Design, Implementation and Validation of Resource-Aware and Resilient Wireless Networked Control Systems

JOSÉ ARAÚJO

Doctoral Thesis
Stockholm, Sweden 2014

KTH Royal Institute of Technology
School of Electrical Engineering
Department of Automatic Control
SE-100 44 Stockholm
SWEDEN

TRITA-EE 2014:047
ISSN 1653-5146
ISBN 978-91-7595-295-6

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framläggs till offentlig granskning för avläggande av teknologie doktorsexamen i reglerteknik fredagen den 24 oktober 2014, klockan 10:15 i sal F3, Kungliga Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Networked control over wireless networks is of growing importance in many application domains such as industrial control, building automation and transportation systems. Wide deployment however, requires systematic design tools to enable efficient resource usage while guaranteeing close-loop control performance. The control system may be greatly affected by the inherent imperfections and limitations of the wireless medium and malfunction of system components. In this thesis, we make five important contributions that address these issues.

In the first contribution, we consider event- and self-triggered control and investigate how to efficiently tune and execute these paradigms for appropriate control performance. Communication strategies for aperiodic control are devised, where we jointly address the selection of medium-access control and scheduling policies. Experimental results show that the best trade-off is obtained by a hybrid scheme, combining event- and self-triggered control together with contention-based and contention-free medium access control.

The second contribution proposes an event-based method to select between fast and slow periodic sampling rates. The approach is based on linear quadratic control and the event condition is a quadratic function of the system state. Numerical and experimental results show that this hybrid controller is able to reduce the average sampling rate in comparison to a traditional periodic controller, while achieving the same closed-loop control performance.

In the third contribution, we develop compensation methods for out-of-order communications and time-varying delays using a game-theoretic minimax control framework. We devise a linear temporal coding strategy where the sensor combines the current and previous measurements into a single packet to be transmitted. An experimental evaluation is performed in a multi-hop networked control scenario with a routing layer vulnerability exploited by a malicious application. The experimental and numerical results show the advantages of the proposed compensation schemes.

The fourth contribution proposes a distributed reconfiguration method for sensor and actuator networks. We consider systems where sensors and actuators cooperate to recover from faults. Reconfiguration is performed to achieve model-matching, while minimizing the steady-state estimation error covariance and a linear quadratic control cost. The reconfiguration scheme is implemented in a room heating testbed, and experimental results demonstrate the method's ability to automatically reconfigure the faulty system in a distributed and fast manner.

The final contribution is a co-simulator, which combines the control system simulator Simulink with the wireless network simulator COOJA. The co-simulator integrates physical plant dynamics with realistic wireless network models and the actual embedded software running on the networked devices. Hence, it allows for the validation of the complete wireless networked control system, including the study of the interactions between software and hardware components.

Sammanfattning

Nätverksreglering över trådlösa nätverk blir allt viktigare inom många tillämpningsområden, såsom industriella styrsystem, fastighetsautomation och transportsystem. För att nå en omfattande spridning krävs dock systematiska designverktyg som möjliggör en effektiv resursanvändning och samtidigt garanterar att reglerprestandan uppfylls. Styrsystemen påverkas i hög grad av begränsningar i det trådlösa mediet och av eventuella fel hos systemkomponenterna. I denna avhandling presenteras fem viktiga bidrag som behandlar detta problem.

I det första bidraget studerar vi händelsestyrd reglering. Vi undersöker hur man på ett effektivt sätt kan justera dessa metoder för att nå en önskad reglerprestanda. Vi utformar kommunikationsstrategier för icke-periodisk reglering, och väljer trådlös mediumaccess (MAC) och schemaläggning tillsammans. Experimentella resultat visar att den bästa avvägningen erhålles genom ett hybridschema, som kombinerar händelsestyrd reglering med så kallad hybrid MAC.

Det andra bidraget föreslår en ny händelsebaserad metod för att byta mellan snabb och långsam periodisk sampling. Händelsevillkoret är en kvadratisk funktion av reglersystemets tillstånd. Numeriska och experimentella resultat visar att denna hybridstyrning klarar av att minska den genomsnittliga samplingshastigheten i jämförelse med en traditionell periodisk styrning, samtidigt som man uppnår samma reglerprestanda.

I det tredje bidraget utvecklar vi en metod för att ta hand om sensordata som över en kommunikationskanal får tidsvarierande fördröjning och omkastad ordning. Vi föreslår en spelteoretisk minimax reglerformulering för att lösa detta problem. Vi utformar en linjär tidskodningsstrategi, där sensorn kombinerar nuvarande och tidigare mätningar i varje sänt packet. Metoden utvärderas experimentellt i ett multi-hop nätverk, där en sårbarhet i routinglagret utnyttjas av en skadlig mjukvara. De experimentella och numeriska resultaten visar fördelarna med metoden.

Det fjärde bidraget föreslår distribuerad omkonfigurering av reglernätverk. Vi studerar system där sensorer och ställdon samarbetar för att återställa reglersystemet efter ett fel. Omkonfigurering utförs genom att minimera det stationära skattningsfelets kovarians och den kvadratiska kostnadsfunktionen. Metoden implementeras på ett system för temperaturreglering. Experimentella resultat visar att metoden fungerar väl även i praktiken.

Det slutliga bidraget är en ny systemsimulator som bygger på reglersystemsimulatorn Simulink och den trådlösa nätverkssimulatorn COOJA. Simulatorn integrerar dynamiken från det fysiska systemet med realistiska trådlösa nätverksmodeller och inbyggd programvara på nätverksenheterna. Den kan därför användas för validering av kompletta trådlösa nätverksreglersystem, där man kan studera samspelet mellan programvara, maskinvara och den fysiska processen.

Acknowledgements

Many people have greatly contributed to make the work presented in this thesis possible, for which they deserve a huge thanks. First and foremost I would like to thank my advisor Karl Henrik Johansson. We can say that this adventure started in the summer of 2007 when I came to Stockholm for a year of exchange studies and did my master thesis under your guidance. Crazy that seven years have already passed, and so many great things have happened. I am incredibly grateful for the fantastic learning experience, all the support, great ideas and the opportunities you have always provided me with. Without you I would never have reached this goal. I also want to thank my co-advisor Henrik Sandberg for all the enthusiasm, advice and availability to help over the years.

If it would not have been for João Sousa, I would probably not be writing this thesis. Thank you for introducing me to the research world at FEUP and for initiating the contact with Kalle prior to my exchange studies. I would like to thank Paulo Tabuada for hosting me at the Cyber-Physical Systems Laboratory at UCLA and for all the collaboration we did before, during and after my visit. Your patience, enthusiasm, knowledge and hands-on approach have been a great source of inspiration and have truly helped me in the past two years. I would also like to thank Vijay Gupta for the opportunity to visit and interact with his group at Notre Dame, Maben Rabi for the time at Chalmers and Jin Kim and Woojin Kim for hosting me in Seoul. I am also very grateful to Vikram Krishnamurty for inviting me to spend a great research period at UBC before I started my PhD.

I want to thank all my co-authors and collaborators. It has been an amazing privilege to be able to work and learn with such talented people. Thanks to Alex Wang, Yassine Ariba, Pangun Park, João Faria, Aitor Hernandez, David Andreu, Adolfo Anta, Manuel Mazo Jr., Faisal Altaf, Ubaldo Tiberi, James Weimer, Hamza Fawzi, Yasser Shoukry, Alireza Ahmadi, Behdad Aminian, Mikael Johansson, André Teixeira, Erik Henriksson, Chithrupa Ramesh, Daniel Lehmann, A. Khan, I. Shames, M. Amoozadeh and Piergiuseppe Di Marco. I want to especially thank André for all the support, inspiring discussions, motivation and above all, your friendship. A particular thanks to Phoebus, Christian, Euhanna, Håkan, Farhad, Hamid, Burak and Hamza for all the valuable discussions and support. Many thanks also to my past and current office mates for the excellent company and all the fun. Thanks for all the football matches, the dinners, for the serious talks and the not so serious ones. You have been like a big family to me and an essential part of

this experience. The same goes for my labmates at UCLA, and a special thanks to Matthias for all the awesome time in LA and outskirts. I will never, ever, forget about our hike in 1.5 meter deep snow. Adam is also to blame for that one. I would also like to thank Håkan*, Chithrupa, Themis, Adam, Hamid, Farhad, Yasser, André and Nina for careful proofreading of the thesis. Thanks also to all the faculty at Automatic Control for doing all this fantastic work, hiring amazing people and trying to make us the best in the world at what we do. Many thanks to Anneli, Hanna, Karin, Kristina, Niclas and Pontus for all your help with everything at the department and for all the fun times.

I am grateful to the Swedish Research Council (VR), the Knut and Alice Wallenberg Foundation, the Swedish Governmental Agency for Innovation Systems (VINNOVA) through the projects NECS and WiComPI, and the European Commission through the FeedNetBack and HYCON projects, for the financial support that made this work possible.

I would also like to thank all my friends in Sweden, Portugal and around the world for being awesome and for all the encouragement and happy moments throughout the years.

My heartfelt gratitude goes to my parents and brother, for all the love and for always believing in me. I promise to teach Maria Luis all this stuff when she is old enough! Big thanks also to my parents-in-law for all your support. Most of all, I would like to thank my love Nina for all the patience, care and happiness you have given me. Sorry for all the stressful and long working days. I promise to compensate for that and I am so looking forward to our adventures!

Lastly, I have to thank The Magician, Goldroom, Moonlight Matters, RAC, Bag Raiders and Kitsuné for the insanely good remixes, sets and albums that I have listened to every single day of the PhD journey. Your super sounds are greatly acknowledged.

Thank you!

José Araújo
Stockholm, September 2014

Contents

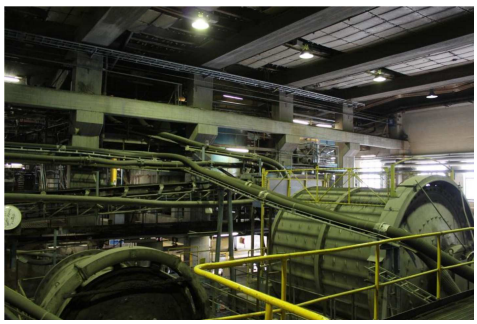
Chapter 1

Introduction

In the last several decades we have seen great advances in computation, communication and control. The proliferation of tiny wireless devices capable of performing computation, communication, sensing and actuation has provided the means to create many intelligent complex systems commonly termed networked control systems (NCSs). In such systems, information from the physical system is sensed and processed using electronic components, and decisions taken are then applied through actuators to the physical world. These systems appear in vast application domains such as industrial control systems, building automation systems, power systems and transportation systems. Such applications are very important, for instance, as they represent the largest energy consumers and CO_2 emission producers. By operating these systems more efficiently one can not only obtain reductions in global energy usage and emissions, but also enhance industrial productivity.

Numerous challenges arise when designing and deploying wireless NCSs. The wireless medium is shared and inherently unreliable. Hence, communications are affected by packet losses, delays and bandwidth limitations. Each wireless device may be required to be in operation for several years on the same battery pack, imposing the need for efficient energy usage. Moreover, some components of a large NCS may necessarily fail during operation. All these issues demand new tools for modeling, design, verification, implementation and validation of NCSs. Research in NCSs has so far only addressed some of these challenges. In this thesis, we focus on the design, implementation and validation of wireless NCSs. We develop theoretical results for such systems, as well as design tools and perform experimental evaluations.

The rest of this chapter is organized as follows. In Section 1.1, we discuss industrial process control systems and building automation systems, which are two motivating applications of wireless NCSs. Illustrative examples of the thesis results are provided in Section 1.2. In Section 1.3, we present the questions which are addressed in the thesis and in Section 1.4, the outline and contributions are given. Finally, Section 1.5 provides the notation and acronyms utilized throughout the thesis.



(a) Boliden Garpenberg ore mine

(b) Iggesund paper mill

Figure 1.1: Boliden's Garpenberg ore mine and Iggesund's paper mill. Wireless deployments have been conducted in both these industrial sites as part of the VINNOVA WiComPi project. (Courtesy of Boliden and Iggesund)

1.1 Motivating applications

Several applications motivate the thesis. In what follows, we discuss the impact of wireless NCSs in industrial process control systems and building automation systems.

Industrial process control

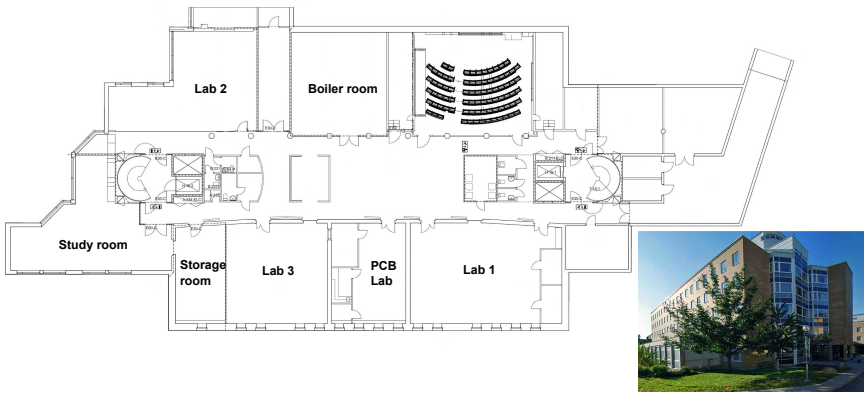
Industrial process control systems can be improved through wireless communications. Typical industrial plants have several hundreds of control loops, with communication between sensors, controllers and actuators performed over wired networks (Samad et al., 2007). In order to improve production quality, more sensing is often required. In general, however, it is hard to introduce new sensors in the wired network but can be accomplished by wireless sensor networks. Additionally, wire elimination in hazardous locations is sometimes desirable for specialized applications. The cost of wiring in an industrial plant can range between 300 to 6000 USD per meter (Samad et al., 2007; Åkerberg et al., 2011).

Figure 1.1 depicts Boliden's Garpenberg ore mine and Iggesund's paper mill. Wireless deployments have been conducted in both these industrial sites as part of the VINNOVA WiComPi project.

Several wireless networking solutions have been proposed for the process industry and are commercially available, e.g., WirelessHART (HART Communication Foundation, 2007) by ABB, Emerson Process Management, Linear Technology and Siemens, as well as Honeywell's OneWireless Network based on the ISA100.11a standard (International Society of Automation, 2010). All these solutions combine platform-specific hardware and protocols, and use the low-power IEEE 802.15.4 standard (IEEE 802.15.4, 2006) as the physical layer. Other standards such as WIA-

(a) The Sino-Italian Ecological and Energy-Efficient Building at Tsinghua University in Beijing was designed to maximize both passive and active solar efficiency, and contains advanced HVAC control systems. (http://www.ecofriend.com)

(b) Wireless control is part of the next generation of intelligent green buildings at the Stockholm Royal Seaport (Courtesy of Folkhem & WingÅrdh Architects).



(c) The floor plan of the KTH HVAC Testbed implemented in the 2nd floor of the School of Electrical Engineering building.

Figure 1.2: The current trend in building automation joins large sensing and actuation capabilities with advanced control algorithms, aiming at large energy consumption and CO_2 emission reductions.

PA (Zhong et al., 2010) and the recent IEEE 802.15.4e (IEEE 802.15.4e, 2012) are also designed for industrial applications.

Even though several products based on wireless communication already exist, only monitoring applications have seen widespread use in industry. This is due to the fact that feedback control applications are intrinsically more complex, and the currently available communication protocols were primarily designed for monitoring applications. The stricter requirements of control applications with respect to reliability and sampling update frequencies are typically not fulfilled by today's systems.

Building automation

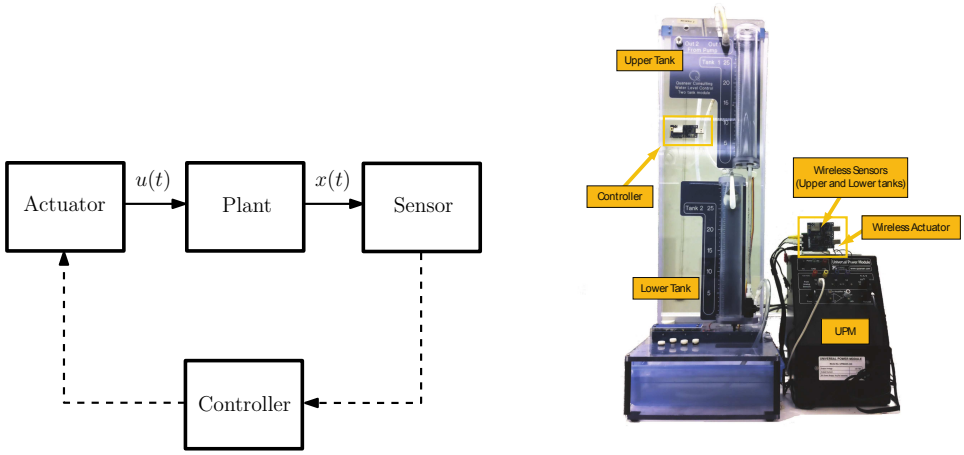
Building automation systems can be considerably enhanced through improved control techniques making use of larger amounts of building, user and weather information. Such data may be gathered by wireless devices, the mobile devices of the building occupants and the internet. Studies indicate that residential, office and commercial buildings account for nearly 40% of the U.S. and European energy consumptions and 30% of the total CO_2 emissions (U.S. Department of Energy, 2008; UK Department of Trade and Industry, 2011; Nest Labs, 2014). Heating, ventilation, and air-conditioning (HVAC) systems are known to be the largest consumers, e.g., accounting for 43% of U.S. residential energy consumption.

Figure 1.2(a) shows the Sino-Italian Ecological and Energy-Efficient Building at Tsinghua University in Beijing. This building was designed to maximize both passive and active solar efficiency, and contains advanced HVAC control systems, making it a zero energy building. The Stockholm Royal Seaport development of residential and office buildings in Stockholm consider wireless control as part of the next generation of intelligent green buildings as depicted in Figure 1.2(b) (Sou et al., 2011). Low-cost wireless sensor and actuator networks are an attractive technology to monitor temperature, humidity, CO_2 levels, light and occupancy (Kim et al., 2009; Lu et al., 2010; Erickson et al., 2011). As the sensors are typically battery powered, their operation must take into account energy constraints. Moreover, system components such as sensors and actuators in HVAC systems are typically subject to high wear and tear and random malfunctions. Such failures may increase operation costs of the HVAC system, cause building damages and user discomfort. Thus, it is required to make the control system resilient to failures. Several wireless products currently exist in the market which aim at providing solutions for the measurement and control of buildings, such as the Nest smart thermostat, Philips Hue LED lights and Belkin WeMo Home Automation products.

Currently, an HVAC control testbed is provided by KTH (Pattarello et al., 2013; KTH, 2014). The testbed is composed of more than 40 wireless sensors measuring temperature, humidity, CO_2 and light intensity in several rooms, as well as many cooling and heating actuators, located in the floor depicted in Figure 1.2(c). From its online platform, users are able to download sensor data and implement advanced control (Parisio et al., 2013; Fabietti, 2014) and fault-detection algorithms (Weimer et al., 2013).

1.2 Illustrative examples

We now provide illustrative examples for some of the problems addressed in this thesis. In the first example, we demonstrate resource-aware control design, where an appropriately devised aperiodic sampling and control mechanism is shown to utilize the NCS resources more efficiently than a classic periodic controller. The second example depicts the consequences of actuator failures in a system with multiple actuators. The impact of delayed and out-of-order communication in a



(a) A wireless networked control system.

(b) Double tank system with wireless sensing, control and actuation.

Figure 1.3: Sensor and actuation data are communicated over a wireless network for the control of the double tank system. Wireless links are depicted by dashed arrows.

control system is finally illustrated in the third example.

Resource-aware control

In this example, we consider the control of a double tank system (Åström and Lundh, 1992) over a wireless network as illustrated in Figure 1.3. Consider the control of the lower-tank level. The time constant of this process is typically between 12 to 20 seconds. The state of the system is denoted as $x(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}$, where $x_1(t)$ and $x_2(t)$ represent the levels at the upper and lower tank, respectively. The control objective is to track a reference value of 8 cm at the lower tank $x_2(t)$. Additionally, a load disturbance is introduced in the upper tank at time $t = 35$ s. We evaluate the control performance under various sampling and transmission strategies. In a wired implementation, a relatively fast sampling rate is typically selected in order to obtain high closed-loop performances and fast disturbance rejection. The response for a sampling period $T = 1$ s is depicted in Figure 1.4. In order to efficiently use the communication bandwidth and the wireless devices energy resources, one can instead use a slow sampling period. The case of a sampling period $T = 5$ s is depicted in the same figure. The closed-loop performance with such slow frequency is clearly deteriorated. Particularly, the disturbance rejection performance is significantly reduced. A question that arises is how can one achieve a similar level of performance as the fast sampling case, but reducing the number of network transmissions. Consider a down-sampled controller strategy, which is based

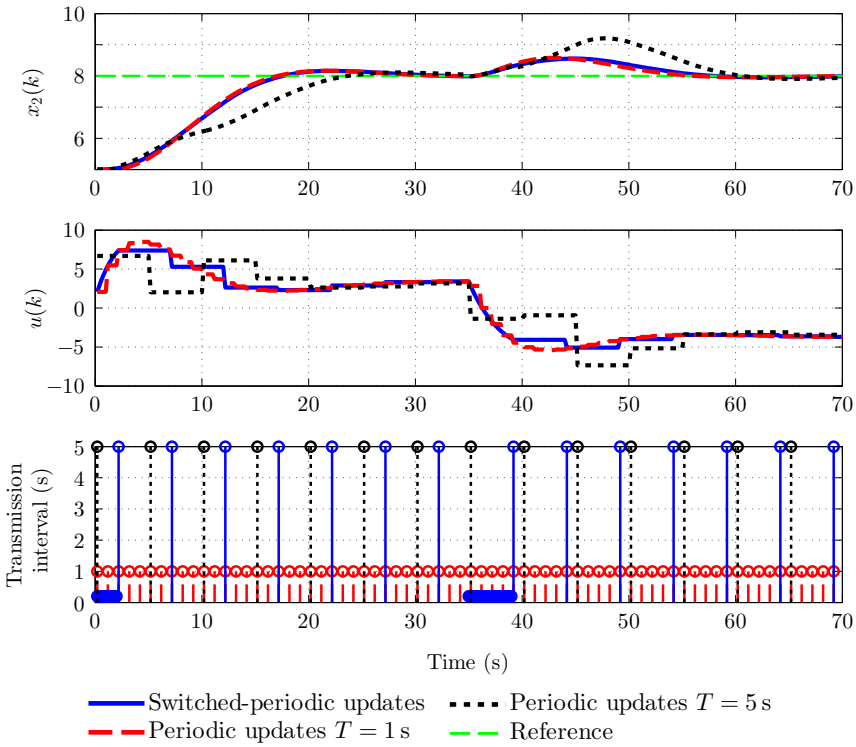


Figure 1.4: Control of the lower tank liquid level $x_2(t)$ of a double tank system under periodic and aperiodic sampling. Sampling the system quickly with a fast sampling of $0.2\,\mathrm{s}$ and then switching to a slow sampling rate of $5\,\mathrm{s}$, achieves similar performances of a system controlled periodically with a period of $1\,\mathrm{s}$, while greatly reducing the number of control loop updates.

on fast periodic transmissions with $T = 0.2\,\mathrm{s}$, followed by slow periodic transmissions every $T = 5\,\mathrm{s}$. Using this strategy, a total of 34 packets is transmitted during the experiment, while using the periodic mechanism with $T = 1\,\mathrm{s}$ period, 70 packets are transmitted, and 15 are transmitted with the slower periodic mechanism. By utilizing the down-sampled mechanism with aperiodic updates, the achieved performance matches the one obtained with the fast periodic mechanism with $T = 1\,\mathrm{s}$, while providing a 50% sampling reduction.

This motivates the study of control techniques that do not rely on the classic periodic control paradigm. The design of these strategies, providing stability/performance guarantees with reduced resource usage, is of great importance in resource constrained wireless NCSs.

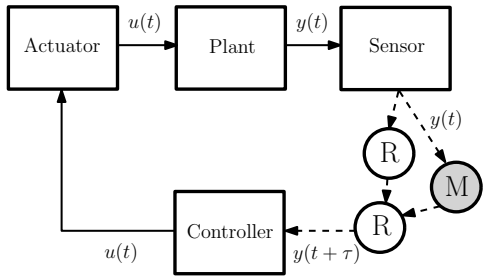


Figure 1.5: Illustration of an NCS where communication between the sensor and controller is performed over a multi-hop relay (R) network. A malicious relay (M) inflicts a variable delay τ on each forwarded packet, affecting the packet delivery order.

Control under out-of-order and delayed communications

We now consider an example where sensor data are subject to variable delay.

Figure 1.5 depicts the scenario where a malicious relay node delays and alters the order of sensor packet deliveries to the controller. Such scenario is applied in the control of the batch-reactor process (Walsh et al., 1999), a fourth-order open-loop unstable system. The sensor is set to periodically sample and transmit sensor data to the controller every 20 ms, over the multi-hop relay network. The malicious relay node is able to influence packet routing in the network so that all packets are relayed through it. In order to remain stealthy, the malicious node delays the packets with a variable delay between 20 ms to 140 ms. A classic output-feedback linear quadratic controller without any delay compensation is used (Åström and Wittenmark, 1990). We now compare the closed-loop performance under: (a) variable delay with out-of-order packet delivery, (b) fixed delay of 140 ms and (c) no delay. Figure 1.6 shows the system outputs, control inputs and the packet delay inflicted in each received packet for the variable delay case. A cross sign indicates a message that is received out of its specified order and a circle indicates a packet received in the correct order. The height of the stem indicates the induced delay. Even though the system remains stable under fixed and varying delay, it is clear that the control performance is deteriorated. Moreover, under variable delay the performance is worse than under fixed delay. Hence, for an efficient operation of wireless NCSs one is required to design and implement mechanisms that guarantee suitable control performances in the presence of variable delay and out-of-order messages. Such mechanisms may be implemented at the sensor, controller, or both, taking advantage of the all the available information.

Control under actuator faults

Wireless NCSs must be resilient to component failures. Figure 1.7 depicts a wireless NCS where several distributed actuators regulate a plant. Sensor data is fused at an

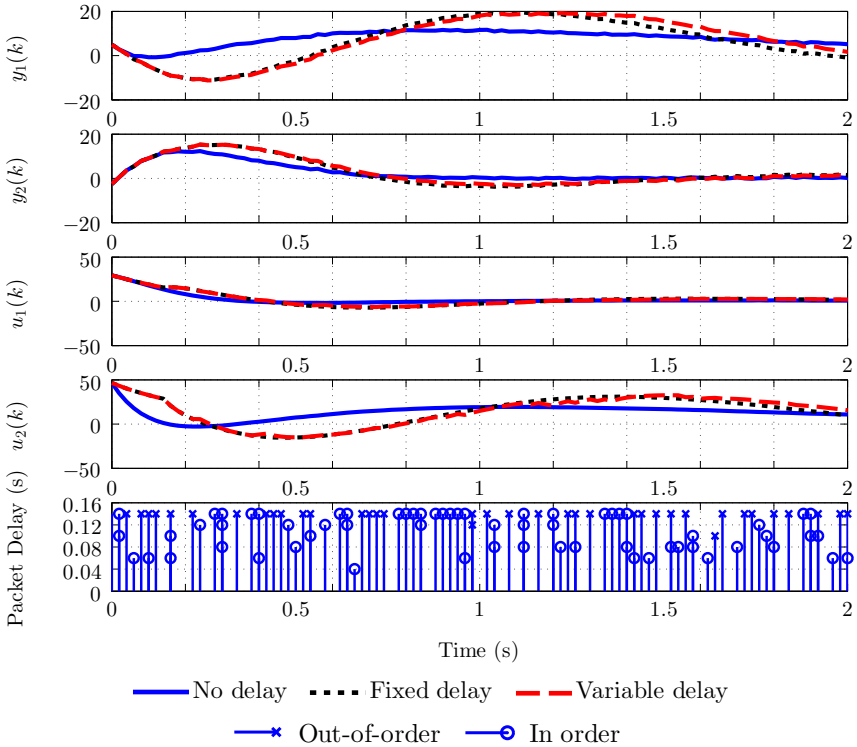


Figure 1.6: Example showing the impact of fixed and variable delay on the closed-loop control performance when regulating a batch-reactor process. The bottom figure shows the effect of the delay on the packet delivery, where a cross mark denotes a packet delivered out of its order, while the circle denotes packet received in the correct order. The height of each stem denotes the delay induced in the received packet.

estimator, which informs the actuators of the current state of the system. This is a scenario typically encountered in building automation (KTH, 2014), where temperature, humidity and CO_2 data is gathered in a centralized unit. This central unit is connected to a supervisory control and data acquisition (SCADA) system, which communicates with several distributed Programmable Logic Controllers (PLCs) through various communication mediums. If actuators are wireless, it is common that feedback from these units to the centralized unit does not exist. Hence, distributed or decentralized fault detection and reconfiguration methods are advantageous in such scenarios. In Figure 1.8 we depict the impact of actuator faults in the temperature control of two adjacent rooms, where $y_1(t)$ is the temperature in room 1 and $y_2(t)$ the temperature in room 2. A temperature of 21°C is to be regulated in both rooms during the first 400 s, followed by a temperature of 20.5°C

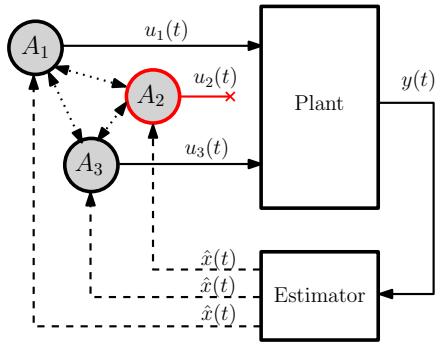


Figure 1.7: A wireless NCS where plant actuation is distributed among several actuator devices. These devices may be subject to failures due to random malfunctions or malicious attacks.

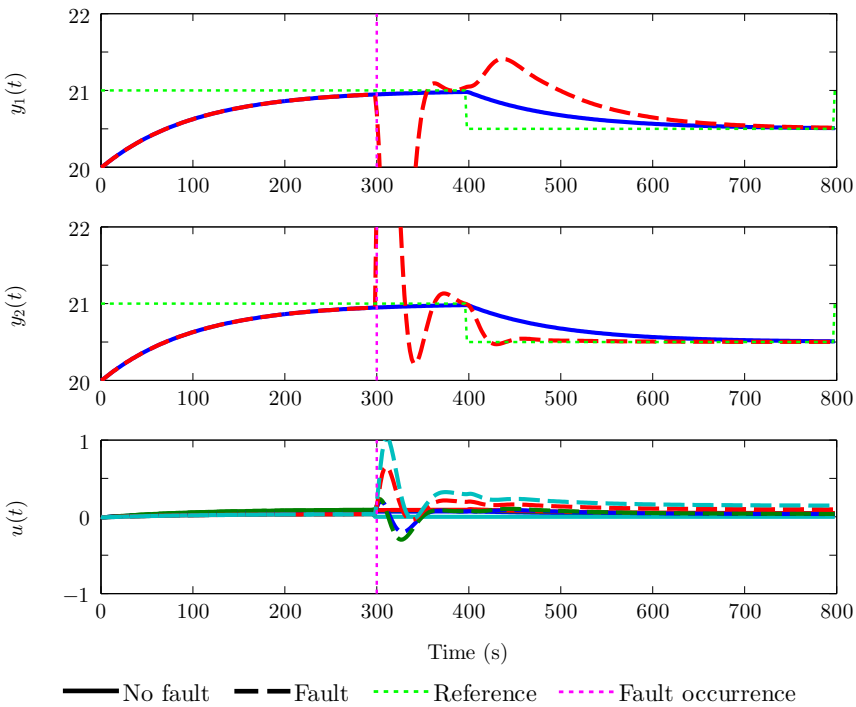


Figure 1.8: Impact of actuator faults in the temperature control of two adjacent rooms. A fault deactivates one actuator at time $t = 300\,\mathrm{s}$.

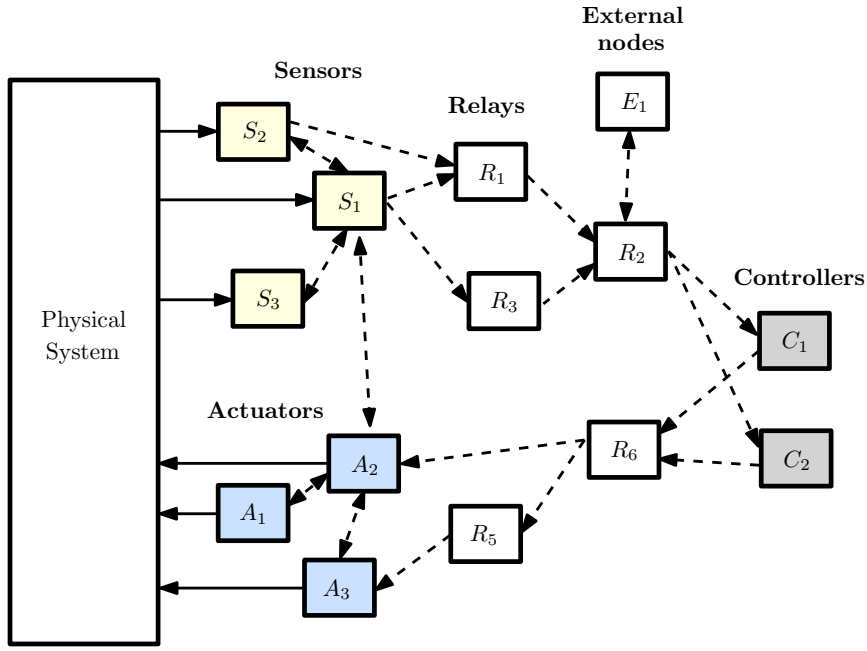


Figure 1.9: Illustration of a wireless networked control system. Sensor data is transmitted to relay nodes, which forward the messages to the controller nodes which compute the control inputs to be applied by the actuator nodes to the physical system. Wireless links are depicted by a dashed arrow, while wired links by a solid arrow. In this thesis, we design, implement and validate techniques that allow the system to use its resources in an efficient manner and be resilient to faults and network imperfections.

for the rest of the time. Four actuators with wireless communications are installed, which perform cooling and heating of the rooms. One of the actuators becomes inactive at time $t = 300\,\mathrm{s}$ because of a fault. If no reconfiguration of the control algorithm at each healthy actuator takes place, the control performance rapidly deteriorates. As the actuators are able to exchange information wirelessly, as well as potentially perform fault detection, automatic reconfiguration schemes may be devised and implemented in a distributed manner. In this way, a safe and efficient system operation would be guaranteed at all times.

1.3 Problem formulation

A wireless networked control system is depicted in Figure 1.9, where the communication between sensors, controllers and actuators is performed over a wireless network which contains relay nodes. The sensor nodes take measurements and transmit them

to controller nodes which compute control inputs to be applied by the actuators. Several external nodes may share the same wireless network, creating additional network traffic.

In the thesis, we address the following questions in the above wireless networked control system setup:

Q1: How can one design and implement triggering conditions which reduce the communication, computation and energy resources, while guaranteeing a desired level of performance of the closed-loop system?

Q2: What are efficient communication mechanisms to perform control with aperiodic sampling?

Q3: What is a suitable compensation scheme for sensor measurements affected by time-varying delay and with out-of-order delivery?

Q4: When sensor and actuator devices fail, how can the remaining nodes handle such faults while maintaining a suitable closed-loop performance?

Q5: How can one perform the implementation and validation of wireless NCSs under realistic plant dynamics and network models, while using the software of the wireless devices?

As illustrated in Figure 1.10, in this thesis we design, implement and validate suitable methods and algorithms which address the above questions. In particular, Chapters 3, 4 and 5 design and implement solutions which allow the system to utilize the wireless medium efficiently by reducing the number of message transmissions while guaranteeing stability and performance properties of the closed-loop system, addressing question **Q1** and **Q2**. In Chapter 6, we design and implement a compensator for delayed and out-of-order communications addressing question **Q3**. Chapter 7 addresses question **Q4** by proposing a distributed reconfiguration method for wireless NCSs under sensor and actuator faults. In Chapter 8 we present a co-simulator which is an implementation and validation tool for wireless NCSs that allows the usage of the software of the wireless devices. This simulator is used to implement and validate the tools presented in Chapters 5 and 6, hence proposing a solution to **Q5**. The validation of methods proposed in Chapters 3 and 7 is performed through lab experiments on process control and building temperature control applications.

1.4 Outline and contributions

We now summarize the remainder of the thesis and introduce the publications for which each chapter is based upon.

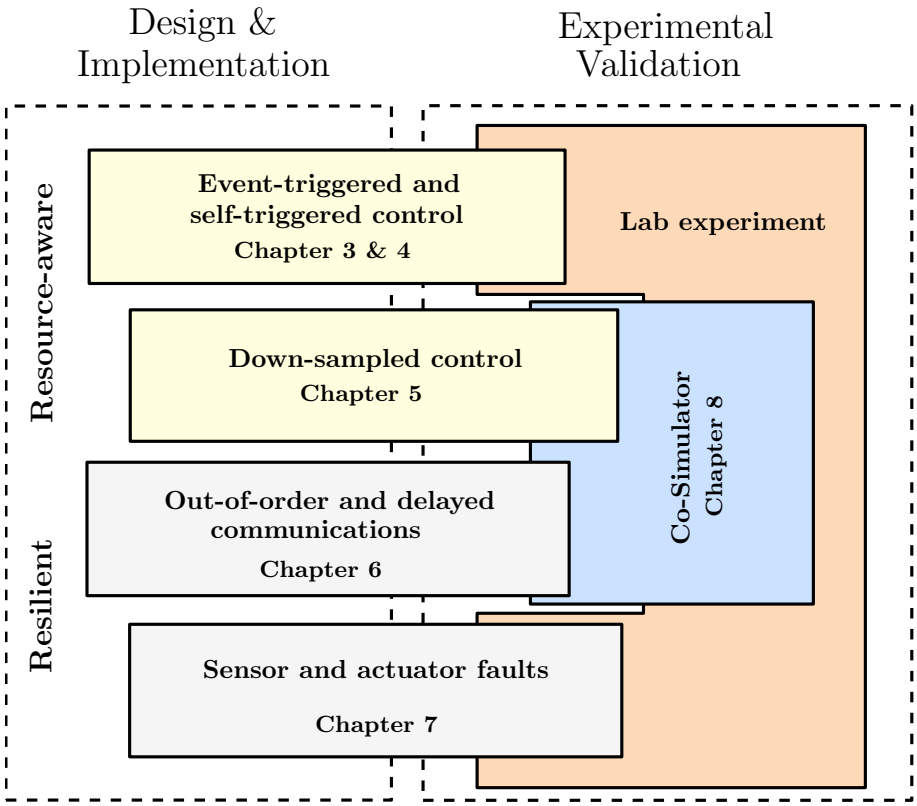


Figure 1.10: Illustration of the contributions of this thesis to the design, implementation and validation of wireless networked control systems.

Background

In Chapter 2, we provide a review of existing literature on the topics covered in the thesis. Particularly, we survey the topics of aperiodic sampling and control design for efficiently utilizing the NCS resources, compensation for network imperfections and reconfiguration of NCSs. Additionally, we review wireless medium-access control schemes and wireless protocols specifically designed for control applications. Finally, we provide a survey of simulators for NCSs, and the experimental platforms used in this thesis for the implementation and validation of the proposed techniques.

Triggering conditions for aperiodic control

Chapter 3 considers self-triggered and event-triggered control techniques and provides methods to tune and execute triggering rules for these aperiodic control strategies. In a self-triggered mechanism, the controller is responsible for computing the

next time instant at which actuation should be updated, by evaluating the control law on fresh sensor measurements. One of the main challenges in self-triggered control is how to perform the exact calculation of the time at which these updates should take place. We present a technique to compute lower bounds on the self-triggered update times in a computationally light manner. Additionally, we propose a Semidefinite Programming-based technique that produces triggering conditions that are less conservative than the existing ones and for which the update times are larger. The algorithms are validated through numerical examples. This work provides solutions to **Q1**.

This chapter is based on the following publication:

- J. Araújo, H. Fawzi, M. Mazo Jr., P. Tabuada, and K. H. Johansson. An improved self-triggered implementation for linear controllers. In *Proceedings of the IFAC Workshop on Distributed Estimation and Control of Networked Systems*, volume 3, pages 37–42, 2012

Communication implementation for aperiodic control

In Chapter 4, we propose an implementation of recently proposed event-triggered and self-triggered control algorithms which have the potential to enable a dynamic trade-off of network resources and closed-loop stability, providing a solution to **Q1** and **Q2**. By showing how these controllers can be implemented over the IEEE 802.15.4 standard, a practical wireless control system architecture with guaranteed closed-loop performance is detailed. Event-based, predictive and hybrid sensor and actuator communication schemes are compared with respect to their capabilities and implementation complexity. A two double tank laboratory experimental setup, mimicking typical industrial process control loops, is used to demonstrate the applicability of the proposed approach. Experimental results show how the sensor communication adapts to the changing demands of the control loops and the network resources, allowing for lower energy consumption and efficient bandwidth utilization.

This chapter is based on the following publications:

- J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson. System architectures, protocols and algorithms for aperiodic wireless control systems. *Industrial Informatics, IEEE Transactions on*, 10(1):175–184, 2014
- J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson. Self-triggered control over wireless sensor and actuator networks. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops*, pages 1–9, 2011

Event-based sampling-rate selection

We propose and evaluate a down-sampled controller where the sampling-rate is selected on an event-based manner in Chapter 5. The devised controller is able to

reduce the network usage while guaranteeing a desired linear quadratic control performance, hence addressing question **Q1** and **Q2**. This method is based on fast and slow periodic sampling intervals, as the closed-system benefits by being brought quickly to steady-state conditions, while behaving satisfactorily when being actuated at a slow rate once at those conditions. The proposed mechanism is shown to provide large savings with respect to network usage when compared to traditional periodic control and other aperiodic controllers proposed in the literature. Additionally, it is shown how the down-sampled controller can be applied to the control of first-order systems with time-delay. Also, a communication implementation strategy is devised for the application of the proposed controller to a scenario where multiple control loops share the same wireless network. Both numerical and experimental evaluations are performed to illustrate the down-sampled controller.

This chapter is based on the following publication:

- J. Araújo, A. Teixeira, E. Henriksson, and K. H. Johansson. A down-sampled controller to reduce network usage with guaranteed closed-loop performance. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014. to appear

Compensator for out-of-order and delayed communications

Chapter 6 studies the design of a minimax controller, where the dynamical system is affected by a disturbance controlled by an adversary, and measurements from the sensor to the controller are affected by time-varying delays and out-of-order delivery. The proposed technique is practically evaluated on the control of a double tank system over a multi-hop wireless network, while a malicious relay node inflicts a random delay in each transmitted packet. The attack is mounted at the routing layer level on the state-of-the-art RPL routing protocol proposed by the Internet Engineering Task Force (IETF). Additionally, we propose a linear temporal coding mechanism for the minimax controller which allows the sensor to combine current and prior sensor measurements in order to increase the estimation performance and robustness to network imperfections. Numerical and experimental results illustrate the performance of the proposed compensator. An answer to **Q3** is given in this chapter.

This chapter is based on the following publication:

- Y. Shoukry, J. Araújo, P. Tabuada, M. Srivastava, and K. H. Johansson. Minimax control for cyber-physical systems under network packet scheduling attacks. In *Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*, pages 93–100, 2013

Distributed reconfiguration for sensor and actuator faults

In Chapter 7, we address the problem of distributed reconfiguration of NCSs under sensor and actuator faults. In particular, we consider systems with redundant sen-

sors and actuators cooperating to recover from faults. The recovery is performed to achieve model-matching, while minimizing a steady-state estimation error covariance and a linear quadratic control cost. It is shown that the reconfiguration and its underlying computation can be distributed. The distributed reconfiguration strategy is then evaluated through a numerical example and also in an HVAC experimental setup for the temperature control of a room. The heaters have wireless capabilities which allows them to detect the fault and perform the distributed reconfiguration in an efficient manner. This chapter provides an answer to question **Q4**.

This chapter is based on the following publication:

- A. Teixeira, J. Araújo, H. Sandberg, and K. H. Johansson. Distributed actuator reconfiguration in networked control systems. In *Proceedings of the IFAC Workshop on Distributed Estimation and Control in Networked Systems*, volume 4, pages 61–68, 2013

A tool for design, implementation and validation of wireless networked control systems

Chapter 8 presents the co-simulator GISOO that integrates Simulink and COOJA and allows for the study and validation of wireless NCSs. GISOO enables users to evaluate actual embedded code for the wireless nodes in realistic networked control experiments, observing the effects of control, communication and computation components, providing an answer to **Q5**. This tool may be used at the design stage in order to, e.g., study suitable sensor transmissions policies, controller parameters and protocol designs, and may also be used during the implementation stage to, e.g., perform trade-off analysis between different protocols, transmission policies and hardware/software interactions. In this way, a wide range of communication solutions can be evaluated without developing abstract models of their control-relevant aspects, and changes made to the networking code in simulations is guaranteed to be translated into production code without errors. Several examples are provided that illustrate the capabilities of the co-simulator. The co-simulator is used in Chapters 5, 7 and 6.

