.g. [10], [15] and references therein. We will therefore use the ideas of MPC with a finite horizon cost function to address the scheduling and controller co-design problem.

5.3 Stochastic control formulation

In this section we want to design a sequence of control laws

$$U_N^* \triangleq (u^*(\ell) : \ell = 0, \dots, N-1) \tag{5.21}$$

and schedules

$$\mathcal{S}_N^* \triangleq (S^*(\ell) \in \mathbb{S}_a : \ell = 0, \dots, N-1), \tag{5.22}$$

such that the LQ cost

$$J_W(x(0), \mathcal{S}_N, U_N) \triangleq \mathbf{E} \left\{ \|x(N)\|_W^2 + \sum_{\ell=0}^{N-1} \|x(\ell)\|_Q^2 + \|u(\ell)\|_{R(\gamma(\ell))}^2 \middle| x(0); \mathcal{S}_N \right\} \tag{5.23}$$

is minimized, where

$$\mathbf{E} \{A; B\} \triangleq \int A \Pr \{A; B\} dA.$$

In (5.23), the positive semi-definite tuning parameters $Q \geq 0$ and $W \geq 0$ penalize the state. The random matrix $R(\gamma(k)) \geq 0$ penalizes the control input for successful transmissions. Each $r_{ij}(\gamma(k))$ of $R(\gamma(k))$ is given by

$$r_{ij}(\gamma) \triangleq \begin{cases} r_{ij} & \text{if } \gamma_i = 1 \text{ and } \gamma_j = 1 \\ 0 & \text{otherwise} \end{cases} \quad 0 < i, j \leq n,$$

where r_{ij} is the i, j 'th element of the tuning parameter $R \geq 0$. The parameters Q , W and R can be tuned to trade-off control performance against control effort. The optimal control and scheduling laws (5.21) and (5.22) are then found by

$$(U_N^*, \mathcal{S}_N^*) \triangleq \arg \min_{U_N, \mathcal{S}_N \in \Xi_N} J(x(0), \mathcal{S}_N, U_N), \quad (5.24)$$

where \mathcal{S}_N is minimized over all admissible scheduling sequences. This results in the optimal cost

$$J_W^*(x(0)) \triangleq J_W(x(0), \mathcal{S}_N^*, U_N^*).$$

5.3.1 Solution to the minimization problem

The joint optimization problem (5.24) is a mixed integer problem, where the scheduling sequence \mathcal{S}_N is a discrete variable. Here recall that each element s_j of each schedule $S(k)$ in the sequence \mathcal{S}_N takes values in the set $\{0, p_{\text{CAP}}, p_{\text{CFP}}\}$. This is therefore a difficult problem to solve. However, we can rewrite problem (5.24) to separate the integer and the continuous part of the optimization problem. This yields

$$J_W^*(x(0)) = \min_{\mathcal{S}_N \in \Xi_N} \left[\min_{U_N} [J_W(x(0), \mathcal{S}_N, U_N)] \right]. \quad (5.25)$$

This means, that we first can solve the inner problem for each possible scheduling sequence \mathcal{S}_N . This problem is convex for each \mathcal{S}_N and has an analytical solution. The optimal control laws for a known scheduling sequence \mathcal{S}_N are presented next.

Theorem 5.3.1. *For a known scheduling sequence $\mathcal{S}_N \in \mathbb{S}$, the optimal controls are given by the policy*

$$u(k) = -L_k^* x(k) \quad k = (0, \dots, N-1), \quad (5.26)$$

where the control gain is given by the $n \times m$ matrix

$$L_k^* = [R(S(k)) + M(P_{k+1}, S(k))]^\dagger B^T(S(k)) P_{k+1} A, \quad (5.27)$$

with

$$M(X, S(k)) \triangleq \sum_{s \in \mathbb{Z}_2^n} B(s)^T X B(s) \Pr\{\gamma = s; S(k)\}, \quad (5.28)$$

and \dagger denoting the pseudo matrix inverse defined in Definition 5.2.1². In (5.27), the matrix P_k is given by the recursion

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B(S(k)) [R(S(k)) + M(P_{k+1}, S(k))]^\dagger B^T(S(k)) P_{k+1} A \quad (5.29)$$

²Since $S(k)$ is zero for actuators that are not addressed, the block diagonal matrices $R(S(k))$ and $M(P_{k+1}, S(k))$ are in general singular.

with $P_N = W$. This results in the optimal cost for the known scheduling sequence \mathcal{S}_N

$$J_W(x(0), \mathcal{S}_N, U_N^*) = x(0)^T P_0 x(0) + \sum_{\ell=1}^N \text{trace}(\Sigma_\omega F^T P_\ell F). \quad (5.30)$$

Proof. The proof is presented in Appendix 5.B. \square

Theorem 5.3.1 presents the closed form solution for a known scheduling sequence $\mathcal{S}_N \in \mathbb{S}$. Note however, that (5.26), (5.27) and (5.29) all depend on the particular scheduling sequence \mathcal{S}_N . We therefore define (5.27) and (5.29) as functions of the scheduling sequence. Thus

$$\begin{aligned} P_k(X_{k,N}) &\triangleq Q + A^T P_k(X_{k+1,N}) A - A^T P_{k+1}(X_{k+1,N}) B(X(k)) \\ &\quad [R(X(k)) + M(P_{k+1}(X_{k+1,N}), X_{k,N})]^\dagger B^T(X(k)) P_{k+1}(X_{k+1,N}) A \quad (5.31) \\ L_k^*(X_{k,N}) &\triangleq [R(X(k)) + M(P_{k+1}(X_{k+1,N}), X_{k,N})]^\dagger B^T(X(k)) P_{k+1}(X_{k+1,N}) A, \end{aligned}$$

such that for a given scheduling sequence \mathcal{S}_N the control input is given by

$$u(k) = -L_k^*(\mathcal{S}_{k,N}) x(k).$$

Also, denote the sequence of optimal control laws

$$U_N^*(\mathcal{S}_{k,N}) = (-L_\ell^*(\mathcal{S}_{\ell,N}) x(\ell) : \ell = 0, \dots, N-1).$$

Now the optimal scheduling sequence \mathcal{S}_N^* can be found by

$$\begin{aligned} \mathcal{S}_N^* &= \arg \min_{\mathcal{S}_N \in \Xi_N} [J_W(x(0), \mathcal{S}_N, U_N^*(\mathcal{S}_N))] \\ &= \arg \min_{\mathcal{S}_N \in \Xi_N} \left[x(0)^T P_0(\mathcal{S}_N) x(0) + \sum_{\ell=1}^N \text{trace}(\Sigma_\omega F^T P_\ell(\mathcal{S}_{\ell,N}) F) \right], \quad (5.32) \end{aligned}$$

where \mathcal{S}_N is minimized over all admissible scheduling sequences. This results in the length N sequence of optimal control laws

$$U_N^*(\mathcal{S}_N^*) = (u^*(\ell) : \ell = 0, \dots, N-1)$$

where each control law is given by

$$u^*(k) = -L_k^*(\mathcal{S}_{k,N}^*) x(k). \quad (5.33)$$

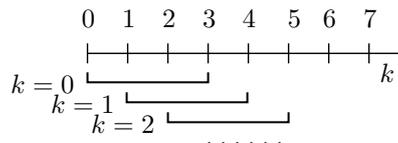
5.3.2 Separation

When output feedback is used instead of state feedback, the controller has at each time-step k access to the following information

$$\begin{aligned} \mathcal{I}(k) &\triangleq (y_k, \gamma_k, v_k, S(k), \mathcal{I}(k-1)) \\ \mathcal{I}(0) &\triangleq (\mathbf{E}\{x(0)\}). \end{aligned}$$

The cost (5.23) then becomes

$$J_W(\mathcal{I}(0), \mathcal{S}_N, U_N) \triangleq \mathbf{E} \left\{ \|x(N)\|_W^2 + \sum_{\ell=0}^{N-1} \|x(\ell)\|_Q^2 + \|u(\ell)\|_{R(\gamma(\ell))}^2 \middle| \mathcal{I}(0) ; \mathcal{S}_N \right\}.$$

Figure 5.4: Illustration of MPC where $N = 3$.

It still holds true, that the joint minimization of this cost with respect to the scheduling sequence and controls can be separated into two problems

$$J_W^*(\mathcal{I}(0)) = \min_{\mathcal{S}_N \in \Xi_N} \left[\min_{U_N} [J_W(\mathcal{I}(0), \mathcal{S}_N, U_N)] \right]. \quad (5.34)$$

The solution to the inner minimization problem is presented for the state feedback case in Theorem 5.3.1. It is straightforward to find the optimal controls for a known scheduling sequence for the inner cost in (5.34) by following the arguments from the work [100] and the steps in the proof of Theorem 5.3.1, which is found in Appendix 5.B. It then shows that when output feedback is used, the optimal control is given by

$$u(k) = -L_k^*(\mathcal{S}_{k,N}) \hat{x}(k|k).$$

The optimal scheduling sequence at each time step is then given by

$$\begin{aligned} \mathcal{S}_N^* &= \arg \min_{\mathcal{S}_N \in \Xi_N} [J_W(\mathcal{I}(k), \mathcal{S}_N, U_N^*(\mathcal{S}_N))] \\ &= \arg \min_{\mathcal{S}_N \in \Xi_N} \left[\hat{x}(k|k)^T P_0(\mathcal{S}_N) \hat{x}(k|k) + \sum_{\ell=1}^N \text{trace}(\Sigma_\omega F^T P_\ell(\mathcal{S}_{\ell,N}) F) \right]. \end{aligned}$$

Thus, as shown in Section 5.2.2, the optimal estimator can be designed independently from the controller. Also, the optimal controller and scheduler can be designed using state feedback, and the same control and scheduling laws apply when the state is replaced by the optimal state estimate. Therefore, the separation principle holds for the controller and scheduler co-design.

5.3.3 Implementation

The finite horizon controller-scheduler (5.32) and (5.33) can be implemented as MPC. In the MPC algorithm, one computes (5.32) at every time-step. Here, only the first schedule $\mathcal{S}^*(0)$ in the sequence \mathcal{S}_N^* and the first control gain $L_0^*(\mathcal{S}_N^*)$ are used. The rest is discarded. The MPC algorithm is stated in Algorithm 5.1 and illustrated in Figure 5.4.

Algorithm 5.1 MPC algorithm.

- Input:** state $x(k)$
Compute: $\mathcal{S}_{0,N}^*$ using (5.32)
Compute: $u^*(k) = -L_0^*(\mathcal{S}_{0,N}^*) x(k)$ as in (5.33)
Output: schedule $\mathcal{S}^*(0)$ and control $u^*(k)$
Discard: schedules $\mathcal{S}_{1,N}^*$ and controls $U_{1,N}^*$
-

Studies show, that a longer prediction horizon N results in better control performance [62]. This however, results in an exponentially increasing complexity to find the optimal scheduling

sequence and control laws. It is therefore desirable to maintain the horizon length N relatively short. Another parameter that has an impact on the control performance is the weighting of the terminal cost W in (5.23). By choosing W with care, one can obtain significant improvements in the control performance of the NCS, and in some cases, guarantee closed-loop stability [55, Theorem 1], [63], [124]. An examples for a choice of W is be discussed in Section 6.3.2.2. This motivates the use of an infinite horizon approximation of the cost (5.23), which is presented in Chapter 6. Next, we will discuss computational considerations of Algorithm 5.1.

5.3.4 Numerical considerations

It is worth to note that the minimization problem (5.32) is a combinatorial problem. Also, while the control laws $L_k^*(\mathcal{S}_{k,N})$ for each scheduling sequence can be computed offline, the optimal scheduling sequence \mathcal{S}_N^* depends directly on the system state $x(k)$. This means that (5.32) can not be computed offline. An exhaustive search has to be performed to evaluate (5.32) where (5.30) has to be evaluated for every scheduling sequence $\mathcal{S}_N \in \Xi_N$. Recall, that Ξ_N is the set that contains all admissible scheduling sequences for a horizon of length N . For N_{CFP} slots in the CFP and N_{CAP} slots in the CAP, the cardinality of Ξ_N , which is equal to the number of admissible scheduling sequences, is given by

$$\text{card}(\Xi_N) = \left[\binom{n}{N_{\text{CFP}}} \binom{n - N_{\text{CFP}}}{N_{\text{CAP}}} \right]^N, \quad (5.35)$$

where

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

is the binomial coefficient and n is the number of actuators to address. It is worth noting that in (5.35), the cardinality increases exponentially as the horizon length N increases. This results in an exponentially increasing computational complexity and is known as the *curse of dimensionality* [15, Chapter 6].

However, the computational requirements can be reduced by some extent, since the Riccati recursion (5.31) does not depend on the state x . This allows for the offline computation of (5.31). Also, the optimal cost (5.30) for a scheduling sequence \mathcal{S}_N can be stated as

$$J_W(x(0), \mathcal{S}_N, U_N^*) = x(0)^T P_0 x(0) + \text{noise}(\mathcal{S}_N),$$

where

$$\text{noise}(\mathcal{S}_N) \triangleq \sum_{\ell=1}^N \text{trace}(\Sigma_\omega F^T P_\ell(\mathcal{S}_{k,N}) F) \quad (5.36)$$

can be computed offline. This reduces the online computations of (5.32) significantly. Algorithm 5.2 shows the online part of (5.32) with the required floating point operations (flops) indicated for each step.

The number of flops that is required in total at each time-step to find the optimal scheduling sequence and control laws is then

$$\frac{3}{2} \text{card}(\Xi_N) (m^2 + m) + \text{card}(\Xi_N) - 1 \quad \text{flops}, \quad (5.37)$$

where $\text{card}(\Xi_N)$ is given by (5.35).

Algorithm 5.2 Computing (5.32) at each time-step

Computed offline:noise(s) $s \in \Xi_N$ $P_0(s)$ $s \in \Xi$ **Input:** $x(k)$ **for** $s \in \Xi_N$ **do** $\text{cost}(s) = x(k)^T P_0(s) x(k)$ $\text{cost}(s) = \text{cost}(s) + \text{noise}(s)$ **end for** $\mathcal{S}_N^* = \arg \min_{s \in \Xi} \text{cost}(s)$ $\triangleright \text{card}(\Xi_N)$ times $\triangleright \frac{3}{2}(m^2 + m) - 1$ flops $\triangleright 1$ flop $\triangleright \text{card}(\Xi_N) - 1$ flops

Remark 5.3.1. Many works that considered actuator selection and scheduling over networks came to the conclusion that an exhaustive search has to be performed to find the optimal scheduling sequence, see for example [42], [92], [93]. These works propose to use simulated annealing [48] as well as branch and bound algorithms to obtain solutions to the combinatorial problem (5.32). A heuristic, that not necessarily results in the optimal schedule is presented in [93].

The controller and scheduler co-design is done using state augmentation. However, when the system (5.1) consists of multiple distinct sub-systems, the matrix A is sparse since the sub-systems are contained on the block diagonal. In this case, the control gain L_k^* (5.27) and Riccati equation P_k (5.29) are also sparse. The computational complexity for computing (5.32) can be reduced significantly by utilising this sparsity in implementations. Another option is to split the system (5.1) into distinct equations for each subsystem. For example, for d decoupled sub-systems, the minimization problem (5.32) can be rewritten as

$$\mathcal{S}_N^* = \arg \min_{\mathcal{S}_N \in \Xi_N} \left[\sum_i^d x_i(0) P_{i,0}(\mathcal{S}_{i,N}) x_i(0) + c_i(\mathcal{S}_{i,N}) \right]. \quad (5.38)$$

This reduces the number of multiplications significantly. In (5.38) the Riccati recursion $P_{i,0}$ can be found as the i 'th block-diagonal of P in (5.31). If one aims at reducing the offline computational complexity, one can instead compute each block diagonal $P_{i,k}$ of P_k individually.

To use Algorithm 5.2, (5.31) and (5.36) have to be computed offline. While it involves no difficulties to compute (5.31) for every possible scheduling sequence $\mathcal{S}_N \in \Xi$, it is possible to reduce the amount of offline computations significantly by using a periodic assumption on the scheduling sequence \mathcal{S}_N . Another method to reduce the length of the scheduling sequence \mathcal{S}_N is to carefully design the final state weighting W in the cost (5.23). These approaches are considered in more detail in Chapter 6.

5.4 Simulations

In this section, we study the performance of the scheduling and controller co-design algorithm that is presented in Algorithm 5.1 through simulation studies. We first shortly present a branch and bound approach that can reduce the amount of computations needed in the exhaustive search in (5.32). This is presented in Algorithm 5.2. This is followed by a simulation where three actuators compete for one slot in the CFP and one slot in the CAP, after which we simulate a scenario where four actuators compete for one slot in the CFP and one slot in the CAP.

5.4.1 Branch and bound algorithm

Although the presented algorithm requires an exhaustive search to be performed, where the amount of flops required is given by (5.37), the amount of searches required can be reduced using a simple branch and bound algorithm. When the system is in steady-state and $x(k)$ is zero, the only part contributing to the cost (5.30) is the constant part (5.36), that depends on the variance of the disturbance and the selected schedule. These parts can be calculated offline. When the cost (5.30) is computed for every possible scheduling sequence, we sort the costs $J_W(\mathbf{0}, \mathcal{S}_N, U_N(\mathcal{S}_N))$ in a list in ascending order. That is, the first cost in the list,

$$J_W(\mathbf{0}, \mathcal{S}_N^1, U_N(\mathcal{S}_N^1)) = \min_{\mathcal{S}_N \in \Xi_N} J_W(\mathbf{0}, \mathcal{S}_N, U_N(\mathcal{S}_N)),$$

where Ξ is the set of all possible scheduling sequences. Cost d in the list is then given by

$$\begin{aligned} J_W(\mathbf{0}, \mathcal{S}_N^d, U_N(\mathcal{S}_N^d)) &= \min_{\mathcal{S}_N \in \{\Xi_N \setminus \mathcal{S}_N^{1:d-1}\}} J_W(\mathbf{0}, \mathcal{S}_N, U_N(\mathcal{S}_N)) \\ &= \min_{\mathcal{S}_N \in \{\Xi_N \setminus \mathcal{S}_N^{1:d-1}\}} \text{noise}(\mathcal{S}_N), \end{aligned}$$

where by a slight abuse of notation, the set $\mathcal{S}_N^{1:d} = \{\mathcal{S}_N^i : i = 1, \dots, d\}$. By sorting the scheduling sequences, such that the constant part (5.36) is sorted from low to high values, (5.32) can be terminated when for some index d

$$\min_{\mathcal{S}_N \in \mathcal{S}_N^{1:d-1}} J_W(x(k), \mathcal{S}_N, U_N^*(\mathcal{S}_N)) \leq J_W(\mathbf{0}, \mathcal{S}_N^d, U_N(\mathcal{S}_N^d)).$$

The algorithm is illustrated in Algorithm 5.3.

Algorithm 5.3 Branch and bound approach used for simulations

Input: state $x(k)$

Variable: $i^* = 1$

Compute: $J_W^{i^*} \triangleq J_W(x(k), \mathcal{S}_N^1, U_N^*(\mathcal{S}_N^1))$.

▷ Initial optimal state

for $j = 2, \text{card}(\Xi_N)$ **do**

if $J_N^{i^*} \leq J_W(\mathbf{0}, \mathcal{S}_N^j, U_N^*(\mathcal{S}_N^j))$ **then**

▷ Optimal scheduling sequence found

 Optimal scheduling sequence $\mathcal{S}_N^* = \mathcal{S}_N^j$.

 Optimal control laws $U_N^*(\mathcal{S}_N^*)$

Break

▷ Exit the loop

end if

$J_W^{i^*} = \arg \min_{\{i^*, j\}} \{J_W^{i^*}, J_W(x(k), \mathcal{S}_N^j, U_N^*(\mathcal{S}_N^j))\}$

▷ Test next scheduling sequence

end for

5.4.2 Numerical example

In this section, we illustrate the idea of the online scheduling and controller co-design, as well as the advantage of online scheduling over offline scheduling heuristics. We consider a NCS with three actuators that compete for one slot in the CFP one slot in the CAP. The parameters of the

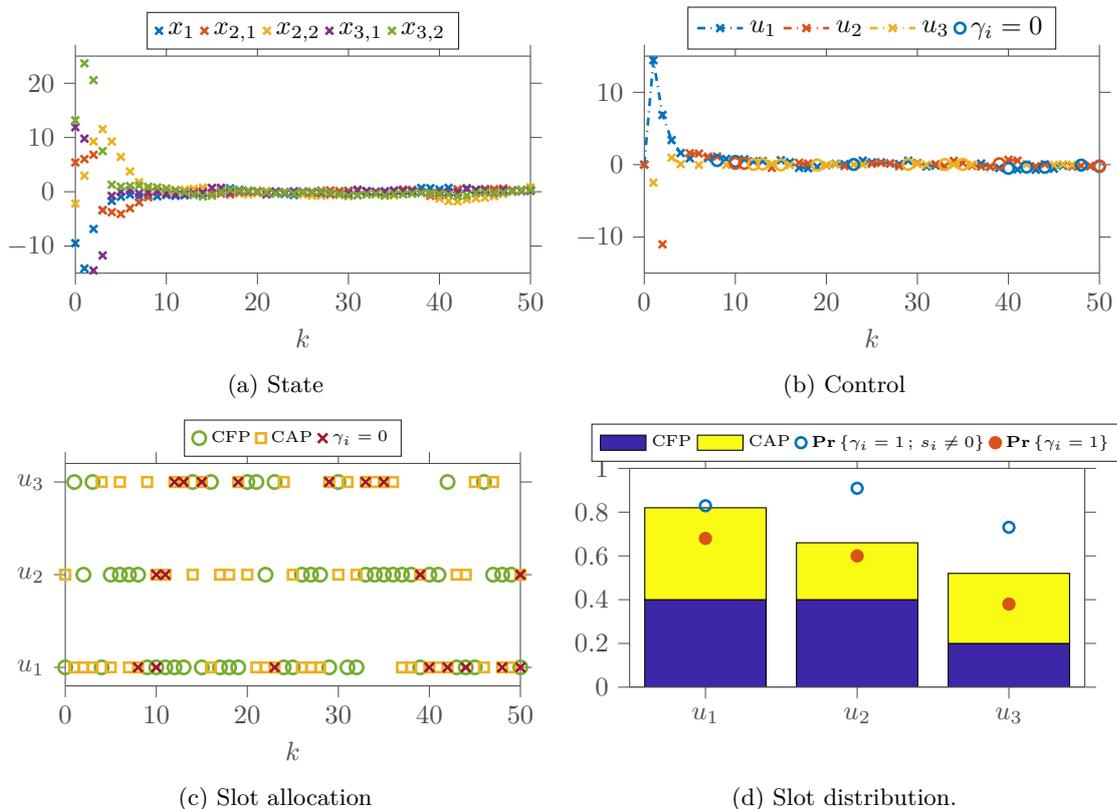


Figure 5.5: The evolution of the state Figure 5.5a and control Figure 5.5b for a simulation of (5.39) using the co-design presented in Algorithm 5.1. In Figure 5.5b, the circles indicate a packet loss on a scheduled transmission. The allocation of slots at each time-step is shown in Figure 5.5c, where a red cross indicates a packet loss. Each actuator's share of the CAP and CFP is shown in Figure 5.5d, where also the average sampled probability for successful transmission when the actuator is addressed ($\Pr\{\gamma_i = 1; s_i \neq 0\}$) and the overall sampled probability for a successful transmission on each actuator ($\Pr\{\gamma_i = 1\}$).

system model (5.1) are given by

$$A = \begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 \\ 0 & 1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0.819 & 0 \\ 0 & 0 & 0 & 0.906 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 9.063 \\ 0 & 0 & 4.683 \end{bmatrix}. \quad (5.39)$$

The plant disturbance $\omega(k) \sim \mathcal{N}(\mathbf{0}_5, 0.01I_5)$ and $F = I_5$. We can observe all states such that $C = I_5$ and the measurement noise $\nu(k) \sim \mathcal{N}(\mathbf{0}_5, I_5)$. In this setting, we select the weighting matrices as $Q = I_5$, $R = 2I_3$ and $W = I_5$. This system has open loop eigenvalues at 1.5, 1, 0.82 and two eigenvalues located at 1.1. The probabilities for successful transmissions are $p_{\text{CFP}} = 0.95$, $p_{\text{CAP}} = 0.75$. We select the horizon length to $N = 3$.

We consider the superframe structure that is presented in Section 5.2.1. This means, that we use the state estimator (5.14) to compute the prediction $\mathbf{E}\{x(k+1)|\hat{y}(k)\}$. This prediction is then used in Algorithm 5.1 to find the optimal scheduling sequence and control law. We assume

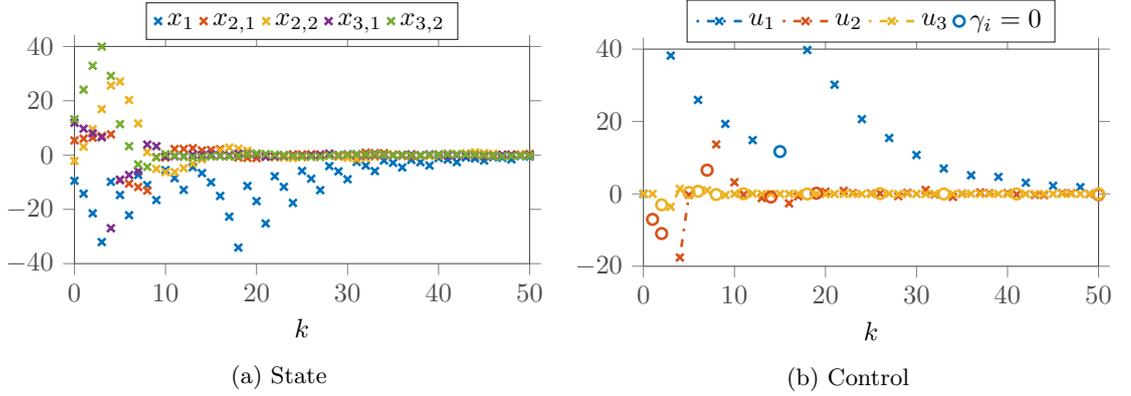


Figure 5.6: The evolution of the state Figure 5.6a and control Figure 5.6b for a simulation of (5.39) using RR schedule. In Figure 5.6b, the circles indicate that a transmission was scheduled, but a packet dropout occurred.

for ease of exposure that all measurements are transmitted in one packet during the CAP in the estimation superframe. The probability for successful transmission $p_{\text{est}} = p_{\text{CAP}} = 0.75$.

Figures 5.5a and 5.5b shows the evolution of the state and control inputs for system (5.39) using the co-design in Algorithm 5.1. This MPC algorithm computes at each time-step k both an optimal scheduling sequence and optimal control laws. We compare the developed MPC algorithm to a simple RR heuristic, where the access to the CFP and CAP is shared equally among the actuators. This means that we offline select a scheduling sequence, which then is repeated periodically. The control laws are then for each cyclic perturbation of the periodic schedule obtained using Theorem 5.3.1. Thus, the RR control laws are also implemented using MPC, but the scheduling is fixed. Figure 5.6 depicts the evolution of the state and control inputs for such a RR algorithm that features the predetermined scheduling sequence

$$\mathcal{S}_3 = \left(\begin{array}{c} \begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ p_{\text{CFP}} \\ p_{\text{CAP}} \end{bmatrix} \\ \begin{bmatrix} p_{\text{CAP}} \\ 0 \\ p_{\text{CFP}} \end{bmatrix} \end{array} \right). \quad (5.40)$$

The control laws are obtained by Theorem 5.3.1 for a horizon length $N = 3$. Comparing the results from Algorithm 5.1 in Figure 5.5 to the RR in Figure 5.6, the co-design algorithm is able to achieve steady-state significantly faster than the RR algorithm.

Figure 5.5c shows the transmission slot allocation for the co-design algorithm. An interesting observation is, that following a packet drop event, the same schedule is not necessarily repeated, and often a new schedule is utilized. The reason for this is, that the controller has new data available in the form of a new state estimate. Algorithm 5.1 then finds that a different schedule and control input combination is optimal. Figure 5.5d shows the transmission slot allocation distribution during the simulation. It shows that, while the actuator that addresses the first subsystem in (5.39) receives 80% of the total attention, it receives the same amount of attention in the CFP as the actuator of the second subsystem, which features lower eigenvalues. The actuator that addresses the third subsystem in (5.39) receives the smallest amount of attention. This subsystem also features marginally stable eigenvalues. However, the third subsystem still gets 20% of the attention of the CFP.

5.4.3 Performance of small coupled systems.

In this section, the performance of the presented controller-scheduler co-design algorithm is compared through Monte Carlo simulation studies. We consider the NCS (5.39), that is used in Section 5.4.2 where the actuators compete for one slot in the CFP and one slot in the CAP. The disturbance $\omega(k) \sim \mathcal{N}(\mathbf{0}_5, I_5)$. The system is simulated with both a set-to-zero and a hold-input strategy at the actuators. The hold-input strategy is described in Section 2.3.1. The probabilities for successful transmissions are $p_{\text{CFP}} = 0.95$, $p_{\text{CAP}} = 0.75$. The state weighting penalty is in this study chosen as $R = I_3$, while the remaining parameters are as in Section 5.4.2.