This chapter is based on the following publication:

- B. Aminian, J. Araújo, M. Johansson, and K. H. Johansson. GISOO: a virtual testbed for wireless cyber-physical systems. In *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 5588–5593, 2013

Other publications

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

- A. Khan, J. Araújo, P. di Marco, D. Lehmann, E. Henriksson, H. Sandberg, and K. H. Johansson. Design and implementation of multi-hop wireless pro-

tocols for process control applications. *Industrial Informatics, Transactions on*, 2014. submitted

- J. Weimer, S. A. Ahmadi, J. Araújo, F. M. Mele, D. Papale, I. Shames, H. Sandberg, and K. H. Johansson. Active actuator fault detection and diagnostics in hvac systems. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 107–114, 2012a

- J. Weimer, J. Araújo, M. Amoozadeh, S. Ahmadi, H. Sandberg, and K. Johansson. Parameter-invariant actuator fault diagnostics in cyber-physical systems with application to building automation. In D. C. Tarraf, editor, *Control of Cyber-Physical Systems*, volume 449 of *Lecture Notes in Control and Information Sciences*, pages 179–196. Springer International Publishing, 2013

- J. Weimer, J. Araújo, and K. H. Johansson. Distributed event-triggered estimation in networked systems. In *Proceedings of the 4th IFAC conference on Analysis and Design of Hybrid Systems*, pages 178–185, 2012b

- U. Tiberi, J. Araujo, and K. H. Johansson. On event-based PI control of first-order processes. In *Proceedings of the IFAC Conference on Advances in PID control*, 2012a

- M. Larsson, J. Lindberg, J. Lycke, K. Hansson, A. Khakulov, E. Ringh, F. Svensson, I. Tjernberg, A. Alam, J. Araújo, F. Farokhi, E. Ghadimi, A. Teixeira, D. V. Dimarogonas, and K. H. Johansson. Towards an indoor testbed for mobile networked control systems. In *Proceedings of the First Workshop on Research, Development and Education on Unmanned Aerial Systems*, 2011

- J. Weimer, J. Araújo, A. Hernandez, and K. H. Johansson. Periodic constraint-based control using dynamic wireless sensor scheduling. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 4789–4796, 2011

- F. Altaf, J. Araújo, A. Hernandez, H. Sandberg, and K. H. Johansson. Wireless event-triggered controller for a 3d tower crane lab process. In *Proceedings of the 19th Mediterranean Conference on Control Automation*, pages 994–1001, 2011

- P. Park, J. Araújo, and K. H. Johansson. Wireless networked control system co-design. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pages 486 –491, 2011

- A. Hernandez, J. Faria, J. Araújo, P. Park, H. Sandberg, and K. H. Johansson. Inverted pendulum control over an IEEE 802.15.4 wireless sensor and actuator network. In *Proceedings of the European Conference on Wireless Sensor Networks*, 2011

- J. Araújo, Y. Ariba, P. Park, H. Sandberg, and K. H. Johansson. Control over a hybrid mac wireless network. In *Proceedings of the First IEEE International Conference on Smart Grid Communications*, pages 197 –202, 2010
- J. Araújo, H. Sandberg, and K. H. Johansson. Experimental validation of a localization system based on a heterogeneous sensor network. In *Proceedings of the 7th Asian Control Conference*, pages 465–470, 2009

Contributions by the author

The scientific contribution of the thesis is mainly the author's own work. The co-author order of the papers listed above indicates in most cases the relative contribution to problem formulation, solution, evaluation and paper writing. The implementation and experimental evaluation of the results in the thesis were performed by the author, with the exception of Chapter 5, where David Andreu performed the experiments under the author's supervision. The co-simulator provided in Chapter 8 was designed by the author and implemented by Behdad Aminian.

1.5 Notation and acronyms

Notation

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	n-dimensional vector space over $\mathbb{R}$
$\mathbb{R}^{n\times m}$	Set of real matrices of dimensions $n \times m$
$\mathbb{R}^+$	Set of non-negative real numbers, $\mathbb{R}^+ = \{a \in \mathbb{R} \mid a \geq 0\}$
$\mathbb{N}$	Set of positive natural numbers, $\mathbb{N} = \{1, 2, 3, \dots\}$
$\mathbb{N}_0$	Set of natural numbers, $\mathbb{N}_0 = \{0, 1, 2, 3, \dots\}$
$\mathbb{Z}$	Set of integer numbers
$\det(A)$	Determinant of matrix A
$\operatorname{rank}(A)$	Rank of matrix A
$\operatorname{tr}(A)$	Trace of matrix A
$\operatorname{vec}(A)$	Vectorization operation to matrix A
$\|x\|$	Eucledian norm of vector x: $\sqrt{x^T x}$
$\|x\|_A$	Eucledian norm of vector x, induced by matrix A: $\sqrt{x^T A x}$
$\|A\|_2$	Induced 2-norm of matrix A, $\|A\|_2 = \max_{u\neq 0} \frac{\|Au\|}{\|u\|}$
$\|A\|_F$	Frobenius norm of matrix A, $\|A\|_F = \left(\operatorname{tr}(A^\top A)\right)^{\frac{1}{2}}$
ℓ_2	Hilbert space of all $x \in \mathbb{R}^n$ such that $\|x\| < \infty$
$A^\dagger$	The pseudoinverse of A
$\kappa(A)$	Condition number of matrix A, $\kappa(A) = \|A\|_2 \|A^\dagger\|_2$
$\lambda_{\min}(A)$	Smallest eigenvalue of matrix A
$\lambda_{\max}(A)$	Largest eigenvalue of matrix A
$A \otimes B$	The Kronecker product of matrix A and B
$A \succ 0$	Positive definite matrix A
$A \succeq 0$	Positive semi-definite matrix A
$\operatorname{diag}(a_i, \dots, a_n)$	Diagonal $n \times n$ matrix with a_i to a_n as the diagonal elements
$\operatorname{blkdiag}(A, B)$	Block diagonal matrix with matrices A and B
$\lfloor a \rfloor$	The floor function, $\lfloor a \rfloor = \max\{n \in \mathbb{Z} \mid n \leq a\}$
$\lceil a \rceil$	The ceiling function, $\lceil a \rceil = \min\{n \in \mathbb{Z} \mid n \geq a\}$
$\operatorname{rem}(a, b)$	Remainder after division of a and b, $\operatorname{rem}(a, b) = a - \left\lfloor \frac{a}{b} \right\rfloor b$

Acronyms

ACK	Acknowledgement
ADC	Analog-to-Digital Converter
CAP	Contention Access Period
CFP	Contention Free Period
CSMA	Carrier Sense Multiple Access
CTP	Collection Tree Protocol
DAC	Digital-to-Analog Converter
EDF	Earliest Deadline First
GTS	Guaranteed Time Slot
HVAC	Heating, Ventilation, and Air Conditioning
IAE	Integral Absolute Error
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
LQ	Linear Quadratic
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
MAC	Medium Access Control
NCS	Networked Control Systems
NM	Network Manager
PI	Proportional Integral
PID	Proportional Integral Derivative
RPL	Routing Protocol for Low-power Lossy Networks
SDP	Semidefinite Programming
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Networks
ZOH	Zero-Order Hold

Chapter 2

Background

In this chapter, we provide a short background to wireless networked control systems (NCSs). The thesis builds from existing methods in automatic control and wireless communications. Particularly, the chapter focus on contributions to resource-aware control design through the development of efficient sampling techniques, compensation methods for network imperfections, reconfiguration strategies for control systems under faults, channel access methods and wireless protocols for control, and simulation tools which allow for the validation of NCSs. In the end of the chapter, we provide details on the experimental platforms that are used throughout the thesis.

2.1 Control over wireless networks

We now present an overview of relevant research that addresses control over wireless networks. In wireless NCSs, the interconnection between controllers, sensors and actuators is performed over a wireless channel with limited bandwidth that may introduce delays and loss of information. Often, the wireless medium may be shared with other control loops and applications. The wireless network is a common resource, which in many cases cannot be disregarded when designing the control system. Additionally, wireless devices are often battery powered, imposing computation and communication constraints. Surveys on NCSs and wireless control were provided in (Bushnell, 2001; Tipsuwan and Chow, 2003; Ploplys et al., 2004; Hristu-Varsakelis and Levine, 2005; Yang, 2006; Hespanha et al., 2007; Antsaklis and Baillieul, 2007). An overview of past and future research challenges in this field is given (Samad and Annaswamy, 2011; Kim and Kumar, 2012; Åström and Kumar, 2014; Lunze, 2014).

In (Hespanha et al., 2007; Lunze, 2014), the authors give an overview of some of the most important challenges addressed in the NCSs research community: estimation and control with variable sampling, delays and packet losses. They also review methods for controller synthesis under communication constraints. The existing approaches to NCS design can be mainly grouped into two categories: design of the

control algorithm and design of the communication protocol. Research has targeted the design and analysis of controllers and estimators under packet dropouts (Walsh et al., 1999; Zhang et al., 2001; Schenato et al., 2007; Yu et al., 2004), delays (Nilsson, 1998; Lincoln and Bernhardsson, 2000), data rate limitations (Elia and Mitter, 2001; Ishii and Francis, 2002; Tatikonda et al., 2004; Nair et al., 2007), variable sampling (Heemels et al., 2010), network attacks (Amin et al., 2009) and actuator faults (Kambhampati et al., 2007). These contributions deal with simplistic network abstractions with no network optimization. Wireless sensor network (WSN) communication protocols are typically designed to achieve high reliability and high energy efficiency (Al-Karaki and Kamal, 2004; Bachir et al., 2010). Recently, several approaches consider the joint design of communications and control in wireless networks. In (Demirel et al., 2014; Zou et al., 2012), the authors propose to design scheduling policies, the routing scheme and controllers over a multi-hop network under an energy consumption constraint. In (Park et al., 2011), a framework to optimize the communication protocol with guaranteed control performance and energy consumption of the network devices. The authors of (Liu and Goldsmith, 2004) propose an adaptive tuning scheme for parameters of the link layer, MAC layer and sampling period. A near optimal rate selection for wireless NCSs was proposed in (Saifullah et al., 2012) which is particularly suited to the WirelessHART protocol. The design of the channel access scheme for wireless NCSs is considered as a challenging problem which is still open. For a more extensive discussion of this topic see (Ramesh, 2014).

An extensive set of techniques has been developed to reduce power consumption in WSNs. Unfortunately, the situation is more difficult for wireless NCSs. Traditional control engineering does not consider implementation requirements such as the minimization of communication between sensors, controllers and actuators. Such minimization in a large-scale wireless context is crucial both for energy savings and bandwidth reduction. Existing studies on this topic either neglect the dynamics of the physical system (Rozell and Johnson, 2007; Akyildiz and Kasimoglu, 2004) or do not provide guarantees on the stability or performance of the physical systems being controlled (Ploennigs et al., 2010). In particular, most efforts in NCSs have been conducted under the assumption of periodic sensing and actuation (Antsaklis and Baillieul, 2007), which, in general, may be inefficient and require data rates that are infeasible in a wireless system. It is reasonable to search for strategies that dictate when a particular device needs to exchange information with others in the network. Aperiodic sampling techniques for NCSs is one such proposal in the literature, which we review next.

2.2 Aperiodic sampling for control

Aperiodic sampling techniques have the potential to allow for a more efficient utilization of computation, communication and energy resources in wireless NCSs. This topic has garnered much interest in recent years. We now present an overview

of the research in this topic.

In an NCS, the dynamics of the plant evolve continuously with time, while controllers, sensors and actuators are represented by discrete-event dynamics. These systems are hybrid systems (Antsaklis, 2000). A special case of event-driven dynamics is when actions take place after a certain time has elapsed, while general event-driven dynamics are characterized by asynchronous occurrences of either controlled or natural events (Cassandras and Lafortune, 2008). The idea of performing adaptive sampling dates back to the 1960s, when the closed-loop performance under such scheme was evaluated by (Dorf et al., 1962). This work showed that large sampling savings could be achieved by suitably choosing the sampling rule. This topic gained increased interested in the end of the 1990s, when some of the theoretical foundations of event-triggered control was developed in (Åström and Bernhardsson, 1999, 2002) and practically validated in (Årzén, 1999). Instead of periodic sensor transmissions and control updates, update instants are defined by events taking place at the sensor or controller. The events are generated when a triggering condition is violated. In this way, communications may be reduced, and thereby providing an extension of the battery life span of network nodes. Various event-triggering methods have been proposed and analyzed for deterministic and stochastic systems, considering both linear and nonlinear system models. Optimal event-triggered control has been proposed for first-order linear stochastic systems in (Åström and Bernhardsson, 1999; Rabi and Johansson, 2009; Blind and Allgower, 2011; Molin and Hirche, 2010) and for second-order systems in (Meng and Chen, 2012). The analysis of different triggering rules under disturbances was provided in (Heemels et al., 2008). Lyapunov-based triggering methods for deterministic systems have been proposed for continuous-time systems (Tabuada, 2007; Velasco et al., 2009; Fiter et al., 2012; Marchand et al., 2013), output-feedback control (Donkers and Heemels, 2012), tracking (Tallapragada and Chopra, 2013), quantized control (Li et al., 2012; Tallapragada and Cortés, 2014) and discrete-time systems (Heemels et al., 2013; Postoyan et al., 2013). Self-triggered control was introduced in (Velasco et al., 2003) as an approach to aperiodic control. In this case, the next triggering time is based on the current measurement and the plant's model. This can be seen as an emulation of the event-triggered scheme: there is no need for continuously monitoring a triggering condition, but instead sensor nodes can be turned off between sampling instants. Strategies using this triggering technique have been proposed for both linear and nonlinear plants (Wang and Lemmon, 2009; Mazo Jr. et al., 2009; Anta and Tabuada, 2009; Mazo Jr. et al., 2010; Anta and Tabuada, 2010a; Almeida et al., 2014). A model-based approach to event-triggered control design was proposed in (Lunze and Lehmann, 2010; Lehmann et al., 2012; Garcia and Antsaklis, 2013; Garcia et al., 2014). In this case, the event-triggered strategy is designed to be robust to model uncertainties and disturbances. Model-predictive event-triggered (Varutti et al., 2009; Kilkki and Bjorkbom, 2013) and self-triggered (Henriksson et al., 2012; Barradas Berglind et al., 2012; Kobayashi and Hiraishi, 2012) control, has also been considered even in relation to a radio communication model (Cardoso de Castro et al., 2012). Recently, (Yu and Antsaklis,

2013) proposed event-triggered controllers based on passivity theory, under delays and quantization. The above techniques have also been used in decentralized or distributed systems, e.g., (Mazo and Tabuada, 2008; Mazo Jr. and Tabuada, 2011; Wang and Lemmon, 2011; De Persis et al., 2010) with Lyapunov-based techniques, considering multi-agent systems (Teixeira et al., 2010; Seyboth et al., 2011; Dimarogonas et al., 2012; Eqtami et al., 2012; Fan et al., 2013; Nowzari and Cortés, 2012; De Persis and Frasca, 2013) and a model-based approach (Garcia et al., 2013; Stöcker et al., 2013). Moreover, several approaches have considered network constraints and its effects, e.g., channel access (Cervin and Henningsson, 2008; Ramesh et al., 2011; Henriksson et al., 2012; Tiberi et al., 2013), CPU availability (Samii et al., 2010), delays and losses (Wang and Lemmon, 2011; Lehmann and Lunze, 2012; Ramesh et al., 2013) and multiple control loops sharing the same network (Al-Areqi et al., 2013; M.H. Mamduhi, 2014). In (Xu and Hespanha, 2004; Cogill, 2009; Molin and Hirche, 2013, 2014), the authors formulate the event-triggered controller design as an optimization problem, where the cost function considers the transmission rate. Additionally, event-triggered techniques have been applied in estimation problems (Sijs and Lazar, 2012; Weimer et al., 2012b; Trimpe and D'Andrea, 2012; Wu et al., 2013) and optimization (Zhong and Cassandras, 2010; Lemmon, 2011; Cassandras, 2014). They have also gained interest in industrial process control, and the design of PI/PID control has been proposed in (Årzén, 1999; Miskowicz, 2006; Vasyutynskyy and Kabitzsch, 2007; Rabi and Johansson, 2008; Durand and Marchand, 2009; Sánchez et al., 2011; Ploennigs et al., 2010; Ruiz et al., 2014), providing stability guarantees (Tiberi et al., 2012a; Chacón et al., 2013; Beschi et al., 2014), addressing actuator saturation (Kiener et al., 2014) and for self-triggered control (Tiberi et al., 2012b). Some of these methods have also been experimentally validated (Yook et al., 2002; Camacho et al., 2010; Lehmann and Lunze, 2011; Trimpe and D'Andrea, 2011; Altaf et al., 2011). In Chapters 3 and 4 we consider triggering rules and communication implementations for Lyapunov-based event- and self-triggered controllers over a wireless NCS. Surveys of event- and self-triggered control can be found in (Aström, 2008; Heemels et al., 2012; Grüne et al., 2014) and (Fiter et al., 2014).

Scheduling of sensing and control, based on either offline or online optimization techniques have also been proposed in the literature. A problem considered is that of scheduling multiple plants over a single processor or a single communication channel. In (Walsh and Ye, 2001) it was demonstrated that dynamic scheduling of NCSs can outperform fixed schedules. Feedback scheduling was proposed in (Cervin et al., 2002; Cervin, 2003), followed by (Martí et al., 2004; Henriksson and Cervin, 2005; Cervin et al., 2010) considering both deterministic and stochastic systems. See (Lozoya et al., 2013) for an extensive survey. The offline scheduling of both control laws and the access of multiple plants over a single communication medium was analyzed in (Lincoln and Bernhardsson, 2002; Zhang and Hristu-Varsakelis, 2006).

It is essential to trade-off control performance to average transmission rate. In (Antunes and Heemels, 2014), the authors proposed a roll-out event-triggered

controller that guarantees better control performance than periodic control. Due to the numerical complexity of the proposed method, the authors resort to computing the best scheduling sequence in a receding-horizon manner. A self-triggered linear quadratic regulator was proposed in (Barradas Berglind et al., 2012; Gommans et al., 2014) which by design guarantees that the obtained quadratic control cost does not exceed a specific maximum value. The method is shown to achieve lower control costs than periodic control at the same transmission rate. However, no guarantees are provided that the transmission rate does not exceed the periodic control rate. In (Antunes, 2013), lower and upper bounds on performance and transmission rate are computed for a given event-triggering method under impulse disturbances and Gaussian noise. The problem of minimum attention control introduced in (Brockett, 1997) has been recently revisited in (Anta and Tabuada, 2010b; Yépez et al., 2011; Donkers et al., 2014), where the goal is to maximize the time between consecutive executions of the control tasks. In this case, the computation of the sampling instant and the control law is performed online and these mechanisms are jointly designed. Stability and performance guarantees are provided using Lyapunov-based techniques in (Anta and Tabuada, 2010b; Donkers et al., 2014) and Linear Quadratic (LQ) control methods in (Yépez et al., 2011). In the work by (Souza et al., 2014), the authors propose a self-triggered controller which selects a sampling interval within a set of predefined sampling periods in order to optimize the H_2 and H_∞ norm. It was recently shown in (Shi et al., 2013) that the optimal transmission schedule for particular linear systems with a finite amount of transmissions over a finite horizon is to transmit consecutively at the beginning of the interval. This is in line with the event-based sampling-rate selection method we propose in Chapter 5.

2.3 Compensation of network imperfections

Communication over a wireless network gives rise to several issues that must be taken into consideration when designing the wireless NCSs. The network is inherently lossy, causing packets to be lost. Each packet is typically affected by both channel access and transmission delays. Additionally, device failures and malicious behavior may strongly degrade the closed-loop system performance. Hence, methods to properly compensate for such artifacts must be devised.

Since the introduction of NCSs, the compensation of fixed and time-varying delays, as well as data loss, have been the focus of much research. The research on time-delay systems has particularly focused on the study of stability of time-varying delays (Gu et al., 2003; Fridman, 2014). Optimal control under communications affected by packet losses has been studied in (Seiler and Sengupta, 2005; Schenato et al., 2007; Gupta et al., 2007, 2009). The network is typically modelled by a Bernoulli or a Markovian process. The stability analysis of the system under delays and losses has been proposed in (Zhang et al., 2001; Heemels et al., 2010). The controller design under time-varying delays has been addressed in (Luck and Ray,

1990, 1994; Nilsson, 1998; Yue et al., 2004). The design of distributed and decentralized controllers with communication delays has received particular attention in the last years (Lamperski and Doyle, 2011; Matni and Doyle, 2013; Feyzmahdavian et al., 2012). Even though the influence of short delays has been extensively studied, the effect of long delays and out-of-sequence messages has received less attention from the control community. The typical solution to out-of-sequence messages is (1) utilize a buffer with length equal to the maximum expected message delay, thus avoiding any out-of-sequence issues (Luck and Ray, 1990, 1994; Nilsson, 1998) or (2) discard any out-of-sequence messages, assuming that the penalty for not using such packets is low (Tang and de Silva, 2006b; Hespanha et al., 2007; Liu et al., 2014). Such approaches may not be suitable since, in case (1), a fixed delay is introduced in the system, and in (2), such strategy may discard a large amount of messages for persistently out-of-sequence message deliveries.

Optimal control under long delays and out-of-sequence measurements for linear stochastic systems is discussed (Lincoln and Bernhardsson, 2000). Even though out-of-sequence measurements are utilized to improve the state estimate, a new control actuation is not performed whenever this occurs, so it can suffer the same drawbacks as the buffer approach. A similar approach is described in (Hirano et al., 2005). In (Schenato, 2008) the author considers the estimation of a linear stochastic system under delays and packet losses where reordering of packets occurs. Several researchers have addressed the problem of optimal estimation under out-of-sequence measurements (Bar-Shalom, 2002; Zhang et al., 2005; Westenberger et al., 2012) but with no consideration of the control counterpart. In Chapter 6 we propose an opportunistic robust output-feedback controller with packet reordering which outperforms the buffer approach mentioned above.

Several other methods have been proposed to improve the estimation and control performance in networked systems. Anytime control was proposed in (Quevedo and Gupta, 2013), packetized predictive control in (Tang and de Silva, 2006b; Quevedo and Nesic, 2011; Fischer et al., 2013; Cunguara et al., 2013) and the transmission of redundant data in (Mesquita et al., 2012). Additionally, compensators to deal with communication outages and packet loss have been designed in (Yu and Fu, 2013; Moayedi et al., 2013; Henriksson, 2014; Gommans et al., 2013). Linear temporal coding was investigated in linear stochastic lossy systems in (Robinson and Kumar, 2007) scalar, and in (He et al., 2013) and (Suia et al., 2014) for the general case. Using this technique, the current and previous measurements at the sensor node are combined in a single packet and transmitted to the estimator in order to achieve larger robustness to packet losses. In (Blind et al., 2009), the authors investigate how to combine multiple sensor values from the current time instant, in order to also robustify the control system when data is transmitted over a lossy network. A linear temporal coding mechanism is devised in Chapter 6, which aims at improving the estimation performance of the proposed controller.

2.4 Control system reconfiguration

Wireless NCSs are composed of components which are prone to failure. In the event of malfunction in actuators, sensors or other system components, the control system may exhibit poor performances or even become unstable (Blanke et al., 2006; Poovendran et al., 2012). Resilient and fault tolerant control is targeting this challenge (Blanke et al., 2006; Isermann, 2005; Ding, 2008).

Since the 1970s, much research has been conducted in fault-tolerant control systems, fault detection and diagnosis (FDD) and reconfigurable control. Several methods have been documented (Patton, 1997; Blanke et al., 2006; Lunze and Richter, 2008; Zhang and Jiang, 2008; Ding, 2008; Hwang et al., 2010). FDD deals with the identification of faults while reconfigurable control proposes methods to reconfigure a system after a fault has been detected and diagnosed. The typical objectives of reconfiguration are to maintain stability, the closed-loop dynamics (model-matching) or minimize the loss in performance. Many types of faults in actuators, sensors and other system components have been considered in both linear and nonlinear systems. The proposed methods are based on a pseudo-inverse approach (Gao and Antsaklis, 1991; Staroswiecki, 2005b; Ciubotaru and Staroswiecki, 2010), optimization based (Maciejowski, 1997; Kambhampati et al., 2007; Härkegård and Glad, 2005; Staroswiecki and Cazaurang, 2008), adaptive control (Tao et al., 2001; Chen et al., 2002), requiring that the nominal controller remains in operation (Lunze and Steffen, 2006; Richter et al., 2011; Seron et al., 2013), analyzing the impact of the fault detection duration in the reconfiguration (Staroswiecki et al., 2007), considering the case when full recovery is not possible (Staroswiecki and Cazaurang, 2008; Staroswiecki and Berdjag, 2010; Wu et al., 2000). Additionally, authors have addressed FDD and reconfiguration under sensor faults (Hoblos et al., 2000; Staroswiecki, 2003, 2005a), sensor and actuator failures (Richter et al., 2011; Seron et al., 2013) and actuator failures with sensor bias (Joshi and Patre, 2013). However, the vast majority of solutions rely on a centralized approach. Such techniques may be impractical due to technical and economical constraints (Åkerberg et al., 2011; Kim and Kumar, 2012). Distributed FDD and reconfiguration has been much less explored. The architecture of such systems is discussed in (Campelo et al., 1999; Staroswiecki, 2005a; Voulgaris and Jiang, 2004; Patton et al., 2007; Jin and Yang, 2009), while in (Yang et al., 2010) a distributed FDD is employed to perform a centralized reconfiguration. In (Patton et al., 2007) the authors propose a hierarchical architecture for reconfiguration and study the performance of a decentralized reconfiguration scheme. More recently, authors have considered distributed and decentralized reconfiguration via plug-and-play methods when new sensors, actuators or physical systems are added to the complete control system (Bendtsen et al., 2013; Riverso et al., 2013; Bodenburg and Lunze, 2013). In Chapter 7 we design and implement a distributed reconfiguration method to handle sensor and actuator faults.

2.5 Medium-access control

For the system-theoretic analysis and design of wireless NCSs, it is necessary to abstract the communication network by appropriate mathematical models. The thesis mainly deals with network models that capture the influence of medium-access control (MAC). In this section, we present a brief overview of wireless multiple access schemes. Since the wireless medium cannot support simultaneous transmissions, mechanisms must be provided to define how each wireless device accesses the network. The channel access control mechanisms provided by the MAC are known as multiple access protocols (Rom and Sidi, 1990). These protocols make it possible for several network nodes to be connected to the same physical channel. The MAC protocols are commonly classified as contention-based or conflict free protocols. We describe them below.

Contention-free MAC

Contention-free MAC ensures that a transmission is successful as long as the physical medium does not cause any losses. This is achieved by allocating the channel to one of the users by a centralized network coordinator, using a static or dynamic schedule. The channel resources can be divided among the users in time, frequency or space:

- Time Division Multiple Access (TDMA)
- Frequency Division Multiple Access (FDMA)
- Code Division Multiple Access (CDMA)

Static schedules may waste the available bandwidth of the network, so for this reason dynamic schedules are preferred. Information must then be exchanged regularly between the central scheduler and the network nodes. Time synchronization is important, so synchronization messages are required to be exchanged between the coordinator and the devices.

Contention-based MAC

In contention-based MAC protocols, nodes compete for the medium. The common mechanism to handle channel collisions is the Carrier Sense Multiple Access (CSMA) scheme. A transmitting node then tries to detect the presence of an encoded signal from any other transmitting node before attempting to transmit. If another transmission is sensed, the node keeps on sensing the channel with probability p. This scheme is commonly known as a p-persistent CSMA scheme. CSMA with collision avoidance (CSMA/CA) delays the transmission of a message by a random amount of time whenever the channel is found busy.

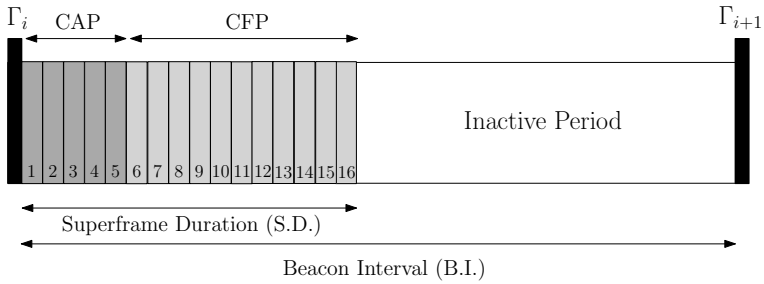


Figure 2.1: Superframe structure of the IEEE 802.15.4 MAC protocol. The time Γ_i is the time at which the beacon message is sent and the superframe i begins. The message transmission takes place during the CAP and CFP. In the inactive mode, the nodes go to a low-power mode to save battery.

Hybrid MAC

Hybrid MACs have both contention-free and contention-based schemes. They allow the trade-off between the advantages of contention-free and contention-based MACs. An example of such MAC is the IEEE 802.15.4 protocol (IEEE 802.15.4, 2006) which we describe in the following section.

IEEE 802.15.4 MAC Protocol

The standardization of low data rate and low power protocols for wireless networks is an ongoing process and there is not yet any widely accepted complete protocol stack for control (Willig, 2008). The IEEE 802.15.4 protocol (IEEE 802.15.4, 2006), which specifies physical and MAC layers, is the base of solutions in industrial automation such as WirelessHART (HART Communication Foundation, 2007), ISA100 (International Society of Automation, 2010) and the TSMP protocol (Pister and Doherty, 2008).

The IEEE 802.15.4 standard specifies two types of medium access mechanisms depending on whether the network is in the beacon-enabled or the non beacon-enabled mode. Here we will focus in the beacon-enabled mode. In such a setup, a centralized coordinator node, the Personal Area Network coordinator or Network Manager (NM), is responsible for synchronizing and configuring all the nodes. The synchronization and configuration takes place periodically at each beacon message which defines the time bounds of the superframe structure defined by the protocol. We denote by Γ_i the time instants at which the beacon is transmitted. The superframe length is named Beacon Interval (B.I.) and has B.I.=aBaseSuperFrameDuration $\times 2^{\text{B.O.}}$ symbols, with $0 \leq \text{B.O.} \leq 14$, where B.O. is the Beacon Order and aBaseSuperFrameDuration is defined by the protocol and specifies the shortest superframe duration. Moreover, each symbol corresponds to $T_{\text{symbol}} = 16\mu\,\text{s}$. The B.I. is further divided in active and inactive periods, as shown

in Figure 2.1. The active period has a time interval defined by Superframe Duration (S.D.) and is divided in 16 equally sized slots of length aBaseSlotDuration. The superframe duration satisfies the equality S.D. = aBaseSuperFrameDuration $\times 2^{\text{S.O.}}$ symbols, with $0 \leq \text{S.O.} \leq 14$, and where S.O. is the Superframe Order. aBaseSlotDuration specifies the shortest slot duration, corresponding to S.O.= 0. The active period is split into a Contention Access Period (CAP) and a Collision Free Period (CFP). During the CAP, nodes transmit best effort messages and the MAC scheme is CSMA/CA. In case the channel is busy, the node applies a random backoff and retries up to macMaxCSMABackoffs attempts. If the channel is idle, the node transmits. A successful transmission is acknowledged by the receiving node by an acknowledgement (ACK) message sent to the transmitter. If the ACK is not received due to collision or network interference, the node retries to transmit the packet up to a maximum of macMaxFrameRetries attempts. The CFP is intended to provide real-time service, by allocating Guaranteed Time Slots (GTS) to the nodes using a TDMA scheme. Since during the CFP there are no packet losses due to collisions or channel congestion, CFP is attractive for real-time control. The total number of GTS slots is limited in the standard to seven. The standard specifies the scheduling of the GTS following a first-come-first-served (FCFS) request-based scheme. At each CAP, the nodes requiring a GTS send a request to the NM which will allocate the slot to the node if there are available GTSs. An inactive period is defined in the end of the active period so that the network nodes and the NM enter a low-power mode to save energy. After this period, all the nodes leave the low-power mode in order to receive the next beacon message.

2.6 Protocols for industrial control applications

Many wireless protocols have been proposed in the literature. In this section we discussed some of the most relevant ones for our studies.

WirelessHART provides reliable and secure networking services for mesh networks (HART Communication Foundation, 2007). The standard is based on a fully compliant IEEE 802.15.4 (IEEE 802.15.4, 2006) physical layer and channel access mechanism. Recent works have proposed methods to improve the performance of WirelessHART by introducing real-time constraints, e.g., (Saifullah et al., 2010), SRDR (Han et al., 2011)). ISA100.11a overcomes some of the limitations of WirelessHART by guaranteeing compatibility with IPv6-based routing solutions to offer more flexibility and interoperability. Both WirelessHART and ISA100.11a (International Society of Automation, 2010) are centralized protocols. The evaluation of a single-hop wireless control of an industrial bioreactor using a single WirelessHART transmitter has been presented in (Blevins et al., 2012). Recent studies have evaluated the application of PID control techniques over WirelessHART (Han et al., 2010; Blevins, 2012; Blevins et al., 2014). However, the experimental validations of a network of WirelessHART or ISA 100.11a devices for closed-loop control, has not been yet reported in the literature.

There are several protocols proposed for control applications. The EARQ protocol (Heo et al., 2009) includes real-time routing constraints. SERAN (Bonivento et al., 2007) and TREnD (Di Marco et al., 2010) are examples of cross-layer MAC and routing solutions for control applications in which protocol parameters are dynamically tuned to minimize the energy consumption while fulfilling latency and reliability constraints. In (Valle et al., 2012), the authors propose a modification of the IEEE 802.15.4 MAC for usage in industrial environments. The protocols in (Burri et al., 2007), (Di Marco et al., 2010) can satisfy control requirements by design, but they are not compliant with the IEEE 802.15.4 standard. The authors of (Silvo et al., 2013) propose a protocol that aims at improving reliability and reducing latency by relying on collaborative retransmissions, "piggybacking" and spatial diversity. In (Shen et al., 2014) and (Shen et al., 2013) the authors propose a priority-based MAC and routing scheme which is compatible with WirelessHART. In a similar research direction, the authors of (Christmann et al., 2014) and (Ramesh et al., 2014) propose a tournament-based MAC and a channel access scheme which aims at allowing efficient wireless NCS deployment. The latest amendment of the standard IEEE 802.15.4e-2012 (IEEE 802.15.4e, 2012) is proposed for factory automation.

Several routing protocols have been proposed, including collection protocols. A major contribution is the collection tree protocol (CTP) (Gnawali et al., 2009). CTP uses link quality estimations for parent selection, data path validation, and adaptive beaconing to provide reliable and efficient operations. The internet engineering task force (IETF) has proposed a routing protocol for low power network (RPL) (Winter et al., 2012), which is based on the core mechanisms of CTP. The protocol stack for the "Internet of Things" has been proposed to be based on IETF RPL and IEEE 802.15.4e MAC (Palattella et al., 2013). An advantage of data collection routing is its implementation of distributed re-routing and fault recovery. However, data-collection protocols currently do not address real-time requirements and do not cope well with node failures. We proposed in (Khan et al., 2014) a wireless protocol for process control applications which combines IEEE 802.15.4e-2012 MAC and a modified CTP routing protocol enabling fast re-routing.

Recently, the design of a high-reliability and low-latency wireless communication system for industrial control has been considered in (Weiner et al., 2014). The system is designed for star network topologies, and does not rely on IEEE 802.15.4 as the physical layer. The authors of (Moraes et al., 2007) propose a virtual token passing procedure to enable real-time communication over the IEEE 802.11 for industrial systems.

2.7 Co-simulators for wireless control

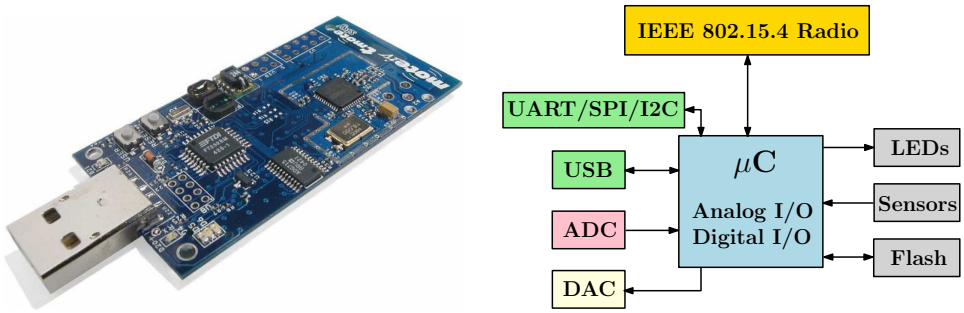
Simulation tools are important for the validation of communication, computation and control aspects of wireless NCSs. Several tools are available at different abstraction levels. We provide an overview of some of these simulators.

MATLAB/Simulink is the most widely used tool for modelling and simulation of control systems. The Truetime simulator (Cervin et al., 2003) is a toolbox implemented in MATLAB/Simulink which facilitates simulation down to the link layer. Ptolemy (Eker et al., 2003) and Modelica (Mattsson et al., 1998) also provide means to perform model-based simulation of wireless NCSs. Packet-level network simulation can be achieved using ns-2 and OMNet++, both of which provide high-level protocol stack implementation. WirelessHART simulators have been implemented for both ns-2 (Zand et al., 2014) and OMNet++ (Ferrari et al., 2014). In (Branicky et al., 2003), the authors developed an extension to ns-2 that allows for the implementation of physical systems and controllers by using an ODE solver. The PiccSIM toolchain (Kohtamaki et al., 2009; Björkbom et al., 2011) provides an integration of MATLAB/Simulink with ns-2, allowing for control system design and automatic C code generation. Similarly, NCSWT (Eyisi et al., 2012) integrates MATLAB/Simulink and ns-2 for simulating networked control systems for a specific application. All the above tools require the application code and communication protocols to be modelled and developed for their native simulation tool. In these cases, the simulation accuracy is dependent on the network modelling capabilities of each simulator. Moreover, the high-level wireless models in ns-2 are not able to capture the low-level communication interactions among network nodes or the interaction between the device's hardware and software.

Emulation of the wireless devices with realistic wireless network models has become a priority in recent simulator developments. Such features are provided in the wireless sensor network tools TOSSIM (Levis et al., 2003), COOJA (Osterlind et al., 2006) and others (Mirkovic and Benzel, 2012). Both TOSSIM and COOJA are able to capture complex communication dynamics and emulate real code. The tool developed in (WCPS, 2013) integrates Simulink and TOSSIM and has been applied to a wireless structural control applications (Li et al., 2013). The TOSSIM simulator emulates the TinyOS operating system (Levis et al., 2004), solely for the micaZ wireless platform. The architecture of the NCS provided by the WCPS simulator is fixed and tailored only for data collection applications, where solely sensor devices and relay nodes are wireless, while the controller and actuators are required to be implemented in Simulink. Additionally, in WCPS, communication between the nodes and Simulink is performed using WCPS-specific functions instead of the native peripherals of the wireless platform. Because of this, considerable modifications to the wireless device code is required before it can be used in real scenarios. On the other hand, COOJA can emulate additional wireless platforms using both TinyOS and ContikiOS (Dunkels et al., 2004) and works at networking, operating system and machine code levels.

2.8 Experimental platforms

We now introduce two experimental platforms used for the implementation and validation of the proposed tools in this thesis.



(a) Tmote sky/Telos device (b) Device schematic

Figure 2.2: Tmote sky/Telos wireless device and its schematic

Wireless devices

Both experimental platforms use the Telos wireless devices depicted in Figure 2.2(a), with the corresponding schematic in Figure 2.2(b) (Polastre et al., 2005). These nodes are equipped with a Texas Instruments MSP430 16-bit, 8 Mhz microcontroller with 48 kB of Flash and 10 kB of RAM memory and a 250 kbps 2.4GHz Chipcon CC2420 IEEE 802.15.4 compliant radio. Furthermore, the node has integrated Analog-to-Digital (ADC), Digital-to-Analog (DAC) converters and serial communication buses, allowing the nodes to be used as sensor and actuator nodes. All the devices are programmed using TinyOS (Levis et al., 2004).

Physical systems

We now describe the physical system model details for the two experimental systems.

Double tank

The double tank system (Åström and Lundh, 1992; Quanser, 2014) consists of a pump, a water basin and two tanks of uniform cross sections. Figure 2.3 depicts the experimental apparatus and a diagram of the physical system. The water in the lower tank flows to the water basin. A pump is responsible for pumping water from the basin to the upper tank, which then flows to the lower tank. The sensing of the water levels is performed by pressure sensors placed under each tank. A wireless node interfaces the sensors with an ADC, in order to sample the sensors for both tanks. The actuation is implemented through the DAC of the wireless actuator node which actuates the pump.

Let L_1 and L_2 denote the levels in the upper and lower tank, respectively, and V_p denotes the voltage applied to the pump. The linear model of the double

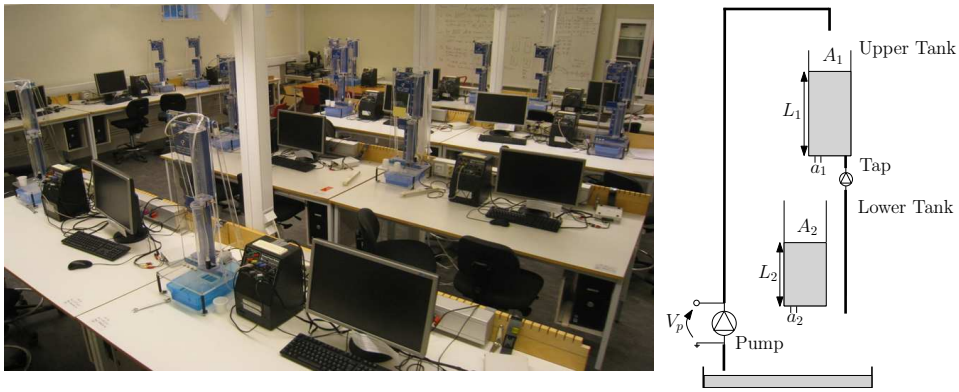


Figure 2.3: Double tank system testbed and its diagram.

tank system obtained by linearizing its nonlinear model around an operating point L_{10}, L_{20} is given by:

$$\dot{x}(t) = Ax(t) + Bu(t),$$

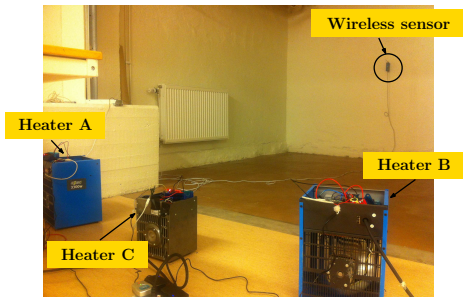
where $x(t) = \begin{bmatrix} \Delta L_1 & \Delta L_2 \end{bmatrix}^T$ is the state of the system, $\Delta L_1 = L_1 - L_{10}$, $\Delta L_2 = L_2 - L_{20}$, $u(t) = \Delta V_p$ the control input and $\Delta V_p = V_p - V_{p_0}$. The system matrices are given by

$$A = \begin{bmatrix} -\frac{a_1}{A_1}\sqrt{\frac{g}{2L_{10}}} & 0 \\ \frac{a_1}{A_2}\sqrt{\frac{g}{2L_{10}}} & -\frac{a_2}{A_2}\sqrt{\frac{g}{2L_{20}}} \end{bmatrix}, \; B = \begin{bmatrix} \frac{K_p}{A_1} \\ 0 \end{bmatrix},$$

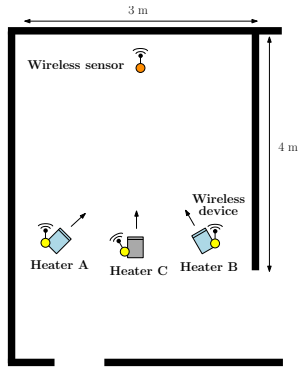
where a_i is the outflow diameter and A_i the diameter of the upper and lower tanks, g is the gravitational acceleration and K_p is the pump motor constant.

Room heating

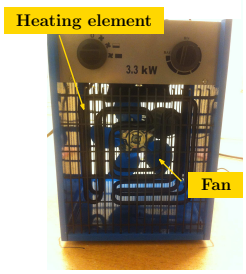
In this experiment, we aim at controlling the temperature of a room using three portable heaters as depicted in Figures 2.4(a) and 2.4(b). There are two types of heaters, two of 3300 W (blue - heaters A and B) and one of 2000 W power (grey - heater C). Figure 2.4(c) shows the 3300 W heater. A Telos wireless device is responsible for switching the heater ON/OFF, see Figure 2.4(d). The actuation is performed by Pulse-Width Modulation (PWM), where the period is set to 120 s, and the duty cycle is computed by the controller. Each wireless actuator receives the temperature measurements from the wireless temperature sensor shown in Fig-



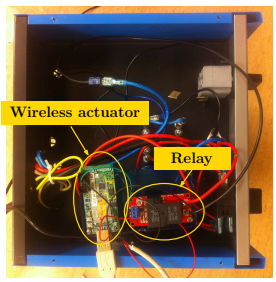
(a) Room and heaters



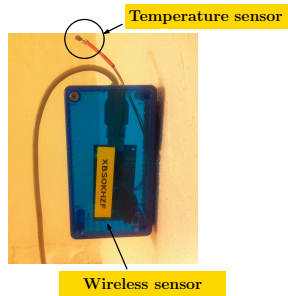
(b) Setup overview



(c) 3300 W heater



(d) Heater setup with wireless actuator



(e) Wireless temperature sensor

Figure 2.4: Room temperature control experimental setup and its components.

ure 2.4(e). The temperature sensor Sensirion SHT11 is connected to the wireless device, providing temperature measurements with an accuracy of ±0.4 °C.