The performance is analyzed by the empirical cost averaged over time, which for each simulation is given by

$$\text{cost} = \frac{1}{M} \left[\sum_{k=0}^{M-1} \|x(k)\|_Q^2 + \|u(k)\|_{R(\gamma(k))}^2 \right], \quad (5.41)$$

where Q and $R(\gamma(k))$ are the weighting matrices of the state and control signal, respectively, and M is the length of the simulation. This is averaged over 100 simulations, each with $M = 20\,000$ time-steps of the same system with different initial conditions and noise realizations.

The MPC co-design algorithm is compared to RR heuristics with different horizon lengths. We select the horizon length to be equal to the scheduling length. The RR schedules are chosen as (5.40),

$$\mathcal{S}_2 = \left(\begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} p_{\text{CFP}} \\ 0 \\ p_{\text{CAP}} \end{bmatrix} \right) \quad \mathcal{S}_4 = \left(\begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} p_{\text{CAP}} \\ 0 \\ p_{\text{CFP}} \end{bmatrix} \begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ p_{\text{CFP}} \\ p_{\text{CAP}} \end{bmatrix} \right),$$

$$\mathcal{S}_5 = \left(\begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} p_{\text{CAP}} \\ 0 \\ p_{\text{CFP}} \end{bmatrix} \begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} p_{\text{CAP}} \\ p_{\text{CFP}} \\ 0 \end{bmatrix} \begin{bmatrix} p_{\text{CAP}} \\ 0 \\ p_{\text{CFP}} \end{bmatrix} \right).$$

These RR scheduling sequences are selected intuitively. For the current system, it is relatively straightforward to select a feasible RR scheduling sequence. However, for larger systems, it can become a difficult task to design a reasonably performing RR scheduling sequence. Also, it is worth noting that there might be scheduling sequences for the RR algorithm that result in a better performance than illustrated. Also, it is not possible to utilize a length 1 RR scheduling sequence for the system (5.39), since there exists no length 1 scheduling sequence that results in a controllable system. Refer to Section 6.1.4 for more details concerning controllability and scheduling.

The results of the simulations are shown in Figure 5.7a, where results for both a set-to-zero and a hold-input policy at the actuators are illustrated. In both situations the simulations show, that the online scheduler and controller co-design algorithm, which is described in Algorithm 5.1 provides significantly better performance than the offline RR design. Also, the performance of the algorithms increases as the optimization horizon N increases. The biggest gains are however in going from $N = 1$ to $N = 2$.

To further illustrate the importance of online scheduling when the system, besides Gaussian noise, is affected by random and unmeasured additive disturbances, we add the disturbance $\chi(k) \in$

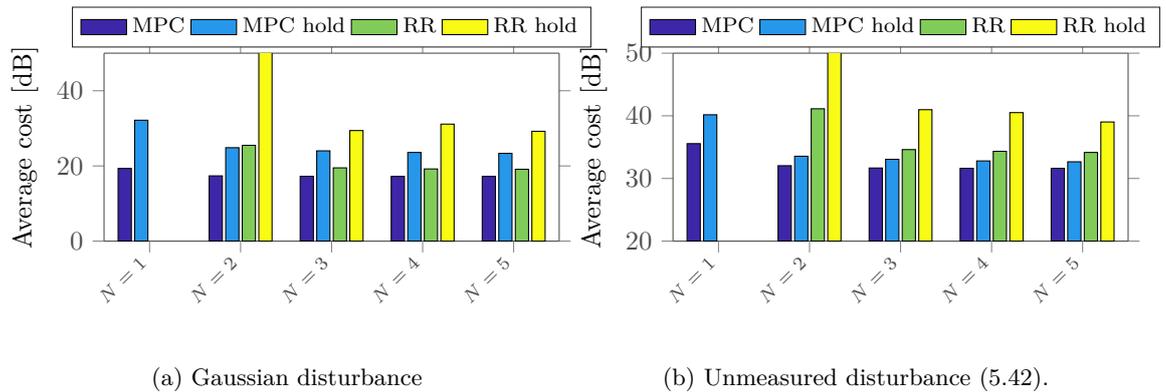


Figure 5.7: The empirical cost of the finite horizon controller and scheduler co-design implemented using Algorithm 5.1. The RR schedules are added for comparison. In Figure 5.7a the RR with hold-input for $N = 2$, attained a cost of 192 dB. In Figure 5.7b the RR with hold-input for $N = 2$ attained a cost of 84 dB.

Horizon length (N)	card (Ξ)	average set-to-zero	average hold
1	6	6	6
2	36	36	36
3	216	215	216
4	1296	1270	1296
5	7776	7414	7768

Table 5.1: Maximum number of scheduling sequences to compare for computing (5.32) calculated using (5.35). The next two columns show the average amount of scheduling sequences to check when using the branch and bound method presented in Algorithm 5.3.

\mathbb{R}^m to (5.1). Every element in the disturbance vector $\chi(k)$ is given by

$$\chi_i(k) = \rho_i(k)v_i(k) + \iota_i(k)\chi_i(k-1), \quad (5.42)$$

where $\Pr\{\rho_i(k) = 1\} = 0.05$, $\Pr\{\iota_i(k) = 1\} = 0.85$, and the step magnitude $v_i(k) \sim \mathcal{U}(-10, 10)$ is uniformly distributed with $\chi_i(-1) = 0$. The noise generated by (5.42) is a random step function with a random magnitude $v_i(k)$.

The results for systems that are affected by the disturbance Figure 5.7b. Both set-to-zero and hold-input actuator policies are used. The online MPC scheduling and controller co-design algorithms outperform the RR heuristic by a significant margin when the horizon length $N > 1$. The performance of the MPC from Algorithm 5.1 with horizon length $N = 1$ is comparable to the performance of the best performance of the RR algorithms. Recall, that the RR heuristic does not feature online scheduling, thus increasing the horizon length does in this case not affect the online computational requirements. It is worth noting, that the performance of the RR heuristic varies significantly depending on which scheduling sequence is utilized. This is avoided completely when Algorithm 5.1 is utilized, in which case the performance improves for longer horizons.

Table 5.1 shows the number of possible scheduling sequences for the different horizon lengths. Further, the efficiency of the branch and bound algorithm, presented in Algorithm 5.3, is presented as well. It here shows, that for the current system, there are no significant reductions in the

computational requirements when the branch and bound method in Algorithm 5.3 is used.

5.4.4 Larger system

Next, we consider a larger system that contains three subsystems and four inputs. The model is given by

$$A = \begin{bmatrix} 0.681 & 0.319 & 0.888 & 0.112 & 0 & 0 & 0 & 0 \\ 0.319 & 0.681 & 0.112 & 0.888 & 0 & 0 & 0 & 0 \\ -0.559 & 0.559 & 0.681 & 0.319 & 0 & 0 & 0 & 0 \\ 0.559 & -0.559 & 0.319 & 0.681 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.72 & 0.72 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.889 & -2.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.54 & 0.841 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.841 & 0.54 \end{bmatrix} \quad (5.43)$$

and

$$B = \begin{bmatrix} 0.471 & 0 & 0 & 0 \\ 0.0286 & 0 & 0 & 0 \\ 0.888 & 0 & 0 & 0 \\ 0.112 & 0 & 0 & 0 \\ 0 & 5.09 & 0 & 0 \\ 0 & 0 & 6.29 & 0 \\ 0 & 0 & 0 & 0.46 \\ 0 & 0 & 0 & 0.841 \end{bmatrix}, \quad (5.44)$$

with $\omega(k) \sim \mathcal{N}(\mathbf{0}_8, I_8)$. This system has eigenvalues at $0.36 \pm -0.93j$, two at 1, -2.73 , -0.40 and $0.54 \pm 0.84j$. The scheduler has access to one slot in the CFP and one in the CAP, and the remaining parameters are as in Section 5.4.3. For this system, the RR scheduling sequences are for $N = 1, \dots, 5$ selected as the scheduling sequence, that satisfies

$$\mathcal{S}_N^{RR} \triangleq \arg \min_{\mathcal{S}_N \in \Xi_N} J_W(\mathbf{0}, \mathcal{S}_N, U_N^*(\mathcal{S}_N)).$$

Figure 5.8a shows that for system (5.43), the MPC co-design results in 15 dB lower cost than the offline RR heuristic in the best case. In this setting, the performance of the hold-input strategy is slightly reduced compared to the set-to-zero strategy. However when the horizon $N = 1$, the MPC with hold-input results in a cost of 557 dB, which is significantly higher than for larger N . When the system is affected by the disturbance (5.42), the online MPC co-design outperforms the RR heuristic by more than 10 dB. Interestingly, in this setting, the hold-input performs on par with set-to-zero when $N > 1$.

The number of scheduling sequences to check in the branch and bound algorithm are displayed in Table 5.2. As with the results in Table 5.1 for system (5.39), the branch and bound algorithm in Algorithm 5.3 results in only a minor reduction in the number of scheduling sequences to compare.

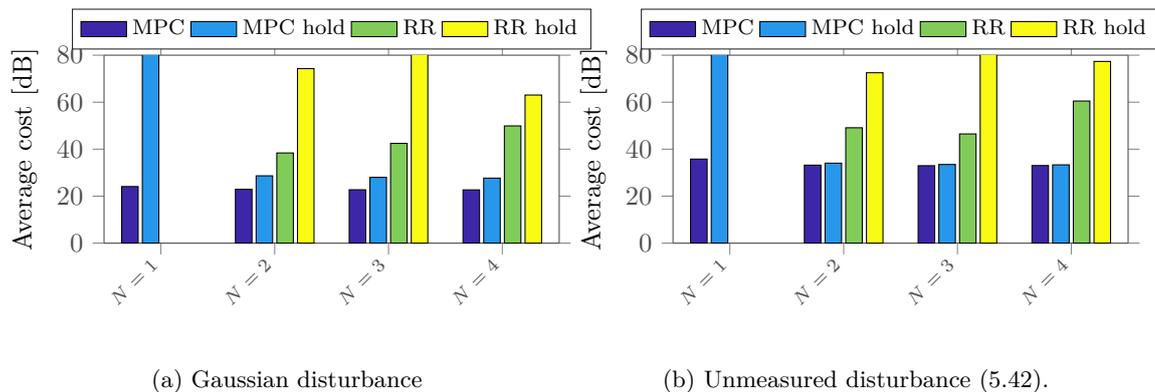


Figure 5.8: The empirical cost of the finite horizon controller and scheduler co-design implemented using Algorithm 5.1. The RR schedules are added for comparison. In Figure 5.8a, the RR with hold-input for $N = 3$, attained a cost of 2230 dB, while MPC with hold-input and $N = 1$ reached 557 dB. In Figure 5.8b the RR with hold-input for $N = 3$ attained a cost of 2233 dB and MPC with hold-input and $N = 1$ reached 508 dB.

Horizon length (N)	card (Ξ)	average set-to-zero	average hold
1	12	12	12
2	144	144	144
3	1728	1723	1728
4	20 736	20 125	20 685

Table 5.2: Maximum number of scheduling sequences to compare for computing (5.32) for system (5.43) and (5.44) calculated using (5.35). The next two columns show the average amount of scheduling sequences to check when using the branch and bound method presented in Algorithm 5.3.

5.5 Summary

We have discussed and illustrated the advantages of online scheduling and controller co-design for systems where the controller and actuators are connected over IEEE 802.15.4 based wireless networks. We considered multiple systems that share a finite amount of transmission bandwidth. The transmission bandwidth is divided in a reliable CFP and a less reliable CAP. While transmission in the CFP is preferred, the limit in the amount of transmission slots in the CFP motivates the use of the less reliable CAP. We demonstrated that optimal scheduling and controller co-design takes advantage of the network bandwidth that is available. This results in a significantly improved performance compared to heuristic round robin (RR) schedules.

Simulation studies showed, that a longer prediction horizon length resulted in improved control performance. However, the computational complexity increases significantly. This limits the practical applicability of the presented co-scheduling algorithm on large scale implementations. In the next chapter, we extend these ideas and aim at providing a suitable choice for the final state weighting of the cost function (5.25), such that good performance can be maintained at a reduced online computational cost.

Appendix

5.A Proof of Lemma 5.2.1

Proof. By the definition of the matrix inverse it follows that for every T , the inverse $(M + T)^{-1}$ is unique. To show that M^\dagger is unique, we have to show that \bar{T} is unique. To do this, assume there exists a $Q \neq \bar{T}$ that satisfies (5.3) to (5.9). It thus satisfies

$$QM = MQ = M.$$

This means by (5.8) that

$$\begin{aligned} QM &= \bar{T}M = QM + QT = \bar{T}M + \bar{T}T \\ &\Rightarrow Q(M + T) = \bar{T}(M + T) \\ &\Rightarrow Q(M + T)(M + T)^{-1} = \bar{T} \\ &\Rightarrow Q = \bar{T}, \end{aligned}$$

which contradicts the assumption that $Q \neq \bar{T}$.

Any non-full rank block diagonal matrix can be permuted using matrix P to have the form

$$\bar{M} = PMP^T = \begin{bmatrix} M_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

This means that T has to be on the form

$$T = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T_1 \end{bmatrix}.$$

Now for block diagonal matrix \bar{M} one \bar{T} that satisfies the conditions in Definition 5.2.1 is the matrix

$$\bar{T} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Recall that \bar{T} is unique. Now by the definition of inverses of partitioned matrices, see for example [107, Lemma A.2] we have that

$$(M + T)^{-1} = \begin{bmatrix} M_i^{-1} & \mathbf{0} \\ \mathbf{0} & T_1^{-1} \end{bmatrix}$$

Pre- and post-multiplying this with \bar{T} results in

$$\bar{T}(M + T)^{-1}\bar{T} = \begin{bmatrix} M_i^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where M_i^{-1} , which is unique by the definition of the matrix inverse. This proves uniqueness of the inverse.

Condition (5.10) follows since

$$MM^\dagger M = M\bar{T}(M + T)^{-1}\bar{T}M$$

where from Definition 5.2.1 $M = \bar{T}M\bar{T} + \bar{T}T\bar{T} = \bar{T}(M+T)\bar{T}$, thus using symmetry and the definitions in (5.3) to (5.9), we have that

$$\begin{aligned}
M\bar{T}(M+T)^{-1}\bar{T}M &= \bar{T}(M+T)\bar{T}\bar{T}(M+T)^{-1}\bar{T}M \\
&= \bar{T}(M\bar{T}+T\bar{T})(M+T)^{-1}\bar{T}M \\
&= \bar{T}(\bar{T}M+\bar{T}T)(M+T)^{-1}M \\
&= \bar{T}(M+T)(M+T)^{-1}M \\
&= \bar{T}M = M.
\end{aligned}$$

Thus proving (5.10). Conditions (5.11) to (5.13) can be shown using a similar argument. \square

5.B Proof of Theorem 5.3.1

This proof follows by applying the dynamic programming algorithm [15]. Consider the cost (5.23) for a horizon length N and a scheduling sequence $\mathcal{S}_N \in \mathbb{S}$. Then, the cost (5.23) can, at any stage $0 \leq k < N$, be stated as

$$\begin{aligned}
&J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) \\
&= \min_{u(k)} \mathbf{E} \left\{ \|x(k)\|_Q^2 + \|u(k)\|_{R(\gamma(k))}^2 + \min_{U_{k+1,N}} \left[\sum_{\ell=k+1}^N \|x(\ell)\|_Q^2 + \|u(\ell)\|_{R(\gamma(\ell))}^2 \right] \middle| x(k); \mathcal{S}_{k,N} \right\} \\
&= \min_{u(k)} \mathbf{E} \left\{ \|x(k)\|_Q^2 + \|u(k)\|_{R(\gamma(k))}^2 + J_W(x(k+1), \mathcal{S}_{k+1,N}, U_{k+1,N}^*) \middle| x(k); \mathcal{S}_{k,N} \right\}. \quad (5.B.1)
\end{aligned}$$

At stage N , the cost for $x(N)$ and schedule $S(N)$ is given by

$$J_W(x(N), S(N)) = x^T(N)Wx(N).$$

Now assume that at stage $k+1$, the optimal cost for a known scheduling sequence $\mathcal{S}_{k+1,N}$ is of the form

$$J_W(x(k+1), \mathcal{S}_{k+1,N}, U_{k+1,N}^*) = x^T(k+1)P_{k+1}x(k+1) + c_{k+1}.$$

Then, at stage k , we have from (5.B.1) and (5.1)

$$\begin{aligned}
J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) &= \min_{u(k)} \mathbf{E} \left\{ \|x(k)\|_Q^2 + \|u(k)\|_{R(\gamma(k))}^2 \right. \\
&\quad \left. + J_W(Ax(k) + B(\gamma(k))u(k) + F\omega(k), \mathcal{S}_{k+1,N}, U_{k+1,N}^*) \middle| x(k); \mathcal{S}_{k,N} \right\}.
\end{aligned}$$

Expanding this yields

$$\begin{aligned}
J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) &= \min_{u(k)} \mathbf{E} \left\{ x^T(k) (Q + A^T P_{k+1} A) x(k) \right. \\
&\quad + u^T(k) \left(R(\gamma(k)) + B^T(\gamma(k)) P_{k+1} B(\gamma(k)) \right) u(k) + 2u^T(k) B^T(\gamma(k)) P_{k+1} A x(k) \\
&\quad \left. + \omega^T(k) F^T P_{k+1} F \omega(k) + c_{k+1} \middle| x(k); \mathcal{S}_{k,N} \right\},
\end{aligned}$$

where we used the fact that $\omega(k)$ is *i.i.d* and uncorrelated to $x(k)$ and $u(k)$. Distributing the expectation yields

$$\begin{aligned} J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) &= x^T(k) (Q + A^T P_{k+1} A) x(k) \\ &+ \min_{u(k)} u^T(k) [\mathbf{E}\{R(\gamma(k)); S(k)\} + \mathbf{E}\{B^T(\gamma(k))P(k+1)B(\gamma(k)); S(k)\}] u(k) \\ &+ 2u^T(k) \mathbf{E}\{B^T(\gamma(k)); S(k)\} P_{k+1} A x(k) + \text{trace}(\Sigma_\omega F^T P_{k+1} F) + c_{k+1}, \end{aligned} \quad (5.B.2)$$

where Σ_ω is the covariance of $\omega(k)$. In (5.B.2), the term

$$\begin{aligned} \mathbf{E}\{B(\gamma(k)); S(k)\} &= \sum_{s \in \mathbb{Z}_2^n} \mathbf{E}\{B(\gamma(k)) | \gamma(k) = s\} \Pr\{\gamma(k) = s; S(k)\} \\ &= \sum_{s \in \mathbb{Z}_2^n} B(s) \Pr\{\gamma(k) = s; S(k)\} \\ &= B(S(k)), \end{aligned}$$

and using the same arguments,

$$\mathbf{E}\{R(\gamma(k)); S(k)\} = R(S(k)).$$

Also

$$\begin{aligned} &\mathbf{E}\{B^T(\gamma(k)) P_{k+1} B(\gamma(k)); S(k)\} \\ &= \sum_{s \in \mathbb{Z}_2^n} \mathbf{E}\{B^T(\gamma(k)) P_{k+1} B(\gamma(k)) | \gamma(k) = s\} \Pr\{\gamma(k) = s; S(k)\} \\ &= \sum_{s \in \mathbb{Z}_2^n} B^T(s) P_{k+1} B(s) \Pr\{\gamma(k) = s; S(k)\} \\ &= M(P_{k+1}, S(k)), \end{aligned}$$

where $M(P_{k+1}, S(k))$ is defined in (5.28). Substituting this back into (5.B.2) yields

$$\begin{aligned} J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) &= x^T(k) (Q + A^T P_{k+1} A) x(k) \\ &+ \min_{u(k)} u^T(k) [R(S(k)) + M(P_{k+1}, S(k))] u(k) \\ &+ 2u^T(k) B^T(S(k)) P_{k+1} A x(k) + \text{trace}(\Sigma_\omega F^T P_{k+1} F) + c_{k+1}. \end{aligned} \quad (5.B.3)$$

Taking the gradient with respect to $u(k)$ and setting this equal to 0 yields the optimal control

$$\begin{aligned} 0 &= [R(S(k)) + M(P_{k+1}, S(k))] u(k) + B^T(S(k)) P_{k+1} A x(k) \\ u^*(k) &= -[R(S(k)) + M(P_{k+1}, S(k))]^\dagger B^T(S(k)) P_{k+1} A x(k). \end{aligned} \quad (5.B.4)$$

This leads to (5.26) and (5.27). The generalized inverse is used since the term

$$R(S(k)) + M(P_{k+1}, S(k))$$

can be singular. Substituting this back into (5.B.3) results in

$$\begin{aligned} J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) &= x^T(k) \left(Q + A^T P_{k+1} A - A^T P_{k+1} B(S(k)) \right. \\ &\quad \left. \times [R(S(k)) + M(P_{k+1}, S(k))]^\dagger B^T(S(k)) P_{k+1} A \right) x(k) + \text{trace}(\Sigma_\omega F^T P_{k+1} F) + c_{k+1}. \end{aligned}$$

This equals

$$J_W(x(k), \mathcal{S}_{k,N}, U_{k,N}^*) = x^T(k) P_k x(k) + c_k, \quad (5.B.5)$$

where P_k is given by (5.29), with $P_N = W$ and

$$\begin{aligned} c_k &= \text{trace}(\Sigma_\omega F^T P_{k+1} F) + c_{k+1} \\ &= \sum_{\ell=k}^{N-1} \text{trace}(\Sigma_\omega F^T P_{\ell+1} F). \end{aligned} \quad (5.B.6)$$

Substituting (5.B.6) into (5.B.5) and repeating the above until step 0 then leads to (5.30).

A periodic approach to scheduling and control co-design

In this chapter, we extend the controller and scheduler co-design solution that is introduced in Chapter 5. Mainly, we present methods that approximate the cost function (5.23) as the prediction horizon goes to infinity. This is done by using a periodic assumption on the scheduling sequence. While we present implementations that directly utilize these infinite horizon approximations, we also present a method that combines the model predictive control (MPC) algorithm from Section 5.3.3 with the periodic infinite horizon approximations. We show a number of stability results and illustrate the performance gains compared to the algorithm that is presented in Chapter 5.

Section 6.1 provides the motivation behind the ideas that are proposed in this chapter. This is followed by a brief summary of the networked control system (NCS) that is considered in Chapter 5. We also provide a linear time-invariant (LTI) formulation of the system, that is utilized throughout the remainder of this chapter. This LTI formulation allows for cleaner and easier expressions. Section 6.2 revises the results from Section 5.3 using the LTI formulation. Section 6.3 presents two methods to approximate the infinite horizon cost function. Implementations of the co-design algorithms are presented in Section 6.4. This includes online MPC algorithms and offline scheduling algorithms. Simulation studies in Section 6.5 illustrate the performance of the proposed designs. Section 6.6 provides a summary of the results.

6.1 Introduction

In Chapter 5, we presented an optimal co-design method for the controller and scheduler for a large, distributed system. The controller and system are connected through a bandwidth limited IEEE 802.15.4 network that is affected by packet loss. The optimal co-design method in Section 5.3 is implemented as MPC. Recall, that the cost-function (6.21) considered in Section 5.3 has a finite horizon. Simulation studies in Section 5.4 showed that longer horizon lengths resulted in increased performance. Ideally, one would design the controller to minimize the cost function over an infinite length horizon. However, as discussed in Sections 5.2 and 5.3, the scheduling part of the co-design problem is combinatorial. Further, the complexity of the problem grows exponentially with the horizon length. This makes it difficult, if not impossible, to evaluate the infinite horizon cost

function. This is commonly referred to as the *curse of dimensionality* [15, Chapter 6].

One method to reduce the complexity of the scheduling problem is, to assume that the scheduling sequence eventually converges to some form of periodic sequence. However, the period of that sequence can be very long. It is therefore difficult to find the optimal period. Instead, we can assume a period length, and compute the infinite horizon cost function for all possible scheduling sequences that have this period length. However, the periodic scheduling sequence that minimizes the infinite horizon cost function, might not be optimal.

To address this, we will based on ideas from MPC, combine the finite horizon co-design approach presented in Section 5.3 with a suitable selection of the final state weighting W in (5.23). In conventional MPC, the final state weighting is frequently designed using a solution to a Riccati equation [17], [40], [62]. However, the system (5.20) presented in Section 5.2, is time-varying due to the scheduling. Also, the Riccati recursion (5.31) is time-varying due to the scheduling. It is therefore non-trivial to find a suitable Riccati equation to use as the final state weighting.

Based on the above discussion and inspired by the work [40], we will assume that after some time, the scheduling sequence will converge to a periodic scheduling sequence with a fixed period. We utilize these periodic scheduling sequences to approximate infinite horizon cost functions. The idea is then, that the finite horizon cost function eventually will converge to this infinite horizon cost approximation. This is implemented by using the solutions to the infinite horizon cost approximations to design the final state weighting W in the finite horizon cost function (5.23). This method is also known as the rollout algorithm, see for example [6], [15]. The results are implemented in a MPC algorithm that provides good performance for short prediction horizons. This results in a significant reduction in the computational requirements compared to Algorithm 5.1 which is presented in Section 5.3.3.

To simplify notation, we first present a LTI formulation of the time-varying system (5.20). After revising the results from Section 5.3 for the LTI system, we present two methods to approximate the infinite horizon cost function by using periodic scheduling sequences. We present both online and offline scheduling co-designs that utilize the solutions to these infinite horizon approximations.

6.1.1 System setup

In this chapter, we consider the system (5.20) from Chapter 5, which is restated here

$$\begin{aligned} x(k+1) &= Ax(k) + B(\gamma(k))u(k) + F\omega(k) \\ y(k) &= Cx(k) + v(k). \end{aligned} \tag{6.1a}$$

In (6.1), $x(k) \in \mathbb{R}^m$, $u(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$ and A , B , F , C are matrices of appropriate dimensions. The plant disturbance $\omega(k) \sim \mathcal{N}(\mathbf{0}_d, \Sigma_\omega)$ is zero-mean Gaussian with covariance Σ_ω and measurement disturbance $v(k) \sim \mathcal{N}(\mathbf{0}_p, \Sigma_v)$. The matrix $B(\gamma(k))$ is, as in (5.19), defined by

$$B(\gamma(k)) \triangleq B \text{diag} \{ \gamma(k) \}, \tag{6.2}$$

where the pair (A, B) is controllable and the density of $\gamma(k)$ depends on the schedule $S(k)$ i.e.

$$\Pr \{ \gamma_j(k) = 1; s_j \} = s_j(k) \quad j \in (1, 2, \dots, n).$$

In the above, $\Pr \{ a; b \}$ means that the distribution of a is parameterized by b .

The schedule at time k is given by the length n vector

$$S(k) \triangleq [s_1(k), \dots, s_n(k)]^T \in \mathbb{S}, \quad (6.3)$$

where, as in (5.17),

$$\mathbb{S} \triangleq \left\{ S = \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \middle| s_i \in \{0, p_{\text{CAP}}, p_{\text{CFP}}\} \right\} \quad (6.4)$$

and $0 \leq p_{\text{CAP}} \leq p_{\text{CFP}} \leq 1$. Denote the set of admissible schedules by $\mathbb{S}_a \subset \mathbb{S}$. An admissible schedule satisfies that N_{CAP} elements are p_{CAP} and N_{CFP} elements are p_{CFP} , while the remaining elements are zero. Recall that N_{CAP} and N_{CFP} are the number of slots that the scheduler can allocate in the contention access period (CAP) and contention free period (CFP), respectively. For more details, refer to Section 5.2.1. The set \mathbb{S}_a is formally stated in (5.18), and is repeated below

$$\mathbb{S}_a \triangleq \left\{ S \in \mathbb{S} \middle| \sum_{i=1}^n \mathbf{1}_{p_{\text{CAP}}}(s_i) = N_{\text{CAP}} \wedge \sum_{j=1}^n \mathbf{1}_{p_{\text{CFP}}}(s_j) = N_{\text{CFP}} \right\},$$

where the indicator function $\mathbf{1}_a(b) = 1$ if $a = b$ and zero otherwise.

In this chapter, we want to co-design an optimal controller and scheduler, such that the selected controller and scheduler minimize a given cost function. We will first present a time-invariant formulation of the above problem, and introduce scheduling sequences that are repeated periodically. Using this LTI formulation, we present a time-invariant revision of the result that is presented in Section 5.3.

The state estimator is designed as in Section 5.2.2 with the optimal state estimate given by (5.14). All measurements are transmitted during the CAP, and the superframe structure from Figure 5.3 is utilized. The separation principle also holds in the current setting. Therefore, the controller design is done using state feedback. In the simulation studies in Section 6.5, the state feedback is replaced by the optimal state prediction.

6.1.2 Periodic scheduling

Denote the periodic scheduling sequence with period T by

$$\mathcal{S} \triangleq (S(\ell) \in \mathbb{S}_a : \ell = 0, \dots, T-1).$$

This means that $S(T) = S(0)$, $S(T+1) = S(1)$ and so on.