We have identified a first-order continuous-time linear time-invariant model for the room temperature dynamics. The identification was performed through step experiments of the system to each individual heater. The identified model is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B \begin{bmatrix} u_A(t) & u_B(t) & u_C(t) \end{bmatrix}^T, \\ y(t) &= x(t), \end{aligned} \tag{2.1}$$

where $x(t)$, represents the variation of temperature from steady-state conditions, $u_A(t)$, $u_B(t)$ and $u_C(t)$ are the control commands and

$$A = -0.0072, \; B = \begin{bmatrix} 0.0317 & 0.0307 & 0.0179 \end{bmatrix}.$$

Experiments showed that the model is accurate in the range 20°C to 26°C. Since the influence of adjacent rooms in the temperature dynamics was small, we neglected their effect in the identified model.

2.9 Summary

In this chapter, we have discussed the background and several aspects related to wireless NCSs. We particularly focused on control over wireless networks, resource-aware design through aperiodic sampling techniques, compensation of network imperfections and fault reconfiguration. A summary of the available MAC schemes was given, with a particular focus on the IEEE 802.15.4 MAC. Protocols for industrial control applications were also presented. With respect to the validation of NCSs, we gave an overview of the current available co-simulation tools. Finally, we presented the experimental platforms which will be utilized throughout the thesis for implementation and validation.

Chapter 3

Triggering conditions for aperiodic control

In this chapter, we consider the execution and tuning of triggering conditions for self-triggered and event-triggered controllers. These aperiodic controllers aim at reducing the amount of sensor transmissions and control loop executions. While in a traditional sampled-data paradigm new controller updates are performed periodically, regardless of the state of the system, event-triggered control is based on events triggered when stability or a pre-specified control performance is about to be lost. This approach has been shown to reduce the number of control loop executions, while providing a high degree of robustness, since the system is permanently monitored. However, by requiring the continuous monitoring of a certain triggering condition, event-triggered controllers still require a large amount of resource consumption, e.g., battery power of sensor devices or sensor/controller computations. To overcome this drawback, several researchers proposed the use of self-triggered control techniques (Velasco et al., 2003; Wang and Lemmon, 2009; Anta and Tabuada, 2010a; Mazo Jr. et al., 2009). The underlying idea is to emulate the event-triggered mechanism and estimate the time at which the next event takes place using the knowledge of the system's dynamics. The sensor nodes are then scheduled for transmission at the expected triggering time. In between triggering times, sensor nodes can simply be in an idle mode thereby greatly reducing its energy consumption. However, this can reduce the robustness of the system to external disturbances (Anta and Tabuada, 2010a). One of the main challenges in self-triggered control is how to perform the exact calculation of the next control update time.

The main contribution of this chapter is the synthesis of an algorithm to compute the next control update time of a self-triggered controller. We particularly consider triggering conditions for systems where the control input is held constant between control updates. This problem has been considered earlier in the literature, e.g., (Wang and Lemmon, 2009; Anta and Tabuada, 2010a) and (Mazo Jr. et al., 2009, 2010). In (Anta and Tabuada, 2010a) a general technique was developed to compute the inter-execution times for any nonlinear polynomial system. Since this technique focused on nonlinear systems, it becomes conservative when used for lin-

ear systems. The papers (Mazo Jr. et al., 2009, 2010) propose, in the context of linear systems, to simply integrate the dynamics using a discretization method in order to find the update time. This method has the drawback that the choice of discretization step greatly influences the complexity of the mechanism. The method proposed by (Wang and Lemmon, 2009) computes a conservative lower bound of the next update time while providing performance guarantees in terms of the closed-loop system's induced $\mathcal{L}_2$ gain. The computed self-triggered update time is shown to be an order of magnitude smaller than the exact inter-executation time. In this chapter, we propose a method that computes the update times for self-triggered control of diagonalizable linear systems. The method builds upon an idea from decentralized event-triggered control in (Mazo Jr. and Tabuada, 2011) and provides a computationally light manner to estimate the update times of self-triggered mechanisms. Additionally, we propose a Semidefinite Programming (SDP)-based technique that produces triggering conditions for event-triggered and self-triggered control that are less conservative than the existing ones and for which the update times are larger.

The rest of the chapter is organized as follows. We begin by reviewing the event-triggered and self-triggered control paradigms in Section 3.1 and introduce the problem setup in Section 3.2. In Section 3.3 we present the algorithm to compute the triggering times of the self-triggered control mechanism and the improved triggering conditions using an SDP-based technique is presented in Section 3.4. The proposed methods are evaluated in Section 3.5 through numerical examples.

3.1 Preliminaries of event-triggered and self-triggered control

We consider networked linear control systems consisting of sensors, actuators and a central controller node. The controller node is responsible for computing the control signal based on the measurements received from the sensor nodes. The *control loop execution* is defined as: the transmission of measurements by the sensor nodes to the controller node, the computation of the control signal by the controller node and the posterior dissemination of the control signal to the actuator nodes. [1]. Additionally, the sensor or controller node is responsible of monitoring and enforcing the triggering condition in the case of event-triggered control, or to compute the update time and schedule the sensor nodes for future transmissions in a self-triggered controller mechanism.

Consider a linear control system of the form:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \tag{3.1}$$

[1] For clarity of presentation we assume all actions are performed instantaneously, where computation and transmission delays are nonexistent. Nevertheless, the paradigms described below are robust to delay and/or can be slightly modified to cope with delays, as it has been shown in the literature (Heemels et al., 2012) and will be shown in Chapter 4.

where A and B are matrices of appropriate dimensions.

In a zero-order-hold (ZOH) implementation of the control signal $u(t)$, the input signal is held constant between control loop executions, i.e., $u(t) = u(t_k)$, for $t \in [t_k, t_{k+1})$, where $t_1, t_2, t_3, ...$ is the sequence of update times at which a new measurement is transmitted to the controller and input signals are applied to the system. We define the *inter-execution time* (also known as *inter-transmission time*) as $\tau_k = t_k - t_{k-1}$, representing the time between control loop executions.

Assuming a controller $u(t) = Kx(t)$ is designed to render the closed-loop system asymptotically stable, there exists a Lyapunov function of the form $V(t) = x(t)^T Px(t)$ satisfying:

$$\dot{V}(t) = \frac{\partial V}{\partial x}(A + BK)x(t) = -x(t)^T Qx(t) \tag{3.2}$$

where Q is a positive definite matrix. The matrix P is the solution of the Lyapunov equation $(A + BK)^T P + P(A + BK) = -Q$, where Q is a chosen symmetric positive definite matrix. The Lyapunov function $V(t)$ can be seen as a certificate of stability, since according to equation (3.2) $V(t)$ is always decreasing, but also of performance since (3.2) also ensures that the rate of decrease is at least $x(t)^T Qx(t)$.

In the event-triggered control mechanism we consider, the design of the event-triggered rule is made so that asymptotic stability of the closed-loop system is guaranteed by the sampled controller $u(t_k) = Kx(t_k)$. In such methods, the input $u(t_k)$ is kept constant until the following quadratic triggering condition

$$z(t)^T \Phi_g z(t) < 0 \tag{3.3}$$

is violated, where $z(t) = [x(t)^T \; x(t_k)^T]^T$ with Φ_g symmetric. The event-triggered mechanism guarantees that $z(t)^T \Phi_g z(t) \leq 0$ and a new control loop execution takes place as soon as $z(t)^T \Phi_g z(t) = 0$. Hence, the event-triggered condition (3.3) defines the sequence of time instants t_k when the control input needs to be updated, as:

$$t_{k+1} = \min\{t > t_k \; : \; z(t)^T \Phi_g z(t) = 0\}.$$

Such triggering rule may be implemented in a self-triggered manner by allowing the sensor or controller node to compute the next execution time t_{k+1} by evaluating (3.3). We now show various event-triggered controller designs proposed in the literature and how they can be cast in the above quadratic form. A similar exercise has been conducted in (Heemels et al., 2013) for the discretized version of the same class of event-triggered controllers.

The mechanisms surveyed in this chapter have been shown in the literature to possess a minimum inter-execution time in the sense that the time between two consecutive execution is always lower bounded by a positive constant, thus avoiding Zeno behaviour (Lygeros et al., 2003). Additionally, this property has also been shown to hold in (Borgers and Heemels, 2014) for the same techniques, with slight adaptations, when the system is affected by Gaussian noise.

3.1.1 Bounding the rate of decay of the Lyapunov function $V(t)$

This method is based on the requirement that the rate of decay of the Lyapunov function $V(t)$ for system (3.1) is bounded by a prescribed rate. The update condition is defined by:

$$\dot{V}(t) < -x(t)^T W x(t), \tag{3.4}$$

for a chosen $0 \preceq W \prec Q$, where matrix W defines the desired rate of decay of the Lyapunov function V.The expression (3.4) can be further represented in the general form (3.3) with Φ_g given by

$$\Phi := \begin{bmatrix} A^T P + PA + W & PBK \\ K^T B^T P & 0 \end{bmatrix}, \tag{3.5}$$

which is derived as follows:

$$\begin{aligned} \dot{V}(t) + x(t)^T W x(t) &= \dot{x}(t)^T P x(t) + x(t)^T P \dot{x}(t) + x(t)^T W x(t) \\ &= (Ax(t) + Bu(t))^T P x(t) + x(t)^T P (Ax(t) + Bu(t)) + \\ &+ x(t)^T W x(t) \\ &= x(t)^T \left(A^T P + PA + W \right) x(t) + 2x(t)^T PBK x(t_k) \\ &= z(t)^T \Phi z(t) \end{aligned}$$

A particular case of the mechanism (3.4) is the simple triggering condition based on the state error proposed in (Tabuada, 2007), which requires the update of the control input when condition (3.3) is violated, for Φ_g given by

$$\tilde{\Phi} = \begin{bmatrix} I - \gamma^2 I & -I \\ -I & I \end{bmatrix}, \tag{3.6}$$

and $0 < \gamma < 1$ is a parameter which depends on the choice of A, B, K, Q and W. Through the derivation of (3.6), the following implication holds for any $z(t)$:

$$z(t)^T \Phi z(t) \geq 0 \Rightarrow z(t)^T \tilde{\Phi} z(t) \geq 0. \tag{3.7}$$

due to the fact that $\Phi \succeq \tilde{\Phi}$, and Φ and $\tilde{\Phi}$ are symmetric matrices (Horn and Johnson, 2012). Other matrices $\tilde{\Phi}$ exist that satisfy the above implication for all $z(t)$. This technique will be the focus of the current chapter and we will say that a triggering matrix $\tilde{\Phi}$ is a *valid bound* to Φ if the implication (3.7) holds for all $z(t)$.

3.1.2 Bounding the Lyapunov function $V(t)$

In (Mazo Jr. et al., 2009), (Mazo Jr. et al., 2010) and (Velasco et al., 2009), a different approach was taken where the desired performance of the mechanism is specified by means of a function $S(t)$ required to upper bound the evolution of $V(t)$:

$$V(t) \leq S(t). \tag{3.8}$$

Provided that (3.8) holds and $S(t)$ is decaying over time, the closed-loop system is stabilized, with a decay rate of its Lyapunov function no lower than the one specified through $S(t)$. One such $S(t)$, as proposed in (Mazo Jr. et al., 2009), is obtained as the Lyapunov function:

$$S(t) = x_s(t)^T P x_s(t), \tag{3.9}$$

for the hybrid system:

$$\begin{aligned} \dot{x}_s(t) &= A_s x_s(t) \qquad t \in [t_k, t_{k+1}) \\ x_s(t_k) &:= x(t_k), \end{aligned} \tag{3.10}$$

where A_s is a Hurwitz matrix satisfying the following Lyapunov equation $A_s^T P + PA_s = -R$, with $Q - R$ positive definite. This parameter can be chosen as $R = \sigma Q$, where $0 < \sigma < 1$ determines the rate of decay of $S(t)$ as a proportion of that of $V(t)$. Again, such condition can be written in the general quadratic form (3.3) with

$$\Phi_g = \begin{bmatrix} P & 0 \\ 0 & -\bar{A}^T P \bar{A} \end{bmatrix},$$

where $\bar{A} = e^{A_s(t-t_k)}$. This follows directly from inequality (3.8) and noticing that from (3.10), $x_s(t) = e^{A_s(t-t_k)} x(t_k)$, for all $t \in [t_k, t_{k+1})$. This technique will be used for the event-triggered and self-triggered implementations proposed in Chapter 4.

3.2 Problem formulation

We now present the problem setup considered in this chapter. Consider the linear control system in (3.1). Supposing A is diagonalizable, we can transform (3.1) into:

$$\dot{x}'(t) = M^{-1} A M x'(t) + M^{-1} B u(t),$$

where $x'(t) = M^{-1} x(t)$ and M is the eigenvector matrix of A. Therefore, we directly assume system (3.1) to have a diagonal A matrix. The main problem we address in this chapter is:

- How can we design an efficient method that computes the next event time for the simple triggering rule given by $\tilde{\Phi}$ of (Tabuada, 2007)?

In fact, we solve the preceding problem not just for the triggering condition defined by $\tilde{\Phi}$ but also for any quadratic triggering condition that has a nice separability structure. This method is presented in detail in Section 3.3.

In the second part of this chapter, we propose an SDP technique to find quadratic triggering conditions that best bound Φ and have the required separability properties to enable the application of the previous devised method. Thus,

this technique allows us to obtain larger inter-execution times that still guarantee (3.4), i.e., $\dot{V} < -x^T W x$.

Note that, in principle, computing the next event time t_{k+1} given by any quadratic triggering condition can always be done since a closed form expression for the evolution of the state is available. However, this is a computationally intensive task if high accuracy is desired. This will be shown in the numerical examples provided in Section 3.5. Instead, we seek a more computationally efficient method which could be potentially be implementable on a microcontroller.

3.3 Main algorithm

We now present our main algorithm to compute a lower bound on the triggering time t_{k+1}, which is the earliest time t for which

$$z^T \tilde{\Phi} z \geq 0,$$

is true. Our algorithm requires the control system (3.1) to be diagonalizable, and the triggering matrix $\tilde{\Phi}$ to be a $2n \times 2n$ symmetric matrix for which the upper-left block is diagonal. This guarantees that the triggering condition is separable. In other words, if we let the *decision gap* G be[2] $G(x) = \begin{bmatrix} x \\ x(t_k) \end{bmatrix}^T \tilde{\Phi} \begin{bmatrix} x \\ x(t_k) \end{bmatrix}$, then G has a separable structure if: $G(x) = \sum_{i=1}^{n} G_i(x_i)$. This separability together with the fact that the system is diagonal (i.e., that $\dot{x}_i$ only depends on x_i and the input) allows for the efficient computation of the triggering time t_{k+1}.

3.3.1 An insight from (Mazo Jr. and Tabuada, 2011)

Our method uses an idea by (Mazo Jr. and Tabuada, 2011) proposed in the context of decentralized event-triggered control.

If we consider $\theta_1, ..., \theta_n \in \mathbb{R}$, such that $\sum_{i=1}^{n} \theta_i = 0$, we can rewrite the decision gap function $G(x)$ as

$$G(x) = \sum_{i=1}^{n} (G_i(x_i) - \theta_i).$$

The following implication then holds since the sum of non-positive quantities is still a non-positive quantity:

$$\forall i = 1, \ldots, n \quad (G_i(x_i) - \theta_i \leq 0) \Rightarrow G(x) \leq 0.$$

Thus, if we define

$$t_i(\theta_i) := \min\{t > t_k : G_i(x_i(t)) - \theta_i = 0\},$$

[2]Since $x(t_k)$ is fixed, we are only interested in the dependence of G on x.

then we clearly have that

$$\min_{i=1,\dots,n} t_i(\theta_i) \leq t_{k+1}.$$

Note that this is true for any choice of θ such that $\sum_{i=1}^n \theta_i = 0$. Thus if we denote by T the quantity: $T(\theta) := \min_{i=1,\dots,n} t_i(\theta_i)$ we have that for any $\theta_1, \dots, \theta_n$ such that $\sum_{i=1}^n \theta_i = 0$,

$$T(\theta) \leq t_{k+1}.$$

Hence, in order to obtain a good lower bound on t_{k+1}, we need to find the value of θ for which $T(\theta)$ is maximal.

Observe that since the system is assumed to be diagonal, and the triggering matrix has a separable structure, one can compute explicitly the value of $T(\theta)$ for any given θ (closed form expressions are provided in Appendix A.1). We will rely on these formulas for the algorithm we present next.

Remark 3.1. *Finding the value of θ for which $T(\theta)$ is maximal is a non-trivial problem that has no analytical solution, in general. In (Mazo Jr. and Tabuada, 2011) the authors proposed to equalize $G_i(x_i) - \theta_i = 0$, for all i in order to find θ. However, such approach does not take into account additional conditions that must be satisfied to correctly find θ and $T(\theta)$, as we show next.*

3.3.2 Finding the best value of θ

In this section we propose an iterative method to compute the value of θ that makes $T(\theta)$ maximal. The method starts with an initial value of θ, say $\theta^{(0)}$, and then keeps improving it at each iteration so that $T(\theta^{(0)}) < T(\theta^{(1)}) < T(\theta^{(2)}) < \dots$ where $\theta^{(\ell)}$ is the value of θ at iteration ℓ of the algorithm.

We now explain how one can improve the value of θ from one iteration to the next, i.e., how one can find $\theta^{(\ell+1)}$ such that $T(\theta^{(\ell+1)}) > T(\theta^{(\ell)})$. The update rule for $\theta_i^{(\ell)}$ will take the form

$$\theta_i^{(\ell+1)} = \theta_i^{(\ell)} - \delta_i,$$

for some appropriately chosen δ_i's. We now show how to choose the values of δ_i's. To lighten the notation, in the rest of this chapter we write $G_i(t)$, $G(t)$ instead of $G_i(x_i(t))$, $G(x(t))$. Observe that for a given $\theta^{(\ell)}$ we have, for any[3] i

$$G_i(T(\theta^\ell)) - \theta_i^{(\ell)} \leq 0,$$

with equality for at least one subsystem i (this is the "bottleneck" subsystem for which $t_i(\theta_i^{(\ell)})$ is minimal, see Figure 3.1).

We observe that if we can make the inequalities (3.3.2) strict for all i then we can make progress, in the sense that there exists $\theta^{(\ell+1)}$ such that $T(\theta^{(\ell+1)}) > T(\theta^{(\ell)})$.

[3]

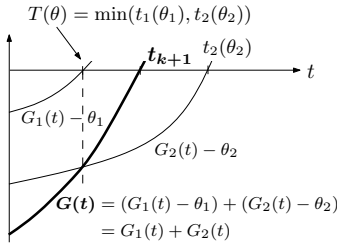


Figure 3.1: For a given θ, we have $G_i(T(\theta)) - \theta_i \leq 0$ for all i, with equality for at least one subsystem i (in the figure, this is subsystem 1 since $t_1(\theta_1) < t_2(\theta_2)$)

Indeed assume that we can find $\delta_1, \ldots, \delta_n,\ \delta_i \in \mathbb{R}$ such that :

$$\sum_{i=1}^{n} \delta_i = 0, \tag{3.11}$$

and such that for all i:

$$G_i(t) - \theta_i^{(\ell)} + \delta_i < 0 \quad \forall t \in [t_k, T(\theta^{(\ell)})]. \tag{3.12}$$

The important property about the δ's here is that the inequalities above are strict for all i. The coefficients δ_i correspond to a vertical displacement of the graphs of $G_i - \theta_i^{(\ell)}$, as depicted for example in Figure 3.2. If such values of δ exist, then by defining $\theta_i^{(\ell+1)} = \theta_i^{(\ell)} - \delta_i$, we have that

$$\forall t \in [t_k, T(\theta^{\ell})],\ G_i(t) - \theta_i^{(\ell+1)} < 0,$$

for all i, and so this means that for all i, $t_i(\theta_i^{(\ell+1)}) > T(\theta^{(\ell)})$ which in turn means that $T(\theta^{(\ell+1)}) > T(\theta^{(\ell)})$.

The question now remains of how to find values $\delta_1, \ldots, \delta_n$ that satisfy conditions (3.11) and (3.12). Note that condition (3.12) on the δ_i's can be rewritten as

$$\max_{t \in [t_k, T(\theta^{(\ell)})]} G_i(t) - \theta_i^{(\ell)} + \delta_i < 0,$$

or equivalently

$$\delta_i < \overline{\delta_i},$$

where $\overline{\delta_i} := \theta_i^{(\ell)} - \max_{t \in [t_k, T(\theta^{(\ell)}))} G_i(t)$. Since we assumed our system to be diagonal, the quantity $\overline{\delta_i}$ can be computed explicitly and the formulas are given in Appendix A.1. The problem of finding δ_i's therefore corresponds to solving the following feasibility problem:

$$\text{find } \delta_1, \ldots, \delta_n \text{ such that } \delta_i < \overline{\delta_i} \text{ and } \sum_{i=1}^{n} \delta_i = 0.$$

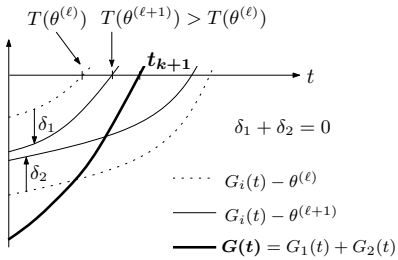


Figure 3.2: This figure shows how to improve the value of the θ's from iteration ℓ to iteration $\ell + 1$, in the sense that $T(\theta^{(\ell+1)}) > T(\theta^{(\ell)})$. Observe that δ_1 and δ_2 on the figure are such that $\delta_1 + \delta_2 = 0$ (condition (3.11)), and $G_i(T(\theta_i^{(\ell)})) - \theta_i^{(\ell)} + \delta_i < 0$ (condition (3.12)). As can be seen in the figure, these conditions guarantee that for $\theta^{(\ell+1)} := \theta^{(\ell)} + \delta$, we have $T(\theta^{(\ell+1)}) > T(\theta^{(\ell)})$.

If there are no solutions to this problem, we stop the algorithm. Otherwise, there can be many solutions to this feasibility problem, and so it would be preferable to find the "best" possible δ. For example one can introduce an objective function that makes the inequalities $G_i(T(\theta^{(\ell)})) - \theta_i^{(\ell)} + \delta_i < 0$ as far from zero as possible. This therefore yields the following optimization problem for the δ's:

$$\begin{array}{ll} \underset{\delta_1,\ldots,\delta_n}{\text{minimize}} & \max_{i=1,\ldots,n} G_i(T(\theta^{(\ell)})) - \theta_i^{(\ell)} + \delta_i \\ \text{subject to} & \delta_i \leq \overline{\delta_i} \text{ for all } i = 1, \ldots, n \\ & \sum_{i=1}^n \delta_i = 0 \end{array} \tag{3.13}$$

This optimization problem can be solved efficiently using a simple bisection algorithm. The details of the algorithm are given in Appendix A.2 by Algorithm A.1. The complete iterative algorithm outlined in this section to compute a value of θ such that $T(\theta)$ is as large as possible is given in Algorithm 3.1. Note that the initial value $\theta^{(0)}$ in Algorithm 3.1 satisfies $\sum \theta_i^{(0)} = 0$ and $G_i(t_k) - \theta_i^{(0)} < 0$.

From the previous discussion, the behavior of the algorithm can be summarized in the following proposition. We omit a formal proof since it is clear from the previous discussion.

Proposition 3.1. *Consider a system $\dot{x}(t) = Ax(t) + Bu(t)$ where A is diagonal, and a triggering matrix $\tilde{\Phi} \in \mathbb{R}^{2n \times 2n}$ whose upper-left block is diagonal. Let $x(t_k)$ be the state at time t_k satisfying $\begin{bmatrix} x(t_k) \\ x(t_k) \end{bmatrix}^T \tilde{\Phi} \begin{bmatrix} x(t_k) \\ x(t_k) \end{bmatrix} < 0$. Let now t_{k+1} be the earliest time $t > t_k$ such that $\begin{bmatrix} x(t) \\ x(t_k) \end{bmatrix}^T \tilde{\Phi} \begin{bmatrix} x(t) \\ x(t_k) \end{bmatrix} = 0$. The sequence $\theta^{(\ell)}$ generated*

by Algorithm 3.1 satisfies

$$T(\theta^{(\ell+1)}) > T(\theta^{(\ell)}),$$

and also $T(\theta^{(\ell)}) < t_{k+1}$ *for all* ℓ *such that* $\theta^{(\ell)}$ *is defined. In other words, the algorithm 3.1 produces a sequence of increasingly tight lower bounds to* t_{k+1}*.*

Algorithm 3.1 Iterative algorithm to approach the value of θ for which $T(\theta)$ is maximal. This yields a lower bound on the triggering time since $T(\theta) < t_{k+1}$ for any θ.

Initialize $\theta_i^{(0)} = G_i(t_k) - \sum_j G_j(t_k)/n$
for $\ell = 0$ to `numiter` **do**
  $t_i(\theta_i^{(\ell)}) \leftarrow \min\{t \geq t_k \colon G_i(t) - \theta_i^{(\ell)} = 0 \text{ and } G_i(s) - \theta_i^{(\ell)} < 0 \; \forall s \in [0,t)\}$
  $T(\theta^{(\ell)}) \leftarrow \min_{i=1,\ldots,n} t_i(\theta_i^{(\ell)})$
  $\overline{\delta}_i \leftarrow \min\{|G_i(t) - \theta_i| \colon t \in [0, T(\theta^{(\ell)})]\}$
  $\delta \leftarrow$ solution to optimization problem (3.13), if feasible
  if problem (3.13) infeasible **then**
    break
  end if
  $\theta_i^{(\ell+1)} = \theta_i^{(\ell)} - \delta_i$
end for

Algorithm 3.1 is typically run for a constant number of iterations `numiter` no larger than 10. Furthermore, each iteration of the algorithm has a computational cost that is linear in n (where n is the dimension of the system state) and hence the algorithm has a total cost that scales linearly. The bisection algorithm to find the δ's is run until a desired accuracy is obtained, which is typically obtained in log number of iterations, i.e., better than linear. This is the case in our evaluations of the proposed algorithm.

3.4 Improved triggering condition

The algorithm presented in the previous section works for any triggering matrix $\tilde{\Phi}$ that has its upper-left block diagonal. Recall from Section 3.1 however that ideally we would like to compute the earliest time t_{k+1} for the original triggering matrix Φ which corresponds to the triggering rule (3.5). Since in general Φ does not have the required diagonal structure, we need to find a valid bound $\tilde{\Phi}$ of the original Φ before using the algorithm of the previous section. One valid bound $\tilde{\Phi}$ that has the required structure is the one proposed in (Tabuada, 2007) that we mentioned earlier (equation (3.6)). In this section however we will try to find somehow "optimal" valid approximations $\tilde{\Phi}$ that have the required diagonal structure using SDP.

Recall that $\tilde{\Phi}$ is a valid bound to Φ if it holds for any $z(t)$:

$$z(t)^T \Phi z(t) \geq 0 \Rightarrow z(t)^T \tilde{\Phi} z(t) \geq 0.$$

This guarantees that the triggering time associated with $\tilde{\Phi}$ will always be smaller than the triggering time associated with Φ. By the S-lemma (Boyd and Vandenberghe, 2004), the constraint above is equivalent to the LMI

$$\exists \lambda \geq 0 \text{ such that } \tilde{\Phi} \succeq \lambda \Phi.$$

From this observation, one way of finding a valid bound $\tilde{\Phi}$ with the required structure is by solving the following SDP problem:

$$\begin{aligned} \underset{\tilde{\Phi},\lambda}{\text{minimize}} \quad & \operatorname{tr}(\tilde{\Phi}) \\ \text{subject to} \quad & \tilde{\Phi} \succeq \lambda \Phi, \ \lambda \geq 0 \\ & \tilde{\Phi} \preceq \begin{bmatrix} I - \sigma^2 I & -I \\ -I & I \end{bmatrix}, \\ & \text{Upper-left block of } \tilde{\Phi} \text{ diagonal} \end{aligned} \tag{3.14}$$

The second constraint of the SDP above guarantees that the optimal $\tilde{\Phi}$ will be at least as good as the one of (Tabuada, 2007) given in equation (3.6). Also, in the SDP above, we are interested in the minimization of $\operatorname{tr}(\tilde{\Phi})$ since we aim at finding the tightest triggering matrix which guarantees the given constraints. Other objective functions can of course be envisaged. Problem (3.14) can be solved efficiently using standard optimization software and is solved offline. In the example section, we utilize the SeDuMi solver (Sturm, 1999) in CVX (Grant et al., 2008).

3.5 Numerical examples

We now present examples illustrating the effectiveness of the proposed techniques. All examples are implemented in MATLAB. The first example will serve to verify the performance of the self-triggered mechanism with respect to its accuracy to compute the event times. Afterwards, we evaluate how the inter-execution times are enlarged by using the improved triggering condition of Section 3.4. A time-response analysis of a well known system is performed to assess the performance of both contributions.

3.5.1 Unstable second-order system

We consider the following unstable second-order system from (Garcia and Antsaklis, 2011) given by

$$A = \begin{bmatrix} 0.55 & -0.4 \\ 0.3 & -0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

and $K = -\begin{bmatrix} 1.3424 & 0.0095 \end{bmatrix}$, $Q = I$ and P obtained from solving the Lyapunov equation. The performance parameter W and σ are set to $W = 0.5I$ and $\sigma = 0.22$.

Table 3.1: Maximum and minimum error in ms of $t_k - t_{k-1}$, compared to the value obtained by the discretization method with 0.01 ms accuracy.

	Mazo Jr. and Tabuada (2011)	Main algorithm
max	65.8	0.048
min	0.0077	0.0016

Table 3.2: Number of elementary arithmetic operations for the unstable second-order system

	Discretization 0.01 ms	Discretization 0.5 ms	Main algorithm
min	3.8×10^5	7638	
mean	7.7×10^5	15458	254
max	20.1×10^5	40242	

The number of iterations for algorithm 1 is set to numiter = 9 in Algorithm 3.1 and numIterDelta = 15 and in Algorithm A.1.

We compare the inter-execution times computed using three different methods: the main algorithm from Section 3.3, the algorithm proposed in (Mazo Jr. and Tabuada, 2011) and an exact computation of the times with an accuracy of 0.01 ms. The comparison is performed using the same triggering condition from (Tabuada, 2007) in (3.6) for all computation methods. We evaluate these mechanisms for distinct initial conditions $x(0) = [\cos(\alpha) \quad \sin(\alpha)]^T$ for $\alpha \in [\frac{\pi}{30}, 2\pi]$ with $\frac{\pi}{30}$ increments and the inter-execution time $t_{k+1} - t_k$ concerns only the first triggering instant. The results are shown in Figure 3.3. The maximum and minimum error given with respect to the discretization method with 0.01 ms of accuracy are depicted in Table 3.1, for the method in (Mazo Jr. and Tabuada, 2011) and for the proposed main algorithm. The main algorithm proposed in Section 3.3 is able to provide a tight bound on the exact inter-execution time for this system. The maximum obtained error is 0.048 ms, hence, it is reasonable to ask how many elementary arithmetic operations would be required by a discretization method under the same accuracy level. We provide a comparison of the number of elementary arithmetic operations performed by a discretization method with 0.01 ms accuracy, as well as the discretization method with 0.05 ms accuracy. The results are shown in Table 3.2. Note that the proposed algorithm has a fixed number of operations as these iterations are independent of the accuracy, but dependent on the fixed order n of the system and the fixed parameters numiter and numiterDelta for both Algorithms 3.1 and A.1, respectively. The results show that the number of operations can be significantly reduced by using the proposed algorithm.

We now evaluate the method proposed in Section 3.4. Figure 3.4 depicts the results for the same set of initial conditions as in the previous example. We compare the inter-execution time given by an exact computation for different triggering ma-

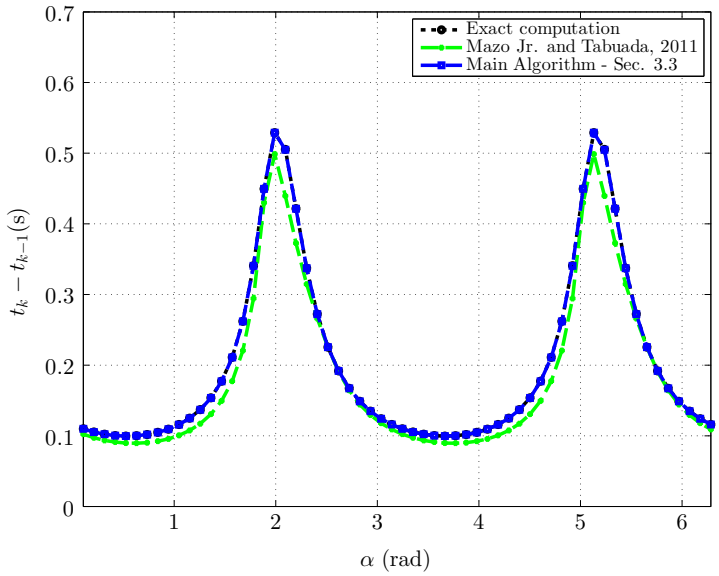


Figure 3.3: Inter-execution time for the triggering condition (Tabuada, 2007) in (3.6) for different computation methods. Times for distinct initial conditions $x(0) = [\cos\alpha \quad \sin(\alpha)]^T$, $\alpha \in [\frac{\pi}{30}, 2\pi]$ with $\frac{\pi}{30}$ increments.

trices: original triggering matrix Φ in (3.5), the triggering matrix $\tilde{\Phi}$ from (Tabuada, 2007) in (3.6) and $\tilde{\Phi}$ from Section 3.4. The maximum time difference for the triggering matrix $\tilde{\Phi}$ from Section 3.4 with respect to the time given by the the triggering matrix Φ is of 3.4 s, while the minimum difference is zero, as they completely match for certain initial conditions. For the triggering matrix $\tilde{\Phi}$ from (Tabuada, 2007) in (3.6), the maximum difference was of 7.3 s and the minimum difference of 1.3 s. Clearly, there is an advantage of performing the SDP technique to compute $\tilde{\Phi}$ as we are able to greatly enlarge the inter-execution times. This is achieved while guaranteeing the same rate of decay of the Lyapunov equation $V(t)$, as also guaranteed for the other triggering conditions.

3.5.2 Batch reactor

To illustrate the performance of the proposed techniques in a time-response experiment, we borrow the Batch Reactor model from (Walsh et al., 1999). The same system and controller parameters are used. The initial condition is set $x(0) = \begin{bmatrix} -15 & 14 & -23 & 15 \end{bmatrix}^T$. Matrix Q is set to be the identity $Q = I$ and P is obtained from solving the Lyapunov equation. The performance parameter S and σ were $0.1I$ and 0.387, respectively. Additionally, numiter = 10 and numIterDelta = 15.

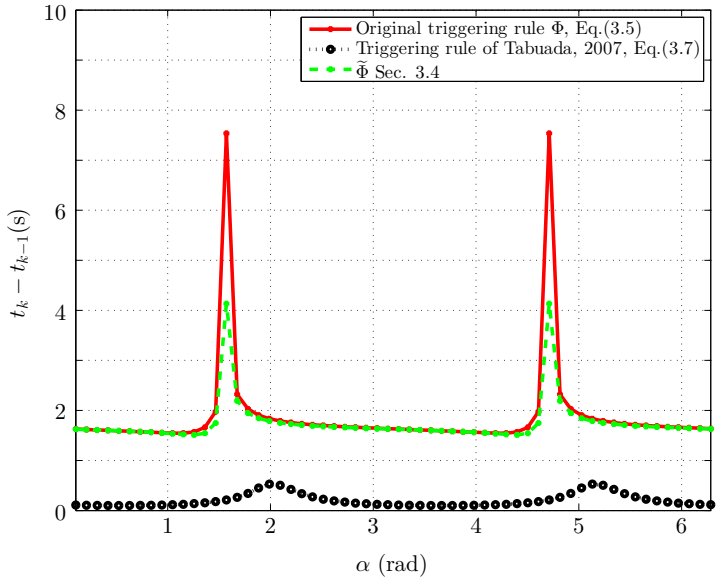


Figure 3.4: Exact computation of the inter-execution time for different triggering rules: Φ in (3.5), $\tilde{\Phi}$ from (Tabuada, 2007) in (3.6) and $\tilde{\Phi}$ from Section 3.4.

The closed-loop system is set to run for 1 second and Table 3.3 presents the number of control loop executions performed by the triggering mechanisms. We compare the event-triggered and self-triggered mechanisms for three triggering rules: Φ in (3.5), $\tilde{\Phi}$ from (Tabuada, 2007) in (3.6) and $\tilde{\Phi}$ from Section 3.4. The self-triggered mechanism is implemented for the main algorithm in Section 3.3 and the method in (Mazo Jr. and Tabuada, 2011).

As the results show, for the event-triggered mechanism, the improved triggering condition allows to achieve the same number of events as the least conservative condition Φ in (3.5), while having the imposed separability structure, and is able to reduce the number of events when compared to condition $\tilde{\Phi}$ from (Tabuada, 2007). Furthermore, the self-triggered mechanism using our algorithm is able to achieve approximately the same number of control-executions as the event-triggered mechanism for both triggering conditions $\tilde{\Phi}$ from (Tabuada, 2007) in (3.6) and $\tilde{\Phi}$ from Section 3.4.

3.6 Summary

In this chapter, we developed a computationally efficient method to compute the inter-execution times of self-triggered controllers of diagonalizable systems, where the triggering conditions are quadratic and have a nice separability structure. This

Table 3.3: Event-triggered and self-triggered control of the batch reactor.

Scheme / Number of control executions	
Event-triggered Φ given by (3.5)	5
Event-triggered of $\tilde{\Phi}$ given by (3.6)	10
Event-triggered with improved $\tilde{\Phi}$ from Section 3.4	5
Our algorithm with $\tilde{\Phi}$ given by (3.6)	11
Our algorithm with improved $\tilde{\Phi}$ from Section 3.4	5
Algorithm of (Mazo Jr. and Tabuada, 2011) with $\tilde{\Phi}$ given by (3.6)	17

method uses an idea from (Mazo Jr. and Tabuada, 2011) which was proposed in the context of decentralized event-triggered control. Through numerical examples we have shown that the self-triggered mechanism using this approach is able to provide a tight bound on the exact inter-execution times, while requiring a significantly lower number of computations. Additionally, in the second part of this chapter, we proposed an SDP technique to find quadratic triggering conditions that best bounded Φ as well as possessed the required separability properties to enable the application of the previously developed method. We have shown that this technique allows us to obtain larger inter-execution times of a self-triggered mechanism of the triggering condition proposed in (Tabuada, 2007), while providing the same performance guarantees with respect to the rate of decay of the Lyapunov function.

Chapter 4

Communication implementation for aperiodic control

In this chapter, we propose a wireless networked control system (NCSs) architecture and communication strategies that guarantee the stability of the closed-loop system while reducing energy consumption and network bandwidth usage. The overall wireless networked control system is illustrated in Figure 4.1, where several plants share a controller node, and a network manager node directs the access to the shared wireless medium. External nodes are also deployed and share the wireless network. Relying on the event- and self-triggered control techniques surveyed in Chapter 3, we propose three communication strategies: *event-based*, *predictive* and *hybrid communication*. Each mechanism defines an aperiodic control scheme and a communication scheduling. The underlying idea in each of these cases is to introduce a mechanism to decide the control update times, thus linking control and communication. Our design relies on the IEEE 802.15.4 protocol (IEEE 802.15.4, 2006), described in Chapter 2.5, as the communication medium between sensors, controller and actuators. We propose a few required modifications to the IEEE 802.15.4 standard that enable the implementation of the proposed strategies. Additionally, we implement all the proposals in a set of double tank systems and perform experiments demonstrating the efficiency of the proposed mechanisms in terms of energy consumption and communication bandwidth usage.

The rest of the chapter is organized as follows. Section 4.1 presents the event-triggered and self-triggered controllers utilized while Section 4.2 presents the channel access methods which are used by each of the techniques. In Section 4.3 we describe the necessary modifications to IEEE 802.15.4 MAC and the communication schedule implementation details. The experimental setup and results are described in Sections 4.4 and 4.5. The chapter ends with a brief summary in Section 4.6.

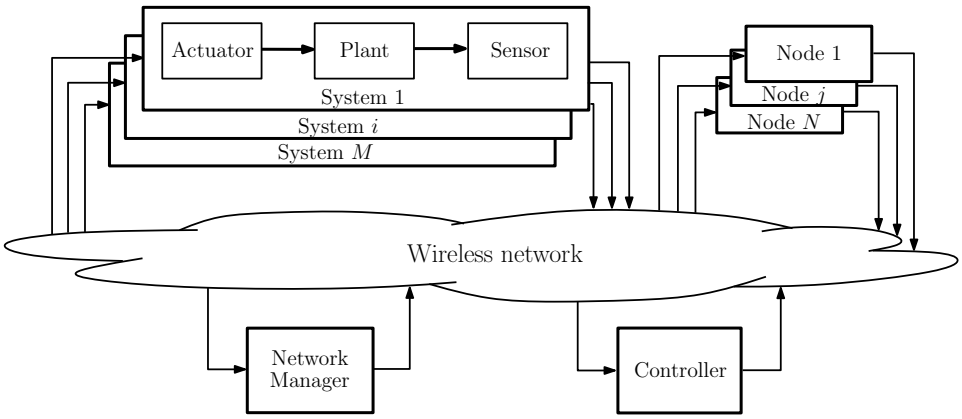


Figure 4.1: Wireless networked control system architecture.

4.1 Event-triggered and self-triggered control

In the current section, we recall the specific event-triggered and self-triggered control techniques presented in Chapter 3, Section 3.1.2, which are the base for the aperiodic communication strategies developed in this chapter. Additionally, we provide modifications to these techniques to take into account communication and computation delays.

4.1.1 Event-triggered control

We consider the linear control system (3.1) of the form:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \tag{4.1}$$

where a controller $u(t) = Kx(t)$ is designed to render the system asymptotically stable, hence, there exists a Lyapunov function of the form $V(t) = x(t)^T Px(t)$ satisfying (3.2). The event-triggered condition considered in this chapter follows the design requirement that the Lyapunov function $V(t)$ is bounded by another Lyapunov function $S(t)$, as given by (3.8) in Section 3.1.2. To enforce the stability condition (3.8), an event-triggered mechanism defines the sequence of time instants t_k when the control input needs to be updated, as:

$$t_{k+1} = \min\{t > t_k \ : \ V(x(t)) - S(x_s(t)) = 0\}. \tag{4.2}$$

For convenience, let us denote by $g_S : \mathbb{R}^n \to \mathbb{R}^+$ the function defined by $g_S(x(t_k)) = t_{k+1} - t_k$, when employing the performance function $S(t)$. Moreover, recall that $\tau_k = t_{k+1} - t_k$ is defined as the inter-transmission time.

Following the proposed design, it can be guaranteed that the minimum inter-transmission time is strictly greater than zero over all possible initial conditions in

the operating region Ω:

$$\tau_{\min} = \inf_{x_0 \in \Omega} g_S(x_0) > 0, \tag{4.3}$$

This time always guarantees a certain level of performance (as defined by $S(t)$) and stability of the closed-loop system, and can be computed using Theorem 5.1 in (Mazo Jr. et al., 2009).

In our implementation, the triggering condition (4.2) is checked periodically, at the speed at which measurements are acquired. We denote this period by T. This implies that the performance that can actually be guaranteed is slightly reduced, as analyzed in (Mazo Jr. et al., 2009).

4.1.2 Self-triggered control

As introduced in Chapter 3, self-triggered controllers relax the requirement of continuously monitoring the triggering condition (4.2) by predicting when it will be violated. A self-triggered technique provides estimates of the inter-transmission times τ_k by relying on the plant model, the last measurement of the state of the system $x(t_k)$, and the performance specification. The prediction of the time between two consecutive updates is embodied in a self-triggering function $\tilde{g}_S : \mathbb{R}^n \to \mathbb{R}^+$ satisfying $\tilde{g}_S(x) \leq g_S(x)$, for all $x \in \mathbb{R}^n$. In self-triggered control it is also customary to impose an upper bound t_{max} to the inter-transmission times in order to provide robustness guarantees of the self-triggered controller. We employ in our strategies the method proposed in (Mazo Jr. et al., 2009), which is aimed at predicting the violation of the event-triggering conditions for linear systems, reviewed in Section 3.1.2. Nevertheless, our proposals also applies to other available techniques, including those for nonlinear systems (Anta and Tabuada, 2010a). Instead of resorting to the algorithm proposed in Chapter 3 we instead use the discretization method to compute the solution of (4.1) and (4.2) at times $t = nT$, $n = 1, 2, \ldots$. The reason for resorting to this technique instead of the one devised in Chapter 3 comes from the fact that (a) the process used in the experimental evaluation in this chapter is slow and so T can be selected to be large, allowing the discretization method to be computed in a quick manner and that (b) the algorithm designed in Chapter 3 is posterior to the work developed in this chapter.

4.1.3 Delay compensation

In the following we propose two simple solutions to compensate for delays between measurement acquisition and actuation updates, applicable to the event-triggered and self-triggered techniques reviewed in the previous sections. For convenience, in the explanations that follow the delays present in the wireless NCSs are divided into two types: delays in the access to the communication channel, with an upper bound Δ_a; and delays due to the actual transmission and computation at the controller, with an upper bound denoted by Δ_{tc}. Furthermore, we denote the total delay as $\Delta = \Delta_a + \Delta_{tc}$. For the purpose of a proper comparison between event-triggered and

self-triggered techniques we have selected techniques sharing the same triggering condition. As the event-triggered technique requires collocated sensors, we assume in what follows that the sensing node has access to the whole state vector. We remark that completely decentralized event-triggered techniques also exist which can be adapted to tolerate upper-bounded delays (Mazo Jr. and Tabuada, 2011; Wang and Lemmon, 2011).

Event-triggered control

In order to compensate for delays in the event-triggered controller, we propose to check condition (4.2) ahead of time, by predicting the value of V and S some amount of time in advance so that we can guarantee that the control update will take place before the condition is violated. This approach has a predictive flavor, and requires the sensing node to compute the control input locally. Let $A_\Delta = e^{A\Delta}$ and $B_\Delta = \int_0^\Delta e^{A(\Delta-r)}B \, \mathrm{d}r$. From a sample acquired at $t_s = nT \in [t_k, t_{k+1})$ one can estimate the value of the state, Δ units of time in the future as:

$$\hat{x}(t_s + \Delta) = A_\Delta x(t_s) + B_\Delta u(t_k).$$

As we wish to guarantee $V(nT) \leq S(nT)$ until the next controller update, we check instead if $\hat{V}(t_s+\Delta) \leq S(t_s+\Delta)$, where $\hat{V}(t) = \hat{x}(t)^T P \hat{x}(t)$. If $\hat{V}(t_s+\Delta) > S(t_s+\Delta)$ then the next controller update time is $t_{k+1} = t_s + \Delta - T$, and the sensor sends the predicted value $\hat{x}(t_s + \Delta - T)$ computed from the measurement $x(t_s - T)$. The controller then applies the new control input $u(t_{k+1}) = K\hat{x}(t_s + \Delta)$ at the time $t_{k+1} = t_s + \Delta$. Note that we impose the upper bound of the delay to the system by waiting until $t_s + \Delta$. This allows us to utilize the previously developed aperiodic techniques that allow for the minimum inter-transmission time to be guaranteed.

Self-triggered control

Due to the predictive nature of self-triggered controller, channel access delays can be prevented, i.e., $\Delta_a = 0$. This can be done by dynamically scheduling their channel access making use of the prediction of the next update event. However, computation and transmission delays may still affect the control loop. In this case, the sensor nodes are scheduled to transmit their measurements at $t_k - \Delta_{tc}$. The controller then receives the measurements early enough to compute $u(t_k) = K\hat{x}(t_k)$ so that the actuator can apply the input at t_k.

The controller estimates $\hat{x}(t_k)$ from the actual measurement $x(t_k - \Delta_{tc})$ as shown in the previous case. Then, the estimate is used to compute the control signal, and also to obtain the next update time $t_{k+1} = t_k + \tilde{g}_S(\hat{x}(t_k))$.

4.2 Communication strategies

We are now ready to introduce aperiodic communication strategies to perform control over wireless NCSs. Each communication strategy defines the usage of event-triggered and/or self-triggered control, specifies the scheduling policy and MAC characteristics. The proposed mechanisms follow the architecture presented in Figure 4.2, in which a plant to be controlled is instrumented with wireless sensors and actuators. Each sensor provides, through a Zero-Order-Hold (ZOH), measurements to the embedded event-generator. The controller receives wireless measurements from the sensor and produces a control input that is sent wirelessly to the actuators, which ZOH the input value. Finally, a Network Manager (NM) is in charge of scheduling the communications between the elements of the system.

4.2.1 Event-based communication

Under this paradigm, the control update times for the system are based on the event-triggered controller presented in Section 4.1.1 and are not known a priori. It is the sensors that decide on-the-fly when it is time to update the controller with new measurements by evaluating (4.2). While event-based controllers are certainly robust to disturbances, as there is a continuous supervision of the state of the plant, they also have clear shortcomings. First, the continuous supervision of the triggering condition imposes the availability of specific hardware dedicated to this task; and second, as the update times are not available a priori there is no possibility of implementing any dynamic scheduling policy. In order to guarantee the reliable communication between sensor, controller and actuator we propose the use of a TDMA MAC with a fixed scheduling policy for the assignment of communication slots to the sensors.

4.2.2 Predictive communication

Motivated by the fact that event-based communication imposes the restrictions of continuous supervision of the triggering condition, as well as a fixed scheduling policy, we propose a predictive communication mechanism.

A predictive strategy utilizes the self-triggered controller presented in Section 4.1.2. In this scheme, the controller will be responsible for computing the value $\tilde{g}_S(x(t_k))$ and calculating t_k in (4.2) for all plants in the network and transmitting these values to the NM. After all values of t_k are transmitted, the scheduling algorithms proposed in Section 4.3.3 are executed at the NM. Hence, a dynamic network bandwidth allocation is performed. This node then informs all the sensor, controller and actuator nodes of the message transmission/reception slots. We remark that the sensor node may also compute t_k, and transmit this information to the centralized unit which performs the scheduling. A drawback of this mechanism when compared to event-based communications is that it is in general less robust to disturbances due to the fact that sensor nodes in between transmissions are set to

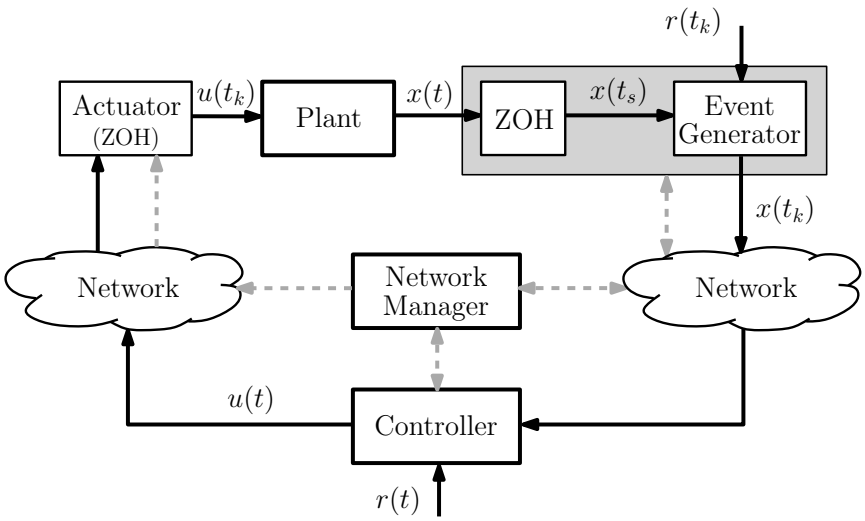


Figure 4.2: System architecture for an aperiodic networked control-loop.

a sleep mode in order to save battery life. If a disturbance affects the plant within this interval, no rejection takes place until a new sensor transmission. This is the motivation for the proposed hybrid communication mechanism which we discuss next.

4.2.3 Hybrid communication

In order to merge the benefits of the two previous mechanisms, we propose the use of a mechanism operating essentially as in the predictive communication case, but with the addition of a number of slots in the communication schedule not assigned a priori to any sensor. These extra communication slots are used to attempt transmissions on an event-triggered fashion when a disturbance, not accounted for in the self-triggered technique, takes place. The access to these slots is performed by a contention based channel access mechanism, as supported by IEEE 802.15.4. Thus, no guarantees on channel access can be provided for event-based transmissions, which means that any strict performance guarantees are only provided by the predictive communication side, while the event-based part of this hybrid mechanism provides a best-effort approach to reduce latency when responding to disturbances. We note that the predictive and hybrid approaches share the concept first proposed in (Cervin and Eker, 2003) in the context of control over wired networks: resources are reserved for the worst case scenario and during the execution one is able to dynamically allocate these resources to the control tasks according to the current state of the system. The unused resources are then utilized to improve the control performance or be allocated to soft tasks with no hard real-time constraints, e.g.,

Table 4.1: Current consumption (mA) of a Telos wireless platform for various microcontroller (μC) and radio operation modes.

Mode	Description	Measured Current
1	μC active, Radio Tx[1]	21.7 mA
2	μC active, Radio Rx/listen	22.8 mA
3	μC active, Radio OFF	2.4 mA
4	μC idle, Radio OFF	40 μA

monitoring messages. This will be further explored in the next section.

4.3 Implementation of communication strategies

4.3.1 IEEE 802.15.4 MAC

The implementation of the wireless protocol for aperiodic communication strategies builds upon the IEEE 802.15.4 MAC layer (IEEE 802.15.4, 2006) which was introduced in Chapter 2.5. We now propose required modifications to the standard which allow the implementation of the aperiodic communication strategies developed and a more efficient utilization of the communication resources.

4.3.2 MAC limitations

Sensors are assumed to be battery powered, and so we aim at maximizing their life span by reducing their power consumption. In order to design energy efficient communications, we look at how energy is spent in typical wireless nodes. The power consumption of the wireless sensor platform Telos (Polastre et al., 2005) is given in Table 4.1 (Prayati et al., 2010). The table clearly shows that communication is power expensive, and moreover, that the cost of listening and receiving messages is even more expensive than transmitting.

Naturally, to save large amounts of energy the nodes should keep their radios off and the microcontroller (μC) idle for as long as possible. This is achieved in two ways: reducing the communication slot size to reduce listening, reception and transmission times; and increasing the length of the inactive period. In the standard, two parameters are available to adjust the structure of the superframes which are the Superframe Duration (S.D.) and the Beacon Interval (B.I.). The rest of the protocol parameters are inferred from these two parameters.

While the standard covers most of our needs, it also poses some limitations which we address through the following modifications:

[1] This value represents the current consumption for the maximum transmission power. Hence, this value can be reduced for a lower transmission power.

1. Allowing the number of slots assigned to Contention-Access Period (CAP) and the Collision-Free Period (CFP) to be a free design parameter. This is opposed to the maximum of 7 Guaranteed Time Slots (GTSs) in the CFP, and a total maximum of 16 slots during CAP and CFP imposed by the IEEE 802.15.4 MAC standard. Moreover, we also allow the complete removal of the inactive period, which is not contemplated by the standard. These changes provide the desired increased versatility in adjusting latency in channel access and slot-size adjustments.

2. The use of a scheduling mechanism at the NM for the assignment of GTSs instead of the first-come-first-served (FCFS) request-based scheme in the IEEE 802.15.4 MAC specification.

Note that these modifications do not require changing the overall structure of the IEEE 802.15.4 MAC protocol, but only adjustments on the maximum allowed number of superframe slots, and the GTSs scheduling methodology. The modified superframe structure is presented in Figure 2.1 in Chapter 2.5. For more details on the GTS modification and implementation details see (Hernandez, 2011).

4.3.3 Proposed implementation

In this section we present the scheduling policies for the three proposed communication mechanisms.

Event-based communication

Under the event-based communication strategy, the time instants at which communication needs to occur are not known beforehand. We propose the use of a static TDMA scheme in which GTSs are reserved for each control loop in the network. Communication is then established at a particular GTS (assigned to sensor node i) only if the corresponding condition (4.2) is not satisfied for node i. This TDMA mechanism may introduce a delay between event generation and transmission of measurements, as a sensor may have to wait a certain time from the triggering of an event until a transmission in its assigned slot. Nevertheless, this delay is bounded and can be accommodated, as presented in Section 4.1.3.

In order to choose the B.I. length and the number of GTSs assigned to each control loop, the worst-case inter-event time for each triggering condition needs to be considered. The time between two consecutive slots assigned to node i must be lower than the minimum inter-event time $\tau_{\min}$ (given by (4.3)) of the condition of node i. This guarantees that no more than one transmission is needed for node i between two consecutive GTSs allocated to node i. Notice that, since the schedule is static, the GTSs that are not used cannot be reassigned to other nodes. The next mechanism overcomes this drawback.

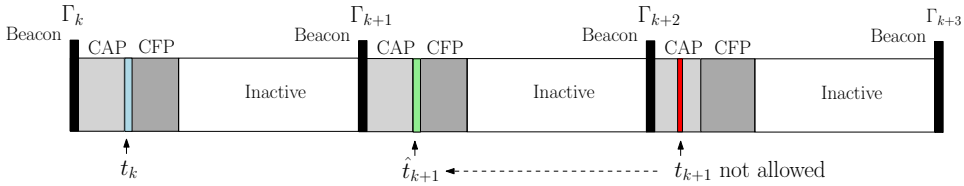


Figure 4.3: Earliest Deadline First (EDF) scheduling over wireless NCSs. Here we describe the scheduling of one of the nodes in the network using the self-triggered control scheme. The required triggering time t_{k+1} is not allowed since it does not occur during a CFP. An adjustment of this time to $\hat{t}_{k+1}$ is made in order to allocate t_{k+1} inside a GTS and guarantee (4.2) under EDF scheduling.

Predictive communication

As proposed in Section 4.3.1 the NM is responsible for the scheduling of GTSs for the nodes in the network and configuration of the wireless network. This node is connected to the PAN coordinator, having access to all the information transmitted by the sensor nodes. In order to allow for efficient usage of the available network resources, we propose to schedule the messages in the network according to an Earliest Deadline First (EDF) approach, which is known to be optimal for time-constrained schedules (Buttazzo, 2005). In what follows we denote by ΔCAP and ΔCFP the values of the CAP and CFP duration, respectively. The design of the GTS scheduling should take into account the following facts:

1. There are two types of messages: hard messages with high priority and hard deadlines, and soft messages with lower priority. The scheduling of both hard and soft messages should be done according to independent EDF schemes.

2. The GTS scheduling algorithm should only schedule the triggering times t_{k+1} given by (4.2), when $t_{k+1} \in [\Gamma_k + \Delta\text{CAP}, \Gamma_{k+1} + \Delta\text{CAP}]$, where Γ_k denotes the k-th superframe start time. If $t_{k+1} > \Gamma_{k+1} + \Delta\text{CAP}$, then the scheduler will only assign a GTS slot to the requesting node in a later superframe.

3. The triggering times t_{k+1} need to be adjusted to new values $\hat{t}_{k+1} \leq t_{k+1}$ if $\hat{t}_{k+1} \leq \Gamma_{k+1} + \Delta\text{CAP}$ in order to fit the triggering time inside the GTS.

This last condition is illustrated in Figure 4.3, where the triggering time t_{k+1} is adjusted to be $\hat{t}_{k+1} \leq t_{k+1}$ since $\hat{t}_{k+1} \leq \Gamma_{k+1} + \Delta\text{CAP}$, to prevent the transmission from falling in the CFP time two superframes ahead.

We now analyze the schedulability of a set of hard messages under the IEEE 802.15.4 protocol. As mentioned before, the active period is divided into the CAP and the CFP. Since no guarantees can be provided during the CAP, we assume that all hard messages are sent during the CFP. During this window, messages are scheduled according to the EDF algorithm.

Each message can be characterized by a triple (τ_k, C, d), where τ_k represents the period of a message (or minimum inter-transmission time for aperiodic messages), C is the maximum transmission time and d is the relative deadline (not necessarily equal to the period τ_k). Notice that control loops involve at least two types of messages: sensor to controller and controller to actuator. Sensor messages are always followed by actuator messages, hence they are never sent at the same time. To model this precedence constraint, we assume an offset φ_a for the actuator messages, equal to the deadline of the sensor message plus the computation time of the control law at the controller node.

The inter-transmission times of sensor-actuator message pair is defined by equation (4.2). Since when offline the scheduler is not aware of the evolution of the state, worst-case inter-transmission times need to be considered in the schedulability analysis. The minimum inter-transmission time for both the sensor and the actuator message is given by $\tau_{\min}$. However, since the transmission times t_k depend on the state of the plant and are not known in advance, we need to select $\tau_{\min}$ as the worst inter-transmission time over all possible initial conditions as defined in (4.3). Indeed, as in the case of a periodic technique, enough resources need to be reserved beforehand assuming worst-case conditions, even though these might rarely occur. However, as the state of the plant is measured, the predictive policy modifies these requirements at run-time, and reserved bandwidth can be reallocated among existing nodes. This property represents the main advantage of the predictive paradigm.

The deadline of the actuator message represents the maximum admissible bound on the delay between a sensor message is received by the controller and the arrival of the actuator message. The deadline of the sensor message represents the maximum admissible bound between the measurement of a sensor and the arrival of its corresponding sensor message. Notice that for control systems only the delay δ between measurement and actuation is relevant, *i.e.*, the sum of the sensor and actuator deadlines.

Given a set of n hard messages plus p control loops, the schedulability conditions (sufficient and necessary) under non-preemptive EDF are (Zheng and Shin, 1994):

$$\sum_{i=1}^{n+2p} \frac{C_i}{\tau_{\min i}} \leq 1$$

$$\sum_{i=1}^{n+2p} \left\lceil \frac{t - d_i - \varphi_i}{\tau_{\min i}} \right\rceil^{+} C_i + C_m \leq t, \quad \forall t \in \mathcal{S}$$

where the set $\mathcal{S}$ is defined as:

$$\mathcal{S} = \cup_{i=1}^{n+2p} \mathcal{S}_i,$$

$$\mathcal{S}_i = \left\{ d_i + m\tau_{\min i} : m = 0, 1, \ldots, \left\lfloor \frac{\tau_{max} - d_i - \varphi_i}{\tau_{\min i}} \right\rfloor \right\},$$

$$\tau_{max} = \max \left\{ d_1, \ldots, d_{n+2p}, (C_m + \sum_{i=1}^{n+2p} (1 - d_i/\tau_{\min i}) C_i)/(1 - \sum_{i=1}^{n+2p} C_i/\tau_{\min i}) \right\}$$

where $C_m := \max_i C_i$ is the maximum transmission time for all possible messages and $\lceil z \rceil^+ = \min\{n \in \mathbb{N}_0 | n \geq z\}$ and $\lfloor z \rfloor = \max\{n \in \mathbb{Z} | n \leq z\}$, and $\varphi_s = 0$ for all sensor messages.

The previous set of equations assume that messages can be transmitted at any time. However, under IEEE 802.15.4 hard messages are not transmitted during the CAP, since guarantees cannot be provided, and during the inactive period to save energy. We model this property by means of two dummy tasks with periods $\tau_{\min i}$ equal to the superframe duration S.D., and deadlines equal the inactive period length and ΔCAP, respectively. Moreover, the dummy task modeling the inactive period should have an offset equal to the length of S.D.. In this way, equations (4.4) and (4.5) can be used to analyze the schedulability under the IEEE 802.15.4 protocol, where now n represents the number of hard messages plus these two dummy messages. For other related scheduling issues we refer the interested reader to (Anta and Tabuada, 2009).

As mentioned before, the schedulability analysis has to be based on the worst-case inter-transmission time $\tau_{\min}$ as defined in (4.3) since the initial condition is in general not known in advance, or disturbances might steer the system to the worst-case condition. Nevertheless, the inherent dynamic nature of the predictive policy allows the scheduler to reallocate resources in an online manner. We note that fifferent strategies could be applied for the dynamic bandwidth allocation: allocate those GTS to soft messages, or assign the GTS among all existing messages according to the needs of each node. For instance, a control system could take hold of GTS slots previously assigned to other control loops in order to improve its own performance.

Hybrid communication

The hybrid communication mechanism utilizes the same scheduling policy defined for the predictive communication strategy, where GTSs are scheduled in an EDF fashion. Additionally, no GTSs are provided for the event-triggered messages since these are only granted access during the CAP, for "best-effort" data transmission.

4.4 Experimental setup

In order to evaluate the performance of the proposed aperiodic communication mechanisms, we built a lab process with a wireless network shared by two control loops and several independent monitoring nodes transmitting auxiliary messages, with no hard deadlines. The control loops are regulating the two double tank processes introduced in Chapter 2. Figure 2.3 shows the setup of two double tank systems and eight independent monitoring nodes. Each double tank is composed of one sensor and one actuator node. The controller node is also the NM in our setup. A scheduler node is added and connected to the NM. This unit performs scheduling computations for each mechanism, reducing the computation load of the NM.

4.4.1 Double tank system

The goal of the experiment is to control the water level of the lower tank L_2 by adjusting the motor voltage V_p accordingly. Tracking of a reference signal $r(t)$ can be achieved by using feedforward tracking, with the control input defined as,

$$u(t) = Kx(t) + Mr(t),$$

where $x = \begin{bmatrix} \Delta L_1 & \Delta L_2 \end{bmatrix}^T$ and the state-feedback matrix K is assumed to be chosen so that the closed-loop system matrix $\bar{A} = A - BK$ is Hurwitz. Matrix M is calculated to ensure setpoint tracking of the undisturbed closed-loop system for a constant command signal $r(t) = \bar{r}$.

In order to apply the aperiodic techniques proposed earlier, the state $x(t)$ must be converging to the origin and $u(t)$ must be a state-feedback controller. This is achieved by shifting the system's origin to the reference value we wish to track. If we assume that the reference is constant, we have the new continuous-time state-space system:

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B\Delta u, \\ \Delta u &= K\Delta x, \end{aligned}$$

for appropriate values of A and B, where $\Delta x(t) = x(t) - \bar{r} = [\Delta L_1 - \Delta L_{1\text{ref}}, \Delta L_2 - \Delta L_{2\text{ref}}]$, and $\Delta L_{i\text{ref}}$, for all $i = 1, 2$, is the steady-state value of the upper and lower tank and $\Delta u(t) = u(t) - M\bar{r}$, which achieves $\lim_{t\to+\infty} \|\Delta x(t)\| = 0$. By selecting the desired reference value of the lower tank $\Delta L_{2\text{ref}}$, the upper tank reference $\Delta L_{1\text{ref}}$ follows by solving the state-space equations in steady-state, i.e., $\Delta\dot{x} = 0$.

In order to guarantee robustness of the predictive scheme with respect to disturbances, an upper bound on the inter-sampling times $\tau_{\max} = 10\,\text{s}$ is imposed. The performance function $S(t)$ in (3.9) is defined by $R = 0.1Q$ and Q is selected as the identity matrix, for all mechanisms. The state-feedback matrix $K = \begin{bmatrix} -0.39 & -0.95 \end{bmatrix}$. We compute the minimum inter-transmission time $\tau_{\min}$ for this system (4.3) using Theorem 5.1 in (Mazo Jr. et al., 2009), which, for this physical system, gives a minimum time $\tau_{\min} = 1$ s. Hence, the inter-transmission times τ_k for the control-related messages will be in the range $[1, 10]$ s. However, these times may be (conservatively) adjusted to be allocated at a GTS.

A periodic technique of the control loops is implemented for comparison purposes. The sampling period of the periodic implementation is set to 0.64 s, which is the closest value to $\tau_{\min} = 1\,\text{s}$ allowed by the protocol.

4.4.2 Communication network

The IEEE 802.15.4 protocol has been partially implemented in TinyOS by (Hauer, 2009) and validated in the Telos platform. An extension of the work by (Hauer, 2009) to include the CFP and the GTS mechanism has been performed at KTH by (Hernandez and Park, 2011) for the same platform. The implementation of

Table 4.2: MAC Parameters

Scheme	CAP	CFP
Event-based	1	10
Predictive	1	10
Hybrid	7	4

the protocol used in our setup is available and documented in (Hernandez, 2011), where the modification of the IEEE 802.15.4 standard as detailed in Section 4.3.2 was performed.

MAC parameters

The number of CAP slots and GTSs are defined in Table 4.2 for each of the communication mechanisms. Moreover, S.D.= 323.1 ms, B.I.= 646.3 ms and each of the 11 slots has a duration of 29.4 ms. When performing feedback control of the double tank system, two slots for sensing and actuation communication must be defined for the mechanisms. Therefore, the minimum number of GTSs to be defined must be larger or equal to four.

Auxiliary messages from monitoring nodes, are scheduled during the available GTSs for the event-based and predictive mechanisms, while in the hybrid mechanism monitoring nodes communication takes place during the CAP.

4.5 Experimental results

We now provide a description of the performed experiments and the obtained results with respect to control performance and energy efficiency as well as the network bandwidth utilization of the proposed strategies.

The initial water level of the lower tank is set to 5 cm and the reference $\bar{r} = 10$. An input disturbance of magnitude 1 V is applied continuously to the pump actuation starting at time $t = 130$ s which will allow an analysis on how well each mechanism rejects disturbances. Whenever the disturbance is detected at the sensor, the node identifies its magnitude and transmits this information to the controller. The controller then adjusts the control input to reject this disturbance. The experiment has a total duration of 220 s. The expected upper bound on the communication and computation delay is set to $\Delta_{tc} = 35$ ms and will be compensated by each communication mechanism.

4.5.1 Control performance and energy consumption

The control performance and energy efficiency is evaluated for one double tank system. With respect to control performance, we analyze both the time response of the water levels and the Integral of the Absolute Error (IAE) of the lower tank

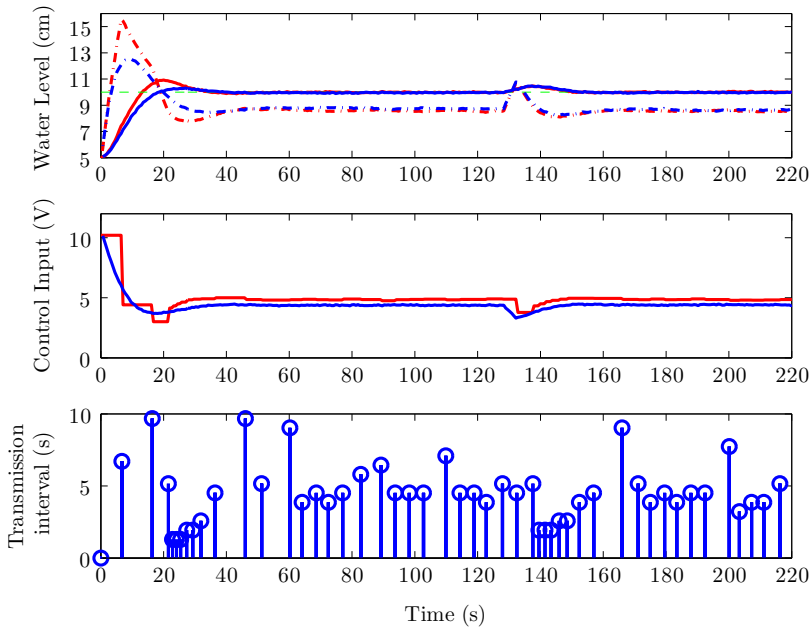


Figure 4.4: Experimental results for event-based communication. The upper plot depicts the evolution of the water tank level and middle plot the control input values for event-based mechanism (red) and a periodic controller (blue). Upper (dash-dotted line) and lower (solid line) water levels are presented.

water level, which is calculated as IAE $= \int_0^\infty |r - \Delta L_2(s)| \, \mathrm{d}s$. The energy efficiency of each communication mechanism is given by the wireless sensor battery lifetime expectation. For this calculation we sum the total current consumption of the wireless node over the complete experiment and repeat it until the full battery capacity of 2900 mAh is consumed.

Event-based communication

The event-based communication mechanism is implemented with $T = 0.64\,\mathrm{s}$ as a new measurement is acquired 5 ms before the start of the GTS allocated to the sensor. In this case, the time delay in the access of the communication channel will be $\Delta_a = 0.64\,\mathrm{s}$ as this is the time between consecutive GTSs allocated to the same control loop. Figure 4.4 shows the time response and inter-transmission times of one double tank system for event-based and periodic mechanisms. It is observed that both control techniques track the reference signal with similar behavior. Table 4.3 depicts the values for the IAE, number of transmissions and battery life span of the wireless sensor nodes. The IAE analysis show that the event-based scheme

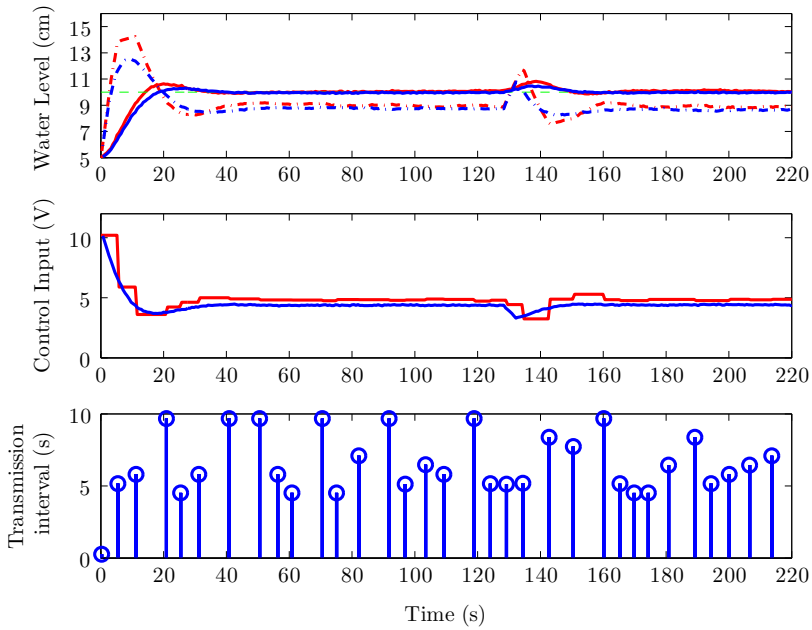


Figure 4.5: Experimental results for predictive communication. Signal legends are shown as in Figure 4.4.

outperforms the periodic technique. A faster rise time is achieved, followed by a fine adjustment of the water flow when closer to the reference. Additionally, the disturbance is efficiently rejected. Even though the number of transmissions of the event-based scheme is only 14.1% of the periodic, the battery lifetime increase is of 38% and not 700% as it could be expected if only the number of transmissions would consume energy. This difference originates from the fact that the wireless nodes still need to turn on the radio and μC to receive beacon messages at each B.I., and spend energy during the inactive period. Thus, high reductions in the number of transmissions do not imply the same ratio of energy savings.

Predictive Communication

In order to provide the same performance guarantees as the event-based mechanism, the predictive mechanisms is implemented with $T = 0.64\,\text{s}$. Figure 4.5 shows the time response and inter-transmission times of the double tank system for the predictive and periodic mechanisms. As in the previous case, both control techniques track the reference signal with similar behavior. From Table 4.3, IAE analysis show that the predictive scheme outperforms the periodic and event-based schemes during the transient but has a much worse performance when rejecting the disturbance. This

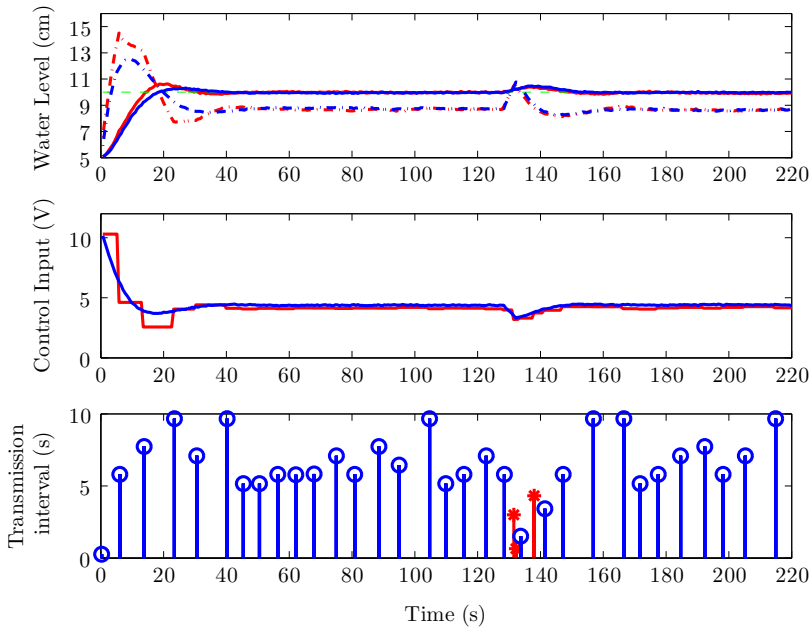


Figure 4.6: Experimental results for hybrid communication. Signal legends are shown as in Figure 4.4. The red star marker represents an event-based transmission and the blue round marker represents a predictive transmission.

occurs due to the fact that the sensor node is only active at scheduled transmission times, and not at every superframe as the periodic and event-based schemes. Since the event-based mechanism and the predictive mechanism share the same performance criterion, both are expected in theory to have similar behavior, in the absence of disturbances. Different performances before the introduction of the disturbance can be explained by model inaccuracies and noise affecting the real plant. The number of transmissions using this scheme is 9.8% of the periodic, and lower than the event-based mechanism. This can be explained using the above arguments, due to the noise affecting the sensor readings. The battery lifetime is increased by 58.6% compared to the periodic scheme, while maintaining good control performances.

Hybrid Communication

The hybrid communication mechanism was implemented with two measurement acquisition periods in its event-based component: a fast acquisition with $T = 10\,\text{ms}$ (Hybrid^{F}) and slow acquisition every $T = 0.64\,\text{s}$ (Hybrid^{S}). In the Hybrid^{F} case, the event-based component will "continuously" check if (4.2) is violated throughout the whole superframe. On the other hand, in the slow technique Hybrid^{S}, (4.2)

is checked once, at the beginning of the CAP. Figure 4.6 depicts the results for the implementation of Hybrid$^{\mathrm{F}}$. The implementation of Hybrid$^{\mathrm{S}}$ showed a similar behavior. Moreover, the hybrid mechanism tracks the reference signal and rejects the disturbance.

The inter-transmission times τ_{ki} are depicted for the case in which the transmission was generated by the event-based (blue circle) or the predictive mechanism (red star). As seen in the figure, only predictive transmissions take place during transient, and event-based transmissions occur during the disturbance rejection phase. From Table 4.3, the IAE during the transient is kept close to the predictive scheme for both hybrid communication strategies as expected. The benefit of using the hybrid scheme become clear when the disturbance occurs. In this case, event-based transmissions occur when rejecting the disturbance. In addition, the Hybrid$^{\mathrm{F}}$ has a very low battery life duration since the μC computes at all times (mode 3 in Table 4.1) and never sleeps. In the Hybrid$^{\mathrm{S}}$ implementation, the battery life increases to the same levels as the other aperiodic schemes since the node is set to sleep if no transmission takes place. The number of transmissions of the Hybrid$^{\mathrm{S}}$ is 10.1% of the periodic, lower than the event-based scheme and close to the predictive scheme. Note that a higher battery consumption is obtained by the hybrid mechanism when compared to a predictive scheme since the triggering condition is verified at every superframe.

4.5.2 Network Bandwidth Utilization

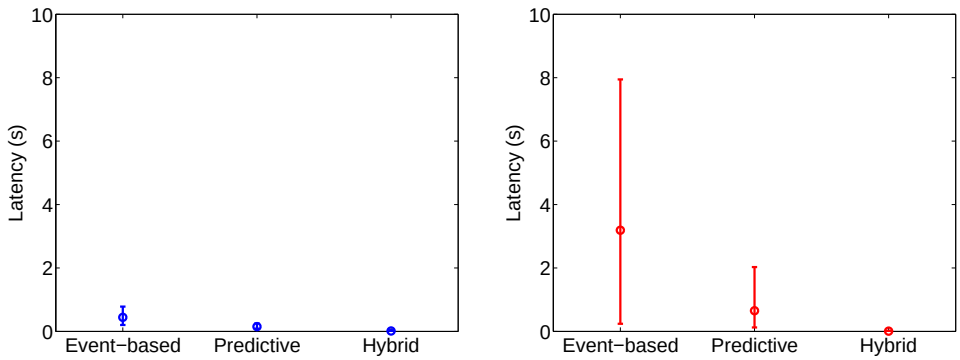
The network bandwidth utilization is characterized by how well the network is shared among the wireless nodes. To evaluate the network bandwidth utilization of each mechanism we define two message deadline types for the soft messages, which represent traffic patterns that could be found in real wireless NCSs. Each soft message has a size of 64 bytes.

- Slow traffic: Slow periodic transmissions, with period T_m = 5B.I. = 3.3 s, through the whole experiment.
- Bursty traffic: Fast periodic transmissions, with period T_m = 1B.I. = 0.64 s, during 25B.I. = 16 s, starting at $t = \{0, 120, 200\}$ s and slow periodic transmissions during the rest of experiment.

We analyze the latency experienced by these nodes in each of the mechanisms, where by latency we mean the time between each transmission is generated by the application and an acknowledgment was received for that particular message. Figure 4.7 depicts the latency analysis for both traffic patterns. For each mechanism it is shown the minimum, maximum and mean value of the latency for all the eight nodes during the experiments. These values are the averages of three experimental runs. By using a bursty traffic pattern, the latency increases in all schemes, with a higher impact in the event-based mechanism as it is based on a static scheduling mechanism. In this case, the queue of soft messages is large, since only six slots

Table 4.3: Performance evaluation of the proposed aperiodic mechanisms. The IAE performance indicator for different experiment phases, number of updates (N_{updates}) and battery life in days, are depicted for each of the mechanisms.

Scheme	$\text{IAE}^{[0,130]}$	$\text{IAE}^{[130,220]}$	$\text{IAE}^{[0,220]}$	N_{updates}	Battery life (days)
Event-based	75.78	12.48	88.26	49	879.84
Predictive	72.87	28.78	101.65	34	1010.18
Hybrid^{F}	67.05	16.51	83.55	36	63.66
Hybrid^{S}	68.19	16.87	85.06	35	910.14
Periodic	73.08	15.42	88.50	347	636.81



(a) Latency for Slow Traffic

(b) Latency for Bursty Traffic

Figure 4.7: Latency analysis of the monitoring wireless nodes, with respect to the traffic pattern. For each mechanism, the plot represents the minimum (lower bar), mean (round marker), and maximum delay (upper bar).

of the superframe are available to be shared among eight monitoring nodes. The periodic scheme latency results are the same as the event-based scheme, since static scheduling is also performed. For the predictive scheme, the benefit of using a dynamic scheduling mechanism is clearly observed. By adjusting the GTS scheduling as a function of the control requirements, more space is available for the monitoring nodes to transmit. The hybrid mechanism was evaluated with eight, ten and twelve monitoring nodes, but no differences in the latency values were observed. This mechanism shows the most efficient network bandwidth utilization since its latency is low, when comparing to the other schemes and is in the interval $[2, 40]$ ms. Each node is able to transmit messages during the CAP, where several other nodes may attempt to transmit. However, no GTS schedule queueing will occur in this case. Naturally, there is an advantage to allow monitoring nodes to transmit soft messages during the CAP, instead of GTSs.

4.6 Summary

In this chapter, we have proposed aperiodic communication strategies for control systems that are specially designed for wireless NCSs. We provided a joint selection of the aperiodic sampling technique, the controller, the wireless MAC protocol and the scheduling algorithm, that together guarantee a required control performance while efficiently using the network resources. In order to implement these mechanisms, we also identified limitations of the current IEEE 802.15.4 MAC protocol and propose slight modifications to enable the proposed techniques.

Experimental results demonstrated the efficiency of the proposed communication mechanisms with respect to control and communication performance. All the mechanisms achieve set-point tracking and disturbance rejection, with closed-loop control performances close to the ones obtained with a traditional periodic paradigm. Finally, we illustrated how these improvements translate in terms of energy savings and network bandwidth utilization. While in the present work we focus on linear systems, we remark that nonlinear systems can be addressed in a similar fashion (Tabuada, 2007; Anta and Tabuada, 2010a).

Chapter 5

Event-based sampling-rate selection

In this chapter, we propose an event-based sampling-rate selection to select between fast and slow periodic sampling rates. The approach is based on linear quadratic control (LQ) methods and performance guarantees given with respect to a quadratic control cost function. Particularly, the devised method guarantees that the same control performance of a periodically sampled controller is obtained, while reducing the rate of sampling/transmission. We denote the controller using such event-based sampling-rate selection policy as a down-sampled controller. The technique is developed considering the disturbance-free case, as well as when sporadic disturbances affect the system. Using this strategy, the transient period is performed with fast control updates, while the steady-state control is performed at a slower rate. The proposed scheme is applied to the tracking of piecewise constant references, rejection of piecewise constant disturbances, as well as to the control of first-order systems with time-delay. Additionally, we propose a scheduling algorithm and MAC scheme so that the down-sampled controller can be used when multiple control-loops share the same wireless network. The application of the down-sampled controller to linear stochastic systems is also discussed. Numerical examples are proposed to validate the down-sampled controller, where we provide comparisons to the recent work of (Antunes and Heemels, 2014) and (Gommans et al., 2014). Furthermore, we implement and evaluate the down-sampled controller in a wireless networked control system setup composed of two double tank systems which share the same wireless network. This setup is implemented in the co-simulator introduced in Chapter 8.

The rest of this chapter is organized as follows. Section 5.1 presents the system setup, the problem formulation and preliminaries on LQ control. The down-sampled controller is presented in Section 5.2 for the case where no disturbances affect the system. The design under disturbances is considered in Section 5.3. In Section 5.4 we briefly discuss the application of the down-sampled controller to linear stochastic systems and in Section 5.5 to the tracking of piecewise constant reference signals, rejection of piecewise constant disturbances and to first-order systems with time-delay. Section 5.6 proposes an implementation of the down-sampled controller when several control loops share the same wireless network. Numerical

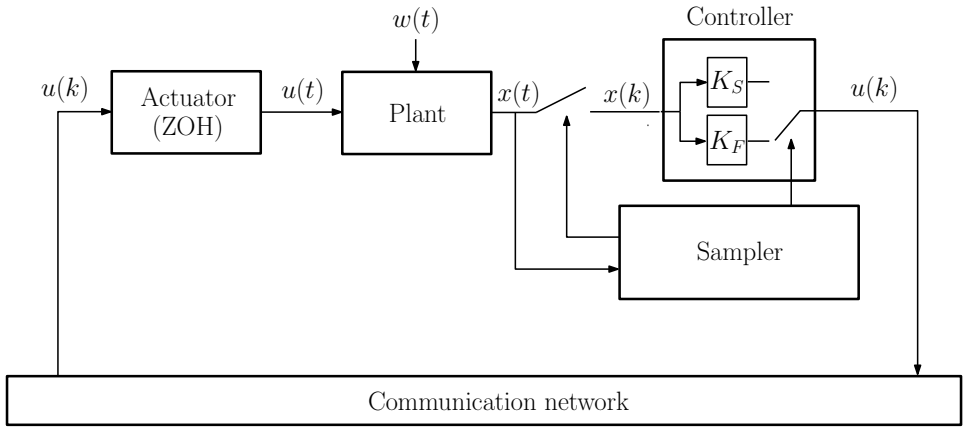


Figure 5.1: Down-sampled controller architecture. The sampler is responsible for selecting when a sample is taken, which control law to use and when a control input is sent to the actuator.

examples are provided in Section 5.7 which validate the proposed mechanism and illustrate its benefits. In Section 5.8 an experimental scenario is proposed to evaluated the down-sampled controller in a multiple control loop scenario. The results for this evaluation are presented in Section 5.9. Finally, Section 5.10 concludes this chapter.

5.1 Problem formulation

5.1.1 System model

Assume the plant is a continuous-time linear time-invariant system,

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{5.1}$$

with a state $x(t) \in \mathbb{R}^n$ and input $u(t) \in \mathbb{R}^m$. The control performance is defined by the average quadratic cost

$$\bar{J} = \lim_{T\to\infty} \frac{1}{T} J^{[0,T]}, \tag{5.2}$$

where

$$J^{[0,T]} = \int_0^T \left(x(t)^T Q_c x(t) + u(t)^T R_c u(t) \right) \mathrm{d}t,$$

with $Q_c \succeq 0$, $R_c \succ 0$. When the system is not affected by disturbances, the control performance is given instead by $\bar{J} = J^{[0,T]}$, as in that case (5.2) would converge to zero. We assume that (A, B) is controllable.

Sampling system (5.1) with a zero-order-hold (ZOH) for a baseline sampling period $h = 1$, gives the discrete-time system

$$x(k+1) = \Phi_1 x(k) + \Gamma_1 u(k), \tag{5.3}$$

where $\Phi_h = \mathbf{e}^{Ah}$, $\Gamma_h = \int_0^h \mathbf{e}^{As} B \, \mathrm{d}s$. For $h = 1$ we drop the subscript of Φ and Γ.

5.1.2 Scheduling and control

The sampling and control of the system is performed synchronously as depicted in Figure 5.1. Whenever a new control input is computed, it is transmitted to the actuator over a communication network and applied to the plant. Hence, we assume that communication and computation delays are inexistent for now. Later in Section 5.5, we analyze the required modifications to the down-sampled controller when the system is affected by known delays.

Let $q(k) \in \{q_F, q_S\}$ represent the current sampling mode of the sampler, which can be a fast sampling mode q_F or slow sampling mode q_S and its evolution is governed by the transition map

$$q(k) = \Pi\left(x(k), q(k-1)\right), \tag{5.4}$$

to be designed in Section 5.2. The time of the next sampling instant τ_{k+1} is dependent on the current mode and is given by

$$\tau_{k+1} = \begin{cases} \tau_k + 1 & \text{if } q(k) = q_F, \\ \tau_k + \delta_S & \text{if } q(k) = q_S, \end{cases} \tag{5.5}$$

where $\delta_S \geq 1$. Hence, the system is either sampled consecutively at every step, or down-sampled with interval δ_S. Note that $\delta_S = 1$ corresponds to the conventional equidistantly sampled control system.