Denote a length NT , with N being possibly infinity, scheduling sequence that contains the periodic scheduling sequence \mathcal{S} repeatedly by

$$\mathcal{S}_N \triangleq (\mathcal{S} : \ell = 1, \dots, N).$$

Define the length NT scheduling sequence \mathcal{S}_N with the first k schedules trimmed of by

$$\mathcal{S}_{k,N} \triangleq \left(S(k \% T), \dots, S(T-1), \mathcal{S}_{N - \lceil \frac{k}{T} \rceil} \right), \quad (6.5)$$

where T is the period length, $\lceil x \rceil$ rounds x up to its nearest larger integer and $\%$ is the modulo operator, such that $a\%b$ returns the remainder after division of a/b . The first part in (6.5) contains the schedules $(S(k\%T), S(k+1\%T), \dots, S(T-1))$.

We will now present a cyclic formulation of the system (6.1a) that uses a periodic scheduling sequence (6.5).

6.1.3 A time-invariant formulation

The system in (6.1a) is time-varying since the schedule $S(k)$ can change at every time-step. Further, (6.1a) is stochastic due to the noise $\omega(k)$ and the packet loss $\gamma(k)$. The distribution of the packet loss $\gamma(k)$ is parameterized by the schedule $S(k)$. Thus, the expected value of the input matrix $B(\gamma(k))$ will change over time, i.e. $\mathbf{Pr}\{\gamma(k) = 1; S(k)\} = S(k)$. However, when a length T scheduling sequence \mathcal{S} is periodic, $\mathbf{E}\{B(\gamma(k)); \mathcal{S}\} = \mathbf{E}\{B(\gamma(k+T)); \mathcal{S}\}$, where

$$\mathbf{E}\{A; \mathcal{S}\} \triangleq \int A \mathbf{Pr}\{A; \mathcal{S}\} dA.$$

In this section, we reformulate system (6.1a) to be stochastic, but time-invariant. Using this, we obtain an augmented input matrix $\bar{B}(\bar{\gamma}(k))$, such that for all k ,

$$\mathbf{E}\{\bar{B}(\bar{\gamma}(k)); \mathcal{S}\} = \mathbf{E}\{\bar{B}(\bar{\gamma}(k+1)); \mathcal{S}\}.$$

We follow the cyclic formulation principles presented in [16, Section 6.3].

Define the augmented state vector

$$\bar{x}(k) \triangleq \begin{bmatrix} \bar{x}_0(k) \\ \bar{x}_1(k) \\ \vdots \\ \bar{x}_{T-1}(k) \end{bmatrix} \in \mathbb{R}^{mT}, \quad (6.6)$$

such that, as in Table 6.1, the state $x(k)$ at each time k is given by

$$\bar{x}_i(k) = \begin{cases} x(k) & \text{if } k\%T = i \\ 0 & \text{otherwise.} \end{cases} \quad (6.7)$$

This means, that all elements of the augmented state $\bar{x}(k)$ are 0, except for $\bar{x}_i(k)$. Define the augmented system matrices

$$\bar{A} \triangleq \begin{bmatrix} 0 & 0 & \cdots & 0 & A \\ A & 0 & \cdots & 0 & 0 \\ 0 & A & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & A & 0 \end{bmatrix} \in \mathbb{R}^{mT \times mT}, \quad \bar{F} \triangleq \begin{bmatrix} 0 & 0 & \cdots & 0 & F \\ F & 0 & \cdots & \cdots & 0 \\ 0 & F & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & F & 0 \end{bmatrix} \in \mathbb{R}^{mT \times dT}$$

and

$$\bar{B}(\bar{\gamma}) \triangleq \begin{bmatrix} 0 & 0 & \cdots & 0 & B(\bar{\gamma}_{T-1}) \\ B(\bar{\gamma}_0) & 0 & \cdots & 0 & 0 \\ 0 & B(\bar{\gamma}_1) & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & B(\bar{\gamma}_{T-2}) & 0 \end{bmatrix} \in \mathbb{R}^{mT \times nT}. \quad (6.8)$$

Time-varying \rightarrow LTI	LTI \rightarrow Time-varying
$\bar{x}_i(k) = \begin{cases} x(k) & \text{if } i = k \% T \\ 0 & \text{otherwise} \end{cases}$	$x(k) = \bar{x}_i(k) \quad \text{if } k \% T = i$
$\bar{u}_i(k) = \begin{cases} u(k) & \text{if } i = k \% T \\ 0 & \text{otherwise} \end{cases}$	$u(k) = \bar{u}_i(k) \quad \text{if } k \% T = i$
$\bar{\gamma}_i(k) = \begin{cases} \gamma(k) & \text{if } i = k \% T \\ 0 & \text{otherwise} \end{cases}$	$\gamma(k) = \bar{\gamma}_i(k) \quad \text{if } k \% T = i$
$\bar{\omega}_i(k) = \begin{cases} \omega(k) & \text{if } i = k \% T \\ 0 & \text{otherwise} \end{cases}$	$\omega(k) = \bar{\omega}_i(k) \quad \text{if } k \% T = i$

Table 6.1: Map between the time-varying and LTI formulation of the system (6.1a). Here $k \% T$ returns the remainder after division of k/T .

Define the vectors

$$\bar{u}(k) \triangleq \begin{bmatrix} \bar{u}_0^T & \bar{u}_1^T & \cdots & \bar{u}_{T-1}^T \end{bmatrix}^T \in \mathbb{R}^{nT} \quad (6.9)$$

$$\bar{\gamma}(k) \triangleq \begin{bmatrix} \bar{\gamma}_0^T & \bar{\gamma}_1^T & \cdots & \bar{\gamma}_{T-1}^T \end{bmatrix}^T \in \mathbb{R}^{nT} \quad (6.10)$$

$$\bar{\omega}(k) \triangleq \begin{bmatrix} \bar{\omega}_0^T & \bar{\omega}_1^T & \cdots & \bar{\omega}_{T-1}^T \end{bmatrix}^T \in \mathbb{R}^{dT}.$$

The mappings between the LTI and time-varying formulation are stated in Table 6.1. It is worth noting that, as with $\bar{x}(k)$, all elements in $\bar{u}(k)$, $\bar{\gamma}(k)$ and $\bar{\omega}(k)$ are zero, besides elements $\bar{u}_i(k)$, $\bar{\gamma}_i(k)$ and $\bar{\omega}_i(k)$, where $i = k \% T$.

The above formulation and the mappings in Table 6.1 allow us to restate the time-varying system (6.1a) as the LTI system

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}(\bar{\gamma}(k))\bar{u}(k) + \bar{F}\bar{\omega}(k), \quad (6.11)$$

where

$$\bar{\omega}(k) \sim \mathcal{N}(\mathbf{0}, \bar{\Sigma}_{\bar{\omega}}(k)).$$

The covariance of $\bar{\omega}$ is, using the maps in Table 6.1, given by

$$\bar{\Sigma}_{\bar{\omega}}(k) \triangleq \text{diag} \left\{ \left[\mathbf{1}_0(k \% T) \Sigma_{\omega} \quad \cdots \quad \mathbf{1}_{T-1}(k \% T) \Sigma_{\omega} \right] \right\} \in \mathbb{R}^{dT \times dT}, \quad (6.12)$$

where the indicator function $\mathbf{1}_i(k) = 1$ if $k = i$ and zero otherwise. Further, define

$$\bar{\Sigma}_{\bar{\omega}} \triangleq \text{diag} \left\{ \left[\Sigma_{\omega} \quad \cdots \quad \Sigma_{\omega} \right] \right\} \in \mathbb{R}^{dT \times dT}. \quad (6.13)$$

Remark 6.1.1. Note that in the definitions in the first column of Table 6.1, the LTI formulation $\bar{x}_j(k) = 0$, $j \neq k \% T$ is merely introduced to ensure that the LTI vector $\bar{x}(k)$ is well defined. However, it is not necessary to force $\bar{x}_j(k) = 0$ and, in fact, not forcing this allows for the simultaneous computation of all cyclic shifts in a periodic schedule. This also holds true for \bar{u} , $\bar{\gamma}$ and $\bar{\omega}$.

For the periodic scheduling sequence \mathcal{S} with period T , define the augmented schedule as

$$\bar{\mathcal{S}} \triangleq \begin{bmatrix} \mathcal{S}(0)^T & \mathcal{S}(1)^T & \cdots & \mathcal{S}(T-1)^T \end{bmatrix}^T \in \bar{\mathbb{S}}^T, \quad (6.14)$$

where the set with period T

$$\bar{\mathbb{S}}^T \triangleq \left\{ \bar{S} = \begin{bmatrix} S_1 \\ \vdots \\ S_T \end{bmatrix} \middle| S_i \in \mathbb{S} \right\},$$

and \mathbb{S} is defined in (6.4). Define $\bar{\mathbb{S}}_a^T$ as the set of admissible schedules with period T as

$$\bar{\mathbb{S}}_a^T \triangleq \{ \bar{S} \in \bar{\mathbb{S}}^T \mid S_i \in \mathbb{S}_a \}$$

where \mathbb{S}_a is defined in (6.3). Note that (6.14) does not depend on the time k . Thus, we have for every entry of $\bar{\gamma}(k)$ that

$$\Pr \{ \gamma_{in+j}(k) = 1; \bar{S} \} = s_{in+j} \quad \begin{array}{l} i = 0, \dots, T-1 \\ j = 0, \dots, n-1. \end{array} \quad (6.15)$$

In the above, s_{in+j} is the j 'th element of the schedule $S(i)$ in the augmented schedule \bar{S} , and the vector $\bar{\gamma}(k) \in \mathbb{Z}_2^{nT}$, where the binary set $\mathbb{Z}_2 \triangleq \{0, 1\}$. It follows directly from (6.15) that

$$\mathbf{E} \{ \bar{\gamma}(k); \bar{S} \} = \bar{S}.$$

Consider a control law

$$\bar{u}(k) = -\bar{L}(k) \bar{x}(k), \quad (6.16)$$

where

$$\bar{L}(k) \triangleq \text{diag} \left\{ \left[\bar{L}_0(k) \quad \dots \quad \bar{L}_{T-1}(k) \right] \right\} \in \mathbb{R}^{nT \times mT}. \quad (6.17)$$

We can then state the closed-loop response of (6.11) with control law (6.16) as

$$\bar{x}(k+1) = \bar{H}_{\bar{L}(k)}(\bar{\gamma}(k)) \bar{x}(k) + \bar{F}\bar{\omega}, \quad (6.18)$$

with

$$\bar{H}_{\bar{L}}(\bar{\gamma}) \triangleq \bar{A} - \bar{B}(\bar{\gamma}) \bar{L}. \quad (6.19)$$

For easier notation, in the remainder of this chapter we omit the time indexing in $\bar{\gamma}(k)$ when not needed.

Also, by the maps in Table 6.1 one can easily obtain the time-varying control law L_k for the system (6.1a) from the block diagonal of the LTI control law (6.16). The time-varying control laws are then given by

$$u(k) = -\bar{L}_{k\%T}(k) x(k).$$

Finally, we define the selector vector e_i , where $i = 0, \dots, T-1$, as the length mT vector that contains ones in the index range im to $(i+1)m-1$, with m being the length of $x(k)$, and zeros elsewhere. Then, define the selector matrix

$$\bar{e}_i \triangleq \text{diag} \{ e_i \}. \quad (6.20)$$

This means that, for example, $\bar{e}_{k\%T} \bar{x}(k)$ ensures that only the entries that correspond to $x(k)$ in $\bar{x}(k)$ are non-zero. These definitions will be used in some of the results presented later in this chapter.

Remark 6.1.2. *Another common used method to reformulate periodic (or cyclostationary) systems into time-invariant systems is lifting. See for example [16, Section 6.2]. However, in the current setting, the lifted formulation will not result in control laws of the desired diagonal form. We, for this reason, resort to the cyclic formulation that is presented above.*

6.1.4 A few notes on controllability

In the system (6.11), the input matrix $\bar{B}(\bar{\gamma})$ depends on the schedule \bar{S} . This can affect the controllability and stabilizability of system (6.11) for particular schedules. Therefore, the following notion on stabilizable schedules is introduced.

Definition 6.1.1 (Stochastic stabilizable schedule). *A schedule $\bar{S} \in \bar{\mathcal{S}}^T$ is called a stochastic stabilizable schedule for system (6.11) if there exists control laws on the feedback form $\bar{u}(k) = -\bar{L}\bar{x}(k)$, $k = (0, 1, \dots)$, such that*

$$\lim_{k \rightarrow \infty} \mathbf{E} \left\{ \|\bar{x}(k)\|^2 ; \bar{S} \right\} < \infty.$$

It is however not straight forward to verify whether Definition 6.1.1 is satisfied for a schedule \bar{S} . In fact, we are not familiar with suitable testable conditions to check stochastic stabilizability for the system (6.11) that does not require the simultaneous design of control laws. However, recall from Section 2.3, that stochastic stabilizability requires the pair (A, B) to be stabilizable in a deterministic sense. A particular schedule $\bar{S} \in \bar{\mathcal{S}}^T$ does not necessarily address all actuators. Thus, for some elements s_i in the schedule \bar{S} , the conditional probability $\mathbf{Pr} \{ \gamma_i = 1 ; s_i \} = 0$. Therefore, by (6.2) and (6.8) some rows of the matrix $\bar{B}(\bar{\gamma})$ equal zero, i.e. $\bar{b}_j = \mathbf{0}$ for some j . Thus, deterministic stabilizability of the pair $(\bar{A}, \mathbf{E} \{ \bar{B}(\bar{\gamma}) ; \bar{S} \})$ is a necessary condition for \bar{S} to be a stochastic stabilizable schedule. This allows for an easy method to discard a number of schedules that will not lead to a stochastic stabilizable system.

An easy to check method for deterministic controllability (see Definition 2.3.1 in Chapter 2) is that the controllability matrix \mathcal{C} is full rank [4, Appendix B]. The controllability matrix for the pair $(\bar{A}, \mathbf{E} \{ \bar{B}(\bar{\gamma}) ; \bar{S} \})$ is defined by

$$\mathcal{C} \triangleq \left[\mathbf{E} \{ \bar{B}(\bar{\gamma}) ; \bar{S} \} \quad \bar{A} \mathbf{E} \{ \bar{B}(\bar{\gamma}) ; \bar{S} \} \quad \dots \quad \bar{A}^{mT-1} \mathbf{E} \{ \bar{B}(\bar{\gamma}) ; \bar{S} \} \right].$$

However, if \mathcal{C} is not full rank for a particular schedule \bar{S} , the schedule is stabilizable by Definition 2.3.2 if the uncontrollable subspace of \bar{A} is stable. If a schedule \bar{S} is not stabilizable in a deterministic sense, it can be discarded.

Next, we present a LTI version of the optimal control problem that we considered in Section 5.3.

6.2 Finite horizon stochastic control

Consider the system (6.11) and the cost function (5.23). Then, for a length NT scheduling sequence \mathcal{S}_N with period T we can, using the time-invariant formulation in Section 6.1.3, restate the cost function (5.23) for a horizon length NT to

$$\bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}) = \mathbf{E} \left\{ \|\bar{x}(NT)\|_{\bar{W}}^2 + \sum_{\ell=0}^{NT-1} \|\bar{x}(\ell)\|_{\bar{Q}}^2 + \|\bar{u}(\ell)\|_{\bar{R}(\bar{\gamma}(\ell))}^2 \middle| \bar{x}(0) ; \bar{S} \right\}, \quad (6.21)$$

where $\bar{x}(k)$ and $\bar{u}(k)$ are given by (6.6) and (6.9), respectively. The cost (6.21) is conditioned by $\bar{x}(0)$ and the schedule \bar{S} . The length NT sequence of control inputs is defined as

$$\bar{U}_{NT} \triangleq (\bar{u}(\ell) : \ell = 0, \dots, NT - 1)$$

and the weighting matrices by

$$\begin{aligned} \bar{Q} &\triangleq \text{diag} \left\{ \begin{bmatrix} Q & \dots & Q \end{bmatrix} \right\} \in \mathbb{R}^{mT \times mT} & Q &\geq 0 \\ \bar{R}(\bar{\gamma}) &\triangleq \text{diag} \left\{ \begin{bmatrix} R(\bar{\gamma}_0) & \dots & R(\bar{\gamma}_{T-1}) \end{bmatrix} \right\} \in \mathbb{R}^{nT \times nT} & R &\geq 0 \\ \bar{W} &\triangleq \text{diag} \left\{ \begin{bmatrix} W & \dots & W \end{bmatrix} \right\} \in \mathbb{R}^{mT \times mT} & W &\geq 0, \end{aligned}$$

with $\bar{\gamma}$ given in (6.10). Also note, that for a periodic schedule $\bar{S} \in \bar{\mathcal{S}}^T$, $\mathbf{E} \{ \bar{\gamma}(k) ; \bar{S} \}$ does not depend on k . The time-indexing k in $\bar{\gamma}(k)$ is therefore omitted in the remainder of the chapter.

It is, using the maps in Table 6.1, easy to verify that (6.21) and (5.23) in Section 5.3 are equivalent. We want to find the optimal sequence of controls \bar{U}_{NT}^* and optimal admissible periodic schedule \bar{S}_T^* that satisfy the joint minimization problem

$$(\bar{U}_{NT}^*, \bar{S}_T^*) = \arg \min_{\bar{U}_{NT}, \bar{S} \in \bar{\mathcal{S}}_a^T} \bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}). \quad (6.22)$$

This then results in the optimal cost

$$\bar{J}_W^*(\bar{x}(0)) \triangleq \bar{J}_W(\bar{x}(0), \bar{S}^*, \bar{U}_{NT}^*).$$

6.2.1 Solution to the minimization problem

The joint optimization problem (6.22) is a mixed integer problem, where the variable \bar{S} is discrete. Recall, that each element s_j of \bar{S} takes values in the set $\{0, p_{\text{CAP}}, p_{\text{CFP}}\}$. This makes (6.22) a difficult problem to solve. However, as in Section 5.3.1, we can rewrite problem (6.22) to separate the integer and the continuous part of the optimization problem. This yields

$$\bar{J}_W^*(\bar{x}(0)) = \min_{\bar{S} \in \bar{\mathcal{S}}_a^T} \left[\min_{\bar{U}_{NT}} [\bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT})] \right],$$

which allows us to first solve the inner problem for each possible periodic schedule \bar{S} , and then find the optimal periodic schedule afterwards. The inner problem is convex for each \bar{S} and has an analytical solution, as we present next.

Theorem 6.2.1. *For a known periodic schedule $\bar{S} \in \bar{\mathcal{S}}^T$ with period T , the optimal controls are given by the policy*

$$\bar{u}(k) = -\bar{L}^*(k) \bar{x}(k) \quad k = 0, \dots, NT - 1, \quad (6.23)$$

where the control gain is given by the $nN \times mN$ block diagonal matrix

$$\bar{L}^*(k) = [\bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S})]^\dagger \bar{B}(\bar{S}) \bar{P}_{k+1} \bar{A}, \quad (6.24)$$

with \dagger meaning the pseudo inverse defined in Definition 5.2.1.¹, and for some square matrix X ,

$$M(X, \bar{S}) \triangleq \sum_{s \in \mathbb{Z}_2^{n_T}} \bar{B}(s)^T X \bar{B}(s) \mathbf{Pr}\{\bar{\gamma} = s; \bar{S}\}. \quad (6.25)$$

The $mT \times mT$ block diagonal matrix \bar{P}_k is given by the recursion

$$\bar{P}_k = \bar{Q} + \bar{A}^T \bar{P}_{k+1} \bar{A} - \bar{A}^T \bar{P}_{k+1} \bar{B}(\bar{S}) [\bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S})]^\dagger \bar{B}(\bar{S})^T \bar{P}_{k+1} \bar{A} \quad (6.26)$$

with $\bar{P}_{NT} = \bar{W}$. This results in the optimal cost for the known periodic schedule \bar{S}

$$\bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \bar{x}(0)^T \bar{P}_0 \bar{x}(0) + \sum_{\ell=0}^{NT-1} \text{trace}(\bar{\Sigma}_\omega(\ell) \bar{F}^T \bar{P}_{\ell+1} \bar{F}),$$

where $\bar{\Sigma}_\omega(k)$ is defined in (6.12).

Proof. The proof is presented in Appendix 6.A. □

Theorem 6.2.1 presents a closed form solution for the optimal controls and optimal cost for a known schedule \bar{S} . However, note that, as in Theorem 5.3.1, (6.23), (6.24) and (6.26) in Theorem 6.2.1 all depend on the particular schedule \bar{S} . We therefore define the functions

$$\begin{aligned} \bar{P}_k(S) &\triangleq \bar{Q} + \bar{A}^T \bar{P}_{k+1}(S) \bar{A} \\ &\quad - \bar{A}^T \bar{P}_{k+1}(S) \bar{B}(S) [\bar{R}(S) + M(\bar{P}_{k+1}(S), S)]^\dagger \bar{B}(S)^T \bar{P}_{k+1}(S) \bar{A} \\ \bar{L}_k^*(S) &\triangleq [\bar{R}(S) + M(\bar{P}_{k+1}(S), S)]^\dagger \bar{B}(S) \bar{P}_{k+1}(S) \bar{A}, \end{aligned}$$

such that for a given schedule \bar{S} , the control input is given by

$$\bar{u}(k) = -\bar{L}_k^*(\bar{S}) \bar{x}(k). \quad (6.27)$$

Also, denote the sequence of optimal control laws by

$$\bar{U}_{NT}^*(\bar{S}) \triangleq (-\bar{L}_\ell^*(\bar{S}) \bar{x}(\ell) : \ell = 0, \dots, NT - 1).$$

Then, the optimal admissible schedule \bar{S}^* can be found by

$$\begin{aligned} \bar{S}^* &= \arg \min_{\bar{S} \in \bar{\mathcal{S}}_d^T} [\bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*(\bar{S}))] \\ &= \arg \min_{\bar{S} \in \bar{\mathcal{S}}_d^T} \left[\bar{x}(0)^T \bar{P}_0(\bar{S}) \bar{x}(0) + \sum_{\ell=1}^{NT} \text{trace}(\bar{\Sigma}_\omega(\ell) \bar{F}^T \bar{P}_\ell(\bar{S}) \bar{F}) \right]. \end{aligned} \quad (6.28)$$

This results in the length NT sequence of optimal co-design control laws

$$\bar{U}_{NT}^*(\bar{S}^*) = (-\bar{L}_\ell^*(\bar{S}^*) \bar{x}(\ell) : \ell = 0, \dots, NT - 1).$$

As discussed in Section 5.3.3, possible performance gains can be achieved by selecting the final state weighting W carefully. For conventional MPC algorithms, W is often related to an optimum of some infinite horizon cost function [15], [37], [56]. This motivates the use of an infinite horizon cost approximation to select the weighting W , which is presented in Section 6.3.

Next, we link the results that we obtained in this section to the results for the time-varying system (6.1a) that was obtained in Section 5.3.1.

¹Since \bar{S} is zero for actuators that we do not address, the matrices $\bar{R}(\bar{S})$ and $M(\bar{P}_{k+1}, \bar{S})$ are in general singular.

6.2.2 Link between time-varying and LTI formulation

The optimal controls for system (6.1a) (and (5.20) in Section 5.1) can be obtained directly from (6.27) using the maps in Table 6.1. For the scheduling sequence $\mathcal{S}_{k,N}$ with period T , this results in the time-varying control laws from Theorem 5.3.1, which are of the form

$$u(k) = -L_k^*(\mathcal{S}_{k,N})x(k),$$

where, as described in (6.17) and the definitions in (6.7) and (6.9), the LTI control law at each time k is of the form

$$\bar{L}_k^*(\bar{S}) = \text{diag} \left\{ \left[\mathbf{0}_{n,m} \quad \cdots \quad \mathbf{0}_{n,m} \quad L_k^*(\mathcal{S}_{k,N}) \quad \mathbf{0}_{n,m} \quad \cdots \quad \mathbf{0}_{n,m} \right] \right\}. \quad (6.29)$$

The control law $L_k^*(\mathcal{S}_{k,N})$ from Theorem 5.3.1 for the time-varying system (6.1a) appears on the i 'th $n \times m$ block on the diagonal of $\bar{L}_k^*(\bar{S})$ in (6.29), where $i = k \% T$. The schedules in the sequence \mathcal{S}_N are extracted directly from \bar{S} , which is defined in (6.14).

The LTI matrix $\bar{P}_k(\bar{S})$ is for $k = (0, \dots, NT - 1)$ structured as

$$\bar{P}_k(\bar{S}) = \text{diag} \left\{ \left[\mathbf{0}_{m,m} \quad \cdots \quad \mathbf{0}_{m,m} \quad P_k(\mathcal{S}_{k,N}) \quad \mathbf{0}_{m,m} \quad \cdots \quad \mathbf{0}_{m,m} \right] \right\},$$

with the time-varying recursion (5.31) appearing on the i 'th $m \times m$ block of $\bar{P}_k(\bar{S})$ where $i = k \% T$.

Remark 6.2.1. *Note, that while the dimension of the matrix (6.26) in the cost (6.28) is $mT \times mT$, the matrix is sparse. In fact, only one $m \times m$ diagonal block is non-zero. Therefore, the online computational cost for computing (6.28) is similar to the cost for computing (5.32), which is given by (5.37).*

It is worth noting, that the optimal controls and solutions to the recursion (6.26) in Theorem 6.2.1 can be computed simultaneously for all possible cyclic rotations of a scheduling sequence \mathcal{S}_T . Consider a periodic scheduling sequence \mathcal{S}_N , as described in Section 6.1.2, and a horizon length of NT , $N \in \mathbb{N}$. Denote the cyclic shift of the periodic schedule \mathcal{S} by ℓ steps

$$\mathcal{S}(\ell) \triangleq (S(\ell), \dots, S(T-1), S(0), \dots, S(\ell-1)) \quad \ell \in \{0, \dots, T-1\}.$$

And denote for the length NT scheduling sequence with periodic scheduling sequence $\mathcal{S}(\ell)$ repeated N times and starting from $S(k)$ by $\mathcal{S}_{k,N}(\ell)$. Then, if we compute the controller using Theorem 6.2.1, but omit the mappings from the time-varying to LTI formulation in the first column of Table 6.1, the vector $\bar{L}_k^*(\bar{S})$ will be of the form

$$\bar{L}_k^*(\bar{S}) = \text{diag} \left\{ \left[L_k^*(\mathcal{S}_{k,N}(0)) \quad L_k^*(\mathcal{S}_{k,N}(1)) \quad \cdots \quad L_k^*(\mathcal{S}_{k,N}(T-1)) \right] \right\}.$$

For a phase offset $0 \leq \ell < T$, the control laws for the time-varying system (6.1a) are found as the $i = (k + \ell) \% T$ $n \times m$ block on the diagonal of $\bar{L}_k^*(\bar{S})$. Similarly, the solution to the recursion (6.26) is at each time-step given by

$$\bar{P}_k(\bar{S}) = \text{diag} \left\{ \left[P_k(\mathcal{S}_{k,N}(0)) \quad P_k(\mathcal{S}_{k,N}(1)) \quad \cdots \quad P_k(\mathcal{S}_{k,N}(T-1)) \right] \right\}, \quad (6.30)$$

where for a phase offset ℓ , the solution to the time-varying recursion (5.31) for each k is found on the i 'th diagonal of $\bar{P}_k(\bar{S})$ with $i = (k + \ell) \% T$. Then, the time-varying cost (5.30) for each phase

$0 \leq \ell < T$ of a scheduling sequence $\mathcal{S}_N(\ell)$, can be stated as

$$\begin{aligned} J_W(x(0), \mathcal{S}_N(\ell), U_{NT}^*(\mathcal{S}_N(\ell))) &= x(0)^T P_0(\mathcal{S}_N(\ell)) x(0) + \sum_{k=1}^{NT} \text{trace}(\Sigma_\omega F^T P_k(\mathcal{S}_{k,N}(\ell)) F) \\ &= x(0)^T P_0(\mathcal{S}_N(\ell)) x(0) + \sum_{k=1}^{NT} \text{trace}(\bar{\Sigma}_\omega(k+\ell) \bar{F}^T \bar{P}_k(\bar{S}) \bar{F}), \end{aligned}$$

where P_k can be found on the diagonal of (6.30). The optimal controls are then given by

$$u_k = -L_k^*(\mathcal{S}_{k,N}^*(\ell)) x(k).$$

6.3 Infinite horizon approximation

The optimal controls and scheduling sequence that minimize the finite horizon cost function (5.23) in Section 5.3, tend to result in improved control performance when the prediction horizon N is increased. This is illustrated by the simulations in Section 5.4. Increasing N in (5.23) corresponds to increasing the period T in the LTI cost function (6.21). In (6.21), we ideally want to let $T \rightarrow \infty$, such that the control cost is minimized over all time. Letting $T \rightarrow \infty$ means, that there is no periodic assumption on the scheduling sequence that minimizes the cost function. However, recall that the set $\bar{\mathcal{S}}_a^T$ of scheduling sequences is a discrete set. To find the optimal scheduling sequence and control law, one has to solve a combinatorial optimization problem, see (6.28). In this optimization problem, every possible scheduling sequence in the set $\bar{\mathcal{S}}_a^T$ has to be tested. As shown in Section 5.3.4, the number of scheduling sequences in this set grows exponentially with the period T . This makes it prohibitive, if not impossible, to minimize (6.21) for large T .