The control input computed at the controller is given by the switched controller

$$u(k) = \begin{cases} K_F x(k) & \text{if } q(k) = q_F, \\ K_S x(k) & \text{if } q(k) = q_S, \end{cases} \tag{5.6}$$

where K_F and K_S are the controller gains. Moreover, the switching instant between sampling modes is denoted by t_s and the last switching time as t_s^-.

5.1.3 Problem statement

Let the average down-sampled controller cost (5.2) be denoted by $\bar{J}_{DS}$ and its transmission rate by

$$\bar{R}_{DS} = \lim_{T \to \infty} \frac{1}{T} \sum_{k=0}^{T} 1_{\tau_k \leq T}.$$

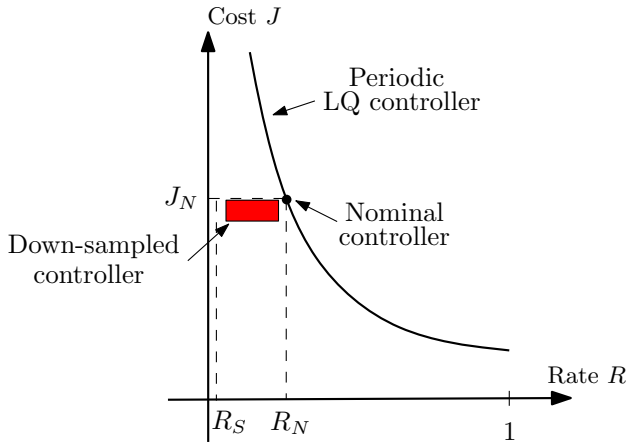


Figure 5.2: Problem illustration. The down-sampled controller is designed to achieve a lower or equal cost, at a lower sampling rate, than that of any periodic LQ controller sampled at a nominal rate. Solution region represented in red.

In the same manner, let us denote by $\bar{J}_N$ the average cost (5.2) of an LQ controller periodically sampled at a nominal period δ_N, where $1 < \delta_N < \delta_S$, and by $\bar{R}_N$ the average transmission rate of the nominal controller.

Problem 5.1. *Given the slow sampling interval δ_S, design a transition map Π (5.4) and a state-feedback controller (5.6) that guarantees that the down-sampled controller achieves*

1. $\bar{J}_{DS} \leq \bar{J}_N$ *and*
2. $\bar{R}_{DS} < \bar{R}_N$.

Problem 5.1 corresponds to the design of a down-sampled controller which achieves a control cost lower or equal than that of any periodic LQ controller sampled with nominal period δ_N, while utilizing a lower transmission rate. An illustration of the problem is given in Figure 5.2.

5.1.4 Preliminaries

We now present a revision of the classic discrete-time LQ control methods which is essential to the development of the methods proposed in this chapter.

Assuming that system (5.3), sampled with a baseline period $h = 1$, is controlled periodically with a fixed period $\delta > 0$, the state-feedback control law given by

$$u(k) = \begin{cases} K_\delta x(k) & \text{if } k \in \mathcal{K}_\delta^{[0,\infty)}, \\ u(k-1) & \text{otherwise,} \end{cases} \tag{5.7}$$

where $\mathcal{K}_\delta^{[l,l']} = \{k \in \mathbb{N}_0 : \mathrm{rem}(k,\delta) = 0 \ \wedge \ l \leq k \leq l'\}$, minimizes the following average quadratic cost function

$$\bar{J}_\delta = \lim_{T\to\infty} \frac{1}{T} J_\delta^{[0,T]}$$

where

$$J_\delta^{[0,T]} = \sum_{k\in\mathcal{K}_\delta^{[0,T]}} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_\delta & N_\delta \\ N_\delta^T & R_\delta \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \tag{5.8}$$

and

$$\begin{bmatrix} Q_\delta & N_\delta \\ N_\delta^T & R_\delta \end{bmatrix} = \int_0^\delta \mathbf{e}^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}^T s} \begin{bmatrix} Q_c & 0 \\ 0 & R_c \end{bmatrix} \mathbf{e}^{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} s} \mathrm{d}s.$$

Let $J_\delta^{[l,\infty)}$ denote the infinite-horizon control cost calculated from time $k = l$, $l \in \mathcal{K}_\delta^{[l,\infty)}$. It is known that the infinite-horizon control cost given by (5.8) under the optimal control policy (5.7) is $J_\delta^{[l,\infty)} = x(l)^T P_\delta x(l)$, where $P_\delta \succ 0$ is the solution to the algebraic Riccati equation,

$$P_\delta = \Phi_\delta^T P_\delta \Phi_\delta + Q_\delta^T - (\Phi_\delta^T P_\delta \Gamma_\delta + N_\delta)(\Gamma_\delta^T P_\delta \Gamma_\delta + R_\delta)^{-1}(\Gamma_\delta^T P_\delta \Phi_\delta + N_\delta^T), \tag{5.9}$$

and the feedback gain is given by

$$K_\delta = -(\Gamma_\delta^T P_\delta \Gamma_\delta + R_\delta)^{-1}(\Gamma_\delta^T P_\delta \Phi_\delta + N_\delta^T). \tag{5.10}$$

See (Åström and Wittenmark, 1990). The finite-horizon control cost for the interval $k \in [l, l']$, for a system actuated with period $\delta > 0$ but discretized with baseline period $h = 1$ as in (5.3), is

$$J_\delta^{[l,l']} = \sum_{k=l}^{l'} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_1 & N_1 \\ N_1^T & R_1 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \tag{5.11}$$

where from now on we drop the subscript in Q, R and N for $h = 1$. Assuming that the controller is given by (5.7) with K_δ as (5.10) and that a sampling and control instant takes place at time $k = l$ and $k = l'$, by induction one can show that the cost (5.11) can be expressed as

$$J_\delta^{[l,l']} = J_\delta^{[l,\infty)} - J_\delta^{[l',\infty)} = x(l)^T P_\delta x(l) - x(l')^T P_\delta x(l'). \tag{5.12}$$

Note that using the state-feedback controller (5.7) with K_δ given by (5.10) does not minimize (5.11).

5.2 Down-sampled controller

In this section, we propose a down-sampled controller which solves Problem 5.1. The design is split between the case where disturbances are absent, which we consider in this section, and when disturbances affect the system, which we discuss in the following section.

Let $J_F^{[l,l']}$, P_F and K_F denote the finite-horizon costs (5.11), the solution of the algebraic Riccati equation (5.9) and the state-feedback gain (5.10), respectively, for the baseline period. In the same manner, these parameters are denoted with subscript S and N for the slow and nominal periods δ_S and δ_N, respectively. Additionally, the down-sampled controller cost is denoted with subscript DS.

Assumption 5.1. *We assume that δ_N, δ_S and the baseline period are non-pathologic, so that the system remains controllable (Chen and Francis, 1995) when sampled with these sampling periods. Moreover, we assume that $J_F^{[0,\infty]} < J_N^{[0,\infty]} < J_S^{[0,\infty]}, \forall x \in \mathbb{R}^n$, which implies $P_F \prec P_N \prec P_S$.*

Theorem 5.1. *Consider system* (5.3) *with sampling and actuation governed by* (5.5)*. Let the down-sampled controller transition map be defined by*

$$\begin{aligned} &\Pi\left(x(k), q(k-1) = q_F\right) = \begin{cases} q_S & \text{if } x(k) \in \mathcal{G}_F \\ q_F & \text{otherwise} \end{cases} \\ &\Pi\left(x(k), q(k-1) = q_S\right) = q_S \end{aligned} \tag{5.13}$$

where

$$\mathcal{G}_F = \{x(k) \in \mathbb{R}^n \mid x(k)^T \Lambda_F x(k) \leq \sigma_F\}, \tag{5.14}$$

$\Lambda_F = P_S - P_F$, $\sigma_F = x(0)^T (P_N - P_F) x(0)$ *and let the system be initialized with $q(0) = q_F$. If $x(0) \neq 0$ there is a switching between the fast and slow mode at the switching time*

$$t_s = \inf\{k \geq 0 : x(k)^T \Lambda_F x(k) \leq \sigma_F\}. \tag{5.15}$$

Let the control input $u(k)$ in (5.6) *have K_F and K_S given by* (5.10) *for the baseline period $h = 1$ and slow period $\delta_S > h$, respectively. Through this design, the down-sampled controller achieves:*

1. *a cost no larger than a nominal LQ controller and hence stability of the closed-loop system, while*
2. *utilizing a lower number of samples than the nominal controller,*

thus providing a solution to Problem 5.1.

Proof. We start by devising the transition map (5.13) which allows for the down-sampled controller to solve Problem 5.1. Afterwards, we show that under this transition map, a switching occurs at time t_s and it occurs only once. The system is

initialized in the fast mode, i.e., $q(0) = q_F$, and suppose a switching from the fast q_F to slow mode q_S occurs at time $t_s > 0$. Hence, by using (5.12), the control cost of the down-sampled controller from time $k = 0$ is given by

$$\begin{aligned} J_{DS}^{[0,\infty)} &= J_F^{[0,\infty)} - J_F^{[t_s,\infty)} + J_S^{[t_s,\infty)}, \\ &= x(0)^T P_F x(0) - x(t_s)^T P_F x(t_s) + x(t_s)^T P_S x(t_s), \end{aligned} \tag{5.16}$$

which has to be guaranteed to be smaller or equal than the nominal control cost

$$J_N^{[0,\infty)} = x(0)^T P_N x(0). \tag{5.17}$$

The transition map with region $\mathcal{G}_F$ in (5.14), as well as the switching instant (5.15), are achieved by re-arranging the above terms and setting the inequality. At the switching instant t_s, the cost-to-go is $J_{DS}^{[t_s,\infty)} = x(t_s)^T P_S x(t_s)$. Hence, the system will asymptotically converge to the origin and a single switch occurs. We now prove that the switching instant t_s given by (5.15) exists. Due to the fact that $\delta_S > \delta_F$ and Assumption 5.1, it holds that $P_S \succ P_F$. Additionally, the state $x(k) = \bar{\Phi}^k x(0)$, where $\bar{\Phi} = \Phi + \Gamma K_F$ is the closed-loop system matrix for the baseline period, and $\bar{\Phi}$ is a Schur matrix since K_F is a stabilizing state-feedback gain. By taking the limit of the left-hand side of the switching condition (5.15) with $\Lambda_F = P_S - P_F \succ 0$, it holds that

$$\lim_{k\to\infty} x(k)^T \Lambda_F x(k) = \lim_{k\to\infty} x(0)^T {\bar{\Phi}^k}^T \Lambda_F \bar{\Phi}^k x(0) = 0,$$

since $\bar{\Phi}$ is Schur and $\Lambda_F \succ 0$. For the switching instant t_s to exist, $\sigma_F > 0$, which is always true since $P_N \succ P_F$ as $\delta_N > \delta_F$.

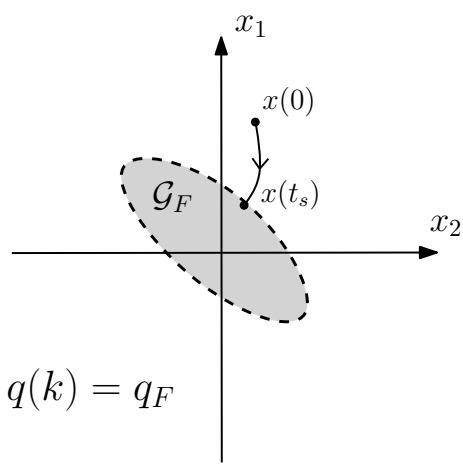
The switching condition guarantees by design the asymptotic stability of the closed-loop system since the infinite-horizon cost of the down-sampled controller (5.16) is guaranteed to be bounded by a bounded infinite-horizon cost (5.17).

The number of transmissions performed by the down-sampled controller over an horizon T is $\Sigma_{DS}^{[0,T]} = t_s + \left\lfloor \frac{T-t_s}{\delta_S} \right\rfloor + 1$, while for the nominal controller is $\Sigma_N^{[0,T]} = \left\lfloor \frac{T}{\delta_N} \right\rfloor + 1$. The rates of both controllers are then given by $R_{DS}^{[0,T]} = \frac{1}{T}\Sigma_{DS}^{[0,T]}$ and $R_N^{[0,T]} = \frac{1}{T}\Sigma_N^{[0,T]}$. Since $\delta_N < \delta_S$, it holds that

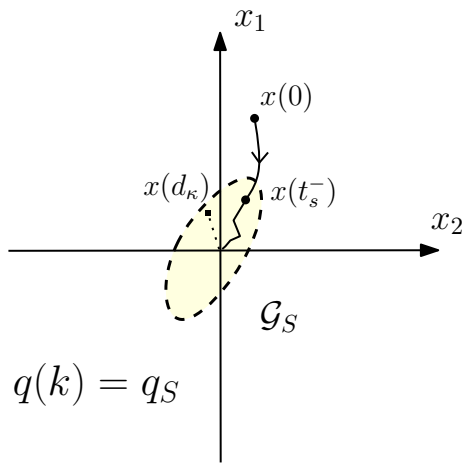
$$\begin{aligned} \lim_{T\to\infty} R_N^{[0,T]} - R_{DS}^{[0,T]} &= \lim_{T\to\infty} \frac{1}{T}\left(\Sigma_N^{[0,T]} - \Sigma_{DS}^{[0,T]}\right) \\ &= \frac{1}{\delta_N} - \frac{1}{\delta_S} > 0, \end{aligned}$$

guaranteeing that a lower number of samples are transmitted over the communication network with the down-sampled controller. □

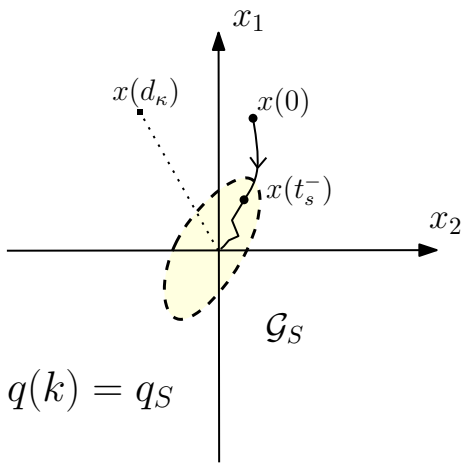
An illustration of the behavior of the down-sampled controller until time t_s is given in Figure 5.3(a). Afterwards, the system state converges to the origin.



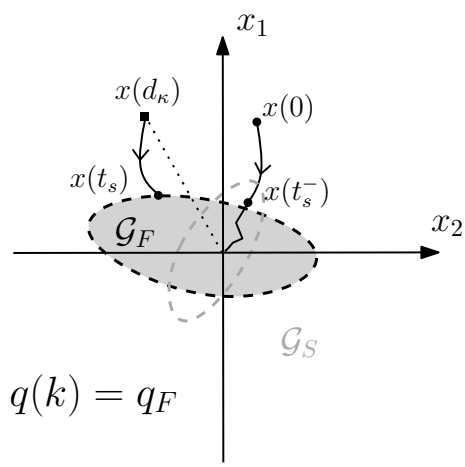
(a) For time $k \in [0, t_s)$



(b) Small disturbance - state remains inside slow region, for time $k \in [t_s^-, d_\kappa]$



(c) Large disturbance, for time $k \in [t_s^-, d_\kappa)$



(d) Large disturbance - trajectory after disturbance, for time $k \in [d_\kappa, t_s)$

Figure 5.3: Illustration of the different behaviors of the down-sampled controller with and without disturbances. The evolution of the system from its initial condition until the first mode transition is depicted in (a). The disturbance in (b) is not large enough to require fast sampling. The case in (c) depicts when the disturbance takes the state outside of the slow region, requiring the usage of fast sampling followed by slow sampling which is depicted in (d).

5.3 Down-sampled controller: disturbance case

We now introduce the design and analysis of the down-sampled controller when disturbances affect the system. The system is affected by sporadic impulsive disturbances $w(k) \in \mathbb{R}^n$, occurring at times $d_\kappa \geq 0,\ \kappa \in \mathbb{N}_0$, perturbing the state as follows

$$x(k+1) = \begin{cases} w(k) & \text{if } k = d_\kappa, \\ \Phi x(k) + \Gamma u(k) & \text{otherwise,} \end{cases}$$

where the time between disturbances $d_{\kappa+1} - d_\kappa \geq \delta_d$ is unknown, but lower bounded by δ_d. Moreover, we assume that the disturbance occurs during the slow sampling interval, i.e., $d_\kappa + \delta_d > t_s$. Such disturbance is used so that one can have a clear comparison to the nominal controller.

For the analysis presented in this section, we only consider the interval of time between disturbances $k \in [d_\kappa, d_{\kappa+1})$, as the disturbances are impulse disturbances which set the value of $x(k+1) = w(k)$ for $k = d_\kappa,\ \forall \kappa$. Thus, by making sure that the down-sampled controller solves Problem 5.1 under each interval $k \in [d_\kappa, d_{\kappa+1}),\ \forall \kappa$ one guarantees that the down-sampled controller is a solution to Problem 5.1 for the interval $k \in [0, \infty)$.

Due to the above assumption on the disturbance interval, one can devise the following bound for the nominal controller cost which is later used in the design of the down-sampled controller.

Lemma 5.1. *Since the disturbance interval is lower bounded by δ_d, there exists an $\epsilon > 0$ such that*

$$J_N^{(d_{\kappa+1},\infty)} \leq \epsilon \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}, \tag{5.18}$$

where $x(d_\kappa)$ is the state at the last disturbance instant and $Y(d_\kappa)$ is given by (B.3). *The value of ϵ depends on the minimum disturbance interval δ_d and is given by*

$$\epsilon = \lambda_{max}\left(V(n_N^-)\right), \tag{5.19}$$

where $V(n_N^-)$ is given by (B.2), *and* $n_N^- = \left(\left\lfloor \frac{\delta_d}{\delta_N} \right\rfloor + 1\right)\delta_N$.

Proof. The proof is given in Section B.1, in the Appendix. □

Next we give the main result of the chapter, which characterizes the transition map (5.4) illustrated in Figure 5.3.

Theorem 5.2. *Consider system* (5.3) *with sampling and actuation governed by* (5.5) *and* $t_s^- = 0$ *at starting time. Let the transition map be defined as*

$$\Pi\left(x(k), q(k-1) = q_F\right) = \begin{cases} q_S & \text{if } x(k) \in \mathcal{G}_F \\ q_F & \text{otherwise} \end{cases} \tag{5.20}$$

$$\Pi\left(x(k), q(k-1) = q_S\right) = \begin{cases} q_F & \text{if } x(k) \in \mathcal{G}_S \wedge k = d_\kappa, \forall \kappa \\ q_S & \text{otherwise} \end{cases} \tag{5.21}$$

where

$$\mathcal{G}_F = \left\{ x(k) \in \mathbb{R}^n \mid x(k)^T \Lambda_F x(k) \leq \sigma_F \right\}, \tag{5.22}$$

$$\mathcal{G}_S = \left\{ x(k) \in \mathbb{R}^n \mid \begin{bmatrix} x(k) \\ 1 \end{bmatrix}^T \Lambda_S \begin{bmatrix} x(k) \\ 1 \end{bmatrix} > \sigma_S \right\},$$

and

$$\begin{aligned} \Lambda_F &= P_S - P_F, \\ \sigma_F &= \begin{bmatrix} x(t_s^-) \\ 1 \end{bmatrix}^T \left(U_N(d_\kappa) - \epsilon Y(d_\kappa) - \begin{bmatrix} P_F & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x(t_s^-) \\ 1 \end{bmatrix} + J_N^{[0,d_\kappa)} - J_{DS}^{[0,d_\kappa)}, \\ \Lambda_S &= U_S(k - t_s^-) - U_N(k) + \epsilon Y(d_\kappa), \\ \sigma_S &= J_N^{[0,k)} - J_{DS}^{[0,k)}, \end{aligned} \tag{5.23}$$

where $Y(d_\kappa)$ *is given by* (B.3), $U_N(l)$ *by* (B.7) *and* $U_S(l)$ *by* (B.11). *Moreover, let the system be initialized with* $q(0) = q_F$. *The switching instant* t_s *is given by:*

$$t_s = \begin{cases} \inf\{k > t_s^- : \Pi\left(x(k), q(k-1) = q_F\right) = q_S\} \\ \inf\{k > t_s^- : \Pi\left(x(k), q(k-1) = q_S\right) = q_F\} \end{cases} \tag{5.24}$$

The control input $u(k)$ *is defined by* (5.6) *with* K_F *and* K_S *given by* (5.10) *for the baseline period* $h = 1$ *and* $\delta_S > h$, *respectively.*

Through this design, the down-sampled controller achieves a normalized cost smaller or equal to that of the nominal controller, i.e., $\bar{J}_{DS} \leq \bar{J}_N$.

Proof. The goal of the down-sampled controller is to achieve $\bar{J}_{DS} \leq \bar{J}_N$ under sporadic impulse disturbances $w(k)$. Notice that in this case, after entering the slow mode, the state may only be brought to a fast mode by a disturbance and thus a re-switching from slow to fast mode must only occur at the time a disturbance occurs, i.e., $k = d_\kappa$.

Consider the transition map (5.20) and (5.21). When the system is in fast sampling mode, its state will be brought to the region $\mathcal{G}_F$ enabling a switch to the slow

sampling mode, as occurred in the no disturbance case analyzed in Section 5.3 (see Figure 5.3(a)). When the system is in the slow sampling mode, a disturbance must be large enough to bring the system state outside of $\mathcal{G}_S$, as depicted in Figures 5.3(c) and 5.3(d). If the disturbance is not large enough, the system will remain in a slow sampling mode (see Figure 5.3(b)).

The region $\mathcal{G}_F$ and transition map (5.20) is derived in the same manner of the switching condition in Theorem 5.1. Rewriting (5.16) for the interval $[d_\kappa, d_{\kappa+1})$ we have that

$$J_{DS}^{[d_\kappa, d_{\kappa+1})} = J_F^{[d_\kappa,\infty)} - J_F^{[t_s,\infty)} + J_S^{[t_s,\infty)} - J_S^{(d_{\kappa+1},\infty)}. \tag{5.25}$$

Given Lemma 5.1, the fact that $J_S^{(d_{\kappa+1},\infty)} \geq 0$ and (5.25), we can derive the following inequalities:

$$\begin{aligned} J_{DS}^{[d_\kappa, d_{\kappa+1})} &\leq J_F^{[d_\kappa,\infty)} - J_F^{[t_s,\infty)} + J_S^{[t_s,\infty)}, \\ J_N^{[d_\kappa, d_{\kappa+1})} &\geq J_N^{[d_\kappa,\infty)} - \epsilon \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}. \end{aligned}$$

Therefore, to guarantee that $J_{DS}^{[d_\kappa, d_{\kappa+1})} \leq J_N^{[d_\kappa, d_{\kappa+1})}$, it is sufficient to require that

$$J_F^{[d_\kappa,\infty)} - J_F^{[t_s,\infty)} + J_S^{[t_s,\infty)} \leq J_N^{[d_\kappa,\infty)} - \epsilon \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}.$$

In fact, as the control cost of the system until time $t = d_\kappa$ is being logged by the sampler, one may introduce these logged costs in the switching condition. This is performed in order to improve the tightness of the down-sampled controller to the nominal controller, since we remove the "cost slack" ($J_N^{[0,d_\kappa)} - J_{DS}^{[0,d_\kappa)}$) that has been carried from previous intervals due to, e.g., numerical precision errors when evaluating the switching condition and/or from the cost gap between the down-sampled and nominal controllers at the time a previous disturbance occurred. Given this, the down-sampled controller is guaranteed to be no worst than the nominal controller if

$$J_{DS}^{[0,d_\kappa)} + J_F^{[d_\kappa,\infty)} - J_F^{[t_s,\infty)} + J_S^{[t_s,\infty)} \leq J_N^{[0,d_\kappa)} + J_N^{[d_\kappa,\infty)} - \epsilon \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix},$$

where the cost-to-go $J_N^{[d_\kappa,\infty)}$ is given by (B.7) and derived in the Appendix B.2. This inequality is enforced by $\mathcal{G}_F$ in the transition map (5.22) with parameters Λ_F and σ_F given in (5.23).

The transition map for the slow sampling mode is based on the requirement that the slow rate should only keep being used from the time the disturbance occurs at time $k = d_\kappa$ onwards, if and only if, the total cost (current cost until time k plus the cost to-go until the next disturbance time $d_{\kappa+1}$) is kept lower than that of the

nominal controller. The above condition can be formulated as

$$J_{DS}^{[0,d_\kappa)} + J_S^{[d_\kappa,d_{\kappa+1})} \leq J_N^{[0,d_\kappa)} + J_N^{[d_\kappa,d_{\kappa+1})},\ \forall\kappa, \tag{5.26}$$

where we have introduced the cost history $J_{DS}^{[0,d_\kappa)}$ and $J_N^{[0,d_\kappa)}$ for both controllers thus removing any "cost slack" from the previous interval. Using bound (5.18) from Lemma 5.1, inequality (5.26) can be guaranteed if

$$J_{DS}^{[0,k)} + J_S^{[k,\infty)} \leq J_N^{[0,k)} + J_N^{[k,\infty)} - \epsilon \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix},\ k = d_\kappa,$$

where $J_S^{[k,\infty)}$ is given by (B.11) and $J_N^{[d_\kappa,\infty)}$ given by (B.7), and both are derived in the Appendix B.4 and B.2, respectively. Enforcing this inequality in the form of the region $\mathcal{G}_S$, we arrive to the transition map in (5.21) with parameters Λ_S and σ_S given by (5.23).

The switching instant t_s in (5.24) follows directly from the definitions of the transition map in (5.20) and (5.21).

The switching condition guarantees by design that the normalized cost of the down-sampled controller is bounded by the normalized nominal cost at each disturbance interval. Due to this fact, asymptotic converge to the origin is only guarantee during the time between disturbances. This completes the proof. □

The down-sampled controller designed in Theorem 5.2 does not give any guarantees w.r.t. the rate of transmission since it is solely designed to guarantee that the performance of the down-sampled controller is no worse than the performance of the nominal controller at each disturbance interval. We now analyze this issue by providing a minimum allowed disturbance interval which guarantees the fulfillment of improvement w.r.t. the transmission rate.

5.3.1 Minimum allowed disturbance interval $\delta_d^{\min}$

We now provide a minimum allowed disturbance interval $\delta_d^{\min}$ so that for $d_{\kappa+1} - d_\kappa \geq \delta_d^{\min}$ the rate of transmission of the down-sampled controller is lower than the nominal controller, i.e., $\bar{R}_{DS} < \bar{R}_N$. Consequently, the down-sampled controller proposed in Theorem 5.2 is a solution to Problem 5.1 under sporadic impulse disturbances. Note that the larger the disturbance interval is, the smaller the transmission rate of the down-sampled controller will be.

In order to perform this analysis, we first require the knowledge of the largest value (worst-case) of the switching time t_s for any initial condition of the state. Such value provides the case where the largest amount of fast samples/transmissions are made by the down-sampled controller.

Notice that the worst-case switching time with the down-sampled controller should be achieved for the best-case situation for the nominal controller under disturbances. This case is when the disturbance occurs synchronously with the

nominal sampling instant, i.e., the nominal controller is able to apply a control input to reject the disturbance immediately when that disturbance occurs. We show this in the following lemma.

Lemma 5.2. *Let a disturbance occur at time $k = d_\kappa$ and consider the nominal controller. Let $n_N^+ = \left(\left\lfloor \frac{d_\kappa}{\delta_N} \right\rfloor + 1 \right) \delta_N$ denote the nominal sampling instant closest (by above) to the disturbance occurrence time $k = d_\kappa$. Moreover, the cost-to-go $J_N^{[d_\kappa,\infty)}$ evaluated at time $k = d_\kappa$, be given by* (B.5)*. Then, it holds that the cost-to-go $J_N^{[d_\kappa,\infty)}$ has its lowest value given by*

$$J_N^{[d_\kappa,\infty)} = J_N^{[n_N^+,\infty)} = x(n_N^+)^T P_N x(n_N^+),$$

when $\mathrm{rem}(d_\kappa, \delta_N) = 0$ *since in that case,* $d_\kappa = n_N^+$.

Proof. The proof is given in the Appendix in B.3. □

Given the above lemma, we can now evaluate the worst-case switching time t_s through the following result.

Lemma 5.3. *Let the switching instant be given by* (5.24)*, $q(0) = q_F$, the baseline period h, and $\delta_S > \delta_N > h$ and $\delta_N > h$ be fixed. The down-sampled controller has the maximum switching instant $\bar{t}_s$ for a given system* (5.3)*, for any initial condition $x(0)$, defined by:*

$$\bar{t}_s = \inf \left\{ k > 0 : \lambda_{max} \left(\bar{\Phi}^{k^T} (P_S - P_F) \bar{\Phi}^k + P_F - P_N + \epsilon \right) \leq 0 \right\}, \tag{5.27}$$

where $\bar{\Phi} = \Phi + \Gamma K_F$ is the closed-loop system matrix for the baseline fast period h.

Proof. The proof follows directly from the switching rule (5.24), evaluated when the disturbance instant $k = d_\kappa$ occurs synchronously with the nominal sampling, due to Lemma 5.2. Then, we apply the fact that $x^T Z x \leq 0, \forall x \in \mathbb{R}^n$, if and only if, $\lambda_{max}(Z) \leq 0$ for any symmetric matrix $Z = Z^T$ (see (Horn and Johnson, 2012)). □

We now can finally characterize the minimum allowed disturbance interval.

Proposition 5.1. *Let $\delta_S > \delta_N > h$ and $\delta_N > h$ be fixed. The minimum allowed disturbance period $\delta_d^{\min}$ which guarantees a solution to Problem 5.1 for the down-sampled controller designed in Theorem 5.2, is given by the solution to the following problem:*

$$\begin{aligned} &\min_{\delta_d^{\min}} && \delta_d^{\min} \\ &s.t. \quad \bar{t}_s + \left\lfloor \frac{\delta_d^{\min} - \bar{t}_s}{\delta_S} \right\rfloor + 1 < \left\lfloor \frac{\delta_d^{\min}}{\delta_N} \right\rfloor \end{aligned} \tag{5.28}$$

where $\bar{t}_s$ is given by (5.27) *in Lemma 5.3, with ϵ defined in* (5.19)*, for $\delta_d = \delta_d^{\min}$.*

Proof. The constraint comes directly from the requirement that during the interval of length $\delta_d^{\min}$,

$$\Sigma_{DS}^{\delta_d^{\min}} < \Sigma_N^{\delta_d^{\min}} \Rightarrow R_{DS}^{\delta_d^{\min}} < R_N^{\delta_d^{\min}},$$

where

$$\Sigma_{DS}^{\delta_d^{\min}} = \left(\bar{t}_s + \left\lfloor \frac{\delta_d^{\min} - \bar{t}_s}{\delta_S} \right\rfloor + 1 \right) + 1,$$

and

$$\Sigma_N^{\delta_d^{\min}} = \left\lfloor \frac{\delta_d^{\min}}{\delta_N} \right\rfloor + 1.$$

Note that an extra transmission is added to the down-sampled controller since due to a disturbance, this controller switches to the fast period and performs a new actuation. On the other hand, the number of transmissions of the nominal controller is kept the same since this controller does not alter its behavior when a disturbance affects the system. □

We note that the computation of (5.28) is performed offline.

5.4 Application to linear stochastic systems

The results presented above can be directly applied when the system dynamics are affected by Gaussian noise (see (Åström and Wittenmark, 1990, Chapter 11)) and the LQ controller is designed for a discounted quadratic cost (Bertsekas, 1995), as also proposed in (Antunes and Heemels, 2014) and (Gommans et al., 2014). The reason for the requirement that a discounted quadratic cost is used, is that the stationary cost, where the noise covariance is the large contributor, must not be the main component of the control cost, as it occurs in the classic undiscounted case (see (Åström and Wittenmark, 1990, Chapter 11)). Instead, the transient cost must be an important element. The importance factor can be regulated through the discount parameter of the cost function (Bertsekas, 1995). This is due to the fact that when no discount/forgetting factor is introduced, there is no incentive for switching from the fast rate to the slow rate since the slow rate's noise covariance in stationarity is very large when compared to the noise covariance of the nominal rate controller. Nevertheless, we note that in practice the discount framework may be desirable as the system may be in transient conditions often, due to frequent large disturbances and/or reference changes. Therefore, the transient behaviour/-cost should be prioritized.

5.5 Application to tracking of first-order systems with time-delay

In this section, we analyze the required modifications to the down-sampled controller to be applied to tracking of piecewise constant reference disturbances and rejection of piecewise constant disturbances. The modifications are performed considering a first-order system with time-delay model (also known as FOTD, FOPTD or KLT model) so that plant delays are considered. The application of control techniques to such system models is desirable since high-order models of industrial processes can be well represented by this low order model, as discussed in (Åström and Hägglund, 2006). In fact, this model has been used to devise many tuning rules for PID controllers in industry and has been the focus of much of the event-triggered control literature applied to PI/PID control.

The system input-output model dynamics can be expressed by the following transfer function

$$G(s) = \frac{K_P}{T_P s + 1} e^{-L_P s},$$

where $K_P > 0$ is the gain, $T_P > 0$ is the time constant and $L_P \geq 0$ the time-delay of the process.

In order to perform reference tracking of a piecewise constant value $r(t)$ and reject a load disturbance $w(t)$ one may introduce an integral component in the controller which allows the controller to behave as a typical PI controller. We note that a feedforward technique as utilized in Chapter 4 may also be used in this case. This is performed by including the integral state

$$\dot{x}_c = r(t) - x(t)$$

as part of a new augmented system dynamics $z(t) = \begin{bmatrix} x(t)^T & x_c(t)^T \end{bmatrix}^T$. For the application of the down-sampled controller, the transfer function FOTD model is sampled at a given period h and converted into state space form as

$$\begin{bmatrix} z(k+1) \\ u(k-L_P) \\ \vdots \\ u(k-1) \\ u(k) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & \dots & 0 \\ 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} z(k) \\ u(k-L_P+1) \\ \vdots \\ u(k-2) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} D \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} r(k) \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} w(k),$$

where $D = \begin{bmatrix} 0 & h \end{bmatrix}^T$ and we assume for simplicity that the delay L_P is a multiple of the sampling period h (see (Åström and Wittenmark, 1990, Chapter 2) for details). A disturbance observer is implemented at the sensor, which estimates the disturbance magnitude in the same manner as performed by (Lunze and Lehmann, 2010)

and (Tiberi et al., 2012b) in the context of model-based event-triggered control and self-triggered control, respectively.

Let us introduce $\xi = \begin{bmatrix} z(k+1) & u(k-L_P) & \dots & u(k-1) & u(k) \end{bmatrix}$, which allows us to write the system in the form

$$\xi(k+1) = \Phi_\xi \xi(k) + \Gamma_\xi u(k) + D_\xi r(k) + E_\xi w(k). \tag{5.29}$$

For the derivation and application of the down-sampled controller, we consider the auxiliary system of (5.29) with state $\tilde{\xi} = \xi - \xi_{eq}$, where ξ_{eq} is the value of ξ at equilibrium for constant $r(k)$ and $w(k)$, and is derived in Section B.5 in the Appendix. Hence, we may rewrite (5.29) as

$$\tilde{\xi}(k+1) = \Phi_\xi \tilde{\xi}(k) + \Gamma_\xi \tilde{u}(k), \tag{5.30}$$

which is in the form of (5.3). This a typical methodology adopted when designing and analysing LQ controllers with piecewise constant references and load disturbances (Dorato et al., 1994). Additionally, a bumpless transfer method (Åström and Wittenmark, 1990) must be utilized to adjust the integral state when switching between controller gains and sampling modes. We note that this procedure applies for a general linear time-invariant system (5.29), and not just for the FOTD model. This procedure will be used in Section 5.8 for the design of the down-sampled controller for piecewise constant reference tracking.

As the controller design procedure, note that since the time-delay of the system is L_P (known from the identified FOTD model), one is able to apply an observer to estimate the value of the state in L_P steps, and use this value to compute the control input. This is analogous to the commonly used Smith Predictor (Åström and Hägglund, 2006). Hence, the control input is given by $\tilde{u}(k) = K\tilde{z}(k+L_P) = K_\delta F_s \tilde{\xi}(k+L_P)$, where K_δ is the controller gain obtained by the LQ design procedure in Section 5.1.4 and $F_s = \begin{bmatrix} I & 0 & \dots & 0 & 0 \end{bmatrix}$ is a selection matrix. A closed-form solution can then be obtained for $\tilde{\xi}(k+l)$ by iterating the system dynamics (5.30), which give $\tilde{\xi}(k+l) = \Upsilon_\delta(l)\tilde{\xi}(k)$, where $\Upsilon_\delta(l)$ is given by (B.12) and is derived in Section B.6 in the Appendix.

The down-sampled controller developed in Section 5.3 can then be readily applied to system (5.30) taking into account the delay L_P, for the following modified parameters: the LQ design is performed and evaluated for matrices $\tilde{Q}_\delta = F_s^T Q_\delta F_s$, $\tilde{R}_\delta = G_S^T R_\delta G_S$ and $\tilde{N}_\delta = F_s^T N_\delta G_S$ in Section 5.1.4, for selection matrix $G_S = \begin{bmatrix} 0 & I & \dots & 0 & 0 \end{bmatrix}$t; and the controller gains K_F, K_N or K_S are substituted by their predicted counterparts $\tilde{K}_F = K_F F_s \Upsilon_F(l)$, $\tilde{K}_N = K_N F_s \Upsilon_N(l)$ and $\tilde{K}_S = K_S F_s \Upsilon_S(l)$.

5.6 Application to a multiple control loop scenario

We now consider the application of the down-sampled controller when several control loops share the same wireless network.

In the previous discussion, we assume that the sensor and controller are collocated and the actuation signal is transmitted over the network. However, the case with separate sensor, controller and actuator can be implemented as follows. The sensor node implements the sampling-rate selection mechanism and transmits the state $x(k)$ to the controller whenever sampling is performed. Together with the state, the sensor transmits a flag which states the current sampling mode. Then, the controller knows which control gain to apply. In the following discussion, we assume that both sensor-to-controller and controller-to-actuator messages are transmitted over the network.

Let the system architecture be the one considered in Chapter 4, where a centralized network manager (NM) schedules the sensor/controller transmissions in GTSs and that the modified IEEE 802.15.4 MAC standard is utilized. In this manner, the same design scheduling procedure of the predictive implementation can be utilized for the down-sampled controller. Recall the superframe structure of the IEEE 802.15.4 MAC. The following procedures are defined for the communication protocol and the down-sampled controller setup:

- For a dynamic GTS allocation, the beacon interval (B.I.) length must be smaller than the slow period $\delta_S h$, so that a request for a slow transmission, after the switching from fast to slow occurs, may be requested to the NM by the sensor. If a static GTS allocation is considered, then all GTSs in a superframe are pre-allocated to all nodes for their fast mode period (worst-case scenario). Such design however has the drawback of reduced efficiency on the utilization of the network bandwidth, as was the case for the event-based communication scheme, discussed in Chapter 4.

- The CAP and inactive periods are disabled in the superframe;

- Each message pair (sensor-to-controller and controller-to-actuator) is scheduled consecutively. Then, the fast mode rate must be adjusted as a function of the minimum slot duration slotDuration, which in turn is dependent on the size of B.I. as $\text{slotDuration} = \frac{\Delta\text{CFP}}{\text{aNumSuperframeSlots}}$. This is due to the fact that the B.I. length is equal to ΔCFP and $\text{aNumSuperframeSlots} = 32$ in our modified IEEE 802.15.4 MAC implementation. Supposing that the dynamic or static allocation of GTSs to control loops is performed in a round-robin fashion, the minimum fast mode period is given by $h_{\min} = 2n_L \times \text{slotDuration}$, where n_L be the number of control loops to be scheduled;

- As the superframe is only composed of GTSs, the GTS request is sent together with the data message transmitted by the sensors to the NM. Thus, in each superframe, a single round of sensors are allocated GTSs so that they can perform such requests.

The single concern when applying a dynamic scheduling to the down-sampled controller in the multiple control loop scenario is the response-time to disturbances, which is the same issue as with the predictive implementation in Chapter 4. When

a disturbance occurs, the sensor must request fast mode GTSs for itself and the respective controller to the network manager for the next superframe. Therefore, in the worst-case, there will be a maximum request/allocation delay of one B.I.. This issue however will not occur when the GTS allocation is made static and provisioning the resources assuming that a fast mode is always used.

5.7 Numerical examples

We now provide an evaluation of the proposed down-sampled controller in Section 5.2 and 5.3. We start by analysing the performance of the down-sampled controller when compared to other two methods proposed in the literature (Antunes and Heemels, 2014; Gommans et al., 2014) for the case when no disturbances affect the system, as in these works no disturbances are considered. Afterwards, we analyze the performance of the down-sampled controller under sporadic impulse disturbances. The evaluation is performed for two different plants, a two-mass and spring system from (Antunes and Heemels, 2014), and a classic double-integrator system (Åström and Wittenmark, 1990).

Two-mass and spring

The two-mass and spring system from (Antunes and Heemels, 2014) is modelled as a 4^{th}-order continuous-time system (5.1) with parameters

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2\pi^2 & 2\pi^2 & 0 & 0 \\ 2\pi^2 & -2\pi^2 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

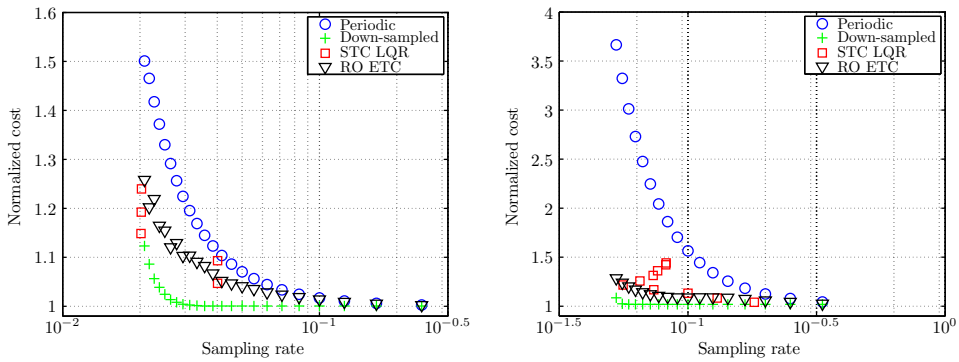
and with initial condition $x(0) = \begin{bmatrix} -1 & 1 & 0 & 0 \end{bmatrix}^T$ and control cost matrices $Q_c = \text{diag}(1, 1, 0, 0)$ and $R_c = 0.1$.

Double integrator

The double integrator (Åström and Wittenmark, 1990) is modelled as a $2^{\text{nd}}-$order continuous-time system (5.1) with parameters

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

and control cost matrices $Q_c = \text{diag}(1, 1)$ and $R_c = 1$.



(a) Two-mass and spring

(b) Double integrator

Figure 5.4: Comparison among different aperiodic control algorithms and the traditional periodic controller in the absence of disturbances.

5.7.1 No disturbance case

Under this formulation, we compare the down-sampled controller to the traditional periodic controller, the roll-out event-triggered controller (RO-ETC) proposed in (Antunes and Heemels, 2014) and the self-triggered linear quadratic regulator (STC LQR) proposed in (Gommans et al., 2014)

Consider the two-mass and spring system. Figure 5.4(a) presents the results of the control performance of the two-mass and spring system for different average sampling periods, under a traditional periodic controller, the down-sampled controller, the STC LQR (Gommans et al., 2014) and the RO ETC (using the same algorithm parameters as (Antunes and Heemels, 2014)). The control performance for all cases is normalized on the continuous-time control performance which is the lowest possible cost under the given Q_c, R_c and $x(0)$. The displayed down-sampled controller cost is the minimum achievable cost for a nominal period $\delta_N \in]h, \delta]$ and the corresponding δ_S was chosen as the minimum period ensuring an average period of δ over the horizon T. In this case, we set $T = 1150\,\text{s}$ and varied $\delta \in [0.02, 0.48]\,\text{s}$. As for the STC LQR, we varied the algorithm's tuning parameter $\beta \in \{1.05, 1.1, ..., 1.25\}$.

As it can be seen, the cost achieved by the down-sampled controller is lower than the other methods.

Consider now the double integrator system. Figure 5.4(b) presents the results of the control performance for this system under the different average sampling periods and controller implementations. In this case, the control cost is averaged over 20 different initial conditions $x(0)$ equally spaced in the unit disk. Moreover, $T = 350\,\text{s}$, $\delta \in [0.25, 5]\,\text{s}$ and baseline period $h = 0.25\,\text{s}$. As for the STC LQR, we varied again the algorithm's tuning parameter $\beta \in \{1.05, 1.1, ..., 1.5\}$. As in

the previous case, the down-sampled controller has a cost smaller than the other algorithms.