However, recall that in (6.21), the prediction horizon is parameterized by both T and N , where N indicates the number of periods T to optimize over. If we here, instead of letting $T \rightarrow \infty$, choose to fix T and let $N \rightarrow \infty$, we obtain an approximation of the infinite horizon cost function under the assumption that the optimal scheduling sequence has period T . By fixing T , the number of possible scheduling sequences remains bounded and the optimal solution can be found. We present two methods which approximate the infinite horizon cost function by assuming periodic scheduling.

A drawback with the above mentioned approach is that we force the scheduling sequence to be periodic. While the optimal scheduling sequence can be periodic, we do not know the period. Also, we do not know whether the optimal scheduling sequence is periodic from the initial state k , or first becomes periodic after some time. To take this into account, we, inspired by the work [40], propose a method that for some steps, say $k+K$, where $K \geq 1$, finds the optimal scheduling sequence among all possible length K scheduling sequences. We then assume, that after $K+1$ steps, the optimal scheduling sequence resorts to a periodic scheduling sequence with a fixed period T . This method is also referred to as the rollout method [6], [15]. We present a co-design algorithm that utilizes this approach in Section 6.4.

In this section we present two methods to approximate the infinite horizon cost for a periodic scheduling sequence with a fixed period. We present co-design algorithms that utilize the solutions to the infinite horizon approximations in Section 6.4.

6.3.1 Discounted cost formulation

The first approach to an infinite horizon co-design cost function is the discounted cost. This cost function will reduce the cost penalty on future states. We consider the discounted cost function

$$\bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}) \triangleq \mathbf{E} \left\{ \sum_{\ell=0}^{NT} \alpha^\ell \left[\|\bar{x}(\ell)\|_{\bar{Q}}^2 + \|\bar{u}(\ell)\|_{\bar{R}(\bar{\gamma})}^2 \right] \middle| \bar{x}(0); \bar{S} \right\}. \quad (6.31)$$

where $\alpha \in (0, 1)$ is the discounting factor. Lower values of α will penalize future state less than more recent states. We want to find the optimal controls and schedule that satisfy

$$(\bar{S}_T^*, \bar{U}_{NT}^*) \triangleq \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T, \bar{U}_{NT}} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}).$$

Similar to the linear quadratic (LQ) cost function in Section 6.2, we can split the joint optimization problem to

$$(\bar{S}_T^*, \bar{U}_{NT}^*) = \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T} \left[\arg \min_{\bar{U}_{NT}} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}) \right]. \quad (6.32)$$

This means that we for a known scheduling sequence \bar{S} can find the sequence of optimal control laws

$$\bar{U}_{NT}^* \triangleq (\bar{u}^*(\ell) : \ell = 0, \dots, NT - 1)$$

by

$$\bar{U}_{NT}^*(\bar{S}) = \arg \min_{\bar{U}_{NT}} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}).$$

6.3.1.1 Solution

This leads to the following result.

Lemma 6.3.1. *For a known schedule $\bar{S} \in \bar{\mathcal{S}}^T$ and horizon length N , the optimal control laws that minimize the discounted cost (6.31) are given by*

$$\bar{u}^*(k) = -\bar{L}_k^* \bar{x}(k) \quad k = 0, \dots, NT - 1$$

where the control gains are given by

$$\bar{L}_k^* = \left[\frac{1}{\alpha} \bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S}) \right]^\dagger \bar{B}(\bar{S})^T \bar{P}_{k+1} \bar{A}$$

where

$$\bar{P}_{k-1} = \bar{Q} + \alpha \bar{A}^T \bar{P}_k \bar{A} - \alpha \bar{A}^T \bar{P}_k \bar{B}(\bar{S}) \left[\frac{1}{\alpha} \bar{R}(\bar{S}) + M(\bar{P}_k, \bar{S}) \right]^\dagger \bar{B}(\bar{S})^T \bar{P}_k \bar{A} \quad (6.33)$$

with $M(P, S)$ defined in (6.25). This results in the optimal cost

$$\bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \bar{x}(0)^T \bar{P}_0 \bar{x}(0) + \sum_{t=1}^{NT} \alpha^t \text{trace}(\bar{\Sigma}_{\bar{\omega}}(t-1) \bar{F}^T \bar{P}_t \bar{F}).$$

Proof. The proof is found in Appendix 6.B. \square

We want minimize the cost (6.31) as the horizon $N \rightarrow \infty$. Define the infinite length sequence of control laws by

$$\bar{U}_\infty \triangleq (\bar{u}(\ell) : \ell = 0, 1, \dots), \quad (6.34)$$

such that for a schedule $\bar{S} \in \bar{\mathcal{S}}_a^T$ the cost

$$\bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_\infty) = \min_{\bar{U}_\infty} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_\infty) \quad (6.35)$$

where the sequence of optimal control laws

$$\bar{U}_\infty^* \triangleq (\bar{u}^*(\ell) ; \ell = 0, 1, \dots)$$

satisfy

$$\bar{U}_\infty^*(\bar{S}) = \arg \min_{\bar{U}_\infty} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_\infty).$$

This leads to the following result:

Theorem 6.3.2. *Suppose that for a known periodic schedule $\bar{S} \in \bar{\mathcal{S}}^T$ the limit $\lim_{N \rightarrow \infty} \bar{P}_N$ exists. Then, the optimal control laws that minimize the infinite horizon discounted cost function (6.35) are given by*

$$\bar{u}(k) = -\bar{L}_d^* \bar{x}(k) \quad k = 0, 1, \dots$$

with the control gain given by

$$\bar{L}_d^* = \left[\frac{1}{\alpha} \bar{R}(\bar{S}) + M(\bar{P}, \bar{S}) \right]^\dagger \bar{B}(\bar{S})^T \bar{P} \bar{A}. \quad (6.36)$$

The matrix \bar{P} is the solution to the discounted Riccati equation which satisfies

$$\bar{P} = \bar{Q} + \alpha \bar{A}^T \bar{P} \bar{A} - \alpha \bar{A}^T \bar{P} \bar{B}(\bar{S}) \left[\frac{1}{\alpha} \bar{R}(\bar{S}) + M(\bar{P}, \bar{S}) \right]^\dagger \bar{B}(\bar{S})^T \bar{P} \bar{A}. \quad (6.37)$$

This results in the optimal discounted cost

$$\bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_\infty^*) = \bar{x}(0)^T \bar{P} \bar{x}(0) + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_\omega(t-1) \bar{F}^T \bar{P} \bar{F}). \quad (6.38)$$

Proof. The proof is presented in Appendix 6.C. \square

The optimal scheduler and controller co-design for the discounted cost function is then done by computing

$$\bar{S}^* = \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_\infty^*(\bar{S})), \quad (6.39)$$

which yields the optimal control law

$$\bar{u}^*(k) = -\bar{L}_d^*(\bar{S}^*) \bar{x}(k) \quad k \in \mathbb{N}. \quad (6.40)$$

This results in the optimal discounted cost

$$\bar{J}_d^*(\bar{x}(0)) \triangleq \bar{J}_d(\bar{x}(0), \bar{S}^*, \bar{U}_\infty^*(\bar{S}^*)). \quad (6.41)$$

Remark 6.3.1. *The solution to the discounted Riccati equation (6.37) can be obtained as the iterative solution to (6.33). Note however, that it is not guaranteed that a solution to \bar{P} exists for every schedule \bar{S} . Also, the existence of a solution to \bar{P} does not guarantee closed-loop stability. Refer to [14] for more information on the existence of solutions to (6.37).*

One drawback with using the discounted cost is, that a finite optimal cost (6.41) does not guarantee closed-loop stability. Next, we present a cost function that instead penalizes the long-run average cost.

6.3.2 Averaged infinite horizon

In this section we co-design the optimal controller and scheduler to minimize the averaged cost function

$$\bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{NT}) \triangleq \frac{1}{NT} \mathbf{E} \left\{ \sum_{\ell=0}^{NT} \left[\|\bar{x}(\ell)\|_{\bar{Q}}^2 + \|\bar{u}(\ell)\|_{\bar{R}(\bar{\gamma})}^2 \right] \middle| \bar{x}(0); \bar{S} \right\}. \quad (6.42)$$

We want to find the optimal control laws and schedule that satisfy

$$(\bar{S}^*, \bar{U}_{NT}^*) = \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T, \bar{U}_{NT}} \bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{NT}).$$

This joint optimization problem can also be rewritten to the sequential optimization problem

$$\begin{aligned} \bar{S}^* &= \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T} \min_{\bar{U}_{NT}} \bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{NT}) \\ \bar{U}_{NT}^*(\bar{S}^*) &\triangleq \arg \min_{\bar{U}_{NT}} \bar{J}_a(\bar{x}(0), \bar{S}^*, \bar{U}_{NT}). \end{aligned}$$

This results in the optimal periodic cost

$$\bar{J}_a^*(\bar{x}(0)) \triangleq \bar{J}_a(\bar{x}(0), \bar{S}^*, \bar{U}_{NT}^*(\bar{S}^*)).$$

6.3.2.1 Solution

We have the following result:

Lemma 6.3.3. *For a known schedule $\bar{S} \in \bar{\mathcal{S}}^T$ and horizon length N , the optimal averaged cost is given by*

$$\bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \frac{1}{NT} \bar{x}(0)^T \bar{P}_0 \bar{x}(0) + \frac{1}{NT} \sum_{\ell=0}^{NT-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P}_{\ell+1} \bar{F}), \quad (6.43)$$

where \bar{P}_k is given by (6.26). This results in the optimal control laws (6.23) with control gains (6.24).

Proof. The proof is presented in Appendix 6.D. □

The optimal infinite horizon cost (6.42) for a known schedule and an infinite length sequence of controls (6.34) is given by

$$\bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{\infty}^*) = \min_{\bar{U}_{\infty}} \bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{\infty}). \quad (6.44)$$

The infinite length sequence \bar{U}_{∞}^* that minimizes (6.44) is presented in the next theorem.

Theorem 6.3.4. *Suppose that for a known schedule $\bar{S} \in \bar{\mathcal{S}}^T$ the limit $\lim_{N \rightarrow \infty} \bar{P}_N$ exists, where \bar{P}_k is given in (6.26). Then, the optimal control law, that minimizes the averaged cost function (6.44), is given by*

$$\bar{u}(k) = -\bar{L}_a^* \bar{x}(k) \quad k = 0, 1, \dots$$

with the control gain

$$\bar{L}_a^* = [\bar{R}(\bar{S}) + M(\bar{P}, \bar{S})]^\dagger \bar{B}(\bar{S})^T \bar{P} \bar{A}. \quad (6.45)$$

Here, \bar{P} is the solution to the Riccati equation

$$\bar{P} = \bar{Q} + \bar{A}^T \bar{P} \bar{A} - \bar{A}^T \bar{P} \bar{B}(\bar{S}) [\bar{R}(\bar{S}) + M(\bar{P}, \bar{S})]^\dagger \bar{B}(\bar{S})^T \bar{P} \bar{A}. \quad (6.46)$$

This results in the optimal cost

$$\bar{J}_a(\bar{S}, \bar{U}_\infty^*) = \frac{1}{T} \sum_{t=0}^{T-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(t) \bar{F}^T \bar{P} \bar{F}). \quad (6.47)$$

Proof. The proof is presented in Appendix 6.E. □

The optimal schedule is then found as

$$\bar{S}^* = \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T} \bar{J}_a(\bar{S}, \bar{U}_\infty^*(\bar{S})), \quad (6.48)$$

which results in the optimal control law

$$\bar{u}(k)^* = -\bar{L}_a^*(\bar{S}^*) \bar{x}(k) \quad k = 0, 1, \dots$$

It is worth noting that for the averaged cost, the optimal schedule does not depend on the state \bar{x} .

6.3.2.2 Stability

For the averaged cost (6.42), we show that for a schedule \bar{S} , the existence of a solution to the Riccati equation (6.46) is a necessary and sufficient condition for mean square stability (MSS) of the closed-loop system. This is presented in the following result.

Theorem 6.3.5. *For a schedule $\bar{S} \in \bar{\mathcal{S}}^T$, the closed-loop system (6.18) with control law (6.45) is MSS if and only if there exists a solution \bar{P} for the Riccati equation (6.46).*

Proof. The proof is presented in Appendix 6.F. □

Corollary 6.3.6. *If the pair $(A, Q^{\frac{1}{2}})$ is detectable, it follows directly by Definition 6.1.1 and Theorem 6.3.5 that stochastic stabilizability is necessary and sufficient for the existence of a solution to the modified Riccati equation (6.46).*

6.4 Implementation

In this section, we present algorithms that utilize the infinite horizon approaches from Section 6.3. First, we present a MPC method that utilizes the above infinite horizon approaches to select a suitable final state weighting for the online algorithm presented in Algorithm 5.1. This is followed by an online implementation of the discounted cost in Section 6.3.1. Finally, a number of offline co-design implementations are presented. These will make it easier to select round robin (RR) scheduling sequences, than the heuristic which was used in the simulations in Section 5.4. The proposed offline approaches will result in the optimal offline performance for the given period T . Further, two of the offline approaches provide MSS co-design solutions.

6.4.1 MPC with final state weighting

Based on the discussions in Sections 6.1 and 6.3, we combine the results from Sections 6.2 and 6.3. More specifically, we present a method that utilizes the solutions to the infinite horizon cost approximations in Section 6.3 as a final state weighting in the finite horizon cost function (6.21) in Section 6.2.

As commonly done in MPC, the final state weighting W in (6.21) is tuned to represent the control cost as the horizon $N \rightarrow \infty$. Ideally, one would like to compute the cost (6.21) as the period $T \rightarrow \infty$. This means, that there is no periodic assumption on the schedule. However, as stated in Section 5.3.4, the scheduling part of the co-design algorithm amounts to a non-deterministic polynomial-time (NP)-hard problem. It is therefore impossible to compute (6.21) as $T \rightarrow \infty$.

To address this issue, we instead assume that for a limited amount of time, say $k + K - 1$, $K \geq 1$, the optimal scheduling sequence can be any possible admissible sequence. Then, from time $k + K$ and on, the scheduling sequence converges to a periodic schedule with a fixed period T . The cost from time $k + K$ and on can then be represented by one of the infinite horizon approximations (6.42) or (6.31), that are presented in Section 6.3. We implement this by using the periodic approximations (6.31) and (6.42) to design the final state weighting W in (6.21).

Remark 6.4.1. *An alternative idea is to assume that the scheduling sequence always is periodic. Implementations based on this are presented in Sections 6.4.2 and 6.4.3.*

We first explain how the solutions to the LTI Riccati equations (6.37) and (6.46) are utilized to select the final state weighting for the finite horizon cost function. This is non-trivial, since the LTI Riccati equations each represent a periodic solution to a periodic Riccati equation. This is followed by the design of the online part of the MPC algorithm. Section 6.4.1.2 presents a method to select which periodic schedule \bar{S} to used for the Riccati equation that is used for the final state weighting. We present three cost functions that can be used to select \bar{S} . Section 6.4.1.3 presents a stability result for the MPC algorithm and Section 6.4.1.4 shows a derivation of the steady-state discounted cost (6.38), which is used to select the final state weighting in Section 6.4.1.2.

6.4.1.1 MPC with final state weighting

In this section, we utilize a solution to Riccati equation (6.37) or (6.46) for a periodic schedule $\bar{S} \in \bar{\mathbb{S}}_a^T$ to design the final state weighting W in the cost function (6.21). This part of the design is done offline.

It is however worth noting, that the final state weighting $W \in \mathbb{R}^{m \times m}$ in (6.21), while the solutions to the Riccati equations (6.37) and (6.46) are in $\mathbb{R}^{mT \times mT}$. The reason for this is, as explained in Section 6.2.2, that the solutions to (6.37) and (6.46) are in fact augmented matrices that contain the periodic solutions to a periodic Riccati equation on the block-diagonal. Recall from Section 6.2.2, that by computing the Riccati equation \bar{P} for \bar{S} , all possible cyclic shifts of the periodic scheduling sequence \mathcal{S} can be computed simultaneously. Thus, for a periodic schedule \bar{S} with period T ,

$$\bar{P} = \text{diag} \left\{ \begin{bmatrix} P_0 & P_1 & \cdots & P_{T-1} \end{bmatrix} \right\}.$$

The question that arises is, which P_i , $i \in \{0, \dots, T-1\}$ to choose as the final state weighting W ?

In MPC, the parameter W in (6.21) is tuned to represent the behaviour or cost of the system at time $k+K$, where $K \geq 1$. As discussed in Section 6.1, we assume that after $k+K$ time-steps, the optimal scheduling sequence \mathcal{S}_∞^* converges to a periodic scheduling sequence with a fixed period T . This has to be taken into account in the cost function (6.21). One obvious method to do this, is by making the final state weighting W periodic. This means that for example, at time k , we select $W_{k\%T}$, at time $k+1$ we select $W_{k+1\%T}$ etc. In this case, the cost (6.21) would at each time k be stated as

$$\bar{J}_{W_i}(x(k), \bar{Z}, \bar{U}_K) \quad \begin{array}{l} k \in \mathbb{N} \\ i = k \% T. \end{array} \quad (6.49)$$

However, at the initial state $k=0$, we do not know whether $i = k \% T$ or $i = (k+\ell) \% T$, where $\ell \in \{1, \dots, T-1\}$. Also, recall from the discussion above and in Section 6.3, that we fix the period T , since it is infeasible to find the actual optimal period that the scheduling sequence might converge to. This means, that the optimal periodic schedule $\bar{S} \in \mathbb{S}_a^T$ for a fixed T , might not be optimal among all possible period lengths. Therefore, to potentially increase the control performance, we, instead of cycling through W_i , $i \in \{0, \dots, T-1\}$, choose to create a set \mathbb{W} that is defined as

$$\mathbb{W} \triangleq \{P_\ell | \ell = 0, \dots, T-1\}. \quad (6.50)$$

Then, at every time-step, we find the final state weighting $W \in \mathbb{W}$ that minimizes (6.49) i.e.

$$\bar{J}_{W^*}(\bar{x}(k), \bar{Z}, \bar{U}_K) \triangleq \min_{W \in \mathbb{W}} \bar{J}_W(\bar{x}(k), \bar{Z}, \bar{U}_K). \quad (6.51)$$

It is worth noting, that the worst case cost of (6.51) would be when $W^* = W_i$. In this case (6.51) is equal the cyclic final state weighting cost (6.49). This thus means that

$$\bar{J}_{W^*}(\bar{x}(k), \bar{Z}, \bar{U}_K) \leq \bar{J}_{W_i}(\bar{x}(k), \bar{Z}, \bar{U}_K) \quad \begin{array}{l} k \in \mathbb{N} \\ i = k \% T. \end{array}$$

The disadvantage of the cost (6.51) compared to (6.49) is, that (6.51) requires more online computations.

In the online part of the MPC algorithm we want to, at every time-step k find a feasible finite length scheduling sequence \mathcal{Z}_K with length K , that minimizes the cost (5.23) over all $W \in \mathbb{W}$. It is worth noting, that the scheduling sequence \mathcal{Z}_K can be expressed by the LTI schedule $\bar{Z} \in \mathbb{S}_a^K$

Algorithm 6.1 MPC for the co-design with final state weighting. This algorithm is modified from Algorithm 5.1.

Input: state $x(k)$
Compute: $\bar{Z}^* \in \bar{\mathbb{S}}_a^K$ and $W^* \in \mathbb{W}$ using (6.52)
Compute: $\bar{u}(k) = -\bar{L}_0^*(\bar{Z}^*)\bar{x}(k)$
Obtain: time-varying control input using the mappings from Table 6.1 and schedule $Z^*(0)$ from the LTI schedule \bar{Z}^*
Output: schedule $Z^*(0)$ and control $u(k)$
Discard: the remaining controls $(\bar{L}_1^*, \dots, \bar{L}_{K-1}^*)$ and schedules $\mathcal{Z}_{1,K-1}^*$

using the formulations in Section 6.1.3. Then, the optimal cost is given by

$$\begin{aligned} \bar{J}_{W^*}(\bar{x}(k)) &\triangleq \min_{\bar{Z} \in \bar{\mathbb{S}}_a^K, \bar{U}_K} \bar{J}_{W^*}(\bar{x}(k), \bar{Z}, \bar{U}_K) \\ &= \min_{\bar{Z} \in \bar{\mathbb{S}}_a^K, \bar{U}_K, W \in \mathbb{W}} \bar{J}_W(\bar{x}(k), \bar{Z}, \bar{U}_K), \end{aligned}$$

where $\bar{J}_W(\bar{x}(k), \bar{Z}, \bar{U}_K)$ is the cost function (6.21). The optimal scheduling sequence and optimal controls are then defined as

$$(\bar{Z}^*, \bar{U}_K^*) \triangleq \arg \min_{\bar{Z} \in \bar{\mathbb{S}}_a^K, \bar{U}_K} \min_{W \in \mathbb{W}} \bar{J}_W(\bar{x}(0), \bar{Z}, \bar{U}_K).$$

Recall from Theorem 6.2.1, that the optimal controls can be found analytically for an optimal schedule \bar{Z}^* and final state weighting W^* . Thus, in the MPC algorithm, we find

$$(\bar{Z}^*, W^*) = \arg \min_{\bar{Z} \in \bar{\mathbb{S}}_a^K, W \in \mathbb{W}} \bar{J}_W(\bar{x}(0), \bar{Z}, \bar{U}_K^*(\bar{Z})). \quad (6.52)$$

The implementation of the MPC algorithm that is based on (6.52) requires some modifications to Algorithm 5.1. This results in Algorithm 6.1.

6.4.1.2 Selecting the final state weighting

Now, we present a method to find the solution \bar{P} to the Riccati equation (6.37) or (6.46) which is used to design the set of final state weightings \mathbb{W} that is used in (6.52). Recall from Theorems 6.3.2 and 6.3.4, that there exists a solution \bar{P} to the Riccati equation (6.37) or (6.46) for every stochastically stabilizable schedule², that is, the schedules that satisfy the condition in Definition 6.1.1. The question is thus, which schedule to use?

Inspired by ideas from the work [40], we select \bar{P} from the schedule that minimizes the cost

$$\min_{T \in \{1, \dots, \bar{T}\}, \bar{S} \in \bar{\mathbb{S}}_a^T} [\bar{F}_y(\bar{S}) + \beta T \text{card}(\bar{\mathbb{S}}_a^K)], \quad (6.53)$$

where $y \in \{a, d0, dx\}$ indicates the choice of infinite horizon approach that is used and K is the horizon length for the online MPC cost (6.52). $\bar{T} \geq 1$ is used to set the maximum period to optimize over. The parameter $\beta \geq 0$ is used to trade-off the computational cost and the control performance. The computational cost amounts to the number of schedules $\bar{Z} \in \bar{\mathbb{S}}_a^K$ and final state

²Except for the discounted Riccati equation (6.37) with $\alpha < 1$, since in this case a solution can exist, even if the schedule is not stochastically stabilizable.

weightings $W \in \mathbb{W}$ that have to be compared in (6.52). It is worth noting, that in (6.52), $\text{card}(\mathbb{W})$ final state weightings have to be compared for every admissible schedule $\bar{Z} \in \bar{\mathbb{S}}_a^K$. Recall from Section 5.3.4, that the amount of admissible schedules is given by

$$\text{card}(\bar{\mathbb{S}}_a^K) = \left[\binom{n}{N_{\text{CFP}}} \binom{n - N_{\text{CFP}}}{N_{\text{CAP}}} \right]^K.$$

Thus, in the online part (6.52) of the MPC Algorithm 6.1, $T \text{card}(\bar{\mathbb{S}}_a^K)$ combinations have to be tested at every time-step k . Tuning β in (6.53) allows one to trade-off the amount of combinations that have to be tested in (6.52) and the value of the cost function $\bar{F}_y(\bar{S})$.

The set \mathbb{W} of final state weightings (6.50) is then comprised of the block-diagonal elements of the solution to the Riccati equation

$$\bar{P}^* \triangleq \bar{P}(\bar{S}^*),$$

where \bar{P} is given by (6.46) if $y = a$ and (6.37) if $y \in \{d0, dx\}$ and

$$\bar{S}^* \triangleq \min_{T \in \{1, \dots, \bar{T}\}} \arg \min_{\bar{S}_T \in \bar{\mathbb{S}}_a^T} [\bar{F}_y(\bar{S}) + \beta T \text{card}(\Xi_K)].$$

We then denote the optimal final state weightings by

$$W_y \in \mathbb{W}_y \quad y \in \{a, d0, dx\}$$

to indicate which infinite horizon approximation was used in the design.

Next, we present three possible choices for the infinite horizon cost approximation $\bar{F}_y(\bar{S})$.

Definition 6.4.1. *The infinite horizon cost approximation $\bar{F}_y(\bar{S})$, $y \in \{a, d0, dx\}$, that is used in (6.53) can be chosen as one of the following:*

- **Averaged (a).** *When the final state weighting is based on the averaged cost from Theorem 6.3.4, we can directly use (6.47), i.e.*

$$\bar{F}_a(\bar{S}) \triangleq \bar{J}_a(\bar{S}, \bar{U}_\infty^*(\bar{S})). \quad (6.54)$$

- **Discounted noise (d0).** *The discounted optimal cost for a known schedule (6.41) depends on the state $\bar{x}(k)$. However, the state $\bar{x}(k)$ is not available offline. We therefore assume that as the system converges to steady-state, the system state \bar{x} will converge to zero, and the cost will be noise-driven. We therefore set the state $\bar{x}(k) = \mathbf{0}$, for all k . This results in the cost*

$$\bar{F}_{d0}(\bar{S}) \triangleq \bar{J}_d(\mathbf{0}, \bar{S}, \bar{U}_\infty^*(\bar{S}_T^*)),$$

where $\bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_\infty^*(\bar{S}_T^*))$ is defined in (6.38). This is the approach that we used in the paper [85].

- **Discounted steady-state (dx).** *Realistically however, the state $\bar{x}(k)$ will, due to the noise $\bar{\omega}(k)$, not equal $\mathbf{0}$. A more sophisticated method is to utilize the statistics of the state $\bar{x}(k)$ as $k \rightarrow \infty$ to compute average steady-state cost. This is done using*

$$\bar{F}_{dx}(\bar{S}) \triangleq \min_{\ell \in \{0, \dots, T-1\}} F_{dx}(\ell, \bar{S}, \bar{U}_\infty^*(\bar{S})), \quad (6.55)$$

where F_{dx} is a cost function that is based on the discounted cost function (6.38). This cost is derived in Section 6.4.1.3 and is given by

$$F_{\text{dx}}(\ell, \bar{S}, \bar{U}_\infty^*(\bar{S})) = \text{trace}(\bar{e}_\ell \bar{\Sigma}_x(\bar{S}) \bar{e}_\ell^T \bar{P}(\bar{S})) + \sum_{t=0}^{T-1} \frac{\alpha^{t+1}}{1-\alpha^T} \text{trace}(\bar{\Sigma}_\omega(\ell+t) \bar{F}^T \bar{P}(\bar{S}) \bar{F}).$$

6.4.1.3 A stability result

Next, we have a result on stability for the MPC algorithm with the final state weighting obtained through (6.54).

Theorem 6.4.1. *Consider a system of the form (6.1a) with bounded noise covariance Σ_ω , and a periodic scheduling sequence $\bar{S} \in \bar{\mathcal{S}}^T$, for which there exists a solution to the Riccati equation (6.46). Then, the system (6.1a) using the MPC scheme in Algorithm 6.1 with the final state weighting computed using (6.54) is MSS.*

Proof. The proof follows a stochastic Lyapunov argument, and is presented in Appendix 6.G. \square

Remark 6.4.2. *To reduce the computational requirements in Algorithm 6.1, one can limit the optimal schedule to be found among a subset $\Xi^K \subset \bar{\mathcal{S}}_a^K$ in the online part of the MPC cost (6.52). In this case, the results from Theorem 6.4.1 hold true and Algorithm 6.1 is MSS if the subset Ξ^K contains all cyclic shifts of the periodic scheduling sequence that minimizes the infinite horizon cost (6.54).*

6.4.1.4 Computing the steady-state discounted cost F_{dx}

In this section, we present a method to compute the steady-state discounted cost F_{dx} that is described in (6.55). Recall, that the optimal discounted cost function (6.32) depends on the state $\bar{x}(k)$ at each time k . Therefore, to find the optimal schedule, the optimization problem (6.39) has to be solved at each time-step. However, to utilize the optimal discounted cost function (6.38) to find a periodic schedule that is optimal over the long run, that is, as $x \rightarrow \infty$, we need to compute the steady-state cost. In this section, we obtain the steady state cost, that then is used in (6.55) to find the optimal periodic schedule to select the final state weighting.

For a schedule $\bar{S} \in \bar{\mathcal{S}}^T$, define the expected value of the optimal discounted cost (6.38) at time k as

$$G_{\text{dx}}(k, \bar{S}, \bar{U}_\infty^*(\bar{S})) \triangleq \mathbf{E} \{ \bar{J}_d(\bar{x}(k), \bar{S}, \bar{U}_\infty^*(\bar{S})) ; \bar{S} \}, \quad (6.56)$$

and define, for each $\ell \in \{0, \dots, T-1\}$, where T is the period, the cost (6.56) as $k \rightarrow \infty$ by

$$F_{\text{dx}}(\ell, \bar{S}, \bar{U}_\infty^*(\bar{S})) \triangleq \lim_{k \rightarrow \infty} G_{\text{dx}}(k+\ell, \bar{S}, \bar{U}_\infty^*(\bar{S})). \quad (6.57)$$

We thus, for a schedule $\bar{S} \in \bar{\mathcal{S}}^T$, take the expected value of the optimal discounted cost (6.38) over the state $\bar{x}(k)$ as $k \rightarrow \infty$. Define the covariance of the state

$$\Sigma_{\bar{x}}(k, \bar{S}) \triangleq \mathbf{E} \{ \bar{x}(k) \bar{x}^T(k) ; \bar{S} \} \quad (6.58)$$

where $\Sigma_{\bar{x}}(0, \bar{S})$ is known. Then, we have the following result.

Lemma 6.4.2. For a known schedule $\bar{S} \in \bar{\mathcal{S}}^T$, the cost (6.56) is at any time $k = 0, 1, 2, \dots$ given by

$$G_{\text{dx}}(k, \bar{S}, \bar{U}_\infty^*(\bar{S})) = \text{trace}(\bar{e}_{k\%T} \Sigma_{\bar{x}}(k, \bar{S}) \bar{e}_{k\%T} \bar{P}(\bar{S})) + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(k+t-1) \bar{F}^T \bar{P}(\bar{S}) \bar{F}), \quad (6.59)$$

where \bar{e}_i is defined in (6.20) and $\Sigma_{\bar{x}}(k, \bar{S})$ is for a schedule \bar{S} is given by the recursion

$$\Sigma_{\bar{x}}(k+1, \bar{S}) = \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}) \Sigma_{\bar{x}}(k, \bar{S}) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}) ; \bar{S} \right\} + \bar{F} \bar{\Sigma}_{\bar{\omega}} \bar{F}^T, \quad (6.60)$$

with

$$\Sigma_{\bar{x}}(0, \bar{S}) = \mathbf{E} \{ \bar{x}(0) \bar{x}^T(0) ; \bar{S} \}$$

and $\bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma})$ is defined in (6.19).

Proof. The proof is presented in Appendix 6.H. □

The optimal solution to the steady-state cost function (6.57) is then given by:

Theorem 6.4.3. Consider a known schedule $\bar{S} \in \bar{\mathcal{S}}^T$ with period T for which a solution to the discounted Riccati equation (6.37) exists. Then, for each $\ell = 0, \dots, T-1$, a solution to the cost

$$F_{\text{dx}}(\ell, \bar{S}, \bar{U}_\infty^*) = \lim_{k \rightarrow \infty} G_{\text{dx}}(Tk + \ell, \bar{S}, \bar{U}_\infty^*(\bar{S}))$$

exists if and only if there exists a solution to

$$\Sigma_{\bar{x}}(\bar{S}) = \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}) \Sigma_{\bar{x}}(\bar{S}) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}) ; \bar{S} \right\} + \bar{F} \bar{\Sigma}_{\bar{\omega}} \bar{F}^T. \quad (6.61)$$

This results in the cost

$$F_{\text{dx}}(\ell, \bar{S}, \bar{U}_\infty^*) = \text{trace}(\bar{e}_\ell \Sigma_{\bar{x}}(\bar{S}) \bar{e}_\ell \bar{P}(\bar{S})) + \sum_{t=0}^{T-1} \frac{\alpha^{t+1}}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell+t) \bar{F}^T \bar{P}(\bar{S}) \bar{F}). \quad (6.62)$$

Proof. The proof is found in Appendix 6.I. □

The cost (6.62) in Theorem 6.4.3 is used on the right-hand-side of (6.55). From Lemma 6.4.2 and Theorem 6.4.3 the following result regarding MSS follows directly:

Corollary 6.4.4. By Theorem 6.4.3 it follows that, for a schedule $\bar{S} \in \bar{\mathcal{S}}^T$, the closed-loop system

$$\bar{x}(k) = \bar{H}_{\bar{L}_d^*(\bar{S})} \bar{x}(k) + \bar{F} \bar{\omega}(k),$$

is MSS if and only if there exists a positive definite solution to (6.61).

Remark 6.4.3. It is worth noting, that since (6.61) follows by the definition in (6.58), stochastic stabilizability, as defined in Section 6.1.4 is a necessary and sufficient condition for a schedule \bar{S} in Theorem 6.4.3.

6.4.2 Discounted MPC

This algorithm is based on the MPC algorithm that is presented in Algorithm 5.1. However, it instead uses the periodic schedules and control laws for the discounted cost. These are given by (6.39) and (6.40). The optimal schedule and controls are computed at every time-step k . The discounted MPC approach is presented in Algorithm 6.2.

Algorithm 6.2 Discounted MPC

1. At time-step k , compute \bar{S}^* using (6.39).
 2. Compute the optimal control laws $\bar{U}_\infty^*(\bar{S}^*)$ using Theorem 6.3.2.
 3. Apply schedule $\bar{S}(k) = \bar{S}^*$ and control $\bar{u}(k) = -\bar{L}_d(\bar{S}^*)$.
-

6.4.3 Offline implementation

In the simulation studies in Section 5.4, we compared the optimal MPC co-design to heuristic offline schedules. However, it is not always a trivial task to find an offline schedule that results in good performance and a stable NCS. In this section, we present three methods on how to find the optimal offline co-design. The offline schedules for a period length T are found by

$$\bar{S} = \arg \min_{\bar{S} \in \bar{\mathcal{S}}_a^T} \min_{T \in \{1, \dots, \bar{T}\}} \bar{F}_y(\bar{S}) \quad y \in \{\text{a}, \text{d0}, \text{dx}\}, \quad (6.63)$$

where $\{\text{a}, \text{d0}, \text{dx}\}$ are defined in Definition 6.4.1 and $\bar{T} \geq 1$ is the maximum period to optimize over. This results in three implementations that can be implemented using the offline algorithm presented in Algorithm 6.3.

Of the offline implementations in Algorithm 6.3, selecting the averaged cost a and steady-state discounted cost dx provides a MSS co-design solution. This follows by Theorem 6.3.5 and Corollary 6.4.4, respectively. On the other hand, using the discounted cost that assumes zero state (d0) does not guarantee stability.

6.5 Simulation studies

In this section, we compare the performance of the proposed scheduler and controller co-design algorithms. We first investigate the impact of the choice of the trade-off parameter β in the cost

Algorithm 6.3 Offline (y).

- The optimal schedule \bar{S}^* is found by (6.63) with y selected from the set $\{\text{a}, \text{d0}, \text{dx}\}$.
 - The optimal feedback gains are then given by $\bar{L}_y(\bar{S}^*)$.
 - At every time-step k , apply schedule $S^*(k \% T)$ from the periodic scheduling sequence \mathcal{S}^* that is obtained from $\bar{S}(k) = \bar{S}^*$ and control input $u(k)$ obtained from the periodic control input $\bar{u}(k) = -\bar{L}_y(\bar{S}^*) \bar{x}(k)$ using the mappings from Table 6.1.
-

(6.53) for the choice of final state weighting for the online MPC algorithm that is presented in Algorithm 6.1. Afterwards, the MPC algorithm with different choices of final state weighting is compared to the discounted MPC that is presented in Algorithm 6.2 and the offline algorithm presented in Algorithm 6.3. We present results for three different systems.

6.5.1 The impact of the choice of β on the final state weighting

In this section, we study the weighting parameter β in (6.53). Figures 6.1a and 6.1e show the cost and number of schedules to compare in the MPC algorithm that is presented in Algorithm 5.1. The simulation uses system (5.39), where 3 actuators share one slot in the CFP and one slot in the CAP. The parameters are as set in Section 5.4.3. We only consider a horizon length $N = 1$, since the goal is to achieve good control performance while maintaining a low number of scheduling sequences.

In Figure 6.1, small values for β mean a low penalty for the number of schedules compared to the cost. It shows, that for system (5.39), there is little to no performance improvement in the empirical cost for low β compared to high β . However, when the system is affected by an unmeasured disturbance of the form (5.42), the performance decreases slightly 0.3 dB for \bar{W}_a and 0.06 dB for \bar{W}_{dx} when β increases. These results are shown in Figure 6.1b.

As Figures 6.1a and 6.1b show, any choice of final state weighting from Section 6.4.1 outperforms Q by a significant margin of 2 dB to 3 dB when the horizon $N = 1$. However, in this case the computational cost is higher. Increasing the weighting factor β results in fewer schedules to compare. It is noteworthy, that for this system, the empirical cost does not increase when β is increased, even when the number of schedules are reduced. When the system is affected by the unmeasured disturbance (5.42), shown in Figure 6.1b, the control performance increases in some cases when β is increased. This means, that fewer scheduling sequences result in improved performance. However, the margin is very small (less than 0.2 dB). Another interesting observation is, that when the system is affected by the unmeasured disturbance, the final state weighting that is obtained by either of the discounted costs \bar{F}_{d0} and \bar{F}_{dx} in Definition 6.4.1 outperforms the final state weighting that is obtained using \bar{F}_a . In contrary, the final state weighting that is obtained using \bar{F}_a results in the best performance when the system is only affected by Gaussian noise, as shown in Figure 6.1a.

Figures 6.1c and 6.1d illustrate the performance when a hold-input strategy is used. In this setting, any choice of final state weighting based on an infinite horizon cost outperforms Q by a margin of 7 dB to 8 dB when $N = 1$. However, the final state weightings that utilize the averaged cost provide a notably lower cost than the discounted costs. The number of schedules is shown in Figure 6.1f. As with the set-to-zero case, reducing the number of schedules by increasing β does not result in any notable performance loss.

6.5.2 Comparison of the online and offline implementations

In this section, we investigate the performance and number of scheduling sequences for all algorithms presented in Sections 5.3.3 and 6.4. This includes the MPC Algorithm 5.1 with final state weighting $W = Q$, and the modified MPC Algorithm 6.1 with final state weightings a, d0 and dx, the offline Algorithm 6.3 for the cost approximations a, d0 and dx and the discounted ap-

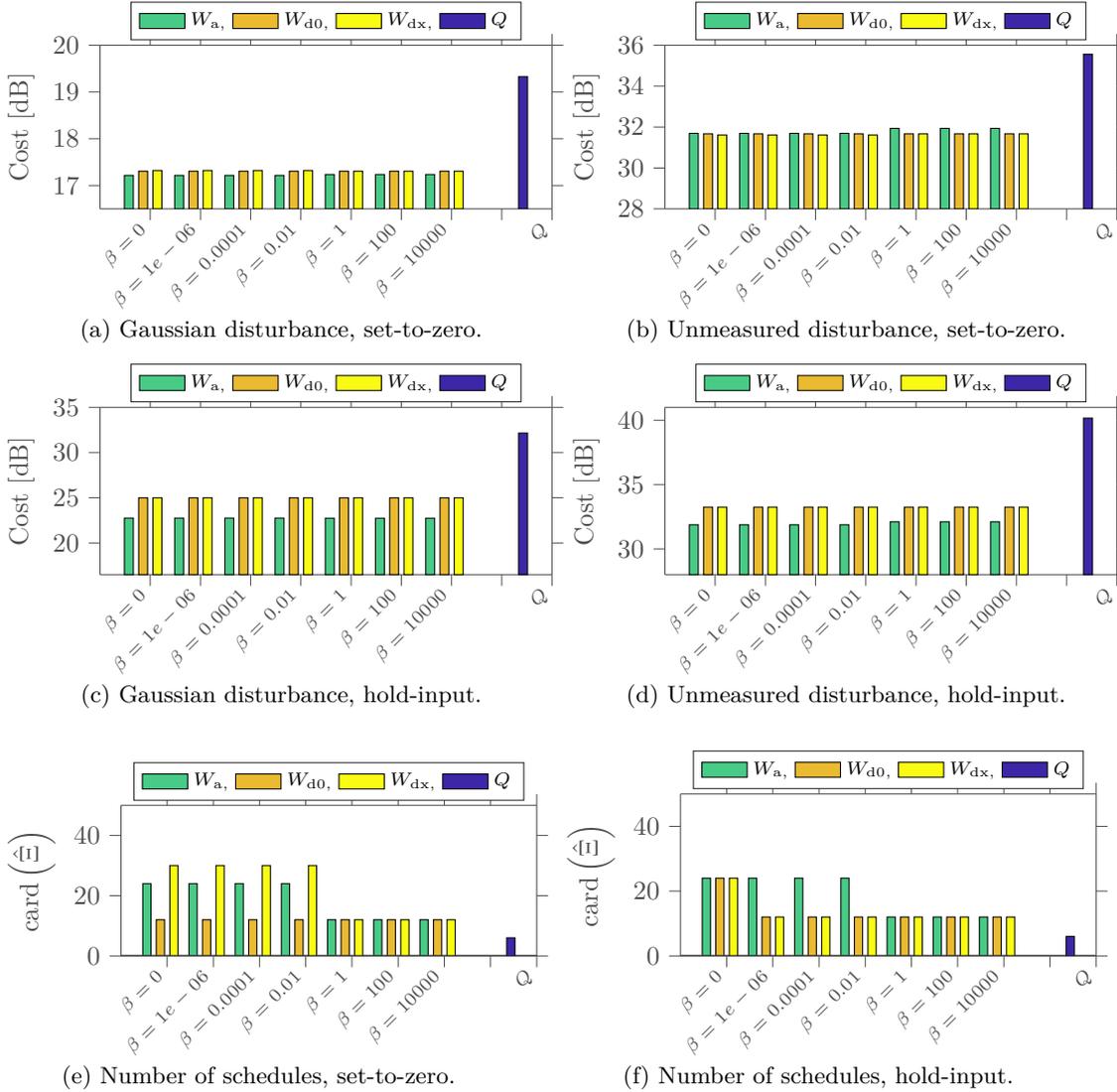


Figure 6.1: A sweep of β for system (5.39) using Algorithm 6.1. Figures 6.1a and 6.1e show the performance and number of schedules with a set-to-zero strategy at the actuators. Figures 6.1c and 6.1f show the performance and number of schedules for a hold-input strategy at the actuators. The system is affected by a Gaussian disturbance. Figures 6.1b and 6.1d show the cost when the system is affected by the unmeasured disturbance (5.42).

proach Algorithm 6.2. We investigate three different systems, with different amounts of actuators and channel capacities. All results are obtained by performing 500 simulations, each of length 20 000 time steps. The empirical cost is computed using (5.41). The final state weightings for Algorithm 5.1 as well as the scheduling sequences for the offline algorithms are selected by (6.53) with $\beta = 0$. That is, the number of scheduling sequences is not penalized. We set $\alpha = 0.9$ for the discounted Riccati equation (6.37). In the MPC Algorithm 6.2, when the final state weighting is given by W_y , the horizon length $N = 1$. Similarly, the offline algorithm is implemented using Algorithm 6.3.

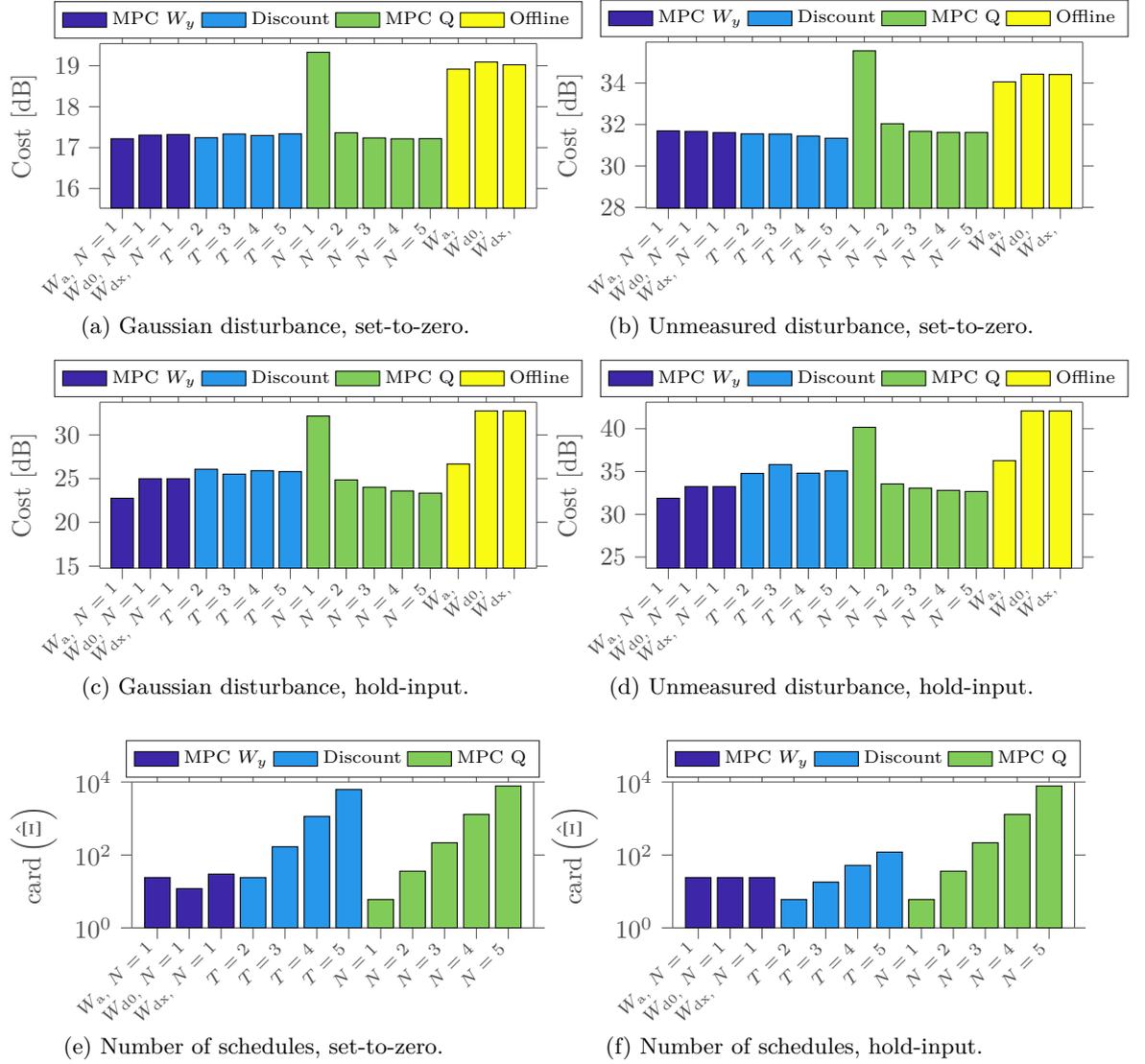


Figure 6.2: Co-design performance for system (5.39). Figures 6.2a and 6.2b utilize a set-to-zero strategy at the actuators, while Figures 6.2c and 6.2d utilize a hold-input strategy that is implemented as described in (2.4). The unmeasured disturbance is generated using (5.42). Figures 6.2e and 6.2f show the number of schedules to compare at each time-step.

6.5.2.1 Small system from Section 5.4.3

For this simulation, we consider the system (5.39) from Section 5.4.3. The results are shown in Figure 6.2. Figures 6.2a and 6.2b show the performance for the system with set-to-zero at the actuators. Here Figure 6.2a shows the cost when Gaussian white noise is present while Figure 6.2b shows the cost when the system is affected by the random disturbance (5.42). The results show, that when there is only Gaussian noise, all online algorithms besides MPC with final state weighting Q and $N = 1$, show performance gains of 2 dB to 3 dB over the offline implementations. Interestingly, when the unmeasured disturbance is present, the performance of the offline algorithms

is also in the range of 3 dB to 4 dB lower than the best online implementations. For this system the discounted MPC (Algorithm 6.2) performs on par with the MPC with final state weighting (Algorithms 5.1 and 6.1). Here, the period length does not affect the performance significantly. Figure 6.2e shows the number of scheduling sequences that have to be compared in the MPC algorithms. It shows, that the computational complexity of the discounted MPC with $T = 2$ is on par with the MPC algorithms that use W_y as final state weighting.

However, when the input is held at the actuators, which is shown in Figures 6.2c and 6.2d, the performance of the online algorithms increases by 8 dB to 10 dB over the offline algorithms. Also, selecting $y = a$ in both the MPC Algorithm 6.1 and the offline Algorithm 6.3 results in significantly improved performance compared to selecting $y \in \{d0, dx\}$. On the other hand, the discounted MPC (Algorithm 6.2) results in reduced performance compared to the MPC Algorithms 5.1 and 6.1 besides the case where $N = 1$ and the weighting Q is used. Interestingly, as Figure 6.2f shows, the discounted MPC Algorithm 6.2 requires fewer scheduling sequences than the other MPC algorithms. This is caused by ill-conditioned matrices that affect the solution to the discounted Riccati equation (6.37). Due to this, there exists solutions to the discounted Riccati equation (6.37) for fewer scheduling sequences than when set-to-zero is used.

6.5.2.2 Larger system from Section 5.4.4

In this section, we show the results of Monte-Carlo simulations for system (5.43) and (5.44) in Section 6.5.2.2. This system consists of three subsystems, of which one system has two actuators and the others each have one. The controller has access to one slot in the CFP and one in the CAP. This means that at each time-step, two out of four actuators are addressed.

Figures 6.3a and 6.3b show the performance of the algorithms with Gaussian noise and with an unmeasured disturbance, respectively when set-to-zero is used at the actuators. For this system, the discounted MPC results in a cost which is approximately 1 dB higher than the best MPC algorithms. Compared to the studies in Section 6.5.2.1, using Q as final state weighting with horizon length $N = 1$ results in improved performance over the offline algorithms. Also, the online algorithms results in a cost that is 3 dB to 6 dB lower than the offline algorithms. Figure 6.3e shows the number of scheduling sequences to compare in the online part of the algorithms. Here the MPC in Algorithm 5.1 with $N = 1$ and $W = Q$ searches among the least amount of scheduling sequences while resulting in similar performance as the discounted MPC in Algorithm 6.2 with $T = 2$. However, the MPC Algorithm 6.1 with final state weighting, results in the best performance while maintaining the number of schedules to compare at each time-step below 100. Among the choices of final state weighting, $y = d0$ yielded the best performance while also requiring the lowest number of scheduling sequences, as shown in Figure 6.3e.

Numerical issues prevented solutions to the discounted Riccati equation (6.37) when a hold-input strategy is used at the actuators. Due to this, no results were obtained for Algorithm 6.2 and Algorithms 6.1 and 6.3 when $y \in \{d0, dx\}$. The obtained results are shown in Figures 6.3c and 6.3d. In both simulations, the MPC Algorithm 5.1 with final state weighting $W = Q$ and horizon length $N = 1$ resulted in a cost larger than 500 dB and are truncated for easier viewing. Also here, the online algorithms outperform the offline algorithm by a margin of 6 dB to 8 dB. Considering the number of scheduling sequences needed in the online algorithms, which is shown

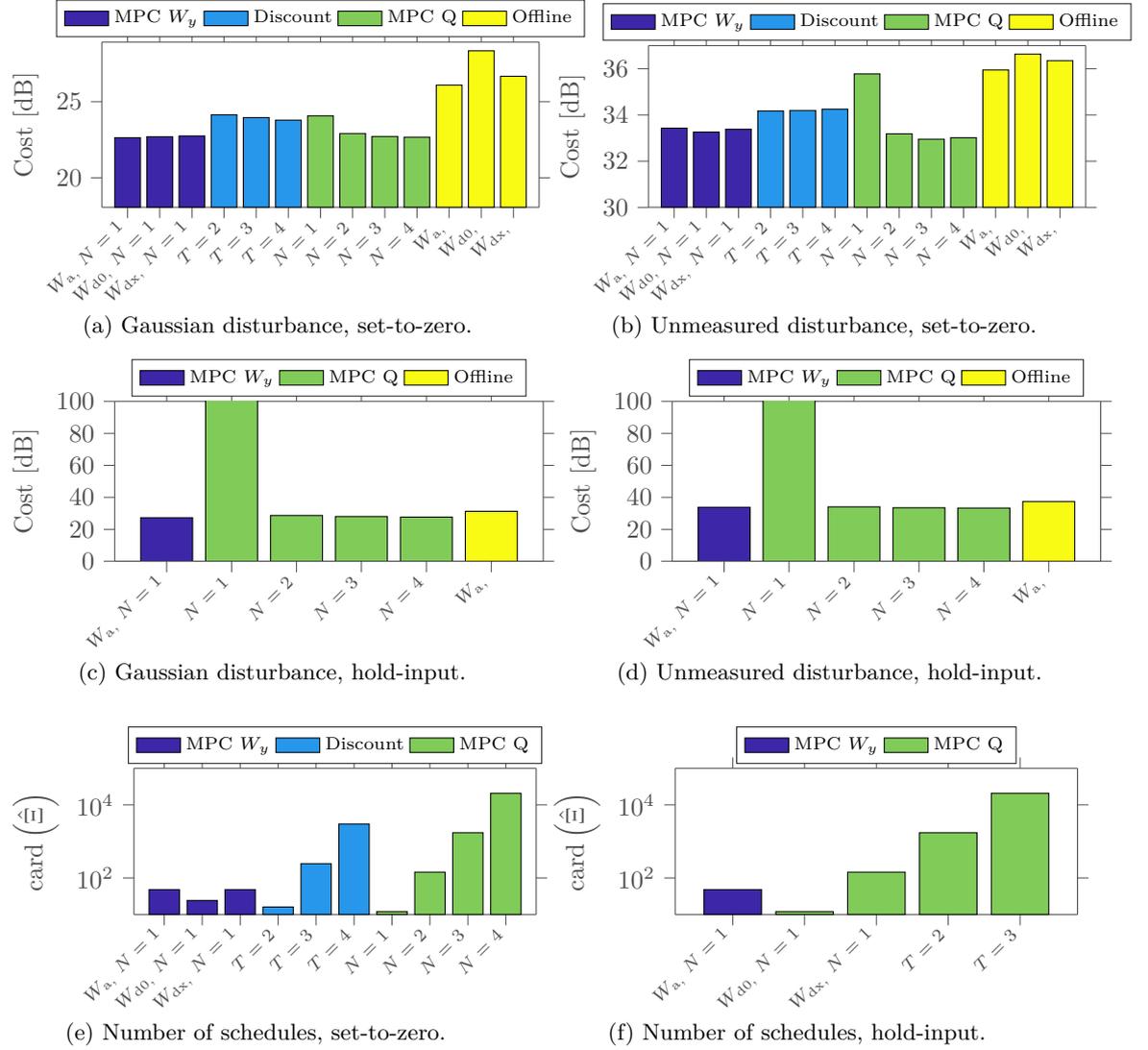


Figure 6.3: Co-design performance for system (5.43) and (5.44) affected by a Gaussian disturbance (Figures 6.3a and 6.3c) and the unmeasured disturbance (5.42) (Figures 6.3b and 6.3d) for both set-to-zero and hold-input actuator policies. In Figures 6.3c and 6.3d, the MPC algorithm with $N = 1$ and Q as final state weighting resulted in a cost of over 500 dB and is truncated for easier viewing. Figures 6.3e and 6.3f show the number of schedules that have to be compared at every time-step.

in Figure 6.3f, Algorithm 6.1 with $y = a$ results in the best performance while requiring the least amount of scheduling sequences.

6.5.2.3 Large scale NCS

In this section we simulate a large scale system with six actuators that share one slot in the CFP and two slots in the CAP. The model is given by

$$A = \text{diag} \{A_1, A_2, A_3, A_4, A_5\}, \quad B = \text{diag} \{B_1, B_2, B_3, B_4, B_5\} \quad (6.64)$$

and

$$C = \text{diag} \{C_1, C_2, C_3, C_4, C_5\}.$$

The subsystems A_i, B_i are given by

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ -0.333 & 0 & 0.667 \\ 1.22 & 0 & -1.22 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0.333 \\ 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.72 & 0.72 \\ 0.889 & -2.4 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 5.09 & 0 \\ 0 & 6.29 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0.54 & 0.841 \\ -0.841 & 0.54 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0.46 \\ 0.841 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1.1 & 0 \\ 1 & 1.1 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} 0.681 & 0.319 & 0.888 & 0.112 \\ 0.319 & 0.681 & 0.112 & 0.888 \\ -0.559 & 0.559 & 0.681 & 0.319 \\ 0.559 & -0.559 & 0.319 & 0.681 \end{bmatrix} \quad \text{and} \quad B_5 = \begin{bmatrix} 0.471 \\ 0.0286 \\ 0.888 \\ 0.112 \end{bmatrix}.$$

In (6.64) the output matrices are given by $C_i = I$ for $i = 2, \dots, 5$ and

$$C_1 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

The matrix $F = I$ is of appropriate dimensions. We set $Q = C^T C$ and $R = I$.