In summary, the down-sampled controller has an advantage over the STC LQR in the fact that, not only a lower control cost is achieved for the same average sampling period, but we can guarantee a specific cost for a selected average sampling period. This is not achieved in the STC LQR since for a specific β value there is no guarantee what the sampling period will be. As expected, the cost difference to the RO ETC is not as large as to the STC LQR since the RO ETC method is based on a roll-out strategy which is known for being efficient on solving combinatorial optimization problems. Nevertheless, the down-sampled controller requires a very low computational effort and is based on simple switching rules, as opposed to the computationally demanding roll-out method proposed in (Antunes and Heemels, 2014).

5.7.2 Disturbance case

The performance of the down-sampled controller under sporadic impulse disturbances is now analyzed on the double integrator system with baseline period $h = 0.1\,\mathrm{s}$, $\delta_N = 10$ and $\delta_S = 50$, during a $T = 140\,\mathrm{s}$ simulation interval. The initial condition is set to $x(0) = \begin{bmatrix} 10 & 10 \end{bmatrix}^T$ and the disturbance occurs three times at $\{d_1, d_2, d_3\} = \{20, 60, 100\}\,\mathrm{s}$ with values:

$$\{w(d_1), w(d_2), w(d_3)\} = \left\{ \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}, \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}, \begin{bmatrix} -6.05 \\ -7.85 \end{bmatrix} \right\}.$$

Using Proposition 5.1 we obtain that the minimum disturbance interval to guarantee that the down-sampled controller utilizes less transmissions than the nominal controller is $\delta_d^{\min} = 500 = 50\,\mathrm{s}$.

Figure 5.5 depicts the time-response of the state and control input of the system with the down-sampled controller and the nominal controller, as well as the sampling instants performed by both controllers. The normalized control cost of both controllers for the same experiment w.r.t. the continuous-time control cost without disturbances is also depicted in this figure. We observe that for both controllers the closed-loop system is stable and that the down-sampled controller cost is always lower or equal than the nominal controller. Moreover, the total number of transmissions during the experiment was $\Sigma_{DS} = 68$ and $\Sigma_N = 141$. As expected, the obtained $\delta_d^{\min}$ is conservative as it is computed for the worst-case scenario, as defined in Section 5.3. We notice that 9 fast sample and actuation instants occur after the experiment is initialized as well as at the moment of the first disturbance. No fast sample is required when the second disturbance occurs. This is due to the fact that the disturbance value is small and the down-sampled controller cost is guaranteed to be below the nominal controller cost by the current slow sampling

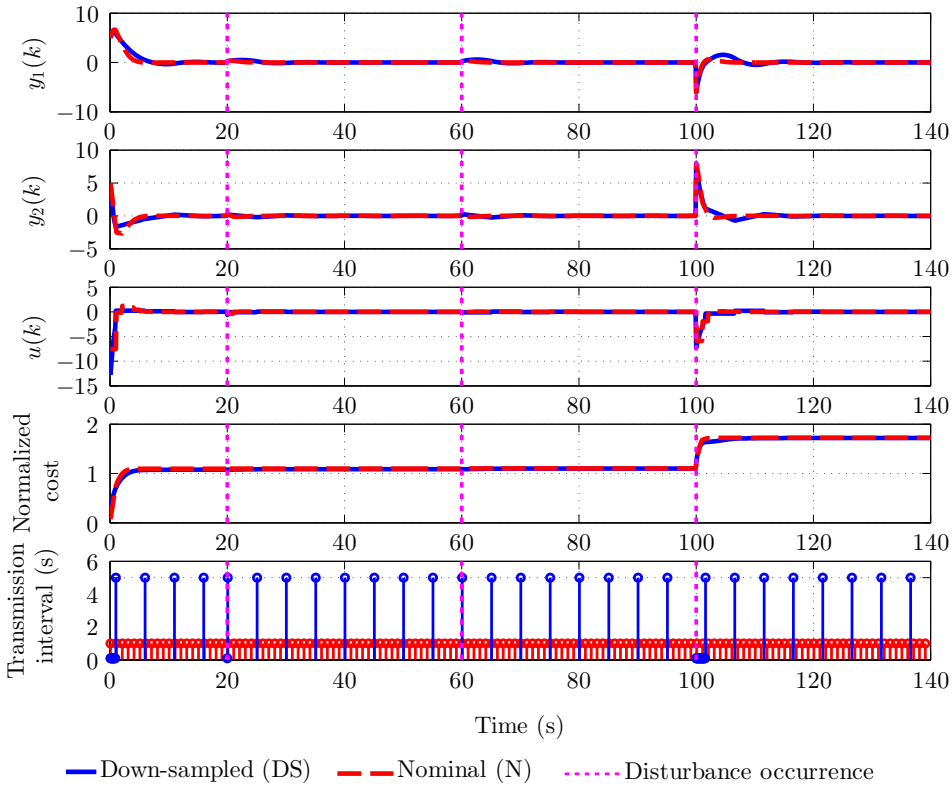


Figure 5.5: Time-response of the state, control input and control cost $J_{DR}^{[0,140]}$ for the down-sampled controller (solid line) and the nominal controller (dashed line) under non-zero disturbances. Transmission intervals shown for both controller strategies, where a circle represents the transmission instant and its height the time since last transmission. Dotted vertical lines denote the instant where a disturbance occurs.

action. When the last and large disturbance affects the system, 22 fast sample and actuation instants are required.

5.8 Experimental setup

We now provide an experimental evaluation of the down-sampled controller when applied to a wireless control scenario with two control loops. The setup is similar to the one used in Chapter 4 where a wireless network is shared by two double tank systems introduced in Chapter 2. This central controller node is also the network manager (NM) which schedules the network transmissions in GTSs. The experimental setup is depicted in Figure 5.6 and was implemented using the GISOO

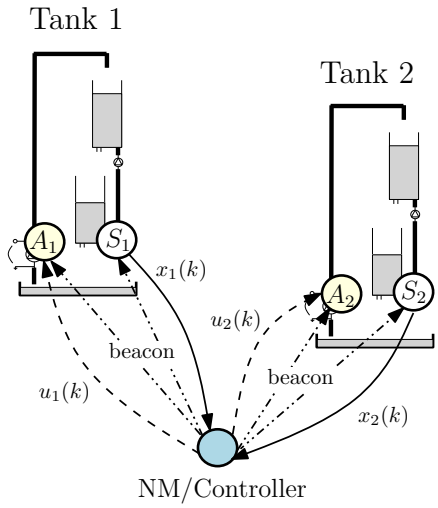


Figure 5.6: Experimental setup comprising of two double tank systems (tank 1 and tank 2) controlled over wireless. The sensor nodes transmit data to the NM/Controller (solid line), which then computes the actuation inputs and transmits these values to each actuator (dashed). At each superframe, the NM transmits a beacon message (dashed-dotted) which maintains all nodes synchronized and updates their GTS schedule.

co-simulator introduced in Chapter 8.

5.8.1 Control system

The goal of the experiment is to control the water level of the lower tank by adjusting the motor voltage accordingly. The tank model is the second-order model derived in Chapter 2.8. In order to perform this task, we utilize a state-feedback controller with an integral state, as discussed in Section 5.5. The water levels at the lower tank in both tank 1 and tank 2 are initialized to 8 cm and 5 cm, respectively. The reference signal for tank 1 is initially set to 10 cm and to 8 cm in tank 2. At time $t = 85$ s, the reference of tank 1 is changed to 8 cm and at $t = 95$ s the reference of tank 2 is changed to 10 cm.

The down-sampled controller is configured with baseline period $h = 0.2$ s and a slow period of 5 s, which corresponds to $\delta_S = 25$. Additionally, the nominal controller to be outperformed is set with a nominal period of 1 s, which corresponds to $\delta_N = 5$. In this experiment we will also compare the down-sample controller to a slow periodic controller with a fixed period of 5 s.

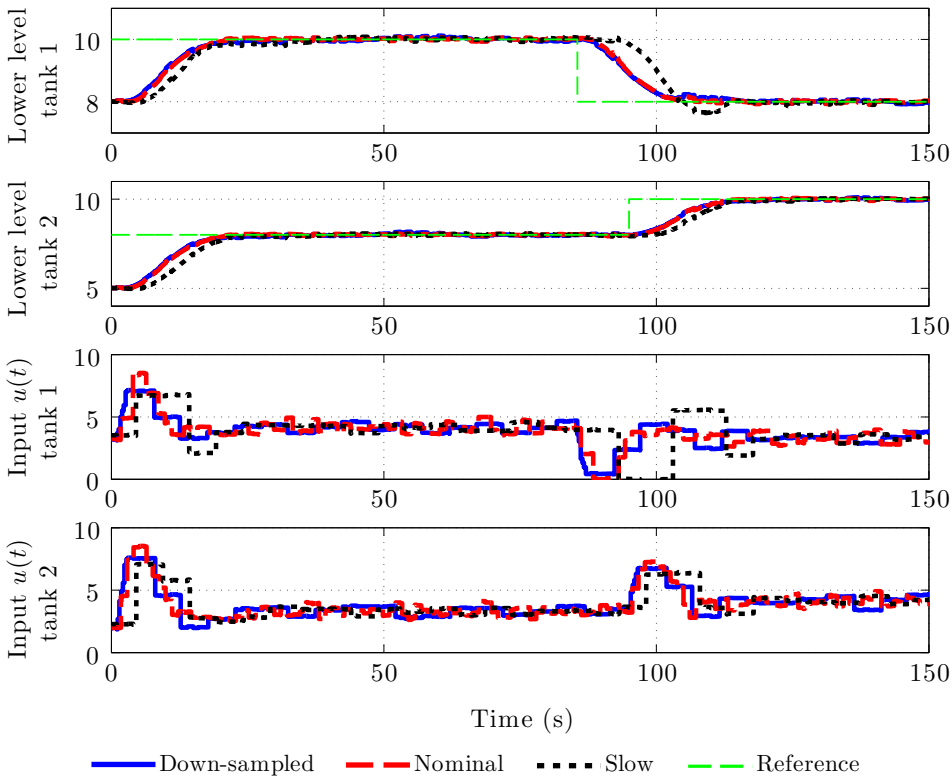


Figure 5.7: Lower tank levels for tank 1 and tank 2 and the control input values applied to the tank. The reference value for each tank system is depicted by the dashed green line.

5.8.2 Wireless network

The communication protocol is the modified IEEE 802.15.4 MAC introduced in Chapter 2 with 32 GTSs and no inactive period. The B.I. is set to 939 ms which gives a slot size of approximately 30 ms. For simplicity, we implemented the static GTS scheduling, where all GTSs are pre-allocated in a round-robin fashion to the two control loops, as was discussed in Section 5.6.

5.9 Experimental results

We now evaluate the performance of the down-sampled controller in comparison to the nominal and a slow controller. The experimental results obtained are depicted in Figures 5.7 and 5.8. Figure 5.7 shows the lower level of both tank 1 and tank 2,

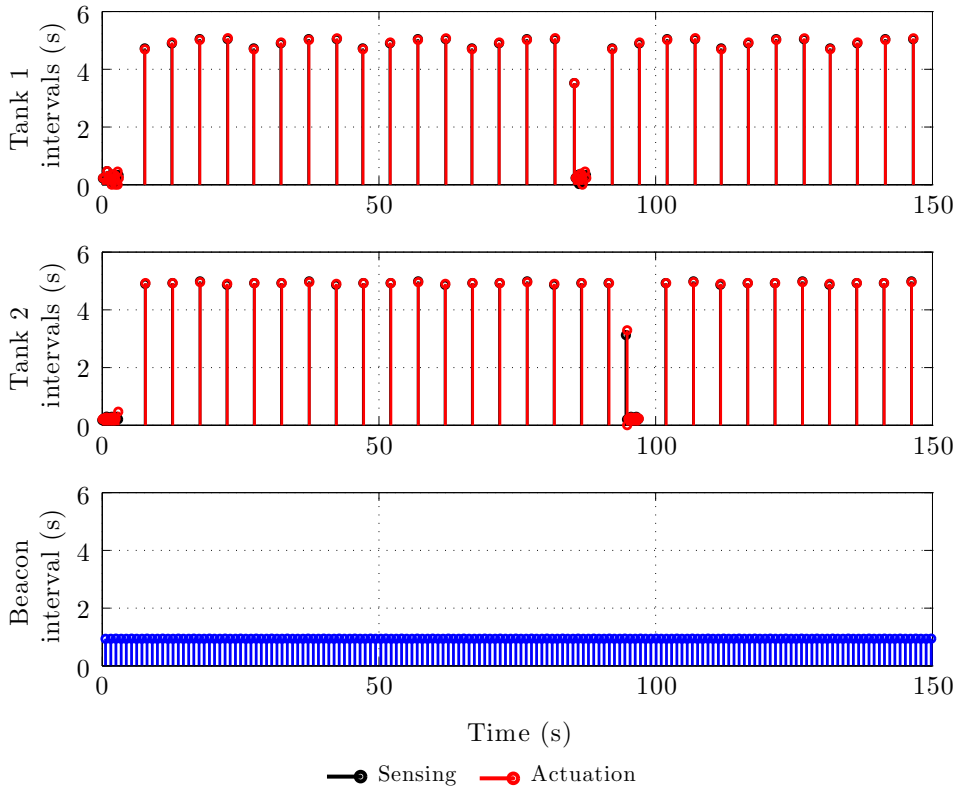


Figure 5.8: Wireless transmissions performed at each tank system, by the sensor (sensing transmission) and NM/Controller (actuation transmissions and beacon transmissions) for the down-sampled implementation. The figure depicts the time interval between two consecutive message transmissions.

as well as the control inputs applied in each of the tanks. The sensing and actuation communication instants and the time interval between consecutive transmissions for both tanks, together with the beacon transmissions are presented in Figure 5.8. In this case, we show only the wireless transmissions performed by the down-sampled controller. As it can be seen, the tracking of the reference signals is well performed in both tanks, where the down-sampled controller actuates at a fast rate in the beginning of the transient, switching to slow afterwards. A total of 13 fast samples are taken by each sensor at tank 1 and tank 2 during the first tracking response, and 8 and 10 fast samples respectively, in the second tracking response. The total number of samples performed by tank 1 was 49 and by tank 2 was 51, throughout the experiment duration. The nominal periodic controller performs 150 samples per tank during the same period, while the slow controller performs 30 per tank. Even

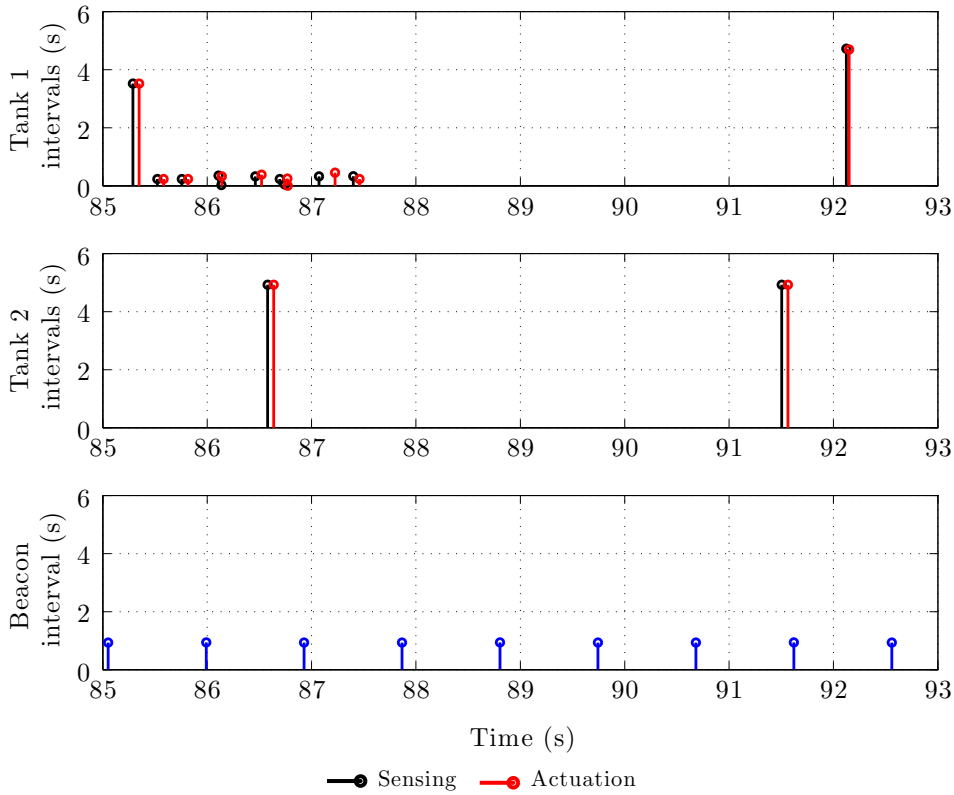


Figure 5.9: Zoom on the wireless communications performed occurring during time $t = [85, 93]$ s for the down-sampled implementation.

though the down-sampled controller uses 20 more samples than the slow controller, the performance obtained by the down-sampled is clearly better. Particularly, the slowly sampled controller is slower to track the reference changes. On the other hand, the down-sampled controller utilizes 30% of the nominal sampling controller, while achieving the same closed-loop performance. It was also confirmed that the quadratic cost of the down-sampled controller is always below the quadratic cost of the nominal controller.

In Figure 5.9, we show the communication which occurs during $t = [85, 93]$ s, which is the interval after a change in reference occurs in tank 1. The interval between fast transmissions in tank 1 is normally around the specified period of $h = 0.2$ s but there are intervals with lower period. This is due to the fact that the implemented sampling timer is not synchronized with the allocated GTS slots and the time required for ADC sampling and computation of the switching condition

is not deterministic. However, the impact of such inaccuracy in the performance of the proposed scheme is negligible.

5.10 Summary

We have introduced an event-based sampling-rate selection method which reduces network usage while guaranteeing a specified quadratic cost under sporadic disturbances. This method is based on fast and slow sampling intervals with the intuition that the closed-system benefits from being brought quickly to steady-state conditions, while behaving satisfactory when being actuated at a slow rate once at those conditions. Such technique can be applied to perform tracking of piecewise constant references and disturbance rejection and, under slight modifications, be used with first-order time-delay systems and linear stochastic systems. Moreover, we show how the down-sampled controller can be implemented in a multiple control loop scenario where several control systems share the same wireless medium. Through numerical simulations and an experimental validation, we demonstrated the benefits of using the down-sampled policy instead of a fixed-rate one. Moreover, we have shown that this simple mechanism provides large savings with respect to the required sampling rate when compared to the traditional periodic controller and other state-of-the-art aperiodic controller mechanisms.

Chapter 6

Compensator for out-of-order communications and time-varying delays

In this chapter, we consider the design of a robust output-feedback controller under delayed and out-of-order sensor measurements in a wireless networked control system (NCS). Such delays may be caused by network congestions, poor connectivity or malicious relay nodes intentionally delaying messages. The controller is designed using the minimax control framework (Başar and Bernhard, 2008). The proposed technique is evaluated on a wireless networked control lab experiment implemented in the co-simulator introduced in Chapter 8. The lab experiment is a double tank system controlled over a multi-hop low power and lossy wireless network. The wireless devices are restricted in processing power, memory and energy and a packet forwarding delay is introduced by a malicious node.

To cope with out-of-order communication and time-varying delays an extension of the minimax controller with a linear temporal coding mechanism is proposed. Each sensor combines its current measurement with previously transmitted measurements and transmits this information in a single packet to the controller. It has been shown in (Robinson and Kumar, 2007), in the context of Kalman filtering under packet drops, that this approach improves estimation performance in scalar systems. Recently in (Suia et al., 2014), the authors have shown that linear temporal coding increases the stability margin in terms of the maximum allowable packet loss rate. In this chapter, the coding is designed to improve the estimation performance of the observer-based controller, which also allows to increase the robustness of the controller to the network imperfections.

The rest of this chapter is organized as follows. Section 6.1 introduces the problem formulation and the controller design framework. The minimax controller design for fixed delay is given in Section 6.2. In Section 6.3, the design of the minimax controller under delayed and out-of-order messages is presented, while the linear temporal coding mechanism is proposed in Section 6.4. Section 6.5 defines the experimental case study comprising of a packet scheduling attack in a wireless multi-hop scenario. Experimental results are presented in Section 6.6. Concluding

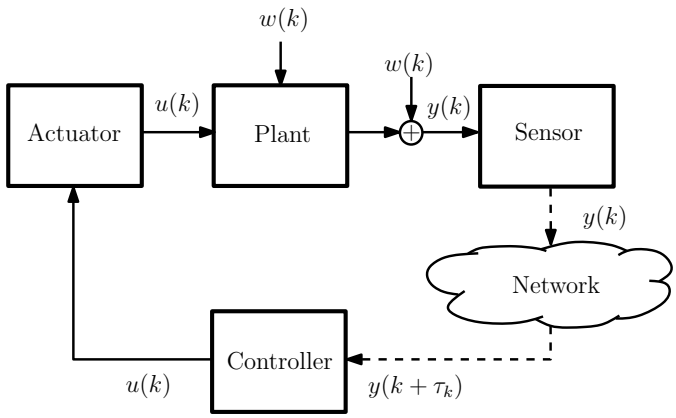


Figure 6.1: Illustration of a wireless NCS where communication between the sensor and controller is performed over a wireless network. A delay τ_k affects each sensor measurement.

remarks are presented in Section 6.7.

6.1 Problem formulation

We consider a discrete-time linear time-invariant system subject to both state and output disturbances:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Dw(k), \\ y(k) &= Cx(k) + Ew(k), \end{aligned} \tag{6.1}$$

where, $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$, $u_k \in \mathbb{R}^m$, and $w_k \in \mathbb{R}^l$ are the system state, output, control input and disturbance input, respectively. Moreover, E is assumed to have full row rank, $N = EE^T \succ 0$ and $DE^T = 0$, as process and measurement disturbances are assumed to be decoupled. The system setup considered is depicted in Figure 6.1. Sensor data is transmitted over a network which induces a delay of $\tau_k \geq 0$ in each packet. The controller computes the control input, transmitting it to the actuator over a wired channel, not affected by any delay. The following standing assumptions are made:

(A1) Packets are timestamped by the sensor.

(A2) The transmission delay τ_k is upper bounded by $\bar{\tau} > 0$ time units, hence no packets are lost.

The delay may be caused by network congestions, poor network connectivity or a malicious relay delaying data delivery. In the presence of long delays, out-of-

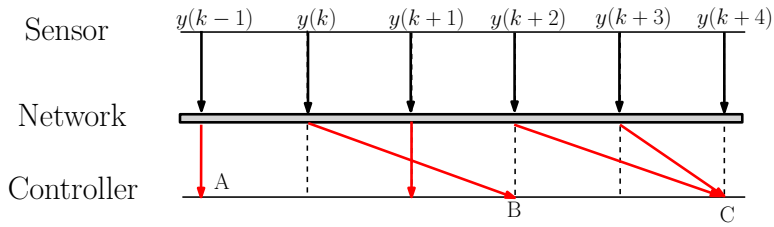


Figure 6.2: An illustration of packet delivery scenarios for sensor to controller transmissions. Time-varying delays may lead to (A) packets delivered within the next transmission instant, (B) Packets delivered out of its order and (C) multiple packets received at the same time instant.

sequence measurement delivery may occur. An illustration of a packet delivery scenario is depicted in Figure 6.2.

We now introduce the information structure. The set of received packets at the controller until time $k \geq 0$ is defined as

$$\Pi_0^k = \left\{ j \in \{0, ..., k\} \mid \text{packet } j \text{ is received at the controller} \right\},$$

and the set of all measurements received at the controller until time k as

$$\mathcal{Y}_0^k = \left\{ y(j) \mid j \in \Pi_0^k \right\}.$$

Then, the set of available information at the controller is

$$\mathcal{I}_0^k = \left\{ \mathcal{Y}_0^k, \mathcal{U}_0^{k-1}, \Pi_0^k \right\}, \tag{6.2}$$

where $\mathcal{U}_0^{k-1}$ represents the sequence $\{u(0), ..., u(k-1)\}$.

To design the robust output-feedback controller for the proposed NCS, we follow the zero-sum dynamic game approach from (Başar and Bernhard, 2008). The control design is viewed as a dynamic game between two players. The controller tries to minimize a given finite horizon quadratic cost, while the disturbance maximizes the same cost. Let $\mathbf{U}$ and $\mathbf{W}$ denote the spaces of control and disturbance policies, respectively. The sequences of control and disturbance policies are denoted as $\mu \in \mathbf{U}$ and $\nu \in \mathbf{W}$, respectively, and are defined by $\mu = \{\mu_0, ..., \mu_{K-1}\}$ and $\nu = \{\nu_0, ..., \nu_{K-1}\}$, over a finite horizon length K. The functions μ_k and ν_k map the information set $\mathcal{I}_0^k$ into the control and disturbance spaces of $\mathbb{R}^m$ and $\mathbb{R}^l$, as $u(k) = \mu_k(\mathcal{I}_k)$ and $w(k) = \nu_k(\mathcal{I}_k)$, respectively. Let us also define $\omega = (x(0), \nu) \in \Omega = \mathbb{R}^n \times \mathbf{W}$. The disturbance is assumed to have the same information knowledge of the controller.

The design problem is that of finding a controller that minimizes the cost function

$$\mathcal{P}_{\mu}^{K} = \sup_{\omega \in \Omega} \frac{\left(J^{K}(\mu,\nu)\right)^{\frac{1}{2}}}{\left(\sigma^{K}\right)^{\frac{1}{2}}}, \tag{6.3}$$

subject to the information structure (6.2), where

$$J^{K}(\mu,\nu) = \|x(K)\|_{Q_K}^2 + \sum_{k=0}^{K-1} \|x(k)\|_{Q}^2 + \|u(k)\|^2,$$

and

$$\sigma^{K} = \|x(0)\|_{Q_0}^2 + \sum_{k=0}^{K-1} \|w(k)\|^2.$$

We use the notation $\|.\|_Q$ to denote the Euclidean norm with positive definite weighting matrices Q, Q_K and Q_0 of appropriate dimension. As shown in (Başar and Bernhard, 2008, Chapter 6), this problem can be solved by utilizing a soft-constrained game approach for a zero-sum game cost function given by

$$J_{\gamma}^{K}(\mu,\nu) = \|x(K)\|_{Q_K}^2 - \gamma^2\|x(0)\|_{Q_0}^2 + \sum_{k=0}^{K-1} \Big(\|x(k)\|_{Q}^2 + \|u(k)\|^2 - \gamma^2\|w(k)\|^2\Big), \tag{6.4}$$

where $\gamma > 0$ is the disturbance attenuation level and $x(0)$ is the unknown initial state of the system. In this chapter, we will find conditions on γ for which the zero-sum game admits a solution. The infimum of all values γ is denoted by $\gamma^{\star}$, for which the corresponding controller is the H^{∞} controller that minimizes (6.3). Additionally, when $\gamma \to \infty$, the minimax controller approaches the solution of the Linear Quadratic Gaussian (LQG) controller (Başar and Bernhard, 2008; Khargonekar, 1991; Moon and Başar, 2014).

As the controller does not have access to full state information, we perform a worst-case minimax estimate under the information structure (6.2). We adopt the *worst-case certainty equivalence principle* (Başar and Bernhard, 2008). Under certainty equivalence, one can split the design problem into two parts: the first is to design an observer which estimates the worst state that matches the sequence of available inputs and outputs; the second is to design a controller which makes use of the estimated state in order to generate the new control input. The worst-case state estimation works as follows: whenever the estimator receives an out-of-order packet, it starts by (i) reordering all the previously received messages, (ii) compute the worst-case disturbance compatible with the available information, (iii) find the corresponding worst-case state estimate, and then (iv) compute the control input as if the actual state is equal to this worst-case estimate.

In the next section, we provide details on the design of the minimax controller solving the posed problem under fixed delays, as introduced in (Başar and Bernhard, 2008). Afterwards, in Section 6.3, we derive the minimax controller under time-varying delay and out-of-order messages.

6.2 Minimax control under fixed delay

We start by reviewing some results proved in (Başar and Bernhard, 2008) upon which we base our control design. Consider a linear time-invariant system subject to disturbance modeled with (6.1), along with the cost function (6.4) and that the time delay is fixed at $\tau = \bar{\tau}$. Thus, at time $k \geq \bar{\tau}$, only information up to time $k-\bar{\tau}$ is available to the controller. In other words, the measurement information set follows $\mathcal{Y}_0^k = \mathcal{Y}_0^{k-\bar{\tau}}$. The minimax controller design is designed under no delay from the initial time until time $k-\bar{\tau}$, while, for the remaining time $[k-\bar{\tau}+1, \ldots, k]$, an estimate is made of the worst-case disturbance, where no observations are available to the controller.

Let us introduce the parameter α_k, taking a value $\alpha_k = 1$ if there is a packet reception at time k, and $\alpha_k = 0$, otherwise. Also, we denote the state-estimate for time $k+1$ with information up to time k as $\hat{x}(k+1|k)$. To improve the clarity of presentation, we denote $\hat{x}(k+1) = \hat{x}(k+1|k)$, whenever the information is available up to the previous time step k. Consider now the minimax controller with $\hat{x}(0) = x_0$, given by

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + A\Lambda(k)\left(\gamma^{-2}Q\hat{x}(k) + \alpha_k C^T N^{-1}\left(y(k) - C\hat{x}(k)\right)\right), \tag{6.5}$$

$$u(k) = -B^T\Gamma(k)A\left(I - \gamma^{-2}\Sigma(k)M(k)\right)^{-1}\hat{x}(k), \tag{6.6}$$

and where

$$\Lambda(k) = \left(\Sigma(k)^{-1} + \alpha_k C^T N^{-1} C - \gamma^{-2}Q\right)^{-1},$$

is the estimator gain, and

$$\Gamma(k) = \left(M(k+1)^{-1} + BB^T - \gamma^{-2}DD^T\right)^{-1}.$$

Additionally, $M(k+1)$ and $\Sigma(k)$ are the solutions to the Game Algebraic Riccati Equations (GAREs), with $M(K) = Q_K$ and $\Sigma(0) = Q_0^{-1}$,

$$M(k) = A^T\left(M(k+1)^{-1} + BB^T - \gamma^{-2}DD^T\right)^{-1}A + Q, \tag{6.7}$$

$$\Sigma(k+1) = A\Lambda(k)A^T + DD^T. \tag{6.8}$$

Moreover, we introduce

$$\widetilde{\Sigma}(k+1) = A(\Sigma(k)^{-1} - \gamma^{-2}Q)^{-1}A^T + DD^T. \tag{6.9}$$

The following result by (Başar and Bernhard, 2008) states the conditions for the existence of the minimax controller.

Proposition 6.1 (Theorem 6.6 (Başar and Bernhard, 2008)). *Consider the dynamic game* (6.4)*, subject to the system dynamics* (6.1)*, the information structure* (6.2) *and a fixed delay* $\tau = \bar{\tau}$*. Let* $\gamma^\star > 0$ *be fixed. The minimax controller exists for all* $\gamma \geq \gamma^\star$*, if all the following conditions hold:*

(a) *GARE* (6.7) *has a solution over* $[0, K]$*;*

(b) *GARE* (6.8)*, with* $\alpha_k = 1$*, has a solution for* $k \in [0, K - \bar{\tau}]$*;*

(c) *GARE* (6.8)*, with* $\alpha_k = 0$*, has a solution for* $k \in [K - \bar{\tau} + 1, K]$*;*

(d) *The solution of* (6.7) *and* (6.8) *satisfies the following conditions*

$$\rho\left(\Sigma(k)Q\right) < \gamma^2, \quad k = 0, ..., K-1, \tag{6.10}$$

and

$$\rho\left(\widetilde{\Sigma}(k+1)M(k+1)\right) < \gamma^2, \quad k = 0, ..., K. \tag{6.11}$$

The minimax controller is then given by

- *the estimator* (6.5)*, with* $\alpha_k = 1$ *for* $k \in [0, K - \bar{\tau}]$*;*
- *the estimator* (6.5)*, with* $\alpha_k = 0$ *for* $k \in [K - \bar{\tau} + 1, K]$*;*
- *the feedback controller* (6.6) *for* $k \in [0, K]$*.*

If any of the conditions ***(a)*** *to* ***(d)*** *fail, then no such controller exists for* $\gamma \geq \gamma^\star$*.*

This proposition shows that, if conditions (6.10) and (6.11) hold at all times, the estimator is able to construct the state-estimate $\hat{x}(k)$ as (6.5) by, using the information received up to time $k - \bar{\tau}$, with $\alpha_k = 1$, for $k \in [0, k - \bar{\tau}]$, and then run a worst-case open-loop estimator for the rest of the period with $\alpha_k = 0$, for $k \in [k - \bar{\tau} + 1, k]$, as no information is available. Note that the separation principle, meaning that the estimator (6.5) and the controller (6.6) can be designed separately, does not hold in the minimax controller design since the estimation and control GAREs (6.7) and (6.9), respectively, are coupled through condition (6.11).

Remark 6.1. *The minimax controller in Proposition 6.1 follows the principle that measurements are used by the estimator at the "correct time", i.e., the time they were transmitted. For the remainder of the time, a worst-case estimate is constructed without any measurement information. Thus, it is possible to apply the same minimax controller to the case of time-varying delay, by utilizing a buffer approach (Luck and Ray, 1990, 1994). In this way, whenever a new measurement is received, it is properly stored in a buffer in its correct order, and presented to the estimator at its transmission time. This approach is conservative as it forces an artificial fixed delay of* $\bar{\tau}$*, even though the delay of each measurement is* $\tau \leq \bar{\tau}$*.*

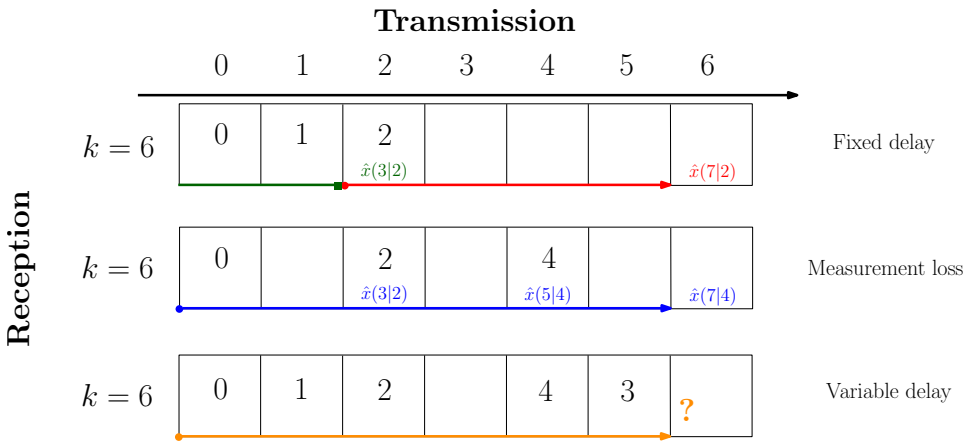


Figure 6.3: Illustration of the behavior of the estimator at current time $k = 6$, under fixed delays and with measurement loss, for the designs provided in (Başar and Bernhard, 2008) and (Moon and Başar, 2014), respectively. The fixed delay value is $\tau = \bar{\tau} = 4$. For the fixed delay scenario, the estimation $\hat{x}(3|2)$ can be correctly performed. The estimation at the current time step must also use only information available up to time $k = 2$. In the loss scenario, the estimation performed a the current step is performed based on all the received information up to time $k = 6$. The final scenario, shows an example of the packet reception under time-varying delay, where an out-of-order packet delivery at time $k = 5$ occurs. We design the compensator to deal with such case. The state estimation at time $k + 1$ with information up to time k (perfect estimation) is represented in green, while the red color represents an estimation with delayed information. The blue color denotes a state estimation with missed information.

6.3 Minimax control under out-of-order packets and time-varying delay

We now provide the first main contribution of this chapter. In the case of time-varying delay and out-of-order messages, the information structure has components from the information structure under fixed delay, but also from a lossy information structure considered by (Moon and Başar, 2014). In (Moon and Başar, 2014), the authors propose a minimax controller solving (6.4), for a lossy information pattern where α_k is stochastic, where actuation input values may be lost. However, no delays are considered. The proposed worst-case certainty equivalent minimax controller and the existence conditions under measurement loss, are the same as the ones from Proposition 6.1 for $\tau_k = 0$, but taking into account the availability (or not) of measurement data at each time step, i.e., the parameter α_k, in the information structure (6.2). To illustrative this, and the proposed opportunistic minimax controller, we provide the following example.

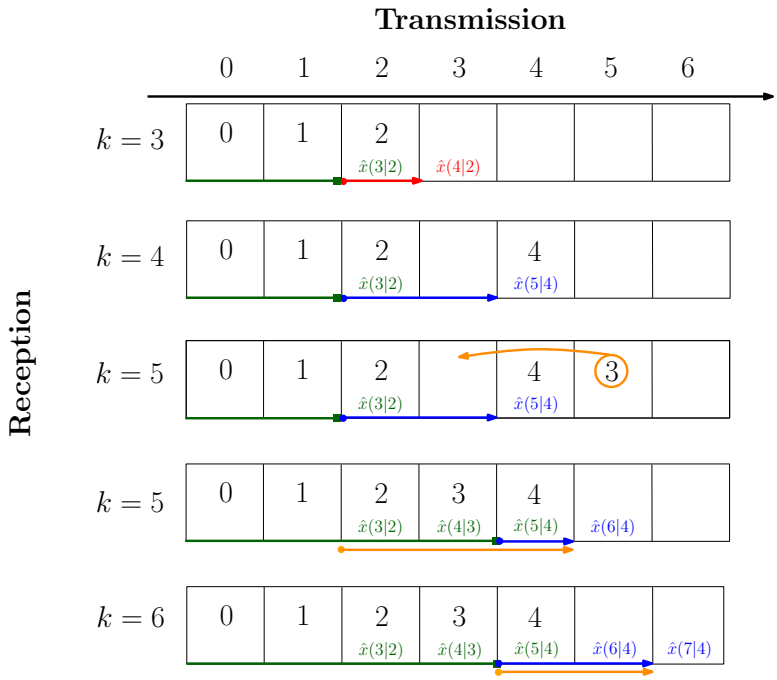


Figure 6.4: Illustration of the behavior of the estimator under varying delay and out-of-order packets, utilizing the proposed estimator, for a specific packet reception sequence. Whenever a packet is received with delay or it is out-of-order, the estimator goes back in time, until the point where all packets have been correctly received. A re-estimation of the state with the new information then follows. If measurement data is missing at a given time step, the estimator performs an open-loop estimation at that point.

Consider the case with fixed-delay where $\tau = \bar{\tau} = 4$, measurement loss and time-varying delay, depicted in Figure 6.3. We analyze their behavior up to time $k = 6$. For the fixed delay case, all packets have arrived up to time $k = 2$, hence the estimate $\hat{x}(2|2)$ is currently available. From that point on, up to time $k = 6$, the minimax controller devised in Proposition 6.1 is utilized with $\alpha_k = 0$, since a perfect estimation $\hat{x}(3|2)$ is available at time $k = 2$. As for the measurement loss case, a packet reception occurs at time $k = \{0, 2, 4\}$. At those times, the minimax controller is used with $\alpha_k = 1$, while for $k = \{1, 3, 4, 6\}$, the value $\alpha_k = 0$. The estimation at time step $k = 6$ can thus be made based on all the information available up to that time.

Consider now the time-varying delay case, with receptions as shown in Figure 6.3. At time $k = 2$, we are able to construct $\hat{x}(3|2)$, as for the fixed delay case. The detailed illustration of the following steps, for $k \in [3, 6]$, is given in Figure 6.4. In the next step $k = 3$, no information arrives at the controller. Taking this step as

a lost measurement ($\alpha_3 = 0$), we are able to estimate the state $\hat{x}(4|2)$. Notice that this is also the same step (missing measurement) as performed by the minimax controller under fixed delay at time $k = 6$, as previously illustrated. In the next step, the measurement arrives with no delay so we estimate $\hat{x}(5|4)$. Again, this step is the same as performed by the minimax controller with lossy measurements by (Moon and Başar, 2014, Theorem 1). At time step $k = 5$, a delayed and out-of-order packet transmitted at $k = 3$ arrives, and its delay is $\tau = 2$. In this case, the proposed estimator introduces this measurement in its correct order in the information structure $\mathcal{I}_0^5$. Moreover, it goes back to time $k = 2$, where a perfect estimation $\hat{x}(3|2)$ is available, and re-estimates the state $\hat{x}(4|3)$, with the newly received information, and also $\hat{x}(5|4)$. Then, it must perform updates with $\alpha_5 = 0$ and $\alpha_6 = 0$, in order to obtain $\hat{x}(6|4)$ and $\hat{x}(7|4)$, respectively, since those measurements have not yet arrived.

We now introduce the algorithm which describes the proposed minimax controller. Let us introduce the variable $\kappa_k \in \mathbb{N}_0$, which is the time value at which all the packets have been correctly derived, obtained at time k. In the example given above, $\kappa_k = 2$ at time $k = \{3, 4\}$ and changes to $\kappa_k = 5$ for $k \geq 5$. The number of packets received at each time interval is denoted as N_{pkts}. Buffers $\Theta_{\mathcal{Y}}$, $\Theta_{\mathcal{U}}$ and Θ_{Π} of appropriate sizes are created in order to store the information structure $\mathcal{I}_{k-\bar{\tau}}^k$. Moreover, buffers $\Theta_{\mathcal{X}}$ and Θ_{Σ} are used to store the state-estimate $\hat{x}(k)$ and GARE (6.8), from time $k - \bar{\tau}$ to time k, respectively. The values is all buffers are stored in ascending order of transmission time. If a measurement does not arrive at the controller at a particular time k, then it is not contained in the information structure and its buffer value is empty. Additionally, we create the temporary variables $\bar{x}(k)$ and $\bar{\Sigma}(k)$ which are used online. Algorithm 6.1 provides an implementation of the proposed minimax controller.

The opportunistic minimax controller is summarized in the following result.

Theorem 6.1. *Consider the dynamic game* (6.4)*, subject to the system dynamics* (6.1) *the information structure* (6.2) *and a varying delay* τ *and reordering of out-of-order packets upon their arrival. Let* $\gamma^\star > 0$*.*

The minimax controller exists for all $\gamma \geq \gamma^\star$*, if all the following conditions hold:*

(a) *GARE* (6.7) *has a solution over* $[0, K]$*;*

(b) *GARE* (6.8)*, with* $\alpha_k = 1$*, has a solution for* $k \in [0, K - \bar{\tau}]$*;*

(c) *GARE* (6.8)*, with varying* α_k*, has a solution for* $k \in [K - \bar{\tau} + 1, K]$*;*

(d) *The solution of* (6.7) *and* (6.8) *satisfies the following conditions*

$$\rho\left(\Sigma(k)Q\right) < \gamma^2, \quad k = 0, ..., K-1, \tag{6.12}$$

and

$$\rho\left(\widetilde{\Sigma}(k+1)M(k+1)\right) < \gamma^2, \quad k = 0, ..., K. \tag{6.13}$$

Algorithm 6.1 Minimax controller under out-of-order packets and time-varying delay

Define N_{pkts}, $\Theta_{\mathcal{Y}}$ based on the packets received at $]k-1,k]$ and initial $\kappa = 0$.
if $N_{pkts} = 0$ **then** ▷ No packet received
  $\alpha_k \leftarrow 0$
  $\bar{x}(k+1) \leftarrow (6.5)$
  $\bar{\Sigma}(k+1) \leftarrow (6.8)$
else ▷ Packet(s) received
  Update $\Theta_{\mathcal{Y}}$
  Update Θ_{Π}
  $\bar{x}(k+1) \leftarrow \Theta_{\mathcal{X}}(\kappa)$ ▷ Initializations
  $\bar{\Sigma}(k+1) \leftarrow \Theta_{\Sigma}(\kappa)$
  $u(t) \leftarrow \Theta_{\mathcal{U}}(\kappa)$
  $y(t) \leftarrow \Theta_{\mathcal{Y}}(\kappa)$
  for $t = \kappa : k$ **do** ▷ Re-compute $\bar{x}$ and $\bar{\Sigma}$
    if $\Theta_{\Pi}(t) \in \emptyset$ **then** ▷ No packet arrived
      $\alpha_t \leftarrow 0$
    else ▷ If packet arrived
      $\alpha_t \leftarrow 1$
    end if
    $\bar{x}(k+1) \leftarrow (6.5)$
    $\bar{\Sigma}(k+1) \leftarrow (6.8)$
    Update $\Theta_{\mathcal{X}}$
    Update Θ_{Σ}
  end for
end if
$M(k) \leftarrow (6.7)$
$u(k) \leftarrow (6.6)$ ▷ Compute new input
Update $\Theta_{\mathcal{U}}$
Update κ

The minimax controller is then given by:

- *the estimator* (6.5)*, with* $\alpha_k = 1$ *for* $k \in [0, K - \bar{\tau}]$*;*
- *the estimator* (6.5)*, with varying* α_k*, for* $k \in [K - \bar{\tau} + 1, K]$*;*
- *the feedback controller* (6.6) *for* $k \in [0, K]$*.*

If any of the conditions ***(a)*** *to* ***(d)*** *fail, then no such controller exists for* $\gamma \geq \gamma^{\star}$*.*

Proof. The proof follows from Propositions 6.1 and (Moon and Başar, 2014, Theorem 1) as follows. Notice that for time $k \in [0, K - \bar{\tau}]$, all measurements are received and the result from Proposition 6.1 holds. Recall that re-ordering occurs whenever out-of-order packets arrive, and the information structure (6.2) is updated at

each time step. Then, for time $k \in [K - \bar{\tau} + 1, K]$, according to the packet reception, the computation (in the case of packet without delay or missed packet) or re-computation (in the case of delayed packet arrival) of the minimax controller is performed, with the updated information structure $\mathcal{I}^K_{K-\bar{\tau}+1}$ (6.2). This procedure is the same as in (Moon and Başar, 2014, Theorem 1) as the update information structure $\mathcal{I}^K_{K-\bar{\tau}+1}$ becomes the same as the lossy information structure considered in (Moon and Başar, 2014, Theorem 1). This concludes the proof. Again, note that the separation principle does not hold in this case since the estimation and control GAREs (6.7) and (6.9), respectively, are coupled through condition (6.13). □

Memory and computation requirements

The proposed estimator and controller under varying delay and out-of-order messages requires the storage of the information $\Theta_{\mathcal{X}}$, $\Theta_{\mathcal{U}}$, $\Theta_{\mathcal{Y}}$ and Θ_{Σ}, at the controller unit as detailed in Algorithm 6.1. Note that the proposed estimator re-estimates the state and computes the respective GARE (6.8), from time κ up to time k, instead of re-estimating from time $k - \bar{\tau}$ to time k. This is done in order to reduce the computation of the minimax controller. However, note that the same memory requirements are necessary as the buffer size cannot be decreased, according to κ.