This system results in a large number of scheduling sequences as shown in Figures 6.3e and 6.3f. When $N = 1$, the MPC algorithm with final state weighting Q has to compare 60 scheduling sequences at each time-step. When $N = 2$, this number increases to 3600 and for $N = 3$ this would be 21 600 scheduling sequences. This makes it prohibitive to perform the optimal co-design for long horizon lengths, and thereby obtaining a low cost with the MPC algorithm with final state weighting Q . However, as the results in Figures 6.3a and 6.3b show, significant performance gains (19 dB) can be obtained by using a suitable selection of the final state weighting. This also holds true when a hold-input strategy is used, which is illustrated in Figures 6.3c and 6.3d. The offline implementations result in a cost of 3 dB more than the best performing online cost. Also, it depends on the scenario which choice of final state weighting from Section 6.4.1 results in the best performance. In Figure 6.3b, when the unmeasured disturbance (5.42) is present, the choice of final state weighting W_{dx} , results in a 2 dB lower cost than W_a .

However, when a hold-input strategy is used, numerical stability prevents computation of the final state weighting W_{dx} . Also, the MPC algorithm with final state weighting Q was unable to maintain stability. The performance of the remaining algorithms is shown in Figures 6.4c and 6.4d. In this setting, the MPC algorithms reduce the cost by 5 dB compared to the optimal offline implementations. The discounted MPC algorithm results in a higher cost than the best choice of offline algorithm. Since this number includes cyclic shifts of the scheduling sequences, there were merely 21 unique scheduling sequences that the discounted algorithm could utilize. The main reason for this is also the numerical instability that affects the inversion of the large matrices when iteratively solving the discounted Riccati equation (6.37).

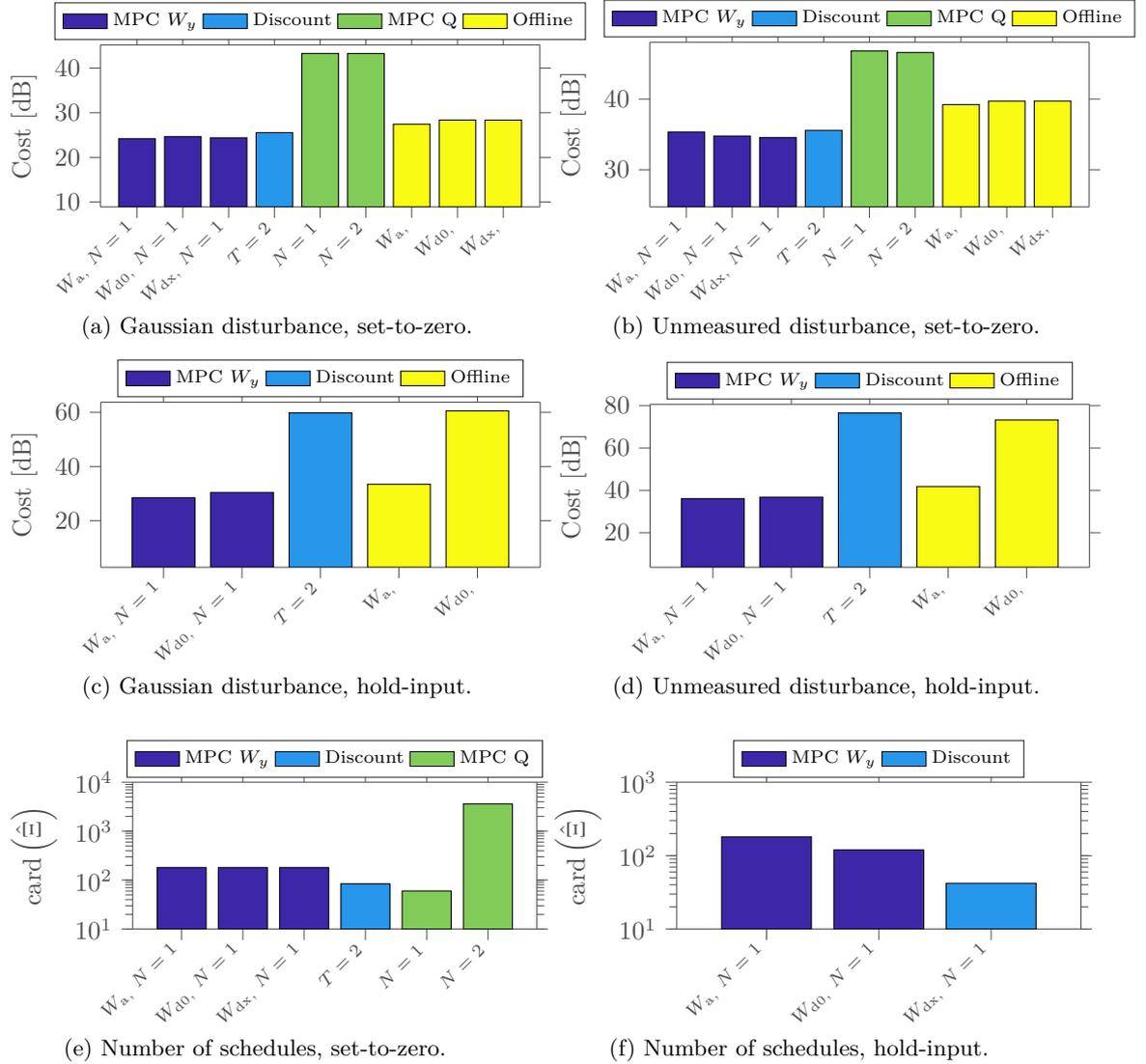


Figure 6.4: Co-design performance for system (6.64) affected by Gaussian noise (Figures 6.4a and 6.4c) and the unmeasured disturbance (5.42) (Figures 6.4b and 6.4d) with both set-to-zero and hold-input actuator strategies. Figures 6.4e and 6.4f show the number of schedules that have to be compared at every time-step.

6.5.2.4 Summary of the simulation results

We have illustrated that a carefully selected final state weighting for the MPC Algorithm 5.1, which is implemented in the modified MPC Algorithm 6.1, indeed results in significant performance gains for a relatively low computational cost. Further, it is using these carefully selected final state weightings, possible to obtain good performance on large scale systems, that else could be prohibitive to design and implement an optimal co-design for.

It however largely depends on the system, actuator strategy and disturbances which choice of infinite horizon approximation in Section 6.4.1 results in the best performance. However, the

performance deviation between the different approximations tends to be within 1.5 dB.

The offline co-designs that were obtained in Section 6.4.3 using the infinite horizon approximations resulted in solid performance. In some cases these outperformed MPC algorithms with short horizons and poorly selected final state weightings. Further, compared to the heuristic approach that we utilized in Section 5.4, the proposed design in Algorithm 6.3 makes it possible to find an optimal offline scheduling sequence. Among the offline implementations, the implementation that uses W_a , that is obtained by the averaged cost function (6.48) results in the best performance.

6.6 Summary

In this chapter, we presented a periodic formulation of the scheduling problem presented in Chapter 5. Using this formulation, the schedules could be treated as time-invariant. After revising the finite horizon solution from Section 5.3 in the time-invariant framework, we presented two methods to approximate an infinite horizon cost function. These approximations utilized the assumption, that a scheduling sequence would be repeated periodically with a fixed period length. These infinite horizon approximations were utilized to select the final state weighting for the online MPC scheduling algorithm, which for short horizon lengths resulted in significant performance gains compared to an arbitrarily selected final state weighting. This meant, that the online computational cost can be reduced significantly compared to the MPC implementation presented in Section 5.4 while providing similar performance. Further, these infinite horizon approximations were utilized to obtain offline scheduling sequences that can outperform the heuristic offline scheduling approach in Section 5.4 by a significant margin.

We have shown stability for two of the three offline scheduling algorithms. Further we have shown stability for the MPC algorithm when the final state weighting is selected to be the solution to the Riccati equation that is obtained from the averaged infinite horizon cost approximation.

Appendix

6.A Proof of Theorem 6.2.1

For a known schedule $\bar{S} \in \bar{S}^T$, the optimal control law for the cost function (6.21) with horizon length NT is given by

$$\bar{U}_{NT}^* \triangleq \arg \min_{\bar{U}_{NT}} \bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}).$$

The optimal cost for the schedule \bar{S} is given by

$$\begin{aligned} \bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) &= \min_{\bar{U}_{NT}} \bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}) \\ &= \min_{\bar{U}_{NT}} \mathbf{E} \left\{ \|\bar{x}(NT)\|_W^2 + \sum_{\ell=0}^{NT-1} \|\bar{x}(\ell)\|_Q^2 + \|\bar{u}(\ell)\|_{R(\bar{\gamma}(\ell))}^2 \middle| \bar{x}(0); \bar{S} \right\}. \end{aligned}$$

This can for $n = 1, 2, \dots, NT$ be stated as

$$\bar{J}_W(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \min_{\bar{U}_{0,n}} \mathbf{E} \left\{ \sum_{\ell=0}^{n-1} \|\bar{x}(\ell)\|_Q^2 + \|\bar{u}(\ell)\|_{R(\bar{\gamma}(\ell))}^2 + \bar{J}_W(\bar{x}(n), \bar{S}, \bar{U}_{n,NT}^*) \middle| \bar{x}(0); \bar{S} \right\}.$$

Thus, the above can be formulated stage-wise for $k = 0, 1, \dots, NT - 1$ as

$$\begin{aligned} \bar{J}_W(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) = \\ \min_{\bar{u}(k)} \mathbf{E} \left\{ \|\bar{x}(k)\|_{\bar{Q}}^2 + \|\bar{u}(k)\|_{\bar{R}(\bar{\gamma}(k))}^2 + \bar{J}_W(\bar{x}(k+1), \bar{S}, \bar{U}_{k+1,NT}^*) \middle| \bar{x}(k); \bar{S} \right\}, \end{aligned} \quad (6.A.1)$$

where

$$\begin{aligned} \bar{J}_W(\bar{x}(NT), \bar{S}_N) &= \mathbf{E} \left\{ \|\bar{x}(NT)\|_{\bar{W}}^2 \middle| \bar{x}(NT); \bar{S} \right\} \\ &= \bar{x}^T(NT) \bar{W} \bar{x}(NT). \end{aligned}$$

The proof now follow by dynamic programming [15]. If at time $k+1$ the optimal cost for a schedule \bar{S} has the form

$$\bar{J}_W(\bar{x}(k+1), \bar{S}, \bar{U}_{k+1,NT}^*) = \bar{x}^T(k+1) \bar{P}_{k+1} \bar{x}(k+1) + c_{k+1},$$

then, at time k , writing out (6.A.1) leads to

$$\begin{aligned} \bar{J}_W(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) = \min_{\bar{u}(k)} \mathbf{E} \left\{ \bar{x}^T(k) \bar{Q} \bar{x}(k) + \bar{u}^T(k) \bar{R}(\bar{\gamma}(k)) \bar{u}(k) \right. \\ \left. \bar{x}^T(k+1) \bar{P}_{k+1} \bar{x}(k+1) + c_{k+1} \middle| \bar{x}(k); \bar{S} \right\}. \end{aligned}$$

Substituting the recursion (6.11) yields

$$\begin{aligned} \bar{J}_W(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) &= \min_{\bar{u}(k)} \mathbf{E} \left\{ \bar{x}^T(k) \bar{Q} \bar{x}(k) + \bar{u}^T(k) \bar{R}(\bar{\gamma}(k)) \bar{u}(k) \right. \\ &\quad \left. (\bar{A} \bar{x}(k) + \bar{B}(\bar{\gamma}(k)) \bar{u}(k) + \bar{F} \bar{\omega}(k))^T \bar{P}_{k+1} (\bar{A} \bar{x}(k) + \bar{B}(\bar{\gamma}(k)) \bar{u}(k) + \bar{F} \bar{\omega}(k)) + c_{k+1} \middle| \bar{x}(k); \bar{S} \right\} \\ &= \min_{\bar{u}(k)} \mathbf{E} \left\{ \bar{x}^T(k) (\bar{Q} + \bar{A}^T \bar{P}_{k+1} \bar{A}) \bar{x}(k) + \bar{u}^T(k) [\bar{R}(\bar{\gamma}(k)) + \bar{B}^T(\bar{\gamma}(k)) \bar{P}_{k+1} \bar{B}(\bar{\gamma}(k))] \bar{u}(k) \right. \\ &\quad \left. + 2\bar{u}^T(k) \bar{B}^T(\bar{\gamma}(k)) \bar{P}_{k+1} \bar{A} \bar{x}(k) + \bar{\omega}^T(k) \bar{F}^T \bar{P}_{k+1} \bar{F} \bar{\omega}(k) + c_{k+1} \middle| \bar{x}(k); \bar{S} \right\}. \end{aligned}$$

Distributing the expectation yields

$$\begin{aligned} \bar{J}_W(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) &= \min_{\bar{u}(k)} \bar{x}^T(k) (\bar{Q} + \bar{A}^T \bar{P}_{k+1} \bar{A}) \bar{x}(k) \\ &\quad + \bar{u}^T(k) \left[\mathbf{E} \{ \bar{R}(\bar{\gamma}(k)); \bar{S} \} + \mathbf{E} \{ \bar{B}^T(\bar{\gamma}(k)) \bar{P}_{k+1} \bar{B}(\bar{\gamma}(k)); \bar{S} \} \right] \bar{u}(k) \\ &\quad + 2\bar{u}^T(k) \mathbf{E} \{ \bar{B}^T(\bar{\gamma}(k)); \bar{S} \} \bar{P}_{k+1} \bar{A} \bar{x}(k) + \text{trace}(\bar{\Sigma}_{\bar{\omega}}(k) \bar{F}^T \bar{P}_{k+1} \bar{F}) + c_{k+1} \end{aligned}$$

with $\bar{\Sigma}_{\bar{\omega}}(k)$ defined in (6.12). Computing the expectations as in the proof in Appendix 5.B and substitution in the above yields

$$\begin{aligned} \bar{J}_W(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) &= \bar{x}^T(k) (\bar{Q} + \bar{A}^T \bar{P}_{k+1} \bar{A}) \bar{x}(k) + \text{trace}(\bar{\Sigma}_{\bar{\omega}}(k) \bar{F}^T \bar{P}_{k+1} \bar{F}) + c_{k+1} \\ &\quad + \min_{\bar{u}(k)} \bar{u}^T(k) \left[\bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S}) \right] \bar{u}(k) + 2\bar{u}^T(k) \bar{B}^T(\bar{S}) \bar{P}_{k+1} \bar{A} \bar{x}(k). \end{aligned}$$

The proof now proceeds from (5.B.4) in Appendix 5.B. □

6.B Proof of Lemma 6.3.1

For a schedule $\bar{S} \in \bar{S}^T$ the minimized cost (6.31) can for $n = 1, 2, \dots, NT - 1$ be stated as

$$\bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \min_{\bar{U}_{0,n}} \mathbf{E} \left\{ \sum_{\ell=0}^{n-1} \alpha^\ell \left[\|\bar{x}(\ell)\|_{\bar{Q}}^2 + \|\bar{u}(\ell)\|_{\bar{R}(\bar{\gamma}(\ell))}^2 \right] + \alpha^n \bar{J}_d(\bar{x}(n), \bar{S}, \bar{U}_{n,NT}^*) \middle| \bar{x}(0); \bar{S} \right\}.$$

This yields the cost at each stage

$$\bar{J}_d(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) = \min_{\bar{u}(k)} \mathbf{E} \left\{ \|\bar{x}(k)\|_{\bar{Q}}^2 + \|\bar{u}(k)\|_{\bar{R}(\bar{\gamma}(k))}^2 + \alpha \bar{J}_d(\bar{x}(k+1), \bar{S}, \bar{U}_{k+1,NT}^*) \middle| \bar{x}(k); \bar{S} \right\}.$$

The result then follows directly by proceeding as in Appendix 6.A. \square

6.C Proof of Theorem 6.3.2

The proof follows standard arguments for infinite horizon optimal stochastic control [15], [49]. Since $\lim_{k \rightarrow \infty} \bar{P}_{-k} \triangleq \bar{P}$ exists, it follows using the arguments in the proof of Lemma 3.3.2, that \bar{P} is the solution to the discounted Riccati equation (6.37). Now computing

$$\lim_{N \rightarrow \infty} \bar{J}_d(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \lim_{N \rightarrow \infty} \bar{x}^T(0) \bar{P}_0 \bar{x}(0) + \lim_{N \rightarrow \infty} \sum_{\ell=1}^{NT} \alpha^\ell \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F} \bar{P}_\ell \bar{F}^T),$$

where,

$$\lim_{N \rightarrow \infty} \bar{x}^T(0) \bar{P}_0 \bar{x}(0) = \bar{x}^T(0) \bar{P} \bar{x}(0) \quad (6.C.1)$$

The term

$$\lim_{N \rightarrow \infty} \sum_{\ell=1}^{NT} \alpha^\ell \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F} \bar{P}_\ell \bar{F}^T) = \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^{kT} \sum_{\ell=1}^T \alpha^\ell \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell-1) \bar{F} \bar{P}_\ell \bar{F}^T)$$

define

$$c \triangleq \sum_{\ell=1}^T \alpha^\ell \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell-1) \bar{F} \bar{P}_\ell \bar{F}^T) \quad (6.C.2)$$

the above leads to the geometric sum which, if $|a| < 1$ has the solution

$$\lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} \alpha^{kT} c = \frac{1}{1 - \alpha^T} c. \quad (6.C.3)$$

Combining (6.C.1) to (6.C.3) then leads to the cost (6.38), where \bar{P} is the solution to (6.37).

6.D Proof of Lemma 6.3.3

For a schedule $\bar{S} \in \bar{S}^T$, the averaged cost (6.42) can for $n = 1, 2, \dots, NT - 1$ be stated as

$$\bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_{NT}^*) = \frac{1}{NT} \min_{\bar{U}_{0,n}} \mathbf{E} \left\{ \sum_{\ell=0}^{n-1} \|\bar{x}(\ell)\|_{\bar{Q}}^2 + \|\bar{u}(\ell)\|_{\bar{R}(\bar{\gamma}(\ell))}^2 + (NT - n) \bar{J}_a(\bar{x}(n), \bar{S}, \bar{U}_{n,NT}^*) \middle| \bar{x}(0); \bar{S} \right\}.$$

At each stage $k = 0, 1, \dots, NT - 1$ this yields

$$\bar{J}_a(\bar{x}(k), \bar{S}, \bar{U}_{k,NT}^*) = \frac{1}{NT-k} \min_{\bar{u}(k)} \mathbf{E} \left\{ \|\bar{x}(\ell)\|_{\bar{Q}}^2 + \|\bar{u}(\ell)\|_{\bar{R}(\bar{\gamma}(\ell))}^2 + \right. \\ \left. (NT-k-1) \bar{J}_a(\bar{x}(k+1), \bar{S}, \bar{U}_{k+1,NT}^*) \mid \bar{x}(k); \bar{S} \right\},$$

where

$$\bar{J}_a(\bar{x}(NT), \bar{S}) = \mathbf{E} \left\{ \|\bar{x}(NT)\|_{\bar{Q}}^2 \mid \bar{x}(NT); \bar{S} \right\}.$$

The result is then obtained by proceeding as in Appendix 6.A. \square

6.E Proof of Theorem 6.3.4

Since $\bar{P} = \lim_{N \rightarrow \infty} \bar{P}_N$ exists for the schedule $\bar{S} \in \bar{\mathcal{S}}^T$, the cost (6.44) using (6.43) becomes

$$\bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_\infty) = \\ \lim_{N \rightarrow \infty} \frac{1}{NT} \bar{x}^T(0) \bar{P}_{NT} \bar{x}(0) + \lim_{N \rightarrow \infty} \frac{1}{NT} \sum_{\ell=0}^{NT-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P}_{\ell+1} \bar{F}). \quad (6.E.1)$$

Since \bar{P} is finite, the first term on the right-hand-side of the above vanishes. For the second term, define the linear map

$$\xi(n) \triangleq \frac{1}{T} \sum_{\ell=0}^{T-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P}_{nT+\ell+1} \bar{F}), \quad (6.E.2)$$

such that

$$\xi(N) = \frac{1}{T} \sum_{\ell=0}^{T-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P}_{NT+\ell+1} \bar{F}).$$

It is easy to verify by (3.26) and (3.27) and Lemma 3.C.1 that \bar{P}_N in (6.26) is monotone. Since also $\bar{P} = \lim_{N \rightarrow \infty} \bar{P}_N$ exists, we have on the limit that

$$\begin{aligned} \lim_{n \rightarrow \infty} \xi(n) &= \lim_{n \rightarrow \infty} \frac{1}{T} \sum_{\ell=0}^{T-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P}_{nT+\ell+1} \bar{F}) \\ &= \frac{1}{T} \sum_{\ell=0}^{T-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P} \bar{F}) \\ &= \frac{1}{T} \text{trace}(\bar{\Sigma}_{\bar{\omega}} \bar{F}^T \bar{P} \bar{F}) \\ &\triangleq \bar{\xi}, \end{aligned}$$

where $\bar{\Sigma}_{\bar{\omega}}$ is defined in (6.13).

Going back to (6.E.1), substituting (6.E.2) yields

$$\begin{aligned} \bar{J}_a(\bar{x}(0), \bar{S}, \bar{U}_\infty) &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \frac{1}{T} \sum_{\ell=0}^{T-1} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell) \bar{F}^T \bar{P}_{nT+\ell+1} \bar{F}) \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \xi(n). \end{aligned}$$

Since \bar{P}_N is monotone and (6.E.2) is a linear map, $\xi(N)$ is also monotone. Therefore

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \xi(n) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \bar{\xi} = \bar{\xi}$$

and the result follows. \square

6.F Proof of Theorem 6.3.5

Before showing the proof of Theorem 6.3.5, we introduce the following definition and lemma:

For a schedule $\bar{S} \in \bar{\mathcal{S}}^T$, define the matrix

$$\mathcal{A}_{\bar{L}_k^*} \triangleq \mathbf{E} \left\{ \bar{H}_{\bar{L}_k^*}(\bar{\gamma}) \otimes \bar{H}_{\bar{L}_k^*}(\bar{\gamma}) ; \bar{S} \right\}. \quad (6.F.1)$$

Then, we have the following result.

Lemma 6.F.1. *For a schedule $\bar{S} \in \bar{\mathcal{S}}^T$, the Riccati equation (6.26) can be stated as*

$$\text{vec}(\bar{P}_k) = \mathcal{A}_{\bar{L}_k^*}^T(\bar{S}) \text{vec}(\bar{P}_{k+1}) + \text{vec} \left(\bar{Q} + (\bar{L}_k^*(\bar{S}))^T \bar{R}(\bar{S}) \bar{L}_k^*(\bar{S}) \right).$$

Proof. The proof follows the same procedure as the proof of Lemma 3.3.2 in Appendix 3.C, i.e.

$$\bar{P}_k = \bar{Q} + \bar{A}^T \bar{P}_{k+1} \bar{A} - \bar{A}^T \bar{P}_{k+1} \bar{B}(\bar{S}) [\bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S})]^\dagger \bar{B}(\bar{S})^T \bar{P}_{k+1} \bar{A}$$

Using (6.24) and multiplying and dividing by $[\bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S})]$ yields

$$\bar{P}_k = \bar{Q} + \bar{A}^T \bar{P}_{k+1} \bar{A} - (\bar{L}_k^*(\bar{S}))^T [\bar{R}(\bar{S}) + M(\bar{P}_{k+1}, \bar{S})] \bar{L}_k^*(\bar{S}). \quad (6.F.2)$$

Using (3.C.5) and (3.C.6) in Appendix 3.C, yields

$$\begin{aligned} \bar{A}^T \bar{P}_{k+1} \bar{A} = \mathbf{E} \left\{ (\bar{A} - \bar{B}(\bar{\gamma}) \bar{L}_k^*(\bar{S}))^T \bar{P}_{k+1} (\bar{A} - \bar{B}(\bar{\gamma}) \bar{L}_k^*(\bar{S})) ; \bar{S} \right\} \\ + (\bar{L}_k^*(\bar{S}))^T M(\bar{P}_{k+1}, \bar{S}) \bar{L}_k^*(\bar{S}) + 2 (\bar{L}_k^*(\bar{S}))^T \bar{R}(\bar{S}) \bar{L}_k^*(\bar{S}). \end{aligned} \quad (6.F.3)$$

Combining (6.F.2) and (6.F.3) then results in

$$\begin{aligned} \bar{P}_k &= \mathbf{E} \left\{ (\bar{A} - \bar{B}(\bar{\gamma}) \bar{L}_k^*(\bar{S}))^T \bar{P}_{k+1} (\bar{A} - \bar{B}(\bar{\gamma}) \bar{L}_k^*(\bar{S})) ; \bar{S} \right\} + \bar{Q} + (\bar{L}_k^*(\bar{S}))^T \bar{R}(\bar{S}) \bar{L}_k^*(\bar{S}) \\ &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_k^*}^T(\bar{S})(\bar{\gamma}) \bar{P}_{k+1} \bar{H}_{\bar{L}_k^*}(\bar{S})(\bar{\gamma}) ; \bar{S} \right\} + \bar{Q} + (\bar{L}_k^*(\bar{S}))^T \bar{R}(\bar{S}) \bar{L}_k^*(\bar{S}). \end{aligned}$$

This can be vectorized as

$$\text{vec}(\bar{P}_k) = \mathbf{E} \left\{ \bar{H}_{\bar{L}_k^*}^T(\bar{S}) \otimes \bar{H}_{\bar{L}_k^*}(\bar{S}) ; \bar{S} \right\} \text{vec}(\bar{P}_{k+1}) + \text{vec} \left(\bar{Q} + (\bar{L}_k^*(\bar{S}))^T \bar{R}(\bar{S}) \bar{L}_k^*(\bar{S}) \right).$$

Then, the result follows by using (6.F.1). \square

We now present the proof of Theorem 6.3.5.

Proof of Theorem 6.3.5. Define the state covariance for a schedule $\bar{S} \in \bar{\mathcal{S}}^T$ by

$$\mathcal{X}_{\bar{S}}(k) \triangleq \mathbf{E} \left\{ \bar{x}(k) \bar{x}^T(k) ; \bar{S} \right\},$$

where $\mathcal{X}_{\bar{S}}(0) \triangleq \mathbf{E} \{ \bar{x}(0) \bar{x}^T(0) ; \bar{S} \}$ is known and the state evolution is given by (6.18).

Evaluating this expectation yields

$$\begin{aligned} \mathcal{X}_{\bar{S}}(k+1) &= \mathbf{E} \left\{ \left(\bar{H}_{\bar{L}_a(\bar{S})}(\bar{\gamma}) \bar{x}(k) + \bar{F} \bar{\omega}(k) \right) \left(\bar{H}_{\bar{L}_a(\bar{S})}(\bar{\gamma}) \bar{x}(k) + \bar{F} \bar{\omega}(k) \right)^T ; \bar{S} \right\} \\ &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_a(\bar{S})}(\bar{\gamma}) \bar{x}(k) \bar{x}^T(k) \bar{H}_{\bar{L}_a(\bar{S})}^T(\bar{\gamma}) + \bar{F} \bar{\omega}(k) \bar{\omega}^T(k) \bar{F}^T ; \bar{S} \right\} \\ &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_a(\bar{S})}(\bar{\gamma}) \mathbf{E} \{ \bar{x}(k) \bar{x}^T | \bar{S} \} (k) \bar{H}_{\bar{L}_a(\bar{S})}^T(\bar{\gamma}) ; \bar{S} \right\} + \mathbf{E} \{ \bar{F} \bar{\omega}(k) \bar{\omega}^T(k) \bar{F}^T ; \bar{S} \} \\ &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_a(\bar{S})}(\bar{\gamma}) \mathcal{X}_{\bar{S}}(k) \bar{H}_{\bar{L}_a(\bar{S})}^T(\bar{\gamma}) \middle| \bar{S} \right\} + \bar{F} \bar{\Sigma}_{\bar{\omega}}(k) \bar{F}^T, \end{aligned}$$

where we used the fact that $\bar{\omega}(k)$ is independent and identically distributed (*i.i.d.*). Using properties of the Kronecker product, the above can be vectorized as

$$\text{vec}(\mathcal{X}_{\bar{S}}(k+1)) = \mathcal{A}_{\bar{L}_a(\bar{S})} \text{vec}(\mathcal{X}_{\bar{S}}(k)) + \text{vec}(\bar{F} \bar{\Sigma}_{\bar{\omega}}(k) \bar{F}^T), \quad (6.F.4)$$

where $\mathcal{A}_{\bar{L}_a(\bar{S})}$ is defined in (6.F.1). A solution to the linear equation (6.F.4) exists if and only if all eigenvalues of $\mathcal{A}_{\bar{L}_a(\bar{S})}$ are within the unit circle.