As discussed in Remark 6.1, a common approach to deal with varying-delays and out-of-order messages in NCSs is to use the conservative buffer strategy (Luck and Ray, 1990, 1994). In this way, the measurements are used by the controller in the correct order but with a fixed delay of $\bar{\tau}$ time units. Such implementation requires the storage of the same information as the solution we propose. This is the case as this approach requires the same information set while applying the method devised in Proposition 6.1. Additionally, both implementations require the storage of the solution of the GARE (6.7).

With respect to the computational complexity, while the proposed opportunistic minimax controller requires the online computation of $\Sigma(k)$, in the buffer case this value can be pre-computed and stored in memory. However, this would slightly increase the memory requirements when compared to the propose solution.

6.4 Linear temporal coding for minimax control

In this section, we propose an algorithm that performs linear temporal coding to improve the estimation quality under the minimax control framework. The underlying idea of the linear temporal coding strategy is that previous measurements can be aggregated together with the current measurement, and then transmitted to the controller, instead of sending only the current measurement. Hence, by receiving additional information, under the varying delay information structure, the state-estimation performed at the controller can be improved.

The intuition behind the linear temporal coding strategy devised is the following. The GARE equation (6.8) controls the estimation uncertainty, and conditions (6.12) and (6.13), necessary for the existence of the minimax controller, depend

on its evolution. Thus, by being able to reduce GARE (6.8), the uncertainty can be reduced, as well as conditions (6.12) and (6.13) are valid for a larger amount of time. The latter property can also be interpreted as that the attenuation level $\gamma^\star$ is reduced. These properties will be demonstrated through numerical simulations in Section 6.6. We first introduce the minimax controller design under linear temporal coding. Afterwards, we devise the linear temporal coding algorithm.

6.4.1 Minimax control under linear temporal coding

Recall that as per assumptions A1 and A2, a maximum delay of $\bar{\tau}$ time steps affects each transmitted packet and that acknowledgements exist between the controller and the sensor, in order to signal successful packet deliveries. The linear temporal coding mechanism may thus encode the information of the last $q \in [1, \bar{\tau}]$ measurements in the current packet to be transmitted. Consider the current measurement given by

$$y(k) = Cx(k) + Ew(k).$$

The current output of the linear temporal coding mechanism, which encodes the last q measurements, is denoted as $z(k)$ and given by

$$z(k) = T_k \bar{C} \begin{bmatrix} x(k) \\ y(k-1) \\ \vdots \\ y(k-q+1) \\ y(k-q) \end{bmatrix} + Ew(k),$$

where $T_k \in \mathbb{R}^{p \times p(q+1)}$ is the linear temporal coding matrix at time k, and $\bar{C} = \text{blkdiag}(C, I_p, \ldots, I_p)$. In order to apply the minimax controller techniques introduced in Section 6.1, an augmentation of the state-space system is required. Consider the new augmented state $\xi(k) = \begin{bmatrix} x(k)^T & \upsilon(k)^T \end{bmatrix}^T$, with $\upsilon(k) = \begin{bmatrix} y(k-q)^T, \ldots, y(k-1)^T \end{bmatrix}^T$. The dynamics of $\upsilon(k)$ can be described by

$$\upsilon(k+1) = F\upsilon(k) + Ly(k),$$

and $F = \begin{bmatrix} 0 & I_{(q-1)p} \\ 0 & 0 \end{bmatrix}$ and $L = \begin{bmatrix} 0 \\ I_p \end{bmatrix}$. Then, the new augmented system has the following dynamics

$$\begin{aligned} \xi(k+1) &= \bar{A}\xi(k) + \bar{B}u(k) + \bar{D}w(k), \\ z(k) &= T_k \bar{C}\xi(k) + \bar{E}w(k), \end{aligned} \tag{6.14}$$

where, $\bar{A} = \begin{bmatrix} A & 0 \\ LC & F \end{bmatrix}$, $\bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$, $\bar{D} = \begin{bmatrix} D \\ LE \end{bmatrix}$ and $\bar{E} = E$.

Note now that in system (6.14), the process and measurement disturbance become coupled, i.e., $P = \bar{D}\bar{E}^T \neq 0$, which was not the case for the minimax controller introduced in Section 6.1. Due to this fact, a redesign of the minimax controller is required, for which we follow (Başar and Bernhard, 2008, Chapter 6.4.1), where such coupling is discussed. For generality and ease of notation, we devise the new minimax controller and the linear temporal coding mechanism for system (6.14), considering the original notation $x = \xi$, $\hat{x} = \hat{\xi}$, $A = \bar{A}$, $B = \bar{B}$, $\ldots$, $E = \bar{E}$. Let us introduce the new matrices

$$\widetilde{A} = A - PN^{-1}T_kC, \quad H = DD^T - PN^{-1}P^T.$$

Then, the estimator equation is no longer (6.5), but is instead given by the modified estimator under linear temporal coding as

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + \widetilde{A}\Lambda(k)\left(\gamma^{-2}Q\hat{x}(k) + \alpha_k C^T T_k^T N^{-1}(z(k) - T_kC\hat{x}(k))\right), \tag{6.15}$$

and where the estimator gain is

$$\Lambda(k) = \left(\Sigma(k)^{-1} + \alpha_k C^T T_k^T N^{-1} T_k C - \gamma^{-2}Q\right)^{-1}. \tag{6.16}$$

Note that the control input equation (6.6) remains the same, as the disturbance coupling does not interfere in the controller design. Additionally, the modified observer GARE becomes:

$$\Sigma(k+1) = \widetilde{A}\Lambda(k)\widetilde{A}^T + H, \tag{6.17}$$

and $\widetilde{\Sigma}(k+1)$ becomes

$$\widetilde{\Sigma}(k+1) = \widetilde{A}(\Sigma(k)^{-1} - \gamma^{-2}Q)^{-1}\widetilde{A}^T + H, \tag{6.18}$$

while the controller GARE (6.7) remains the same. Conditions (6.10), (6.11), (6.12), (6.13) in Proposition 6.1 and Theorem 6.1, are now evaluated for the modified GAREs (6.17) and (6.18). This finalizes the required modifications to the minimax controller, taking into account the disturbance couplings and the linear temporal coding mechanism. Note that the controller must know T_k so that it can be used in the calculation of (6.15), (6.16), (6.17) and (6.18).

6.4.2 Linear temporal coding algorithm

We now turn our attention to the design of the linear temporal coding matrix T_k. Recall that the goal of this mechanism is to improve the quality of the estimation and allow for conditions (6.12) and (6.13) to be valid for a longer open-loop time. The latter goal allows as a consequence, for a reduction of the minimum feasible attenuation level $\gamma^\star$. We do this by minimizing the spectral radius of the GARE

solution (6.17) with $\alpha_k = 1$, which refers to the iteration at time k when a packet is received, i.e., when the measurement has an impact on the solution. Since the GARE is a symmetric matrix, the spectral radius corresponds to the maximum eigenvalue. Hence, by reducing the maximum eigenvalue of the solution of (6.17), the left-hand side of equations (6.12) and (6.13) is decreased. As conditions (6.12) and (6.13) are required to be satisfied for the minimax controller, as per Theorem 6.1, we are thus allowing for a larger number of time steps without the arrival of new measurement data. The same effect was observed in (Robinson and Kumar, 2007) and (Suia et al., 2014), in the context of Kalman filtering estimation with linear temporal coding. Even though we devise the linear temporal coding mechanism considering the proposed opportunistic minimax controller from Theorem 6.1, the same procedure can be applied to the fixed delay case in Proposition 6.1 and the loss case in (Moon and Başar, 2014, Theorem 1). Additionally, we note that since the minimax controller solution with $\gamma \to \infty$ converges to the LQG design, as pointed out in (Başar and Bernhard, 2008; Khargonekar, 1991; Moon and Başar, 2014), this design could also potentially be applied in that framework.

We now introduce the formulation of the optimization problem which finds the linear temporal coding matrix T_k for the above objective.

Consider the new augmented system (6.14), for which the minimax controller is designed. Given the information structure (6.2) under varying delay, the linear temporal coding optimization can be formulated as the following optimization problem:

$$\begin{aligned} \min_{T_k} \quad & \lambda_{\max}(\Sigma(k+1)) \\ \text{s.t.} \quad & 0 \prec \Sigma(k)^{-1} - \gamma^{-2} Q \\ \Sigma(k+1) = {} & (A - PN^{-1}T_k C)\Big(C^T T_k^T N^{-1} T_k C + \Sigma(k)^{-1} \\ & -\gamma^{-2} Q\Big)^{-1} (A - PN^{-1}T_k C)^T + H \end{aligned} \tag{6.19}$$

The first constraint is condition (6.12), while the second constraint is the GARE (6.17), with $\alpha_k = 1$, since the coding matrix T_k is designed assuming that the current packet to be transmitted arrives at the controller. We remark that the linear coding approach we propose is of a greedy nature since we optimize for the GARE solution (6.17) for the next time step. Thus, no guarantees can be given with respect to the global behavior of the linear temporal coding method. Nevertheless, the numerical results show that performance can be improved by utilizing this approach.

In what follows, we assume that $N = EE^T \succ 0$ is a diagonal matrix, which means that there is no coupling among measurement disturbances and that each measurement is affected by a disturbance. Without loss of generality, we then assume that $N = I$, since one can always apply a transformation to $z(k)$ (or $y(k)$) to have $N = I$.

In this formulation, the transmission delay of the transmitted packet at time

k is neglected in the design of T_k. This is due to the fact that the future delay is unknown. A modification could be made by resorting to a predictive, sequence-based approach as proposed in (Tang and de Silva, 2006a; Quevedo and Nesic, 2011), where a set $\{T_k, z(k)\}_\tau,\ \tau \in [0, \bar{\tau}]$ is computed, for each possible delay value, and transmitted to the actuators. Upon arrival of the packet with a certain delay τ, the actuator selects the correct $\{z(k), T_k\}_\tau$ from the received set. However, this would greatly increase the size of the data to be communicated over the network. Nevertheless, this issue would not occur for the application of this method to the fixed delay, and loss cases.

Even though the problem of minimizing the maximum eigenvalue of a matrix can be easily solved using SDP (Boyd and Vandenberghe, 2004), the above problem cannot be solved directly since the decision variable T_k appears in a quadratic form in the GARE $\Sigma(k+1)$. Thus, one has to resort to a heuristic to solve this problem. In the following, we show how the problem can be solved efficiently by using a rank constrained SDP heuristic optimization (Dattorro, 2005, Chapter 4.4).

A heuristic solution to Problem (6.19)

The solution to the above problem, which finds the linear temporal coding matrix T_k, can be obtained through the following heuristic method. The value of the linear temporal coding matrix is given by

$$T_k = UV^T, \tag{6.20}$$

where, $U \in \mathbb{R}^{p \times p+p(q+1)}$, $V \in \mathbb{R}^{p(q+1) \times p+p(q+1)}$ and given by $\begin{bmatrix} U \\ V \end{bmatrix} = Y$, $YY^T = X$. Matrix $X \in \mathbb{R}^{p+p(q+1) \times p+p(q+1)}$ is symmetric with the structure

$$X = \begin{bmatrix} I_p & X_a \\ X_a^T & X_b \end{bmatrix}, \quad X_a \in \mathbb{R}^{p \times p(q+1)}, \quad X_b \in \mathbb{R}^{p(q+1) \times p(q+1)},$$

where X_a and X_b are obtained by solving the following two subproblems 1 and 2, iteratively for a finite number of steps[1]. Let $0 < \sigma < 1$, $\beta > 0$. Moreover, let $\beta_0 = \lambda_{\max}\big(\Sigma(k+1)\big)$ denote the solution to $\lambda_{\max}\big(\Sigma(k+1)\big)$ when no coding is performed, i.e., $T = \begin{bmatrix} I & 0 \end{bmatrix}$.

[1]In both second-order system examples ($n = p = 3$) given in Section (6.6) with $q = 1$, the number of steps required was always less than 5.

Subproblem 1:

$$\min_{X,\beta} \quad (1-\sigma)\mathrm{tr}(W^T X) + \sigma\beta \tag{6.21}$$

$$\text{s.t.} \quad 0 < \beta \leq \beta_0 \tag{6.22}$$

$$0 \prec \begin{bmatrix} C^T X_b C + \Sigma(k)^{-1} - \gamma^{-2} Q & (A - P X_a C)^T \\ A - P X_a C & \beta I - H \end{bmatrix} \tag{6.23}$$

$$0 \prec C^T X_b C + \Sigma(k)^{-1} - \gamma^{-2} Q \tag{6.24}$$

$$X \text{ symmetric}$$

Subproblem 2:

$$\begin{aligned} \min_{W} \quad & \mathrm{tr}(W^T X) \\ \text{s.t.} \quad & \mathrm{tr}(W) = p(q+1) \\ & 0 \prec W \\ & I \succ W \end{aligned} \tag{6.25}$$

We note that both problems above can be efficiently solved using standard SDP solvers. In the example section, we utilize the SeDuMi solver (Sturm, 1999) in CVX (Grant et al., 2008).

We now provide the derivation of the above heuristic method, where the computation of T_k is performed indirectly by first obtaining the value of X, followed by Y, U and V. Suppose a matrix X, to be obtained from the optimization problem, is symmetric and given by $X = \begin{bmatrix} I_p & X_a \\ X_a^T & X_b \end{bmatrix}$. Now let us introduce $\begin{bmatrix} U \\ V \end{bmatrix} = Y$, where $YY^T = X$. The operation $YY^T = X$ can be performed using singular value decomposition (Horn and Johnson, 2012) on X, as

$$X = ZSZ^T = ZS^{\frac{1}{2}} S^{\frac{1}{2}} Z^T = YY^T,$$

where $Y = ZS^{\frac{1}{2}}$. Utilizing now Y, matrix X can be expressed as

$$X = YY^T = \begin{bmatrix} U \\ V \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}^T = \begin{bmatrix} I_p & UV^T \\ VU^T & VV^T \end{bmatrix}.$$

Suppose now that T_k is given by (6.20). Replacing T_k in X, we obtain

$$X = \begin{bmatrix} I_p & T_k \\ T_k^T & T_k^T T_k \end{bmatrix},$$

where the elements of X are $X_a = T_k$ and $X_b = T_k^T T_k$. In order for X to have this defined structure and for T_k to be obtained from it, one must require that X

has rank p, since $T_k \in \mathbb{R}^{p \times p(q+1)}$. To achieve a solution X with rank equal to p, we formulate a rank constrained optimization via SDP as detailed in (Dattorro, 2005, Chapter 4.4). The proposed solution is to iterate between two SDPs, where the first SDP solves $\min_X \operatorname{tr}(WX)$ ((6.21) in subproblem 1, for a matrix $W \in \mathbb{R}^{p+p(q+1) \times p+p(q+1)}$, and the second SDP solves subproblem 2 in (6.25), where it is required that $\operatorname{tr}(W) = p + p(q+1) - p = p(q+1)$, in order to enforce the rank constraint on X to be lower or equal than p.

The maximum eigenvalue minimization in the original problem (6.19), may be rewritten in its equivalent SDP form (Boyd and Vandenberghe, 2004), by introducing a variable $\beta > 0$ and solving instead

$$\begin{aligned} \min_{X,\beta} \quad & \beta \\ \text{s.t.} \quad & 0 \preceq \beta I - \Sigma(k+1), \end{aligned}$$

where $\Sigma(k+1)$ is a function of X through T_k. Substituting now $\Sigma(k+1)$ by the solution of the GARE (6.17) in the above linear inequality constraint, we arrive to the inequality

$$0 \preceq \beta I - (A - PT_kC)(C^T T_k^T T_k C + \Sigma(k)^{-1} - \gamma^{-2}Q)^{-1}(A - PT_kC)^T - H,$$

which can be represented with respect to the elements X_a and X_b as follows:

$$0 \preceq \beta I - (A - PX_aC)(C^T X_b C + \Sigma(k)^{-1} - \gamma^{-2}Q)^{-1}(A - PX_aC)^T - H.$$

Applying the Schur complement to the above inequality, the constraints (6.23) and (6.24) of subproblem 1 are obtained. Moreover, we add the constraint $\beta \leq \beta_0$, since we desire that the obtained $\beta = \lambda_{\max}(\Sigma(k+1))$, using the coding matrix T_k, to be smaller or equal than $\beta_0 = \lambda_{\max}(\Sigma(k+1))$, without coding and using $T = \begin{bmatrix} I & 0 \end{bmatrix}$. To finalize, we choose to perform a joint optimization, subject to a weight σ in subproblem 1.

The above heuristic method is not guaranteed to find the optimal solution $T_k^\star$ to problem (6.19). However, the optimization problem is always guaranteed to have a solution, since the no coding result, where $T_k = T = \begin{bmatrix} I & 0 \end{bmatrix}$, is always a solution. Additionally, by imposing the constraint (6.22) we are guaranteed to at least not do worse than the no coding operation. In the examples performed in Section 6.6, the algorithm has performed well, with improvements achieved with less than 5 iterations of the algorithm.

Finding a stationary solution

The proposed method computes a new coding matrix T_k at each time step. Since the coding matrix utilized by the sensor must be known by the controller, utilizing a different coding matrix at each step is undesirable in practice. We now propose a method to find a fixed and finite set of T_k matrices, to be used by the sensor and

Algorithm 6.2 Linear temporal coding for sensor

Offline
Define horizonOffline, numIter, σ, β_0, p and q and minimax controller parameters.
for $k = 1 :$ horizonOffline **do**
  $\delta \leftarrow$ number of consecutive missed packet deliveries
  for $i = 1 :$ numIter **do**
    β and $X \leftarrow$ solution of subproblem 1
    $W \leftarrow$ solution of subproblem 2
  end for
  $T(\delta) \leftarrow$ (6.20)
end for

Online
Define k as current time step
$\delta_k \leftarrow$ number consecutive missed packet deliveries
$T_k(\delta_k) \leftarrow$ value from table
$z(k) \leftarrow$ (6.14)
Transmit $z(k)$ to controller

controller. Let δ_k denote the number of consecutive packets not delivered to the controller, evaluated at time step k. In order to apply the above mechanism, we propose to compute a set of coding matrices T_k which depend solely on the current number of consecutive packets not delivered to the controller δ_k. This computation is performed offline in a test sequence, where packets are delivered correctly (no delay), and when packet delivery does not occur for a total of $\bar{\tau}$ consecutive steps. In this way, a total of $\bar{\tau} + 1$ coding matrices is obtained. In order to enforce stationarity under consecutive correct packet deliveries (no delay), we modify the objective function (6.21) by introducing the component $\mathrm{tr}(X - X_{k-1})$, which penalizes variations from the previous obtained solution X_{k-1}.

The sensor and controller must then encode and decode the sensor measurements utilizing the pre-calculated $T_k(\delta_k)$ values, according to the number of consecutive steps without packet deliveries at the controller. Then, the controller utilizes the linear coding matrix T_k corresponding to the number of missed packet deliveries at the time of the packet transmitted by the sensor, i.e., $T_k(\delta_{k-\tau})$. The scalar value $\delta_{k-\tau}$ can also be transmitted by the sensor to the controller.

The linear temporal coding mechanism to be implemented at the sensor and at the controller is summarized in Algorithms 6.2 and 6.3, respectively.

6.5 Experimental setup

In this section we discuss an example of a wireless NCS setup where packet scheduling attacks occur, introducing a delayed and out-of-sequence packet delivery. This

Algorithm 6.3 Linear temporal coding for controller

Define k as current time step
if Packet arrived at time k **then**
  Define $\delta = \delta_{k-\tau}$
  $T_k(\delta) \leftarrow$ value from table
  Run Algorithm 6.1 with modified minimax controller (6.15)-(6.17)
  Define δ_k as the number of consecutive packets missing at current time k
end if

experiment was evaluated using the co-simulator presented in Chapter 8, and is used to validate the minimax controller devised in Section 6.3.

6.5.1 Control system

In this experiment, we utilize the double tank system introduced in Chapter 2.8, where the aim is to track a piecewise constant reference value at the lower tank. The double tank system model (6.1) is obtained from the continuous-time double tank system model, sampled with a 2 s sampling period.

The reference values are set to 8 cm and 10 cm, changing every 150 s. In the same way as in Chapter 5, an integral state is introduced in the controller so that reference tracking can be achieved. The integral state at the controller is given by:

$$x_c(k+1) = x_c(k) + T_s\left(r(k) - C_c y(k)\right),$$

where x_c is the controller integral state and $C_c = \begin{bmatrix} 0 & 1 \end{bmatrix}$ The minimax controller is then applied to the new augmented system with state $\xi(k) = \begin{bmatrix} x(k) & x_c(k) \end{bmatrix}^T$. The control input is given by

$$u(k) = K_\xi \hat{\xi}(k),$$

where K_ξ and $\hat{\xi}(k)$ are given by the (6.6) and (6.5), respectively. Following the augmented model, we set

$$D = 0.1\begin{bmatrix} B & B & B & 0 & 0 & 0 \end{bmatrix},\ E = 0.1\begin{bmatrix} 0 & 0 & 0 & I_3 & I_3 & I_3 \end{bmatrix}.$$

Moreover, we selected the matrices $Q = Q_K = Q_0 = 0.1I$.

6.5.2 Wireless network

All the wireless devices are programmed using TinyOS (Levis et al., 2004). As the protocol stack, each wireless device utilizes the CSMA/CA mechanism, while the routing layer is defined by the state-of-the-art routing protocol for low power and lossy networks (RPL), proposed by the internet engineering task force (IETF) (Winter et al., 2012). The implementation of RPL in TinyOS used in the experiments was published and validated in (Ko et al., 2011a).

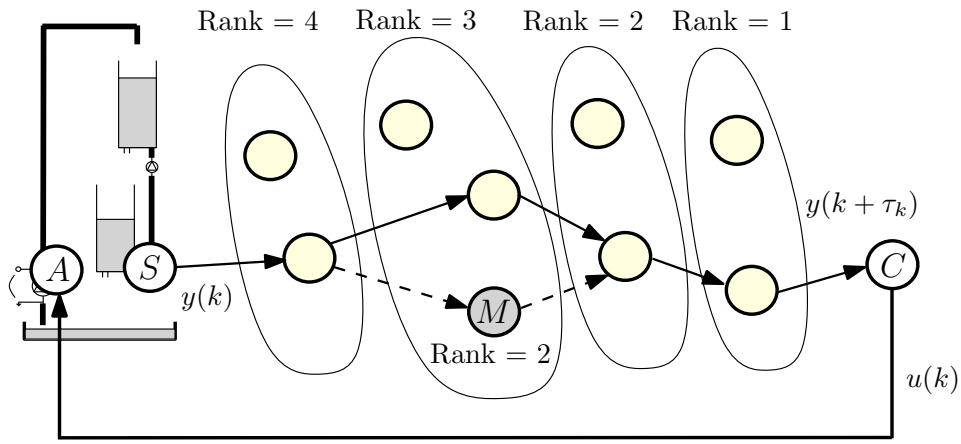


Figure 6.5: Multi-hop networked control scenario under malicious node attack.

The RPL routing protocol is a distributed protocol which defines mechanisms for multipoint-to-point, point-to-multipoint and point-to-point traffic between devices inside the network. In this work, we focus on the latter case, where a sensor node transmits information through a relay network to the controller (sink node), as depicted in Figure 6.5.

During the protocol operation, specific routing messages are exchanged among nodes so that: nodes are aware of their neighbors and their position/depth in the network; they can select the best neighbor (denoted as parent) using specific routing metrics (nodes reliability, latency and the total number of hops of each neighbor to the sink node), and for periodically checking if any re-routing is required due to path failure or instability. [2] The node depth in the network towards the sink is defined as the node rank, where the sink node has rank 0. An example of the RPL ranking assignment is presented in Figure 6.5.

Implementation details

The multi-hop wireless network is setup as depicted in Figure 6.5 in the double tank system testbed presented in Figure 2.3 which is implemented in GISOO. For this experiment, we deployed a wireless sensor and wireless controller, as well as nine relay nodes, specifically displaced to form a 5-hop network as shown in Figure 6.5. A malicious node (M node, in grey) is placed in the network at 3-hop depth from the controller node.

[2]For ease of explanation, we do not use the precise terms defined by the standard and perform an high-level explanation of the protocol. A detailed description of the protocol can be found in (Winter et al., 2012)

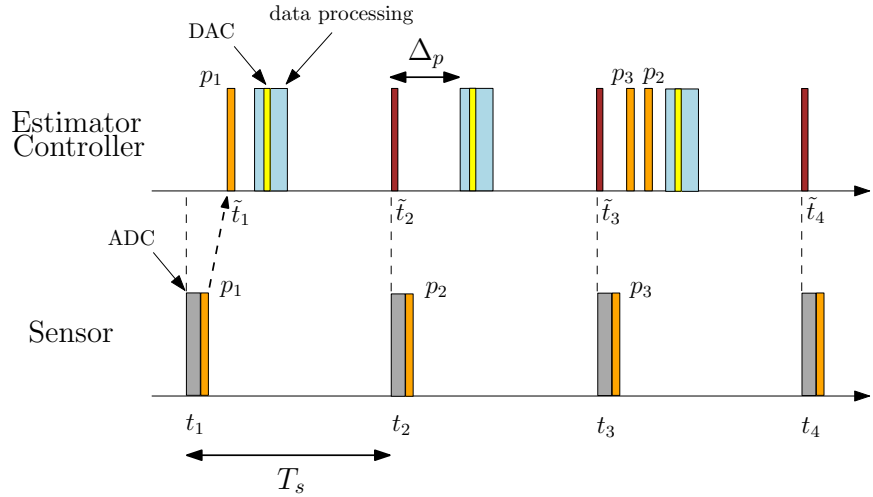


Figure 6.6: Data transmission and synchronization scheme for the sensor node and the estimator/controller node. The sensor node transmits data packets (p_i) every T s to the estimator/controller node.

The sensor node generates a packet every $T_s = 2$ s which has the controller node as the destination. An illustration of the data transmission and synchronization between the sensor and estimator/controller is depicted in Figure 6.6. After starting, the sensor node sets a local periodic timer with period T_s and performs the ADC measurement of the tank levels. This data, together with the time at which the periodic timer was set, is transmitted to the controller node in packet p_1, which arrives after a variable transmission delay.[3] When this packet arrives at time $\tilde{t}_1$, the controller node sets up Timer P to fire once at $\tilde{t}_2 = T_s - (\tilde{t}_1 - t_1)$ s. After this timer is fired it will be set to fire periodically every T_s s. Additionally, another timer (Timer D) is set at this node to fire Δ_p s after t_1, i.e., in $d_1 = \Delta_p - (\tilde{t}_1 - t_1)$ s. Afterwards, Timer D is set to fire Δ_p s after Timer P fires. Whenever Timer D fires, Algorithm 6.1 is run, utilizing all the packets which have arrived in the time interval $[\tilde{t}_i, \tilde{t}_i + \Delta_p]$. Whenever the control input is calculated, the new value is transmitted to the DAC output.

Through pre-evaluation, it was seen that the packet transmission delay through the multi-hop network ranged between approximately 50 ms and 250 ms and so, the value of Δ_p was set to 500 ms. In this way, an unattacked packet transmitted by the sensor node will almost surely arrive and be processed within the current interval. We remark that wireless interference or routing layer updates may cause a higher packet transmission delay, however, this issue did not occur in the experiment.

[3] For simplicity of implementation it is assumed that the first packet transmitted is not attacked by the malicious node.

6.5.3 Scenario definition

Through the analysis of the RPL protocol (Winter et al., 2012), it is noticeable that by being a distributed routing protocol, where each node selects its parents by using local information, it is vulnerable to attacks, as reported in (Le et al., 2012; Gaddour and Koubâa, 2012). Particularly, a malicious node can mount an attack by advertising that it has a very high reliability and very low latency, or that it has a lower rank value that it in practice has. The attacker approach followed in this experiment is the latter, as depicted in Figure 6.5. The malicious node is placed at a rank 3 region of the network, but set to advertise to all neighbors that it has rank 2. Hence, the node forwarding data from the sensor node to the controller, which is placed before the area where the malicious node is deployed, will select the malicious node as its best parent and forward the data packets to it. In the current RPL implementation such attack is not detected and all data packets flowing through the network will be forwarded through the malicious node. We note that the study of the resilience and security of RPL is currently being the subject of much research (Dvir et al., 2011; Le et al., 2012). After the malicious node forces all data packets with sensor data transmitted through the network to pass through it, it is able to implement the packet scheduling attack. The maximum delay is set to $\bar{\tau} = 8\,\text{s}$, allowing each packet to be inflicted with a delay between 0 to 8 s. The delay is introduced at the MAC layer level, by storing in memory all the packet data (routing and sensor data information), to be forwarded with a selected delay, at the corresponding time in the future.

6.6 Evaluation results

We start by providing the experimental results of the application of the proposed minimax controller without the linear temporal coding mechanism to the experimental setup provided in the previous section. Afterwards, we numerically evaluate the minimax controller with linear temporal coding.

6.6.1 Minimax control without linear temporal coding

We evaluate the proposed minimax controller with and without the packet scheduling attack, and we compare it to the minimax controller with a buffer approach (Luck and Ray, 1990, 1994), which introduces a fixed delay at the controller. The buffer length $\bar{\tau}$ is defined to be the maximum number of k-step delay imposed by the attacker which in this experiment is $\frac{8}{2} = 4$.

The buffer minimax controller and our proposed minimax controller are designed following Proposition 6.1 and Theorem (6.1), respectively. The value $\gamma^\star = 47.29$ was found for the fixed delay $\bar{\tau}$ case, while $\gamma^\star = 35.09$ was found for the time-varying delay case. However, one must pick $\gamma = 47.29$ for the proposed design since in the worst-case scenario the delay will be $\bar{\tau}$.

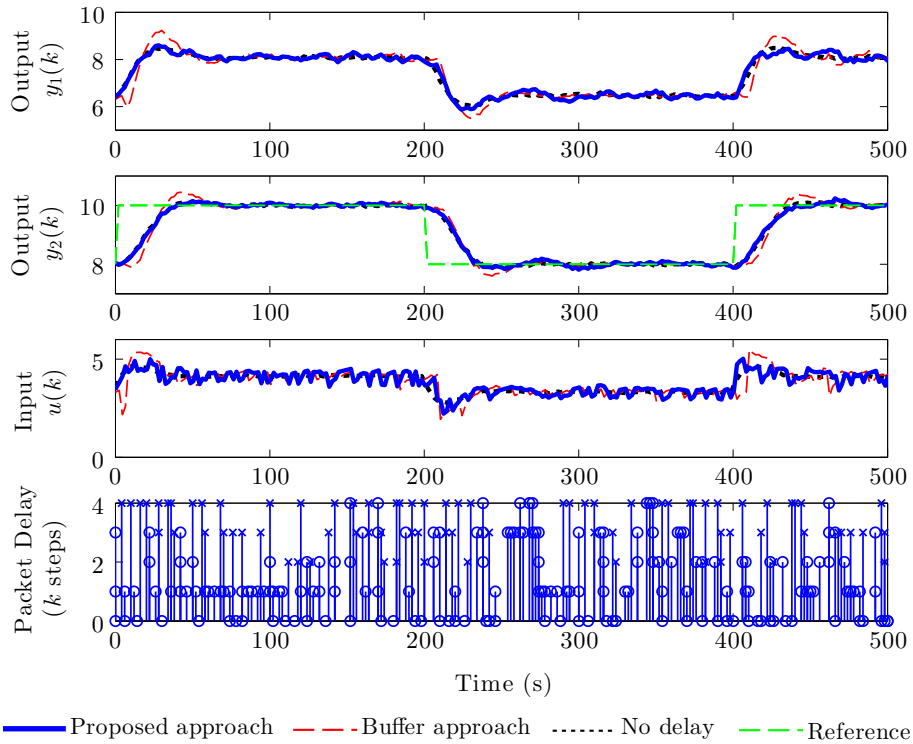


Figure 6.7: The output values and control input for the three different cases evaluated: no delay (dotted line), buffer approach (dashed line) and the proposed approach (solid line). Lower figure depicts the packet delay induced by the attacker to each received packet. A cross represents an out-of-order packet while a circle denotes a packet delivered in the correct order.

We analyze the control cost J^K given by (6.3), the estimation cost

$$J_{\text{est}}^K = \sum_{k=0}^{K-1} \|x(k) - \hat{x}(k)\|^2,$$

and the square of the maximum of the RHS of the necessary conditions (6.12) and (6.13) for the existence of the minimax controller, i.e.,

$$c_\gamma^2 = \max \left\{ \rho\left(\Sigma(k)Q\right), \rho\left(\widetilde{\Sigma}(k+1)M(k+1)\right) \right\}.$$

The experiment was performed for a $K = 85$ steps, which corresponds to a 170 s.

Figure 6.7 depicts the time-response analysis. The figure shows the output values of the system under three different evaluated cases: no delay (dotted line), buffer

Table 6.1: Experimental evaluation: out-of-order communications and time-varying delay

	$\gamma^\star$	c_γ^2	J^K	J_{est}^K
No delay	35.09	35.09	1	1
Buffer approach	47.29	47.29	7.78	1.51
Proposed approach	47.29	35.09	4.14	1.41

approach (dashed line) and the proposed approach (solid line), and the packet delay induced by the attacker to each received packet. As the proposed minimax controller makes use of all information up to time the current time step, an improved performance with respect to the over-designed buffer implementation is obtained. Moreover, the closed-loop response of the proposed approach follows closely the one of the case without delay. The values of the normalized control cost and estimation cost with respect to the no delay case are depicted in Table 6.1, together with the parameters $\gamma^\star$ and c_γ^2.

6.6.2 Minimax control with linear temporal coding

We now evaluate the linear temporal coding mechanism proposed in Section 6.4. For this, we apply this strategy in various different processes and quantify the improvement obtained when using the minimax controller with and without the linear temporal coding scheme. The processes utilized were one stable (system A) and one unstable first-order system (system B), the stable second-order double tank system (system C) and an unstable second-order system (system D), and the second-order double integrator system (system E). The parameters of each system are given in the Appendix C.1. We set the maximum delay to be $\bar{\tau} = 5$. The linear temporal coding is designed with $q = 1$ for all cases, hence, only the previously transmitted measurement is coded in the current measurement. The time horizon is set to $K = 30$. The algorithm parameter numIter $= 5$. For each system, we evaluate the improvement given by the coding mechanism with respect to the case without the coding mechanism in terms of control cost J^K given by (6.3), the estimation cost J_{est}, $\gamma^\star$ and also c_γ^2. The improvement of each parameter is calculated as the weighted difference of the coding value to the no coding value, e.g., for the control cost J^K:

$$\Delta J^K = \frac{J^K_{\text{delay,no coding}} - J^K_{\text{delay,coding}}}{J^K_{\text{delay,no coding}} - J^K_{\text{no delay,no coding}}}.$$

The same calculation and notation follows for the rest of the parameters, and the improvement is depicted in percentage.

We begin by showing the evolution of the squared root of the RHS of conditions (6.12) and (6.13) for the second-order unstable system D, for out-of-order communications and time-varying delays. The results are presented in Figure 6.8.

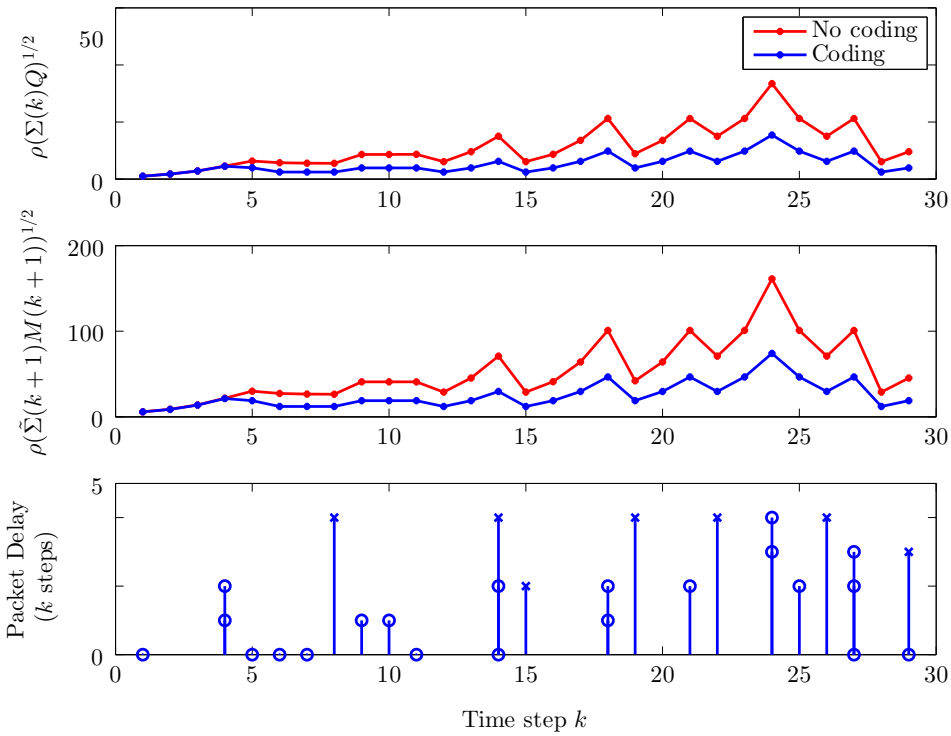


Figure 6.8: The values of $\rho(\Sigma(k)Q)^{1/2}$, $\rho(\widetilde{\Sigma}(k+1)M(k+1))^{1/2}$ and packet delivery information, during the control of the second-order unstable system D with out-of-order communications and time-varying delay.

Recall that these values must be upper bounded by γ for the minimax controller to exist. We find that the parameter $\gamma^\star = 162$ for the coding case and $\gamma^\star = 80$ without coding. The time-delay affects the magnitude of these values, while the absence of a measurement makes the values increase. By utilizing the linear temporal coding method, the magnitude of these values is decreased. For this packet trace, c_γ^2 is 161.05 without coding, and 74.01 with coding. Note also that the most stringent condition is condition (6.13). Table 6.2 summarizes the improvement of all statistics for the considered systems. As it can be seen, improvements were obtained for all the evaluated systems.

6.7 Summary

In this chapter, we have introduced a minimax controller which is robust to out-of-order messages and time-varying delays. The proposed minimax controller is

Table 6.2: Evaluation of the minimax control under linear temporal coding: out-of-order communications and time-varying delay. Improvement in percentage.

	$\Delta\gamma^\star$	Δc_γ^2	ΔJ^K	ΔJ_{est}
System A	42.00	52.88	73.50	96.05
System B	3.43	99.7	23.71	37.73
System C	11.21	12.50	13.49	50.15
System D	60.79	64.97	62.55	81.69
System E	28.57	46.11	45.05	86.51

opportunistic in the sense that it is designed for the worst-case delay, while immediately using the information in the received packets. The ability to immediately use the received information leads to improved control performance. The proposed controller was evaluated in an experimental setup of a multi-hop wireless NCS where a malicious node introduced time-varying delay and out-of-order packet delivery. We have also devised a linear temporal coding scheme for the minimax controller, which aims at improving the estimation performance and the robustness of the minimax controller to network imperfections. This strategy was validated through numerical examples.

Chapter 7

Distributed reconfiguration for sensor and actuator faults

This chapter addresses the problem of distributed reconfiguration of networked control systems with sensor and actuator faults and where there exists sensor and actuator redundancy. Using the proposed scheme, healthy sensors and actuators are able to locally compensate for faults disabling a given set of sensors and actuators in the network. The proposed distributed method is able to minimize the loss in estimation and control performance under faults while achieving model-matching: the desired closed-loop estimation and state trajectory remains the same with and without faults. We evaluate the distributed reconfiguration strategy numerically but also experimentally. Particulary, we implemented the distributed reconfiguration scheme on the room heating testbed introduced in Chapter 2.8. The testbed is comprised of three heaters with wireless and computation capabilities, which control the temperature dynamics of a room at KTH. The results demonstrate that the proposed method is able to quickly reconfigure the system under sensor and actuator faults, while achieving the proposed reconfiguration goals.

The rest of this chapter is organized as follows. Section 7.1 presents the system architecture and formulates the problem. The centralized solution to the reconfiguration problem is presented in Section 7.2. In Section 7.3 it is shown that the reconfiguration can be distributed among the sensor or actuator nodes and an efficient algorithm is devised. Stability properties of the system under the proposed distributed reconfiguration scheme are given in Section 7.4. A numerical example illustrates the distributed sensor and actuator reconfiguration methods in Sections 7.5. In Section 7.6 the experimental setup of the room heating testbed with distributed heater reconfiguration is presented, while the experimental results are discussed in Section 7.7. Finally, Section 7.8 concludes this chapter.

Notation

The notation $|\cdot|$ represents the cardinality of a set, and $\mathcal{A} \setminus \mathcal{B}$ denotes the set obtained by removing set $\mathcal{B}$ from set $\mathcal{A}$, for $\mathcal{B} \subseteq \mathcal{A}$. A network is represented by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The edge $e_k = (i, j) \in \mathcal{E}$ indicates that nodes i and j can exchange information. Denote $\mathcal{N}_i = \{j | j \neq i, (i, j) \in \mathcal{E}\}$ as the neighbor set of node i where we assume that the network has no self-loops. Define $\mathcal{C}$ as the span of real symmetric matrices, $\mathcal{S}^n$, with sparsity pattern induced by the network, i.e., $\mathcal{C} = \{S \in \mathcal{S}^n | S_{ij} = 0 \,\text{if}\, i \neq j \,\text{and}\, (i, j) \notin \mathcal{E}\}$.

7.1 Problem formulation

The architecture of the considered networked control system is depicted in Figure 7.1. This architecture has two networks, one of sensors and one of actuators, each with sufficient redundancy in components. Each network is represented by an undirected graph. Each sensor or actuator is able to exchange information with its neighbors within the network. In typical applications such as building automation and industrial process control, a large number of sensors is expected to be deployed. To reduce the sensor-to-estimator communication, the information from the sensor nodes is fused at aggregator nodes, which connect to the estimator. The estimator is responsible for computing the state-estimate to be broadcasted to the actuators in the network which compute the control input values. The individual components of the system are described below. In this section, we utilize the graph theory introduced in Chapter 1.5.

7.1.1 System model

Suppose the plant is modeled by a stochastic linear time-invariant differential equation,

$$\mathrm{d}x(t) = Ax(t)\,\mathrm{d}t + B\Gamma_u(t)u(t)\,\mathrm{d}t + \mathrm{d}w(t) \tag{7.1}$$

$$y(t)\,\mathrm{d}t = \Gamma_y(t)\left(Cx(t)\,\mathrm{d}t + \mathrm{d}v(t)\right), \tag{7.2}$$

with a state $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^p$ and $u(t) \in \mathbb{R}^m$ are the measurement vector and input vector, respectively, with redundancy in their components, and $w(t) \in \mathbb{R}^n$ and $v(t) \in \mathbb{R}^p$ are independent Wiener processes with uncorrelated increments. The incremental covariances are $W\,\mathrm{d}t$ and $V\,\mathrm{d}t$, respectively. Moreover, processes $w(t)$ and $v(t)$ are assumed to be mutually uncorrelated (Åström, 1970).