Now, easy to see that P_{k+1} in Lemma 6.F.1 is linear in P_k . Then, by Lemma 3.C.1 in Appendix 3.C, it follows directly, that

$$\lim_{k \rightarrow \infty} P_k = \bar{P},$$

where

$$\text{vec}(\bar{P}) = \mathcal{A}_{\bar{L}_a(\bar{S})}^T \text{vec}(\bar{P}) + \text{vec}(\bar{Q} + \bar{L}_a^T(\bar{S}) \bar{R}(\bar{S}) \bar{L}_a(\bar{S})). \quad (6.F.5)$$

Now since $\mathcal{A}_{\bar{L}_a(\bar{S})}$ is identical in (6.F.4) and (6.F.5)

$$\lim_{k \rightarrow \infty} \|\text{vec}(\mathcal{X}_{\bar{S}}(k))\| < \infty$$

if and only if there exists a solution to (6.46), and thereby also (6.F.5) exists, which proves the theorem. \square

6.G Proof of Theorem 6.4.1

For this proof, consider the system (6.1a) without the additive disturbance $\omega(k)$. The reason for this is, that when the state $x(k)$ becomes sufficiently large, the contribution of the noise $\omega(k)$ is insignificant. Thus, stochastic stability results for the system (6.1a) without the disturbance $\omega(k)$ using the MPC in Algorithm 6.1 extends to systems that are affected by disturbance $\omega(k)$ with bounded covariance. The control laws for the MPC algorithm are given by feedback laws of the form $u(k) = -L^*(S(k))x(k)$. This results in the system

$$\begin{aligned} x(k+1) &= Ax(k) - B(\gamma(k))L^*(S(k))x(k) \\ &= H_{L(S(k))}(\gamma(k))x(k). \end{aligned} \quad (6.G.1)$$

The control law $L^*(S(k))$ is the optimal control law for schedule $S(k)$, that is obtained in Theorem 6.2.1. In the below, the argument for $H_{L(S(k))}$ is omitted for easier notation.

Define, for any horizon length $N \in \mathbb{N}$ the value function

$$V(x(k), \mathcal{S}_{k, k+N-1}, i) \triangleq \mathbf{E} \left\{ \sum_{\ell=0}^{N-1} \|x(k+\ell)\|_Q^2 + \|L(S(k+\ell))x(k+\ell)\|_{R\{\gamma(k+\ell)\}}^2 + \|x(k+N)\|_{P_{i+N}}^2 \mid x(k); \mathcal{S}_{k, k+N-1} \right\}, \quad (6.G.2)$$

where $P_{i+N} \triangleq P((i+N) \% T)$ is the $(i+N) \% T$ 'th shift of the solution to the periodic Riccati equation (6.46) for a schedule \bar{S}^* and $\mathcal{S}_{k, k+N-1}$ is a length N scheduling sequence. Thus, by Theorem 6.3.5, the schedule \bar{S}^* is MSS. In (6.G.2) the positive semi-definite matrices Q and $R(\gamma(k))$ are set to the same values as are used to compute \bar{P} as in Theorem 6.3.4. It is further worth noting the similarity between the value function (6.G.2) and the MPC cost functions (5.23) and (6.21).

Let

$$\begin{aligned} \mathcal{S}_{k, N-1}^\infty &\triangleq \arg \min_{\mathcal{S} \in \Xi} V(x(k), \mathcal{S}, i) \\ i_k^*, \mathcal{S}_{k, N-1}^* &\triangleq \arg \min_{i, \mathcal{S} \in \Xi} V(x(k), \mathcal{S}, i), \end{aligned} \quad (6.G.3)$$

where Ξ is the set of all possible length N scheduling sequences. Define the minimum value function

$$V^*(x(k)) \triangleq V(x(k), \mathcal{S}_{k, k+N-1}^*, i_k^*).$$

With some abuse of notation, we denote the schedule S_i to be the i 'th schedule in the periodic scheduling sequence \bar{S} . The sequence \bar{S} is the scheduling sequence that is obtained from the schedule $\bar{S} \in \bar{\mathcal{S}}^T$ that is used to compute the Riccati equation \bar{P} in (6.46). Denote P_i to be the i 'th block of the block diagonal matrix \bar{P} . This block corresponds to the periodic scheduling sequence $\bar{S}_{i, \infty}$ where S_i is the first schedule. It is then, using (6.G.1), easy to verify that

$$\mathbf{E} \left\{ \|x(k)\|_{P_i}^2 \mid x(k) \right\} = \mathbf{E} \left\{ \|x(k)\|_Q^2 + \|L^*(S_i(k))x(k)\|_{R\{\gamma(k)\}}^2 + \|x(k+1)\|_{P_{i+1}}^2 \mid x(k) \right\}, \quad (6.G.4)$$

where $i = k \% T$. In this proof i is the cyclic indexing the periodic scheduling sequence \mathcal{S} . Therefore, we adopt the convention that if $i+1 = T$, we set $i = 0$. See Section 6.1.2 for the notation of periodic scheduling sequences.

We want to show that $V^*(x(k))$ is a Lyapunov function. i.e.

$$V^*(x(k)) \geq 0 \quad \forall x(k) \quad (6.G.5a)$$

$$V^*(x(k)) = 0 \Rightarrow x(k) = 0 \quad (6.G.5b)$$

$$|x| \geq |y| \Rightarrow V^*(x) \geq V^*(y) \quad (6.G.5c)$$

$$V(x(k)) \geq V(x(k+1)) \quad \forall k. \quad (6.G.5d)$$

It is easy to verify that for all $\mathcal{S}_{k, k+N-1}$ and i ,

$$V(x(k), \mathcal{S}_{k, k+N-1}, i) \geq 0 \quad (6.G.6)$$

and

$$V(x(k), \mathcal{S}_{k, k+N-1}, i) = 0 \quad (6.G.7)$$

requires $x(k) = 0$. Further, it follows directly from (6.G.2) that when $|x| \geq |y|$

$$V(x, \mathcal{S}_{k, k+N-1}, i) \geq V(y, \mathcal{S}_{k, k+N-1}, i), \quad (6.G.8)$$

for all $\mathcal{S}_{k, k+N-1}$ and i . Thus, $V(x(k), \mathcal{S}_{k, k+N-1}, i)$ is monotone in $x(k)$. Since (6.G.6) to (6.G.8) hold true for all $\mathcal{S}_{k, k+N-1}$ and i , this must also hold for $\mathcal{S}_{k, k+N-1}^*$ and i_k^* . This proves (6.G.5a) to (6.G.5c).

However, since (6.G.1) and (6.G.2) are stochastic due to the variable $\gamma(k)$, condition (6.G.5d) is difficult to show. Instead, we consider the stochastic Lyapunov candidate $\mathbf{E}\{V^*(x(k))\}$ and verify that it is decreasing for all k , i.e.

$$\mathbf{E}\{V^*(x(k))\} \geq \mathbf{E}\{V^*(x(k+1))\}. \quad (6.G.9)$$

Refer to [52] for more details regarding stochastic Lyapunov functions.

First, we show that for the scheduling sequence $\bar{\mathcal{S}}_{i+1, i+N}$ that consists of the first N schedules of the periodic scheduling sequence that is used to compute the Riccati equation P_i , it holds true for all $i \in \{0, \dots, T-1\}$ that

$$\mathbf{E}\{V(x(k), \bar{\mathcal{S}}_{i, i+N-1}, i)\} \geq \mathbf{E}\{V(x(k+1), \bar{\mathcal{S}}_{i+1, i+N}, i+1)\}. \quad (6.G.10)$$

For this, we have that

$$\begin{aligned} \mathbf{E}\{V(x(k+1), \bar{\mathcal{S}}_{i+1, i+N}, i+1)\} &= \mathbf{E}\left\{\mathbf{E}\left\{\sum_{\ell=0}^{N-1} \|x(k+\ell+1)\|_Q^2 \right. \right. \\ &\quad \left. \left. + \|L(S(i+\ell+1))x(k+\ell+1)\|_{R\{\gamma(k+\ell+1)\}}^2 + \|x(k+N+1)\|_{P_{i+N+1}}^2 \mid x(k+1); \bar{\mathcal{S}}_{i+1, i+N}\right\}\right\} \\ &= \mathbf{E}\left\{\sum_{\ell=0}^{N-1} \|x(k+\ell)\|_Q^2 + \|L(S(i+\ell))x(k+\ell)\|_{R\{\gamma(k+\ell)\}}^2 \right. \\ &\quad \left. + \|x(k+N)\|_Q^2 + \|L(S(i+N))x(k+N)\|_{R\{\gamma(k+N)\}}^2 + \|x(k+N+1)\|_{P_{i+N+1}}^2 \right. \\ &\quad \left. - \left[\|x(k)\|_Q^2 + \|L(S(i))x(k)\|_{R\{\gamma(k)\}}^2\right]\right\}. \end{aligned}$$

Then, using (6.G.4), we have can rewrite the above to

$$\begin{aligned} \mathbf{E}\{V(x(k+1), \bar{\mathcal{S}}_{i+1, i+N}, i+1)\} &= \\ &= \mathbf{E}\left\{\sum_{\ell=0}^{N-1} \|x(k+\ell)\|_Q^2 + \|L(S(i+\ell))x(k+\ell)\|_{R\{\gamma(k+\ell)\}}^2 + \|x(k+N)\|_{P_{i+N}}^2 \right. \\ &\quad \left. - \left[\|x(k)\|_Q^2 + \|L(S(i))x(k)\|_{R\{\gamma(k)\}}^2\right]\right\} \\ &= \mathbf{E}\left\{V(x(k), \bar{\mathcal{S}}_{i, i+N-1}, i) - \underbrace{\left[\|x(k)\|_Q^2 + \|L(S(i))x(k)\|_{R\{\gamma(k)\}}^2\right]}_{\geq 0}\right\} \\ &\leq \mathbf{E}\{V(x(k), \bar{\mathcal{S}}_{i, i+N-1}, i)\}. \end{aligned}$$

It follows directly that this holds for all $i \in \{0, \dots, T-1\}$. Thus, (6.G.10) holds true.

Next, we show that

$$\mathbf{E}\{V(x(k), \bar{\mathcal{S}}_{i, i+N-1}, i)\} \geq \mathbf{E}\{V(x(k), \mathcal{S}_{k, k+N-1}^\circ, i)\} \geq \mathbf{E}\{V^*(x(k))\}. \quad (6.G.11)$$

For the first inequality, we have that

$$\begin{aligned} \mathbf{E} \{V(x(k), \mathcal{S}_{k,k+N-1}^\diamond, i)\} &= \mathbf{E} \left\{ \min_{\mathcal{S}_{k,k+N-1} \in \Xi} V(x(k), \mathcal{S}_{k,k+N-1}, i) \right\} \\ &\leq \mathbf{E} \{V(x(k), \bar{\mathcal{S}}_{i,i+N-1}^i, i)\}, \end{aligned} \quad (6.G.12)$$

since, in worst case, $\mathcal{S}_{k,k+N-1}^\diamond = \bar{\mathcal{S}}_{i,i+N-1}^i$. To show the second inequality in (6.G.11), we have that

$$\begin{aligned} \mathbf{E} \{V^*(x(k))\} &= \mathbf{E} \{V(x(k), \mathcal{S}_{k,k+N-1}^*, i^*)\} \\ &= \mathbf{E} \left\{ \min_{\mathcal{S}_{k,k+N-1} \in \Xi, i} V(x(k), \mathcal{S}_{k,k+N-1}, i) \right\} \\ &\leq \mathbf{E} \left\{ \min_{\mathcal{S}_{k,k+N-1} \in \Xi} V(x(k), \mathcal{S}_{k,k+N-1}, i) \right\} \\ &= \mathbf{E} \{V(x(k), \mathcal{S}_{k,k+N-1}^\diamond, i)\}. \end{aligned} \quad (6.G.13)$$

Equation (6.G.11) then follows by combining (6.G.12) and (6.G.13).

Lastly, we show that

$$\mathbf{E} \{V^*(x(k))\} \geq \mathbf{E} \{V(x(k+1), (\mathcal{S}_{k+1,k+N-1}^*, S(i_k^* + N)), i_k^*)\}. \quad (6.G.14)$$

In (6.G.14) we have that the sequence

$$\mathcal{S}_{k,k+N-1}^* = (S^*(k), \mathcal{S}_{k+1,k+N-1}^*)$$

where $\mathcal{S}_{k,k+N-1}^*$ is obtained using (6.G.3). Also, the schedule $S(i_k^* + N)$ is the schedule that fits with the $i_k^* + N$ 'th offset in the period i_k^* . Now,

$$\begin{aligned} &\mathbf{E} \{V(x(k+1), (\mathcal{S}_{k+1,k+N-1}^*, S(i_k^* + N)), i_k^*)\} \\ &= \mathbf{E} \left\{ \mathbf{E} \left\{ \sum_{\ell=0}^{N-2} \|x(k+\ell+1)\|_Q^2 + \|L(S^*(k+\ell+1))x(k+\ell+1)\|_{R\{\gamma(k+\ell+1)\}}^2 \right. \right. \\ &\quad + \|x(k+N)\|_Q^2 + \|L(S(i_k^* + N))x(k+N)\|_{R\{\gamma(k+N)\}}^2 \\ &\quad \left. \left. + \|x(k+N+1)\|_{P_{i_k^*+N+1}}^2 \mid x(k+1); (\mathcal{S}_{k+1,k+N-1}^*, S(i_k^* + N)) \right\} \right\} \\ &\stackrel{(a)}{=} \mathbf{E} \left\{ \sum_{\ell=0}^{N-2} \|x(k+\ell+1)\|_Q^2 + \|L(S^*(k+\ell+1))x(k+\ell+1)\|_{R\{\gamma(k+\ell+1)\}}^2 \right. \\ &\quad \left. + \|x(k+N)\|_{P_{i_k^*+N}}^2 \right\} \\ &= \mathbf{E} \left\{ \sum_{\ell=0}^{N-1} \|x(k+\ell)\|_Q^2 + \|L(S^*(k+\ell))x(k+\ell)\|_{R\{\gamma(k+\ell)\}}^2 \right. \\ &\quad + \|x(k+N)\|_{P_{i_k^*+N}}^2 \\ &\quad \left. - \left[\|x(k)\|_Q^2 + \|L(S^*(k))x(k)\|_{R\{\gamma(k)\}}^2 \right] \right\} \\ &= \mathbf{E} \left\{ V^*(x(k)) - \left[\|x(k)\|_Q^2 + \|L(S^*(k))x(k)\|_{R\{\gamma(k)\}}^2 \right] \right\} \\ &\leq \mathbf{E} \{V^*(x(k))\} \end{aligned} \quad (6.G.15)$$

where we used (6.G.4) in (a). This shows that (6.G.14) holds true.

It then follows directly by (6.G.11) that

$$\begin{aligned} \mathbf{E} \left\{ V(x(k+1), \mathcal{S}_{k+1, k+N}^\circ, i_k^*) \right\} &= \mathbf{E} \left\{ \min_{\mathcal{S}_{k+1, k+N}} V(x(k+1), \mathcal{S}_{k+1, k+N}, i_k^*) \right\} \\ &\leq \mathbf{E} \left\{ V(x(k+1), (\mathcal{S}_{k+1, k+N-1}^*, S(i_k^* + N)), i_k^*) \right\}. \end{aligned} \quad (6.G.16)$$

Then, (6.G.9) follows directly by combining (6.G.11), (6.G.15) and (6.G.16). Thus, $V^*(x)$ is a stochastic Lyapunov function and the MPC scheme in Algorithm 6.1 using final state weighting (6.54) is MSS [52].

In this proof we also obtained the result, that the MPC algorithm for system (6.G.1) without the disturbance $\omega(k)$, resulted in similar or better performance than periodic scheduling. However, note that the optimal cost in the MPC in Algorithm 6.1 depends on the statistics of the disturbance $\omega(k)$. Thus, when the system (6.G.1) is affected by noise, the cost improvement does not necessarily hold true for small values of $x(k)$. \square

6.H Proof of Lemma 6.4.2

For a schedule $\bar{S} \in \bar{\mathbb{S}}^T$, the cost (6.55) is at time k given by

$$\begin{aligned} G_{\text{dx}}(k, \bar{S}, \bar{U}_\infty^*(\bar{S})) &= \mathbf{E} \left\{ \bar{J}_d^*(\bar{x}(k), \bar{S}, \bar{U}_\infty^*) ; \bar{S} \right\} = \\ &= \mathbf{E} \left\{ \bar{x}(k)^T \bar{P} \bar{x}(k) + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t-1) \bar{F}^T \bar{P} \bar{F}) ; \bar{S} \right\} \\ &= \mathbf{E} \left\{ \text{trace}(\bar{x}(k) \bar{x}(k)^T \bar{P}) ; \bar{S} \right\} + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t-1) \bar{F}^T \bar{P} \bar{F}) \\ &= \text{trace}(\mathbf{E} \left\{ \bar{x}(k) \bar{x}(k)^T ; \bar{S} \right\} \bar{P}) + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t-1) \bar{F}^T \bar{P} \bar{F}) \\ &= \text{trace}(\Sigma_{\bar{x}}(k, \bar{S}) \bar{P}) + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t-1) \bar{F}^T \bar{P} \bar{F}). \end{aligned}$$

where $\Sigma_{\bar{x}}(k, \bar{S})$ is defined in (6.58). Now computing $\Sigma_{\bar{x}}(k, \bar{S})$ with $k+1$ for cleaner notation yields

$$\begin{aligned} \Sigma_{\bar{x}}(k+1, \bar{S}) &= \mathbf{E} \left\{ [(\bar{A} - \bar{B}(\bar{\gamma}(k)) \bar{L}_d) \bar{x}(k) + \bar{F} \bar{\omega}(k)] [(\bar{A} - \bar{B}(\bar{\gamma}(k)) \bar{L}_d) \bar{x}(k) + \bar{F} \bar{\omega}(k)]^T ; \bar{S} \right\} \\ &= \mathbf{E} \left\{ [\bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \bar{x}(k) + \bar{F} \bar{\omega}(k)] [\bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \bar{x}(k) + \bar{F} \bar{\omega}(k)]^T ; \bar{S} \right\} \\ &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \bar{x}(k) \bar{x}^T(k) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)) + \bar{F} \bar{\omega}(k) \bar{\omega}^T(k) \bar{F}^T ; \bar{S} \right\}, \end{aligned} \quad (6.H.1)$$

where the closed-loop response $\bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k))$ is defined in (6.19) and $\bar{L}_d^*(\bar{S})$ is given by (6.36). The above can be restated to

$$\begin{aligned} \Sigma_{\bar{x}}(k+1, \bar{S}) &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \mathbf{E} \left\{ \bar{x}(k) \bar{x}^T(k) ; \bar{S} \right\} \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)) ; \bar{S} \right\} + \bar{F} \bar{\Sigma}_\omega(k) \bar{F}^T \\ &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \Sigma_{\bar{x}}(k, \bar{S}) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)) ; \bar{S} \right\} + \bar{F} \bar{\Sigma}_\omega(k) \bar{F}^T. \end{aligned}$$

Using \bar{e}_i , as defined in (6.20) and $\bar{\Sigma}_\omega$ as defined in (6.13), allows us to rewrite the above to

$$\begin{aligned}\Sigma_{\bar{x}}(k+1, \bar{S}) &= \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \Sigma_{\bar{x}}(k, \bar{S}) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)); \bar{S} \right\} + \bar{e}_{(k+1)\%T} \bar{F} \bar{\Sigma}_\omega \bar{F}^T \bar{e}_{(k+1)\%T} \\ &= \bar{e}_{(k+1)\%T} \left[\mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \Sigma_{\bar{x}}(k, \bar{S}) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)); \bar{S} \right\} + \bar{F} \bar{\Sigma}_\omega \bar{F}^T \right] \bar{e}_{(k+1)\%T},\end{aligned}$$

Substituting this in the cost (6.H.1) yields

$$\begin{aligned}G_{\text{dx}}(k, \bar{S}, \bar{U}_\infty^*(\bar{S})) &= \\ &\text{trace} \left(\bar{e}_k \left[\mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \Sigma_{\bar{x}}(k-1, \bar{S}) \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)); \bar{S} \right\} + \bar{F} \bar{\Sigma}_\omega \bar{F}^T \right] \bar{e}_k \bar{P} \right) \\ &+ \sum_{t=0}^T \frac{\alpha^t}{1-\alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t) \bar{F}^T \bar{P} \bar{F}) \\ &= \text{trace}(\bar{e}_k \Sigma_{\bar{x}}(k, \bar{S}) \bar{e}_k \bar{P}) + \sum_{t=1}^T \frac{\alpha^t}{1-\alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t-1) \bar{F}^T \bar{P} \bar{F}),\end{aligned}$$

which leads to (6.59) and (6.60). \square

6.I Proof of Theorem 6.4.3

Note that the limit

$$\begin{aligned}\lim_{k \rightarrow \infty} G_{\text{dx}}(k, \bar{S}, \bar{U}_\infty^*(\bar{S})) &= \lim_{k \rightarrow \infty} \text{trace}(\bar{e}_{k\%T} \Sigma_{\bar{x}}(k, \bar{S}) \bar{e}_{k\%T} \bar{P}(\bar{S})) \\ &+ \lim_{k \rightarrow \infty} \sum_{t=1}^T \frac{\alpha^t}{1-\alpha^T} \text{trace}(\bar{\Sigma}_\omega(k+t-1) \bar{F}^T \bar{P} \bar{F})\end{aligned}$$

does not exist, due to the periodic switching in $\bar{e}_{k\%T}$. However, assume that for a known schedule $\bar{S} \in \bar{\mathcal{S}}^T$ the limit $\lim_{k \rightarrow \infty} \Sigma_{\bar{x}}(k, \bar{S})$ exists, where $\Sigma_{\bar{x}}(k, \bar{S})$ is given by (6.60). Now, for $\ell = 0, \dots, T-1$ let

$$\begin{aligned}\lim_{k \rightarrow \infty} G_{\text{dx}}(kT + \ell, \bar{S}, \bar{U}_\infty^*(\bar{S})) &= \lim_{k \rightarrow \infty} \text{trace}(\bar{e}_\ell \Sigma_{\bar{x}}(kT + \ell, \bar{S}) \bar{e}_\ell \bar{P}(\bar{S})) \\ &+ \lim_{k \rightarrow \infty} \sum_{t=1}^T \frac{\alpha^t}{1-\alpha^T} \text{trace}(\bar{\Sigma}_\omega(k + \ell + t - 1) \bar{F}^T \bar{P} \bar{F}) \\ &= \text{trace} \left(\bar{e}_\ell \lim_{k \rightarrow \infty} [\Sigma_{\bar{x}}(kT + \ell, \bar{S})] \bar{e}_\ell \bar{P}(\bar{S}) \right) + \sum_{t=1}^T \frac{\alpha^t}{1-\alpha^T} \text{trace}(\bar{\Sigma}_\omega(\ell + t - 1) \bar{F}^T \bar{P} \bar{F}).\end{aligned}\tag{6.I.1}$$

It is easy to verify that (6.60) is monotonic. Stating (6.60) on vectorized form yields

$$\text{vec}(\Sigma_{\bar{x}}(k+1, \bar{S})) = \mathcal{A}(\bar{S}) \text{vec}(\Sigma_{\bar{x}}(k, \bar{S})) + \text{vec}(\bar{F} \bar{\Sigma}_\omega \bar{F}^T),$$

where

$$\mathcal{A}(\bar{S}) \triangleq \mathbf{E} \left\{ \bar{H}_{\bar{L}_d^*(\bar{S})}(\bar{\gamma}(k)) \otimes \bar{H}_{\bar{L}_d^*(\bar{S})}^T(\bar{\gamma}(k)); \bar{S} \right\}.$$

Now, since $\lim_{k \rightarrow \infty} \Sigma_{\bar{x}}(k, \bar{S})$ exists, also $\lim_{k \rightarrow \infty} \text{vec}(\Sigma_{\bar{x}}(k, \bar{S}))$ exists. As shown in [49, Lemma 2.2], this is equivalent to the existence of a solution to the equation

$$\text{vec}(\Sigma_{\bar{x}}(\bar{S})) = \mathcal{A}(\bar{S}) \text{vec}(\Sigma_{\bar{x}}(\bar{S})) + \text{vec}(\bar{F} \bar{\Sigma}_\omega \bar{F}^T),$$

which leads to (6.61).

This means, that in (6.I.1)

$$\lim_{k \rightarrow \infty} \Sigma_{\bar{x}}(kT + \ell, \bar{S}) = \lim_{k \rightarrow \infty} \Sigma_{\bar{x}}(k, \bar{S}) = \Sigma_{\bar{x}}(\bar{S}),$$

holds true for any ℓ and T , where $\Sigma_{\bar{x}}(\bar{S})$ is the solution to (6.61). Substituting this into (6.I.1) for $\ell = 1, \dots, T - 1$ yields

$$\begin{aligned} \lim_{k \rightarrow \infty} G_{\text{dx}}(kT + \ell, \bar{S}, \bar{U}_{\infty}^*(\bar{S})) = \\ \text{trace}(\bar{e}_{\ell} \Sigma_{\bar{x}}(\bar{S}) \bar{e}_{\ell} \bar{P}(\bar{S})) + \sum_{t=1}^T \frac{\alpha^t}{1 - \alpha^T} \text{trace}(\bar{\Sigma}_{\bar{\omega}}(\ell + t - 1) \bar{F}^T \bar{P} \bar{F}), \end{aligned}$$

which leads to (6.62).

The sufficient condition follows directly, since if $\lim_{k \rightarrow \infty} \Sigma_{\bar{x}}(k, \bar{S}) \not\leq \infty$ then

$$\lim_{k \rightarrow \infty} G_{\text{dx}}(kT + \ell, \bar{S}, \bar{U}_{\infty}^*(\bar{S})) \not\leq \infty$$

for all $\ell = 1, \dots, T - 1$.

Conclusions and future directions

We summarize and conclude the results presented in this thesis and present a number of interesting and relevant future directions.

7.1 Conclusions

Initially, we presented a number of challenges related to the design of networked control systems (NCSs) that communicate over wireless networks. This work addresses mainly two of these challenges: packet loss and network bandwidth constraints. We have presented solutions that provide robustness and improved performance for NCSs under these conditions. By the use of tools such as optimal linear-quadratic control, dynamic programming, stochastic control and optimization, we presented frameworks for the analysis and synthesis of the control and scheduling for NCSs which are interconnected with wireless networks affected by packet loss and bandwidth constraints.

In this thesis, we presented a method for optimal controller and offline estimator design that is robust against packet losses that can be modelled using an independent and identically distributed (*i.i.d*) model. While *i.i.d* packet loss models are convenient in the analysis and synthesis of the controller, packet loss on wireless networks tends to be correlated. However, in many network models that consider correlated packet loss, the network state can not be observed by the controller. If, on the other hand, network models are used where the network state depends on a finite history of recent packet arrivals, the network state can be observed by the controller. We presented an optimal controller design that takes these correlated network models into account. This method greatly improves performance over controllers that feature *i.i.d* network models. Lastly, we considered networks that are affected by *i.i.d* packet loss and are bandwidth limited. These networks are based on the IEEE 802.15.4 protocol. To accommodate this, we presented multiple methods for the co-design of an optimal scheduler and controller that can address multiple actuators over bandwidth limited networks that are affected by packet loss. This, however, is a non-deterministic polynomial-time (NP)-hard problem, as discussed in e.g. [93]. To reduce this computational complexity, we presented a method that utilizes infinite horizon cost function approximations to allow for short horizons in an online model predictive control (MPC) algorithm. This provides significantly improved performance over offline scheduling.

Next, we present the conclusions for the results that we obtained in each chapter.

- **Chapter 3** In this chapter, we presented a method for the analysis and synthesis of the optimal controller and offline estimator for NCSs that are connected with wireless networks featuring *i.i.d* packet dropouts. The controller and estimator are offline and optimal in the sense of minimizing a cost function which is solely based on system and network statistics. We pointed out that these designs are dual, which permits the estimator design for a given system to be performed by formulating a dual system, and then design a controller for the latter. We have thereby extended the classical duality that exists between a linear quadratic regulator (LQR) and a Kalman filter to the case of systems that are affected by *i.i.d* packet loss.
- **Chapter 4** In this chapter, we extended the controller design method from Chapter 3 to consider correlated network models. These correlated network models can capture memory effects on the network which cannot be captured by *i.i.d* models. We utilized a network model where the network state depends on a finite history of packet arrival outcomes. This allows the controller to observe the network state. The optimal controller design that was presented, resulted in feedback policies that take into account the packet arrival history by selecting representative gains from a lookup table. Simulation studies illustrate significant performance gains over standard LQR designs as well as designs that merely consider *i.i.d* network models. The implementation of this controller is however complex, since the number of control gains increases exponentially with the length of the relevant history. In an attempt to reduce the computational complexity, we presented three sub-optimal methods that trade-off controller complexity and control performance. Two out of these three methods provide improved performance at similar complexity levels as the controllers that are developed for an *i.i.d* network model.
- **Chapter 5** In this chapter, we presented a novel co-design method that creates both optimal schedules and control inputs for distributed NCSs that are connected through wireless networks. These networks utilize a IEEE 802.15.4 protocol that features both a more reliable contention free period (CFP) and a less reliable contention access period (CAP). While transmission in the CFP is preferred, the limit of the amount of transmission slots in the CFP motivates the use of the less reliable CAP. We presented a novel co-design method that takes both transmission periods into account to design an optimal schedule and feedback control policies. Simulation studies illustrated significant performance gains compared to offline round robin (RR) heuristics with fixed scheduling. Further, these simulation studies showed, that longer prediction horizon lengths yielded improved control performance. However, these longer prediction horizons lead to a significantly increased computational complexity. This is prohibitive for implementations of the presented co-scheduling algorithm on large scale NCSs.
- **Chapter 6** This chapter extends the controller and scheduler co-design that was introduced in Chapter 5. We presented a framework that allows reformulation of the time-varying stochastic systems that appear when finite length scheduling sequences are utilized. This framework allowed the time-varying systems to be restated as linear time-invariant (LTI) stochastic systems. This simplified the synthesis and analysis significantly. After revising the results from Chapter 5 using the LTI framework, we presented methods to approximate

infinite horizon cost functions by the use of fixed length periodic scheduling sequences with a fixed period. We utilized these results to design the final state weighting in the online MPC algorithm and to design offline schedules. Simulation studies showed, that for shorter horizons, the MPC algorithm with a proper choice of final state weighting outperforms MPC with an arbitrarily selected final state weighting. The online MPC algorithm yield significant performance gains for short prediction horizons compared to arbitrarily selected final state weightings. This allowed for greater performance at lower a computational complexity than the method that is presented in Chapter 5. Further, we showed that the online MPC algorithm provides mean square stability (MSS) when the final state weighting is selected to be the periodic solution to a periodic Riccati equation that minimized a long-run averaged cost.

The offline schedules that were designed using the solutions to the infinite horizon cost approximations, resulted in significantly improved performance compared to the reference heuristic scheduling used in Chapter 5. Further, two of the methods that we presented provide MSS.

7.2 Future work

In this thesis, we addressed a number of problems that arise when implementing wireless networks in NCSs. Many of the solutions that we presented lead to new questions and open problems that will have to be addressed in the future. This section describes a number of these future directions.

7.2.1 Correlated packet dropouts

In Chapters 3 and 4, we have illustrated that the controller designs that we presented can yield significant performance gains when more accurate correlated network models are taken into account. Further, even when constraints on the number of controllers are implemented, the proposed methods outperform *i.i.d* network models. While simulation studies show performance gains, analytical performance guarantees and bounds have so far not been provided. Future work could include an explicit characterization that can quantify the trade-off between the number of controllers and the closed-loop performance for different lengths of packet loss history.

Another topic that has not been addressed is closed-loop stability. However, by reformulating the closed-loop system to fit into the Markov jump linear system (MJLS) framework, the results from works such as [23], can be readily applied.

7.2.2 Scheduler and controller co-design

In Chapters 5 and 6, we presented optimal co-design algorithms to jointly design an optimal schedule and controller. We presented methods that reduce the number of scheduling sequences that have to be compared in an online algorithm while providing good control performance. However, the computational requirements for larger systems are still significant. This becomes prohibitive for implementations on systems that have limited computational power available. One obvious method to reduce the computational complexity of the online part, is to reduce the number of scheduling sequences that are compared in the online MPC algorithm. Reducing the number of

scheduling sequences will inevitably deteriorate the performance by some degree. However, clever selection of the scheduling sequences that are contained in the subset might merely compromise the control performance by a small margin while reducing the computational requirements significantly.

In the above, we considered the online computational part in the MPC algorithm. However, for very large-scale systems, the offline computations for solving the involved Riccati equations can also become prohibitive. To find the optimal scheduling sequence, all solutions to the Riccati equations need to be compared. This is a combinatorial problem and therefore, optimality can only be guaranteed after comparing all possible scheduling sequences. While this to some extent can be performed on large computing grids, the exponential growth of the complexity of the problem can make this infeasible for particularly large systems. Approaches that avoid this computational complexity while guaranteeing some degree of optimality would be a significant contribution. Alternatively, an interesting approach could be to use a form of probabilistic scheduling for the cost function that is used to select the final state weighting in the MPC algorithms.

We have presented results that guarantee closed-loop stability for some of the optimal co-design algorithms, both online and offline. However, the applicability of the remaining methods will be limited until stability results have been obtained for these. Further, an analytical analysis of the cost improvement for the MPC algorithm compared to offline schedules will be useful. This can, among others, be utilized to compare the trade-off between the increased computational complexity against the performance gains without performing extensive simulations.

Bibliography

- [1] 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.
- [2] J. Åkerberg, M. Gidlund, and M. Björkman, “Future research challenges in wireless sensor and actuator networks targeting industrial automation,” in *Industrial informatics (INDIN), 2011 9th IEEE international conference on*, 2011, pp. 410–415.
- [3] B. D. O. Anderson and J. B. Moore, *Optimal filtering*, T. Kailath, Ed., ser. Information and system sciences series. Englewood Cliffs, N.J.: Prentice-hall International, 1979, ISBN: 0486136899.
- [4] —, *Optimal control : Linear quadratic methods*, Prentice-Hall International ed., T. Kailath, Ed. Prentice-Hall International, 1989, ISBN: 0136386512.
- [5] D. Antunes, W. P.M. H. Heemels, and P. Tabuada, “Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design,” in *2012 IEEE 51st IEEE conference on decision and control (CDC)*, 2012, pp. 7212–7217.
- [6] D. Antunes and W. Heemels, “Rollout event-triggered control: Beyond periodic control performance,” *Automatic control, IEEE transactions on*, vol. 59, no. 12, pp. 3296–3311, 2014.
- [7] D. Antunes, J. P. Hespanha, and C. Silvestre, “Stochastic networked control systems with dynamic protocols,” *Asian journal of control*, vol. 17, no. 1, pp. 99–110, 2015.
- [8] J. Araujo, Y. Ariba, P. Park, H. Sandberg, and K. Johansson, “Control over a hybrid MAC wireless network,” in *Smart grid communications (SmartGridComm), 2010 first IEEE international conference on*, 2010, pp. 197–202.
- [9] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. Johansson, “System architectures, protocols and algorithms for aperiodic wireless control systems,” *Industrial informatics, IEEE transactions on*, vol. 10, no. 1, pp. 175–184, 2014.
- [10] M. Athans, R. Ku, and S. Gershwin, “The uncertainty threshold principle: Some fundamental limitations of optimal decision making under dynamic uncertainty,” *Automatic control, IEEE transactions on*, vol. 22, no. 3, pp. 491–495, 1977.

- [11] B. Azimi-Sadjadi, “Stability of networked control systems in the presence of packet losses,” in *Decision and control, 2003. proceedings. 42nd IEEE conference on*, vol. 1, 2003, 676–681 Vol.1.
- [12] H. Bai and M. Atiquzzaman, “Error modeling schemes for fading channels in wireless communications: A survey,” *IEEE communications surveys tutorials*, vol. 5, no. 2, pp. 2–9, 2003.
- [13] J. Baillieul and P. J. Antsaklis, “Control and communication challenges in networked real-time systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.
- [14] D. P. Bertsekas, *Dynamic programming & optimal control, vol 2*, 2nd ed. 22000/9d6417d9c889ab635190561d997c23ad: Athena Scientific, 1995, vol. 2.
- [15] —, *Dynamic programming & optimal control*, 3rd ed. Nashua, U.S.A.: Athena Scientific, 2005, vol. 1, ISBN: 1886529267,1886529264.
- [16] S. Bittanti and P. Colaneri, *Periodic systems: Filtering and control*, Springer, Ed., ser. Communications and Control Engineering. Springer-Verlag London, 2009, ISBN: 1848009103, 1848009110.
- [17] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017, ISBN: 1107652871.
- [18] T. Brooks, “Wireless technology for industrial sensor and control networks,” in *SIcon/01. sensors for industry conference. proceedings of the first ISA/IEEE. sensors for industry conference*, 2001, pp. 73–77.
- [19] F. Casiello and K. A. Loparo, “Optimal control of unknown parameter systems,” *IEEE transactions on automatic control*, vol. 34, no. 10, pp. 1092–1094, 1989.
- [20] J. Chen, K. H. Johansson, S. Olariu, I. C. Paschalidis, and I. Stojmenovic, “Guest editorial special issue on wireless sensor and actuator networks,” *IEEE transactions on automatic control*, vol. 56, no. 10, pp. 2244–2246, 2011.
- [21] W. Chen, Y. Zou, N. Xiao, J. Song, and Y. Niu, “Quantised feedback stabilisation of LTI systems over partly unknown Markov fading channels,” *International journal of systems science*, vol. 0, no. 0, pp. 1–9, 2017.
- [22] Y. Q. Chen and Z. Wang, “Formation control: A review and a new consideration,” in *2005 IEEE/RSJ international conference on intelligent robots and systems*, 2005, pp. 3181–3186.
- [23] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*, ser. Probability and Its Applications. 233 Spring Street, New York, NY 10013-1578 USA: Springer Science & Business Media, 2006, ISBN: 1852337613.
- [24] A. Czornik and A. Świerniak, “On direct controllability of discrete time jump linear system,” *Journal of the Franklin institute*, vol. 341, no. 6, pp. 491–503, 2004.
- [25] B. Demirel, “Architectures and performance analysis of wireless control systems,” PhD thesis, KTH Royal Institute of Technology, 2015, ISBN: 978-91-7595-528-5.
- [26] K. Deng, P. G. Mehta, and S. P. Meyn, “Optimal kullback-leibler aggregation via spectral theory of Markov chains,” *IEEE transactions on automatic control*, vol. 56, no. 12, pp. 2793–2808, 2011.

-
- [27] D. Dolz, D. E. Quevedo, I. Peñarrocha, and R. Sanchis, "Performance vs complexity trade-offs for Markovian networked jump estimators," in *Proceedings of the 19th IFAC world congress, 2014*, Cape Town: International Federation of Automatic Control (IFAC), 2014.
- [28] N. Elia, "Remote stabilization over fading channels," *Systems & control letters*, vol. 54, no. 3, pp. 237–249, 2005.
- [29] C. Fischione, P. Park, S. Ergen, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Analytical modeling and optimization of duty-cycles in preamble-based IEEE 802.15.4 wireless sensor networks," *IEEE/ACM transactions on networking*, 2009, QS 20120315.
- [30] H. C. Foundation. (2017). HART application guide. Retrieved 20-06-2017, Fieldcomm.
- [31] M. D. Fragoso, "Discrete-time jump LQG problem," *International journal of systems science*, vol. 20, no. 12, pp. 2539–2545, 1989.
- [32] M. E.M. B. Gaid, A. Cela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system," *IEEE transactions on control systems technology*, vol. 14, no. 4, pp. 776–787, 2006.
- [33] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell system technical journal*, vol. 39, no. 5, pp. 1253–1265, 1960.
- [34] R. D. Gomes, D. V. Queiroz, A. C. L. Filho, I. E. Fonseca, and M. S. Alencar, "Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes," *Ad hoc networks*, vol. 59, pp. 116–133, 2017.
- [35] G. Goodwin, H. Haimovich, D. Quevedo, and J. Welsh, "A moving horizon approach to networked control system design," *Automatic control, IEEE transactions on*, vol. 49, no. 9, pp. 1427–1445, 2004.
- [36] G. C. Goodwin, D. E. Quevedo, and E. I. Silva, "Architectures and coder design for networked control systems," *Automatica*, vol. 44, no. 1, pp. 248–257, 2008.
- [37] L. Grüne and A. Panin, "On non-averaged performance of economic MPC with terminal conditions," in *2015 54th IEEE conference on decision and control (CDC)*, 2015, pp. 4332–4337.
- [38] W. Heemels, M. Donkers, and A. Teel, "Periodic event-triggered control for linear systems," *Automatic control, IEEE transactions on*, vol. 58, no. 4, pp. 847–861, 2013.
- [39] M. Heinen, *The A380 program*, Slides from speech at EADS Global Investor Forum, 2006.
- [40] E. Henriksson, D. E. Quevedo, E. G. W. Peters, H. Sandberg, and K. H. Johansson, "Multiple-loop self-triggered model predictive control for network scheduling and control," *IEEE transactions on control systems technology*, vol. 23, no. 6, pp. 2167–2181, 2015.
- [41] 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.
- [42] D. Hristu and K. Morgansen, "Limited communication control," *Systems & control letters*, vol. 37, no. 4, pp. 193–205, 1999.
- [43] *IEEE standard for local and metropolitan area networks – part 15.4: Networks low-rate wireless (LR-WPANs)*, IEEE Standards Association, 2011.

- [44] *IEEE standard for local and metropolitan area networks - part 15.4: Low-rate wireless personal area networks (LR-WPANs) amendment 1: MAC sublayer*, IEEE Standards Association, 2012.
- [45] O. C. Imer, S. Yüksel, and T. Başar, “Optimal control of LTI systems over unreliable communication links,” *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.
- [46] ISA. (2009). Isa100.11a standard. Retrieved 20-06-2017, ISA100WCI.
- [47] M. Johansson and R. Jaäntti, “Wireless networking for control: Technologies and models,” in *Networked control systems*, ser. Lecture Notes in Control and Information Sciences 406, M. J. Alberto Bemporad Maurice Heemels, Ed., 1st ed., Springer-Verlag London, 2010, ch. 2, pp. 31–74, ISBN: 978-0-85729-033-5.
- [48] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [49] W. L. D. Koning, “Infinite horizon optimal control of linear discrete time systems with stochastic parameters,” *Automatica*, vol. 18, no. 4, pp. 443–453, 1982.
- [50] —, “Compensatability and optimal compensation of systems with white parameters,” *IEEE transactions on automatic control*, vol. 37, no. 5, pp. 579–588, 1992.
- [51] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, “A Markov-based channel model algorithm for wireless networks,” *Wireless networks*, vol. 9, no. 3, pp. 189–199, 2003.
- [52] Y. Li, W. Zhang, and X. Liu, “Stability of nonlinear stochastic discrete-time systems,” *Journal of applied mathematics*, vol. 2013, p. 8, 2013.
- [53] H. Lin, H. Su, Z. Shu, Z. G. Wu, and Y. Xu, “Optimal estimation in UDP-like networked control systems with intermittent inputs: Stability analysis and suboptimal filter design,” *IEEE transactions on automatic control*, vol. 61, no. 7, pp. 1794–1809, 2016.
- [54] L. Lindbom, A. Ahlén, M. Sternad, and M. Falkenström, “Tracking of time-varying mobile radio channels part II: A case study on D-AMPS 1900 channels,” *IEEE trans. commun.*, vol. 49, pp. 2207–2217, 2000.
- [55] K. V. Ling, J. Maciejowski, A. Richards, and B. F. Wu, “Multiplexed model predictive control,” *Automatica*, vol. 48, no. 2, pp. 396–401, 2012.
- [56] —, “Multiplexed model predictive control,” *Automatica*, vol. 48, no. 2, pp. 396–401, 2012.
- [57] Q. Ling and M. D. Lemmon, “Optimal dropout compensation in networked control systems,” in *42nd IEEE international conference on decision and control*, vol. 1, 2003, 670–675 Vol.1.
- [58] M. Lješnjanić, D. E. Quevedo, and D. Nešić, “Packetized MPC with dynamic scheduling constraints and bounded packet dropouts,” *Automatica*, vol. 50, no. 3, pp. 784–797, 2014.
- [59] S. Longo, G. Herrmann, and P. Barber, “Controllability, observability in networked control,” *IFAC proceedings volumes*, vol. 42, no. 6, pp. 295–300, 2009, 6th IFAC Symposium on Robust Control Design.

-
- [60] 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.
- [61] M. Martalò, S. Busanelli, and G. Ferrari, “Markov chain-based performance analysis of multihop IEEE 802.15.4 wireless networks,” *Performance evaluation*, vol. 66, no. 12, pp. 722–741, 2009.
- [62] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
- [63] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [64] M. Mazo and P. Tabuada, “On event-triggered and self-triggered control over sensor/actuator networks,” in *Decision and control, 2008. CDC 2008. 47th IEEE conference on*, 2008, pp. 435–440.
- [65] —, “Decentralized event-triggered control over wireless sensor/actuator networks,” *Automatic control, IEEE transactions on*, vol. 56, no. 10, pp. 2456–2461, 2011.
- [66] X. Meng and T. Chen, “Event-driven communication for sampled-data control systems,” in *American control conference (ACC), 2013*, 2013, pp. 3002–3007.
- [67] P. Mhaskar, J. Liu, and P. D. Christofides, *Fault-tolerant process control: Methods and applications*, ser. Methods and Applications. Springer-Verlag London, 2013, ISBN: 1447148074, 1447159636, 1447148081.
- [68] Y. Mo and B. Sinopoli, “A characterization of the critical value for Kalman filtering with intermittent observations,” in *2008 47th IEEE conference on decision and control*, 2008, pp. 2692–2697.
- [69] —, “Kalman filtering with intermittent observations: Tail distribution and critical value,” *Automatic control, IEEE transactions on*, vol. 57, no. 3, pp. 677–689, 2012.
- [70] Y. Mo, E. Garone, and B. Sinopoli, “LQG control with Markovian packet loss,” in *Control conference (ECC), 2013 european*, Zürich, 2013, pp. 2380–2385.
- [71] A. Molin and S. Hirche, “On the optimality of certainty equivalence for event-triggered control systems,” *Automatic control, IEEE transactions on*, vol. 58, no. 2, pp. 470–474, 2013.
- [72] —, “A bi-level approach for the design of event-triggered control systems over a shared network,” *Discrete event dynamic systems*, vol. 24, no. 2, pp. 153–171, 2014.
- [73] T. Namerikawa and T. Kato, “Distributed load frequency control of electrical power networks via iterative gradient methods,” in *Decision and control and european control conference (CDC-ECC), 2011 50th IEEE conference on*, 2011, pp. 7723–7728.
- [74] P. Neumann, “Communication in industrial automation—what is going on?” *Control engineering practice*, vol. 15, no. 11, pp. 1332–1347, 2007, Special Issue on Manufacturing Plant Control: Challenges and Issues.

- [75] S. Öncü, N. van de Wouw, W. P.M. H. Heemels, and H. Nijmeijer, “String stability of interconnected vehicles under communication constraints,” in *2012 IEEE 51st IEEE conference on decision and control (CDC)*, 2012, pp. 2459–2464.
- [76] R. Oung and R. D’Andrea, “The distributed flight array,” *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.
- [77] P. Park, C. Fischione, and K. H. Johansson, “Performance analysis of GTS allocation in beacon enabled IEEE 802.15.4,” in *Sensor, mesh and ad hoc communications and networks, 2009. SECON ’09. 6th annual IEEE communications society conference on*, 2009, pp. 1–9.
- [78] —, “Modeling and stability analysis of hybrid multiple access in the IEEE 802.15.4 protocol,” *ACM transactions on sensor networks (TOSN)*, vol. 9, no. 2, 13:1–13:55, 2013.
- [79] T. Park, T. Kim, J. Choi, S. Choi, and W. Kwon, “Throughput and energy consumption analysis of IEEE 802.15.4 slotted CSMA/CA,” *Electronics letters*, vol. 41, no. 18, pp. 1017–1019, 2005.
- [80] E. G. W. Peters, D. E. Quevedo, and J. Østergaard, “Shaped Gaussian dictionaries for quantized networked control systems with correlated dropouts,” *IEEE transactions on signal processing*, vol. 64, no. 1, pp. 203–213, 2016.
- [81] E. G. W. Peters, D. Marelli, M. Fu, and D. E. Quevedo, “Unified approach to controller and MMSE estimator design with intermittent communications,” in *55th IEEE conference on decision and control, CDC, Las Vegas, NV, USA*, 2016, pp. 6832–6837.
- [82] E. G. W. Peters, D. Marelli, D. E. Quevedo, and M. Fu, “Controller design for networked control systems affected by correlated packet losses,” in *IFAC world congress, Toulouse, France*, 2017.
- [83] —, “Predictive control for networked systems affected by correlated packet loss,” *International journal of robust and nonlinear control*, 2017, Early access for Special Issue on Stochastic Predictive Control.
- [84] E. G. W. Peters, D. E. Quevedo, and M. Fu, “Co-design for control and scheduling over wireless industrial control networks,” in *54th IEEE conference on decision and control, CDC, Osaka, Japan*, 2015, pp. 2459–2464.
- [85] —, “Controller and scheduler codesign for feedback control over IEEE 802.15.4 networks,” *IEEE transactions on control systems technology*, vol. 24, no. 6, pp. 2016–2030, 2016.
- [86] K. Plarre and F. Bullo, “On Kalman filtering for detectable systems with intermittent observations,” *IEEE transactions on automatic control*, vol. 54, no. 2, pp. 386–390, 2009.
- [87] R. Postoyan, N. Van De Wouw, D. Nesic, and W. Heemels, “Tracking control for nonlinear networked control systems,” *Automatic control, IEEE transactions on*, vol. 59, no. 6, pp. 1539–1554, 2014.
- [88] 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.

-
- [89] D. E. Quevedo and D. Nešić, “Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts,” *Automatic control, IEEE transactions on*, vol. 56, no. 2, pp. 370–375, 2011.
- [90] D. E. Quevedo, J. Østergaard, and D. Nesić, “Packetized predictive control of stochastic systems over bit-rate limited channels with packet loss,” *IEEE transactions on automatic control*, vol. 56, no. 12, pp. 2854–2868, 2011.
- [91] D. E. Quevedo, E. I. Silva, and G. C. Goodwin, “Control over unreliable networks affected by packet erasures and variable transmission delays,” *IEEE journal on selected areas in communications*, vol. 26, no. 4, pp. 672–685, 2008.
- [92] H. Rehbinder and M. Sanfridson, “Integration of off-line scheduling and optimal control,” in *Real-time systems, 2000. euromicro RTS 2000. 12th euromicro conference on*, 2000, pp. 137–143.
- [93] —, “Scheduling of a limited communication channel for optimal control,” *Automatica*, vol. 40, no. 3, pp. 491–500, 2004.
- [94] E. Rohr, D. Marelli, and M. Fu, “Kalman filtering with intermittent observations: On the boundedness of the expected error covariance,” *Automatic control, IEEE transactions on*, vol. 59, no. 10, pp. 2724–2738, 2014.
- [95] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams, “Finite-state Markov modeling of fading channels – A survey of principles and applications,” *IEEE signal processing magazine*, vol. 25, no. 5, pp. 57–80, 2008.
- [96] 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, Special Issue ADCHEM 2006 Symposium.
- [97] H. Sandberg, S. Amin, and K. H. Johansson, “Cyberphysical security in networked control systems: An introduction to the issue,” *IEEE control systems*, vol. 35, no. 1, pp. 20–23, 2015, Special Issue on Cyberphysical Security in Networked Control Systems.
- [98] L. Schenato, “Optimal estimation in networked control systems subject to random delay and packet drop,” *Automatic control, IEEE transactions on*, vol. 53, no. 5, pp. 1311–1317, 2008.
- [99] —, “To zero or to hold control inputs with lossy links?” *Automatic control, IEEE transactions on*, vol. 54, no. 5, pp. 1093–1099, 2009.
- [100] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, “Foundations of control and estimation over lossy networks,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.
- [101] P. J. Schweitzer, “A survey of aggregation-disaggregation in large Markov chains,” in *Numerical solution of Markov chains*, W. Stewart, Ed., vol. 8, Boca Raton, Florida, USA: CRC press, 1991, ch. 4, pp. 63–88, ISBN: 9780691036991.
- [102] X. Shen, D. Rus, and M. H. Ang, “Bounds for Kalman filtering with intermittent observations,” in *Control conference (ECC), 2015 European*, 2015, pp. 2842–2846.

BIBLIOGRAPHY

- [103] L. Shi, Y. Yuan, and J. Chen, “Finite horizon LQR control with limited controller-system communication,” *Automatic control, IEEE transactions on*, vol. 58, no. 7, pp. 1835–1841, 2013.
- [104] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, “Kalman filtering with intermittent observations,” *Automatic control, IEEE transactions on*, vol. 49, no. 9, pp. 1453–1464, 2004.
- [105] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, “Optimal control with unreliable communication: The TCP case,” in *American control conference, 2005. proceedings of the 2005*, 2005, 3354–3359 vol. 5.
- [106] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, “Optimal linear LQG control over lossy networks without packet acknowledgment,” *Asian journal of control*, vol. 10, no. 1, pp. 3–13, 2008.
- [107] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, 2nd ed., Wiley, Ed. Wiley, 2007, ISBN: 978-0-470-01167-6, 978-0-470-01168-3.
- [108] S. Smith and P. Seiler, “Estimation with lossy measurements: Jump estimators for jump systems,” *Automatic control, IEEE transactions on*, vol. 48, no. 12, pp. 2163–2171, 2003.
- [109] H. Song, S. C. Chen, and Y. Yam, “Sliding mode control for discrete-time systems with Markovian packet dropouts,” *IEEE transactions on cybernetics*, pp. 1–11, 2016.
- [110] K. Stouffer, “Guide to industrial control systems (ICS) security,” *NIST special publication*, vol. 800, no. 82, pp. 16–16,
- [111] M. Tabbara, D. Nesic, 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.
- [112] U. Tiberi, C. Fischione, K. Johansson, and M. D. Benedetto, “Energy-efficient sampling of networked control systems over IEEE 802.15.4 wireless networks,” *Automatica*, vol. 49, no. 3, pp. 712–724, 2013.
- [113] Y. Tipsuwan and M.-Y. Chow, “Control methodologies in networked control systems,” in *Control engineering practice*, 2003, pp. 1099–1111.
- [114] A. Ulusoy, O. Gurbuz, and A. Onat, “Wireless model-based predictive networked control system over cooperative wireless network,” *Industrial informatics, IEEE transactions on*, vol. 7, no. 1, pp. 41–51, 2011.
- [115] J. B.R. D. Val and T. Başar, “Receding horizon control of Markov jump linear systems,” in *American control conference, 1997. proceedings of the 1997*, vol. 5, Albuquerque, 1997, 3195–3199 vol.5.
- [116] A. N. Vargas, J. B. R. do Val, and E. F. Costa, “Receding horizon control of Markov jump linear systems subject to noise and unobserved state chain,” in *Decision and control, 2004. cdc. 43rd IEEE conference on*, vol. 4, Paradise Island, 2004, 4381–4386 Vol.4.
- [117] —, “Optimality condition for the receding horizon control of Markov jump linear systems with non-observed chain and linear feedback controls,” in *Proceedings of the 44th IEEE conference on decision and control*, Seville, 2005, pp. 7308–7313.

-
- [118] B. Wang and M. Fu, "Comparison of periodic and event-based sampling for linear state estimation," *IFAC world congress, Cape Town*, vol. 47, no. 3, pp. 5508–5513, 2014.
- [119] H. S. Wang and N. Moayeri, "Finite-state Markov channel—a useful model for radio communication channels," *IEEE transactions on vehicular technology*, vol. 44, no. 1, pp. 163–171, 1995.
- [120] Z. Wang, C. Zhu, W. Pan, and G. Guo, "Observability and controllability of networked control systems," in *2007 IEEE international conference on integration technology*, 2007, pp. 96–99.
- [121] J. Wu and T. Chen, "Design of networked control systems with packet dropouts," *Automatic control, IEEE transactions on*, vol. 52, no. 7, pp. 1314–1319, 2007.
- [122] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the Mbone multicast network," in *Global telecommunications conference, 1996. GLOBECOM 96 communications: The key to global prosperity*, IEEE, London, 1996, pp. 94–99.
- [123] C. Yang, Y. Bar-Shalom, and C. F. Lin, "Control of partially observed discrete-time jump systems," in *American control conference, 1991*, Massachusetts, 1991, pp. 1559–1562.
- [124] E. Yaz, "Stabilization of discrete-time systems with stochastic parameters," *Systems & control letters*, vol. 5, no. 5, pp. 321–326, 1985.
- [125] K. You and L. Xie, "Minimum data rate for mean square stabilizability of linear systems with Markovian packet losses," *IEEE transactions on automatic control*, vol. 56, no. 4, pp. 772–785, 2011.
- [126] K. You, M. Fu, and L. Xie, "Mean square stability for Kalman filtering with Markovian packet losses," *Automatica*, vol. 47, no. 12, pp. 2647–2657, 2011.
- [127] K. You and L. Xie, "Survey of recent progress in networked control systems," *Acta automatica sinica*, vol. 39, no. 2, pp. 101–117, 2013.
- [128] M. Zeeshan, A. Ali, A. Naveed, A. X. Liu, A. Wang, and H. K. Qureshi, "Modeling packet loss probability and busy time in multi-hop wireless networks," *EURASIP journal on wireless communications and networking*, vol. 2016, no. 1, p. 168, 2016.
- [129] D. Zhang, P. Shi, Q.-G. Wang, and L. Yu, "Analysis and synthesis of networked control systems: A survey of recent advances and challenges," *ISA transactions*, vol. 66, pp. 376–392, 2017.
- [130] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems – a survey," *Industrial informatics, IEEE transactions on*, vol. 9, no. 1, pp. 403–416, 2013.
- [131] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE control systems*, vol. 21, no. 1, pp. 84–99, 2001.
- [132] X. M. Zhang, Q. L. Han, and X. Yu, "Survey on recent advances in networked control systems," *IEEE transactions on industrial informatics*, vol. 12, no. 5, pp. 1740–1752, 2016.
- [133] *Zigbee PRO specification*, ZigBee Alliance, 2007.