Sensor and actuator faults are modelled by the diagonal matrices $\Gamma_y(t) \in \mathbb{R}^{p\times p}$ and $\Gamma_u(t) \in \mathbb{R}^{m\times m}$, respectively, with $[\Gamma_y(t)]_{ii} = \gamma_{y_i}(t) \in \{0, 1\}$ and $[\Gamma_u(t)]_{ii} = \gamma_{u_i}(t) \in \{0, 1\}$. Here $\gamma_{y_i}(t)$ ($\gamma_{u_i}(t)$) represents the effectiveness of sensor (actuator) i at time t, where $\gamma_{y_i}(t) = 1$ ($\gamma_{u_i}(t) = 1$) means that the sensor (actuator) is functioning (healthy), while $\gamma_{y_i}(t) = 0$ ($\gamma_{u_i}(t) = 0$) indicates that the sensor (actuator) is faulty. The system is initially under nominal conditions, hence $\Gamma_y(t) = I$ and

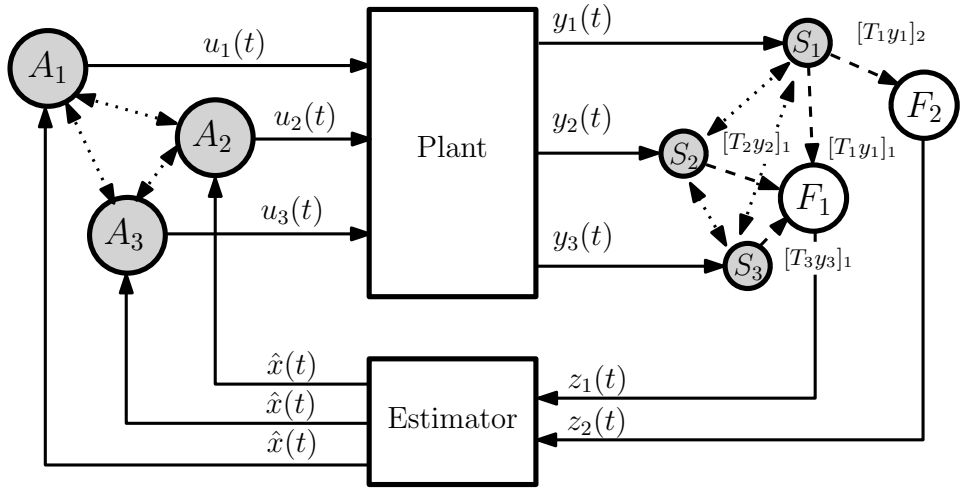


Figure 7.1: Networked control system with a network of sensors S_1, S_2 and S_3, aggregator nodes F_1 and F_2 and actuators A_1, A_2 and A_3. Sensors and actuators are responsible for reconfiguring themselves when system failures occur, through local information exchange in the network.

$\Gamma_u(t) = I$ for $t < 0$. All faults are assumed to occur simultaneously at time $t = 0$ and remain unchanged thereafter, which allows the time argument to be omitted. However, the methods devised in this work directly apply to the non-simultaneous fault case.

The sensor nodes apply a local linear transformation to the sensor measurements and transmit these values through the network to aggregation nodes which fuse the sensor data from several sensors. The fused signal is aggregated as

$$z(t)\,\mathrm{d}t = Ty(t)\,\mathrm{d}t = T\Gamma_y Cx(t)\,\mathrm{d}t + T\Gamma_y\,\mathrm{d}v(t), \tag{7.3}$$

where $T \in \mathbb{R}^{s\times p}$ is the aggregation matrix and $z(t)$ is transmitted to the estimator. It is assumed that the number of fused variables s is strictly smaller than the number of measurements p.

The sensor and actuator networks are represented by the connected and undirected graphs $\mathcal{G}_y(\mathcal{V}_y, \mathcal{E}_y)$ with $|\mathcal{V}_y| = p$ vertices and $\mathcal{G}_u(\mathcal{V}_u, \mathcal{E}_u)$ with $|\mathcal{V}_u| = m$ vertices, respectively. For simplicity of presentation, we assume that each aggregator node is connected to all sensor nodes. The set of sensor and actuator nodes is defined as $\mathcal{V} \triangleq \mathcal{V}_y \cup \mathcal{V}_u$, whereas we denote $\mathcal{V}^f \subseteq \mathcal{V}$ as the set of faulty nodes. Let the set of healthy nodes be $\mathcal{V}^h \triangleq \mathcal{V} \setminus \mathcal{V}^f$ with $\mathcal{E}^h = \{(i,j) \in \mathcal{E} \,|\, i,j \in \mathcal{V}^h\}$. The subgraphs $\mathcal{G}_y^h(\mathcal{V}_y^h, \mathcal{E}_y^h)$ and $\mathcal{G}_u^h(\mathcal{V}_u^h, \mathcal{E}_u^h)$ correspond to the graphs of the healthy sensor and actuator nodes, respectively, where $\mathcal{V}_y^h \triangleq \mathcal{V}^h \cap \mathcal{V}_y$, $\mathcal{V}_u^h \triangleq \mathcal{V}^h \cap \mathcal{V}_u$, $\mathcal{E}_y^h \triangleq \mathcal{E}^h \cap \mathcal{E}_y$, and $\mathcal{E}_u^h \triangleq \mathcal{E}^h \cap \mathcal{E}_u$.

We assume that the controller is given by the continuous-time linear-quadratic Gaussian (LQG) controller (Åström, 1970). Let the pair (TC, A) be observable and (A, B) be controllable. Next we describe the controller and estimator design under nominal conditions with $\Gamma_u = I$ and $\Gamma_y = I$. For LQG control, the feedback gain is obtained as the minimizer of the control cost criterion

$$J_c \triangleq \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau \mathbf{E}\{x(t)^\top Q x(t) + u(t)^\top R u(t)\} \ \mathrm{d}t$$

where $Q \succeq 0$ and $R \succ 0$ are weight matrices. We assume R is diagonal. The optimal LQ controller is given by

$$u(t) = -K\hat{x}(t) = -R^{-1}B^\top P\hat{x}(t) \tag{7.4}$$

where $\hat{x}(t)$ is the state estimate and P the solution to the Riccati equation

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0.$$

The state-estimate is computed by the Kalman-Bucy filter (Åström, 1970) as follows

$$\dot{\hat{x}}(t) = (A - LTC)\hat{x}(t) + Bu(t) + Lz(t), \tag{7.5}$$

with

$$L = \Sigma C^\top T^\top (TVT^\top)^{-1},$$

where $\Sigma = \lim_{t\to\infty} \mathbf{E}\{e(t)e(t)^\top\}$ is the steady-state covariance matrix of the estimation error $e(t) = \hat{x}(t) - x(t)$ given by the Riccati equation

$$A\Sigma + \Sigma A^\top - \Sigma C^\top T^\top (TVT^\top)^{-1} TC\Sigma + W = 0.$$

The Kalman-Bucy filter minimizes the expected mean-squared error, which we denote as the estimation cost function:

$$J_e \triangleq \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^\tau \mathbf{E}\{e(t)^\top e(t)\} \, \mathrm{d}t. \tag{7.6}$$

From now on we drop the time argument in the variables x, u, etc. when it is clear from the context.

7.1.2 Reconfiguration problem

Consider a scenario where faults have disabled several sensor and actuator nodes, yielding $\Gamma_u \neq I$ and $\Gamma_y \neq I$. A possible corrective action is to modify the aggregation matrix T and feedback matrix K so that only the remaining healthy sensors and actuators are used to guarantee a certain level of performance of the system.

Let $\tilde{u} \in \mathbb{R}^m$ and $\tilde{z} \in \mathbb{R}^s$ denote the reconfigured control and sensor fusion signals after the fault. They are given by

$$\begin{aligned} \tilde{z}\,\mathrm{d}t = \tilde{T}y\,\mathrm{d}t = \tilde{T}\Gamma_y C x\,\mathrm{d}t + \tilde{T}\Gamma_y\,\mathrm{d}v, \\ \tilde{u} = -\tilde{K}\hat{x}. \end{aligned}$$

Denote $\tilde{A}_c(\tilde{K}) = A - B\Gamma_u\tilde{K}$ and $\tilde{A}_e(\tilde{T}) = A - L\tilde{T}\Gamma_y C$ as the system matrices for the closed-loop dynamics of the system and estimator, respectively. The objective of the reconfiguration is to achieve model-matching (Gao and Antsaklis, 1991; Staroswiecki and Cazaurang, 2008; Lunze and Richter, 2008) for both the estimation dynamics and the closed-loop system dynamics by computing $\tilde{T}$ and $\tilde{K}$ after the fault occurs, respectively. Model-matching is a common reconfiguration goal as it guarantees maintained system behavior in the presence of faults. The definition of model-matching reconfiguration is as follows. Let us denote the closed-loop estimator dynamics before the fault as $A_e = A - LTC$ and the nominal closed-loop system matrix as $A_c = A - BK$. Then, *model-matching on the estimation error dynamics* is achieved if $\tilde{A}_e(\tilde{T}) = A_e$ for some new aggregation matrix $\tilde{T}$. *Model-matching on the closed-loop system dynamics* is achieved if $\tilde{A}_c(\tilde{K}) = A_c$ for some new feedback gain matrix $\tilde{K}$.

Assumption 7.1. *Model-matching is feasible, i.e.,* $\mathrm{Im}(B\Gamma_u) \subseteq \mathrm{Im}(B)$ *and* $\mathrm{Im}(C^\top\Gamma_y) \subseteq \mathrm{Im}(C^\top)$.

As the model-matching constraints are under-determined, i.e., they admit multiple solutions, we propose to find the model-matching solutions that minimize certain quadratic costs. In particular, the cost function for sensor reconfiguration is the quadratic estimation cost (7.6) under the fault

$$J_e(\tilde{T}) = \lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \mathbf{E}\{\tilde{e}^\top\tilde{e}\}\ \mathrm{d}t \tag{7.7}$$

where $\tilde{e}$ is the estimation error after the fault occurred. Furthermore, we define the objective function of the actuator reconfiguration as the quadratic control cost for the reconfigured control input

$$J_c(\tilde{K}) = \lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \mathbf{E}\{x^\top Q x + \tilde{u}^\top R\tilde{u}\}\ \mathrm{d}t. \tag{7.8}$$

The sensor and actuator networked reconfiguration problem is to find the reconfigured aggregation matrix $\tilde{T}$ and feedback gain matrix $\tilde{K}$ which minimize the estimation (7.7) and control cost (7.8), respectively, subject to the model-matching condition.

The sensor reconfiguration can be re-formulated as

$$\begin{aligned} \min_{\tilde{T}} \quad & J_e(\tilde{T}) \\ \text{s.t.} \quad & A - L\tilde{T}\Gamma_y C = A - LTC, \end{aligned} \tag{7.9}$$

while the actuator reconfiguration problem is

$$\begin{aligned} \min_{\tilde{K}} \quad & J_c(\tilde{K}) \\ \text{s.t.} \quad & A - B\Gamma_u\tilde{K} = A - BK. \end{aligned} \tag{7.10}$$

The solution to the these optimization problems may be achieved in a centralized or distributed fashion. Next we describe a centralized approach to solve them, in which we assume that the reconfiguration takes place instantaneously. Later, we propose an efficient distributed solution based solely on local information exchange among sensor nodes and actuators nodes. In Section 7.4 we analyze the stability properties of the proposed distributed algorithm when the reconfiguration is not instantaneous.

7.2 Centralized sensor and actuator reconfiguration

We now present the centralized sensor and actuator reconfiguration results.

7.2.1 Centralized sensor reconfiguration

The optimal solution to (7.9) can be characterized as follows.

Proposition 7.1. *The solution to the optimization problem* (7.9) *is*

$$\tilde{T}^\star = TC(C^\top V^{-1}\Gamma_y C)^\dagger C^\top \Gamma_y V^{-1}. \tag{7.11}$$

In order to prove Proposition 7.1 we use the following lemma.

Lemma 7.1. *Optimization problem* (7.9) *is equivalent to*

$$\begin{aligned} \min_{\tilde{T}} \quad & \operatorname{tr}\left((W + L\tilde{T}\Gamma_y V\Gamma_y\tilde{T}^\top L^\top)Z_e\right) \\ s.t. \quad & LTC = L\tilde{T}\Gamma_y C \\ & 0 = A_e^\top Z_e + Z_e A_e + I. \end{aligned} \tag{7.12}$$

Proof. The proof is given in the Appendix. □

We now derive the optimal solution to (7.12), which is also the solution to the sensor reconfiguration problem (7.9).

Proof of Proposition 7.1. Consider the optimization problem (7.12), which is convex. Note that the second equality constraint is a Lyapunov equation with the Hurwitz system matrix A_e, determined by the model-matching condition. Hence, the variable Z_e is uniquely defined by the constraint and can be computed before hand. The Lagrangian function for (7.12) is $\mathcal{L}(\tilde{T}, \Lambda) = \operatorname{tr}\left((W + L\tilde{T}\Gamma_y V\Gamma_y\tilde{T}^\top L^\top)Z_e\right) +$

$\text{tr}\left(\Lambda^\top(LTC - L\tilde{T}\Gamma_y C)\right)$, where $\Lambda \in \mathbb{R}^{n\times n}$ represents the Lagrange multipliers. Using the trace derivative expressions, the Karush-Kuhn-Tucker (KKT) optimality conditions can be written as

$$\begin{aligned} 0 &= \frac{\partial}{\partial \tilde{T}}\mathcal{L}(\tilde{T},\Lambda) = 2L^\top Z_e L\tilde{T}\Gamma_y V\Gamma_y - L^\top \Lambda C^\top \Gamma_y \\ 0 &= LTC - L\tilde{T}\Gamma_y C \end{aligned}$$

and can be rewritten as

$$\begin{aligned} 0 &= \tilde{T}\Gamma_y - \frac{1}{2}(L^\top Z_e L)^\dagger L^\top \Lambda C^\top V^{-1}\Gamma_y \\ 0 &= LTC(C^\top V^{-1}\Gamma_y C)^\dagger - \frac{1}{2}L(L^\top Z_e L)^\dagger L^\top \Lambda. \end{aligned}$$

Solving the above equations yields the optimal solution (7.11). □

Figure 7.2 illustrates the centralized reconfiguration that is performed by a system component denoted as reconfiguration manager. A fault occurs at sensor S_2, which detects that it is faulty, reporting it to the reconfiguration manager which now knows Γ_y. The reconfiguration manager solves (7.11) to derive the new aggregation matrix $\tilde{T} = [\tilde{T}_1 \dots \tilde{T}_p]$. Then, $\tilde{T}_1$ is sent to sensor S_1 and $\tilde{T}_3$ to sensor S_3, which compute $\tilde{T}_1 y_1$ and $\tilde{T}_3 y_3$, where $\tilde{T}_i y_i = [[\tilde{T}_i y_i]_1 \dots [\tilde{T}_i y_i]_s]^\top$. Each non-zero component $[\tilde{T}_i y_i]_j$ is sent to the j-th aggregator, allowing each aggregator node to compute z_j and transmit this value to the controller node.

7.2.2 Centralized actuator reconfiguration

The optimal centralized actuator reconfiguration is now presented.

Proposition 7.2. *The solution to the optimization problem* (7.10) *is*

$$\tilde{K}^\star = \Gamma_u R^{-1} B^\top (B\Gamma_u R^{-1} B^\top)^\dagger BK. \tag{7.13}$$

To prove the above result we use the following lemma.

Lemma 7.2. *Optimization problem* (7.10) *is equivalent to*

$$\begin{aligned} \min_{\tilde{K}} \quad & \text{tr}\left((Q + \tilde{K}^\top \Gamma_u R \Gamma_u \tilde{K}) Z_c\right) \\ s.t. \quad & BK = B\Gamma_u \tilde{K} \\ & 0 = A_c Z_c + Z_c A_c^\top + R_0. \end{aligned} \tag{7.14}$$

Proof. The proof is similar to Lemma 7.1 and is thus omitted. □

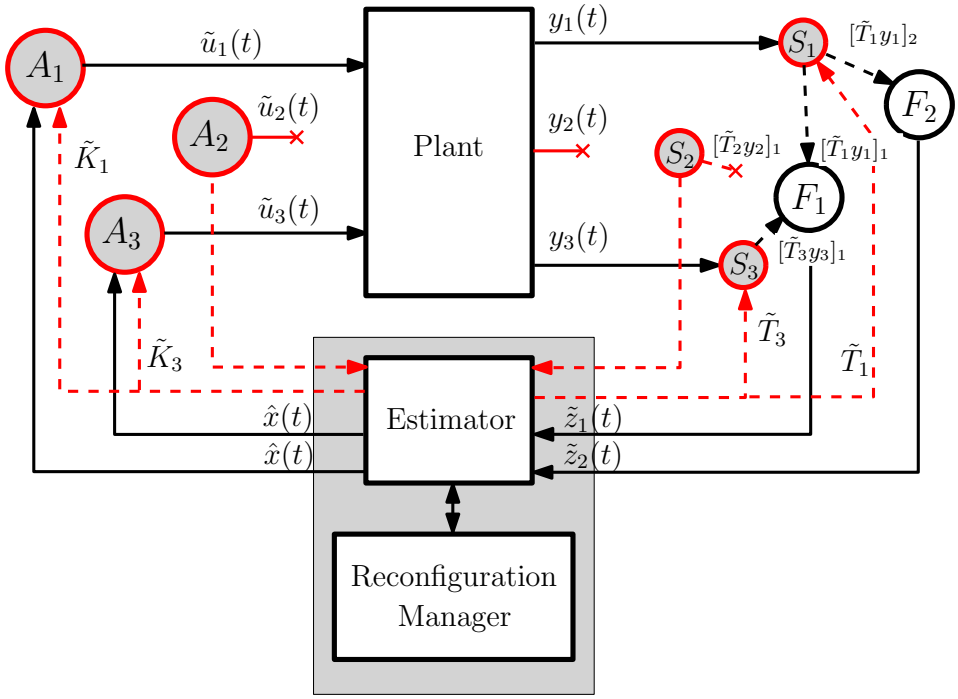


Figure 7.2: Networked control system with centralized sensor and actuator reconfiguration. Faults are reported by the sensors and actuators to the centralized estimator. Red dashed arrows represent the transmission of information related to faults. In the sensor reconfiguration case, the fault information may also be first transmitted from the centralized unit to the aggregators, and then from the aggregators to the sensors.

Proof of Proposition 7.2. Consider the optimization problem (7.14). Similar to the proof of Proposition 7.1, the variable Z_c is the unique solution to the Lyapunov equation given by the second equality constraint. The Lagrangian function for (7.14) is $\mathcal{L}(\tilde{K}, \Lambda) = \operatorname{tr}\Big((Q + \tilde{K}^\top \Gamma_u R \Gamma_u \tilde{K}) Z_c\Big) + \operatorname{tr}\Big(\Lambda^\top (BK - B\Gamma_u \tilde{K})\Big)$, where $\Lambda \in \mathbb{R}^{n \times n}$ represents the Lagrange multipliers. Moreover, the KKT optimality conditions are

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \tilde{K}} \mathcal{L}(\tilde{K}, \Lambda) = 2\Gamma_u R \Gamma_u \tilde{K} Z_c - \Gamma_u B^\top \Lambda \\
0 &= BK - B\Gamma_u \tilde{K}
\end{aligned}
$$

and can be rewritten as

$$
\begin{aligned}
0 &= \Gamma_u \tilde{K} - \frac{1}{2} \Gamma_u R^{-1} B^\top \Lambda Z_c^{-1} \\
0 &= (B\Gamma_u R^{-1} B^\top)^\dagger BK - \frac{1}{2} \Lambda Z_c^{-1}.
\end{aligned}
$$

Solving the above equations yields (7.13). □

Figure 7.2 depicts also a fault in the actuator network. A fault occurs at actuator A_2, which reports to the reconfiguration manager. The reconfiguration manager then solves (7.14) to derive the new controller $\tilde{K} = [\tilde{K}_1^\top \dots \tilde{K}_m^\top]^\top$. Then, $\tilde{K}_1$ is transmitted to to actuator A_1 and $\tilde{K}_3$ to actuator A_3, which allows them to compute and apply $\tilde{u}_1$ and $\tilde{u}_3$, respectively.

We note that the centralized actuator reconfiguration solution may be also obtained through other problem formulations. In (Härkegård and Glad, 2005) the authors propose to solve actuator redundancy through control allocation, formulated as an optimization problem and using the concept of virtual actuators. By appropriately choosing the objective function, the same solution (7.13) can be obtained. Moreover, the same result may be obtained using the pseudo-inverse method (Gao and Antsaklis, 1991; Staroswiecki, 2005b) if the matrix R has identical elements. Otherwise, a modification of the method, to take R into account, is required.

7.3 Distributed sensor and actuator reconfiguration

In this section, we propose a distributed algorithm to solve the reconfiguration problem. We begin by rewriting the centralized sensor and actuator reconfiguration problems in Lemmas 7.1 and 7.2 as quadratic optimization problems with a separable cost function and a global equality constraint.

Lemma 7.3. *Let* $\eta = \begin{bmatrix} \eta_1^\top & \dots & \eta_l^\top \end{bmatrix}^\top$, $\eta_i \in \mathbb{R}^r$ *and* $S \in \mathbb{R}^{l \times l}$ *be a diagonal matrix with non-negative entries. The sensor and actuator reconfiguration problems* (7.12) *and* (7.14) *can be rewritten in the following form:*

$$\begin{aligned} \min_{\eta_1,\dots,\eta_l} \quad & \sum_{i=1}^{l} S_{ii} \|\eta_i\|^2 \\ s.t. \quad & \sum_{i=1}^{l} H_i \eta_i = \omega \end{aligned} \tag{7.15}$$

where $H = \begin{bmatrix} H_1 & \dots & H_l \end{bmatrix}$, $H_i \in \mathbb{R}^{n^2 \times r}$ *and* $\omega \in \mathbb{R}^{n^2}$.

For the sensor case, $\tilde{T} = \begin{bmatrix} \eta_1 & \dots & \eta_p \end{bmatrix}$, $H = (C^\top \Gamma_y^\top) \otimes L$, $\omega = vec\,(LTC)$ *and* $S_{ii} = [\Gamma_y]_{ii} V_{ii}$.

In the actuator case, $\tilde{K} = \begin{bmatrix} \eta_1^\top & \dots & \eta_m^\top \end{bmatrix}^\top$, $H = (I \otimes B\Gamma_u)\, P_r^{-1}$ *with* $P_r \in \mathbb{R}^{mn \times mn}$ *being a permutation matrix such that* $vec\left(\tilde{K}\right) = P_r^{-1}\eta$, $\omega = vec\,(BK)$ *and* $S_{ii} = [\Gamma_u]_{ii} R_{ii}$.

Proof. The proof is given in the Appendix. □

The variables $\eta_i \in \mathbb{R}^r$ and $\omega_i \in \mathbb{R}^{n^2}$ have the following interpretation. For the case of sensor reconfiguration, each η_i represents the aggregation matrix $\tilde{T}$ components for the i-th sensor (i-th column of $\tilde{T}$), i.e., how sensor i transforms its information to be transmitted to each of the fusion nodes that it is connected to. In the same manner, each $\eta_i^\top$ corresponds to the i-th actuator state-feedback matrix $\tilde{K}$ components, i.e., the i-th row of $\tilde{K}$. The value of ω corresponds to the vectorization of the closed-loop estimator dynamics and closed-loop system dynamics before a fault occurs, for the case of sensor and actuator reconfiguration, respectively. This represents the quantity that ideally must be maintained by the combination of all sensor (actuator) nodes during the reconfiguration, which refers to the model-matching constraint.

The optimization problem (7.15) may be solved distributively using dual decomposition (Everett III, 1963; Johansson, 2008; Shor et al., 1985). A requirement is that the network remains connected when faults occur. The optimal solution to problem (7.15) is guaranteed to be achieved asymptotically using dual decomposition algorithms (Boyd et al., 2011). The main drawback is that the global equality constraint of the problem is only ensured asymptotically. Therefore, model-matching is not guaranteed at every iteration. Due to this fact, we later analyze the stability of the system under the distributed reconfiguration in Section 7.4.

To solve the dual optimization problem of (7.15) we resort to the distributed alternating direction method of multipliers (ADMM) algorithm (Boyd et al., 2011).

Theorem 7.3.1. *Define* $q_1, \ldots, q_l \in \mathbb{R}^{n^2}$ *such that* $\sum_{i=1}^{l} q_i = \omega$ *and local variables* $\zeta_1, \ldots, \zeta_l \in \mathbb{R}^{n^2}$. *Let*

$$\eta_i(k) = \frac{1}{2} S_{ii}^{-1} H_i^\top \zeta_i(k)$$

where $\zeta_i(k)$ *is computed by the following algorithm:*

$$\begin{aligned}
\zeta_i(k+1) &= \Big(\frac{1}{2} H_i S_{ii}^{-1} H_i^\top + \rho |\mathcal{N}_i| I\Big)^{-1} \Big(q_i - \rho \sum_{j \in \mathcal{N}_i} \mu_{i,(i,j)}(k) - \pi_{(i,j)}(k)\Big) \\
\xi_{i,(i,j)}(k+1) &= \alpha \zeta_i(k+1) + (1-\alpha)\pi_{(i,j)}(k), \\
\xi_{j,(i,j)}(k+1) &= \alpha \zeta_j(k+1) + (1-\alpha)\pi_{(i,j)}(k), \\
\pi_{(i,j)}(k+1) &= \frac{1}{2}\Big(\xi_{i,(i,j)}(k+1) + \mu_{i,(i,j)}(k) + \xi_{j,(i,j)}(k+1) + \mu_{j,(i,j)}(k)\Big), \\
\mu_{i,(i,j)}(k+1) &= \mu_{i,(i,j)}(k) + \xi_{i,(i,j)}(k+1) - \pi_{(i,j)}(k+1),
\end{aligned} \tag{7.16}$$

where $\rho > 0$ *is the step size,* $\alpha \in (0,2)$ *is a relaxation parameter,* $\rho\mu_{i,(i,j)}$ *is the Lagrange multiplier of node* i *associated with the constraint* $\zeta_i = \pi_{(i,j)}$, *and* $\xi_{i,(i,j)}$ *is an auxiliary variable private to node* i *associated with the edge* (i,j). *Then,* η_i *converges to the solution of* (7.15).

Note that the algorithm in Theorem 7.3.1 is distributed since it only requires communication between neighbors to exchange local values.

To prove Theorem 7.3.1, we first derive the dual form of (7.15).

Lemma 7.4. *Let $f_i(\eta_i) = \eta_i^\top S_{ii}\eta_i$. The optimization problem* (7.15) *can be rewritten in the following dual form:*

$$\begin{aligned} \min_{\{\zeta_i\},\{\pi_{(i,j)}\}} \quad & \sum_{i=1}^{l}\left(\frac{1}{4}\zeta_i^\top H_i S_{ii}^{-1} H_i^\top \zeta_i - q_i^\top \zeta_i\right) \\ s.t. \quad & \zeta_i = \pi_{(i,j)}, \quad \forall i \in \mathcal{V},\ j \in \mathcal{N}_i. \end{aligned} \tag{7.17}$$

Proof. The proof is given in the Appendix. □

Proof of Theorem 7.3.1. The value of $\eta(k)$ is obtained as $\eta(k) = \operatorname{argmin}_{x_i} f_i(x_i) - \zeta^T H_i x_i = \frac{1}{2}S_{ii}^{-1}H_i^\top \zeta_i(k)$. The ADMM algorithm (7.16) follows from (Boyd et al., 2011) and is thus omitted. □

The variables $q_i \in \mathbb{R}^{n^2}$ and $\zeta_i \in \mathbb{R}^{n^2}$ have the following interpretation. Vector q_i describes how the vectorization of the closed-loop dynamics, i.e. ω, is assigned among all nodes in the network. Note that the assignment is only constrained by the condition $\sum_{i=1}^{l} q_i = \omega$, thus admitting several solutions. For instance, one could have the closed-loop dynamics available only to one node, by having $q_1 = \omega$ and $q_j = 0$ for all $j \neq 1$. Variable ζ_i, only available at node i, is a local copy of the Lagrange multiplier associated with the model-matching constraint $H\eta = \omega$.

The following result indicates how the parameters q_i can be updated locally by the healthy nodes after a fault has occurred.

Lemma 7.5. *Let $j \in \mathcal{V}^f$ be an arbitrary faulty node, denote $\mathcal{J} \subseteq \mathcal{N}_j \cap \mathcal{V}_h$ as a subset of its healthy neighbors and assume $\mathcal{J}$ is not empty. Given the set $\{\bar{q}_i\}_{i\in\mathcal{V}}$ such that $\sum_{i\in\mathcal{V}} \bar{q}_i = \omega$, the set $\{q_i\}_{i\in\mathcal{V}}$ satisfying $\sum_{i\in\mathcal{V}_h} q_i = \omega$ can be computed as*

$$q_i = \begin{cases} \bar{q}_i, & i \notin \mathcal{J} \\ \bar{q}_i + \nu_i \bar{q}_j, & i \in \mathcal{J} \end{cases},$$

where $\nu_i \geq 0$ for all $i \in \mathcal{J}$ and $\sum_{i\in\mathcal{J}} \nu_i = 1$.

Proof. The proof is given in the Appendix. □

Notice that, since the sensor and actuator networks are disjoint, in the above scheme the update of q_i for a sensor (actuator) fault is performed within the sensor (actuator) network. The distributed reconfiguration algorithm can be summarized in Algorithm 7.1. An illustration of the distributed sensor and actuator reconfiguration is shown in Figure 7.3 where a fault occurs at sensor S_2 and actuator A_2 at $t = 0$. The sensors locally infer that sensor S_2 is no longer functioning, so sensors S_1 and S_3 reconfigure themselves. This is performed locally by each sensor computing the value of $\tilde{T}_1$ and $\tilde{T}_3$, and calculating $\tilde{T}_1 y_1$ and $\tilde{T}_3 y_3$. Each component $[\tilde{T}_i y_i]_j$

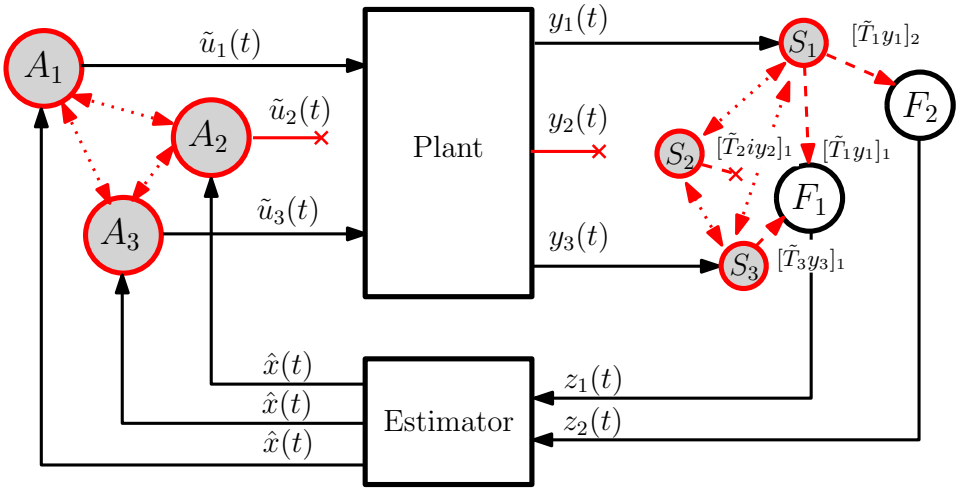


Figure 7.3: Networked control system with distributed sensor and actuator reconfiguration. Faults are detected by the sensors and actuators which are responsible for the reconfiguration. Reconfiguration is achieved through the communication among sensors and among actuators in a distributed manner through the sensor and actuator network, respectively.

Algorithm 7.1 Distributed sensor/actuator reconfiguration

1. Detect and isolate sensor/actuator faults and disconnect the faulty nodes at $t = 0$;
2. Locally compute q_i as per Lemma 7.5;
3. Compute the optimal solution ζ_i to the dual problem (7.17) using the algorithm in Theorem 7.3.1;
4. Compute the primal optimal solution $\eta_i(k) = \frac{1}{2} S_{ii}^{-1} H_i^\top \zeta_i(k)$;
5. Each sensor/actuator node i applies η_i.

is sent to the j-th aggregator, allowing each aggregator node to compute z_j and transmit this value to the controller node. Similarly, the actuators locally infer that actuator A_2 is faulty, so actuators A_1 and A_3 reconfigure themselves. This is a local operation where each actuator computes the value of $\tilde{K}_1$ and $\tilde{K}_3$.

7.4 Closed-loop stability under distributed reconfiguration

The proposed distributed algorithm converges to the optimum asymptotically as it solves the dual problem. Primal feasibility (model-matching), i.e., $H\eta(k) = \omega$, is only achieved in the limit. Therefore, one relevant concern is the system's stability when the dual algorithm is terminated in finite time. The following result shows that stability can be guaranteed in finite time, where the time will depend on the particular set of faults that have occurred.

Consider the general system $\dot{\upsilon} = \big((D + \Delta)\,\upsilon$ with D stable and uncertainty Δ, where $\mathrm{vec}\,(\Delta) = H\eta(k) - \omega$. For the sensor reconfiguration analysis, we have $\upsilon = \hat{x}$, $D = A_e$, $H = (C^\top \Gamma_y^\top) \otimes L$ and $\omega = \mathrm{vec}\,(LTC)$. Similarly, in the actuator reconfiguration case $\upsilon = x$, $D = A_c$, $H = (I \otimes B\Gamma_u)\, P_r^{-1}$ and $\omega = \mathrm{vec}\,(BK)$.

First we recall a necessary and sufficient condition for robust stability with bounded uncertainties.

Lemma 7.6 ((Lee et al., 1996))**.** *Consider the system $\dot{\upsilon} = \big((D + \Delta)\,\upsilon$ with D stable and uncertainty Δ. The system is stable for any norm-bounded uncertainty $\|\Delta\|_F \leq \delta$ with $\delta > 0$ if and only if there exists a positive definite matrix X such that*

$$D^\top X + XD + XX + \delta^2 I \prec 0.$$

Theorem 7.4.1. *Consider the sequence of vectors $\{\eta(k)\}$ converging to $\eta^\star \in \mathcal{H} = \{\eta : H\eta = w\}$ and define $\Delta(k)$ such that $vec\,\big(\Delta(k)\big) = H\eta(k) - w$. Suppose there exist matrices $X \succ 0$ and $M \succ 0$ satisfying the matrix equation $D^\top X + XD + X^2 + M = 0$ and a positive decreasing function of k, $\epsilon(k) > 0$, such that $\|\Delta(k)\|_F \leq \epsilon(k)\|\Delta(0)\|_F$ holds for all k. Let $\bar{k}$ be an integer for which the following inequality holds:*

$$\epsilon(\bar{k}) < \frac{\sqrt{\lambda_{\min}(M)}}{\|H\eta(0) - w\|}.$$

Then, the system under faults with dynamics given by $\dot{\upsilon} = \big(D + \Delta(k)\big)\,\upsilon$ is stable for $k \geq \bar{k}$.

Proof. Suppose that $\|\Delta(k)\|_F \leq \epsilon(k)\|\Delta(0)\|_F$ and consider $\delta(k) = \|\Delta(k)\|_F$. From Lemma 7.6, the closed-loop system at time k is guaranteed to be stable if $D^\top X + XD + X^2 + \delta(k)^2 I = -M + \delta(k)^2 I \prec 0$, which is equivalent to $\delta(k) < \sqrt{\lambda_{\min}(M)}$. Note that the latter is ensured for $\bar{k}$ when $\epsilon(\bar{k})\delta(0) < \sqrt{\lambda_{\min}(M)}$. Since $\epsilon(k)$ is decreasing with k, concludes the proof. □

The above result provides a method to terminate the dual algorithm while ensuring stability. It only requires knowledge of the convergence properties of the dual algorithm, namely the function $\psi(k)$, and the initial distance $\|\Delta(0)\|_F$. The latter can be computed at the beginning, since it only depends on the initial condition of the algorithm and the nominal controller. Furthermore, note that a zero initial condition of the dual algorithm yields $\delta(0) = \|\omega\|$, which can be made locally available

to each agent. Convergence properties of dual algorithms are readily available in the literature, see (Ghadimi et al., 2014; Nedic et al., 2010). Next we apply the results of Theorem 7.4.1 to the ADMM algorithm for the distributed reconfiguration problem formulated in Theorem 7.3.1.

Lemma 7.7. *Consider the optimization problem* (7.15)*, its equivalent dual formulation* (7.17)*, and the ADMM algorithm described in Theorem 7.3.1. Let* $\zeta^\star = \lim_{k\to\infty}\zeta(k)$ *be the optimal solution to* (7.17)*. Then, we have* $\|\zeta(k)-\zeta^\star\| \leq \psi\|\zeta(k-1)-\zeta^\star\|$ *for all* k *with* $\psi \in [0\ 1)$*.*

Proof. The proof follows directly from (Ghadimi et al., 2014, Theorem 1), where the decay rate ψ can be found. □

Theorem 7.4.2. *Consider the optimization problem* (7.15)*, its equivalent dual formulation* (7.17)*, and the ADMM algorithm described in Theorem 7.3.1. The closed-loop system obtained at time* k *from* $\eta(k)$ *is guaranteed to be stable for all* $k \geq \bar{k}$ *with*

$$\bar{k} = \left\lceil \frac{\log(\sqrt{\lambda_{\min}(M)}) - \log\left(\|H\eta(0) - \omega\|\kappa(HS^{-1}H^\top)\right)}{\log(\psi)} \right\rceil.$$

Proof. We have $H\eta(k) = -1/2HS^{-}1H^\top\zeta(k)$ for all k. Furthermore, we can derive the following bound $\|H\eta(k) - H\eta^\star\| = \|1/2HS^{-1}H^\top(\zeta(k) - \zeta^\star)\| \leq \|1/2HS^{-1}H^\top\|_2\|(\zeta(k) - \zeta^\star)\|$. Using Lemma 7.7, we have

$$\begin{aligned}\|H\eta(k) - H\eta^\star\|_2 &\leq \|1/2HS^{-1}H^\top\|_2\psi^k\|(\zeta(0) - \zeta^\star)\| \\ &\leq \kappa(HS^{-1}H^\top)\psi^k\|H\eta(0) - H\eta^\star\|.\end{aligned}$$

Recalling that $\|\Delta(0)\|_F = \|H\eta(0) - w\| = \|H\eta(0) - H\eta^\star\|$ and applying Theorem 7.4.1, we observe that the closed-loop system is stable for all k such that

$$\psi^k < \frac{\sqrt{\lambda_{min}(M)}}{\|H\eta(0) - H\eta^\star\|\kappa(HS^{-1}H^\top)}.$$

The proof concludes by taking the logarithm of both sides and rearranging the terms. □

Next, we compute the matrices X and M that maximize the magnitude of the uncertainty for which stability is ensured.

Proposition 7.3. *Denote $X^\star$ and $\sigma^\star$ as the optimal solution to the convex optimization problem*

$$\begin{aligned} \max_{X,\,\sigma} \quad & \sigma \\ s.\ t. \quad & \sigma > 0 \\ & X \succ 0 \\ & -\sigma I \succ D^\top X + XD \\ & 0 \prec \begin{bmatrix} -D^\top X - XD - \sigma I & X \\ X & I \end{bmatrix}. \end{aligned} \tag{7.18}$$

Then, matrix $X^\star$ satisfies the robust stability constraint $D^\top X + XD + X^2 + \delta^2 I \prec 0$ with $\delta^2 = \sigma^\star$ being the largest disturbance magnitude for which stability is ensured by Proposition 7.6. Additionally, we have that the optimal matrix M is given by $M^\star = -D^\top X^\star - X^\star D - X^{\star 2} \succ 0$.

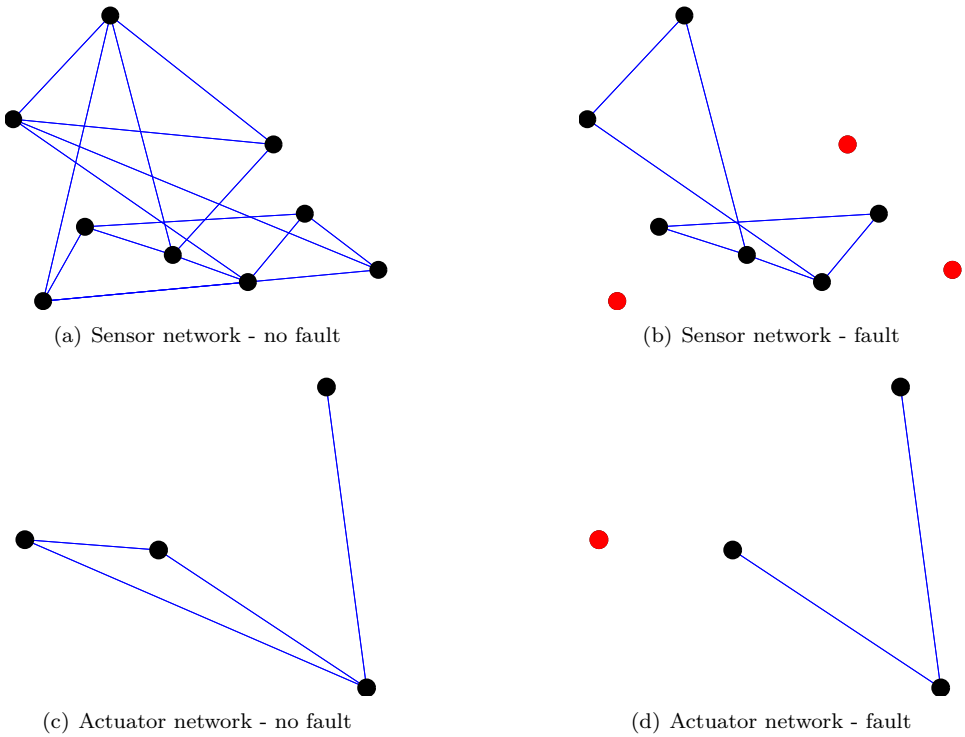
Proof. Note that the largest disturbance magnitude δ for which stability is ensured by Lemma 7.6 can be computed as

$$\begin{aligned} \max_{X \succ 0,\, \delta^2 > 0} \quad & \delta^2 \\ \text{s.t.} \quad & 0 \succ D^\top X + XD + XX + \delta^2 I. \end{aligned}$$

Applying the Schur complement to $-D^\top X - XD - \sigma I - XX \succ 0$ and denoting $\sigma = \delta^2$, the latter optimization problem can be rewritten as (7.18). □

We remark that since Lemma 7.6 used in Theorem 7.4.2, provides a conservative stability guarantee, the obtained $\bar{k}$ is expected to be conservative. This will be later illustrated in the numerical example.

The value $\bar{k}$ assures that stability can be achieved in a finite time. Its calculation can be efficiently performed in a centralized manner, while a distributed computation would require the knowledge of the particular set of faults by all nodes. In a centralized manner, the fault information can be propagated to the node responsible for this calculation, which would then flood the value $\bar{k}$ through the network. One distributed solution is for nodes to flood the complete network with the fault information, and each node to perform the computation of $\bar{k}$. The distributed approach obviously incurs in a large amount of communication among nodes for large networks. Nevertheless, a possible practical approach is for a node i to apply the computed solution $\eta_i(k)$ whenever the node itself and its neighbors are close to stationarity, i.e., $|\eta_i(k) - \eta_i(k)| \leq \varepsilon$, for a sufficiently small $\varepsilon > 0$. The intuition is that since the algorithm guarantees model-matching, and hence stability, asymptotically, when the increments are sufficiently small, this means that the optimal solution is close. This can be easily implemented in practice as follows. All nodes transmit a flag in a beacon which is periodically transmitted whenever reconfiguration takes place. This flag is set to 1 if the bound $|\eta_i(k) - \eta_i(k)| \leq \varepsilon$, or 0 otherwise. Then, the



(a) Sensor network - no fault

(b) Sensor network - fault

(c) Actuator network - no fault

(d) Actuator network - fault

Figure 7.4: Sensor and actuator network graph. The healthy nodes are colored black and the faulty nodes are colored red.

nodes stop running the reconfiguration algorithm if all its neighbors have a value of 1, i.e., commonly known as max-consensus on the flag variable.

We remark that in the case model-matching is not achievable, the distributed sensor and actuator reconfiguration algorithm still converges to (7.11) and (7.13), respectively. Thus, one is able to know a priori what will be the behavior of the system according to a specific set of faults. This information can potentially be utilized for the offline design of aggregator matrices T and state-feedback matrices K. Such matrices could then be stored at the sensors and actuators and be used whenever specific faults occur. Additionally, these matrices could also be designed to guarantee the stability of the system under all possible faults, which, due being a conservative approach, could possibly lead to reduced control and estimation performances.

7.5 Numerical example

We now provide a numerical example in order to validate the proposed distributed reconfiguration method. The aim is to control the temperature dynamics in two adjacent rooms, where 9 sensors are deployed to measure the temperature and 4 heaters actuate the system. The system dynamics, measured outputs and aggregated outputs are given by (7.1), (7.2) and (7.3), respectively, where T is fixed and $V = 0.5I$ and $W = 2I$. Moreover, the state estimate and control input are given by (7.5) and (7.4), respectively. The initial estimation gain L and control input gain K are the solutions to the LQG controller design problem. The ADMM parameters in (7.16) are set to $\rho = 1$ and $\alpha = 1.5$. We note that methods exist to choose the ADMM parameters ρ and α to increase the convergence speed of the algorithm (Ghadimi et al., 2014). These methods are optimal if executed in a centralized manner, but sub-optimal distributed methods are also provided in (Ghadimi et al., 2014).

The sensor network graph is given in Figures 7.4(a) and 7.4(b) while the actuator network is depicted in Figures 7.4(c) and 7.4(d), for the nominal and faulty cases.

We start by analyzing the performance of the distributed reconfiguration scheme presented in Section 7.3 for the sensor and actuator faults depicted in Figure 7.4. As performance indicators, we consider the normalized objective function errors $|J_e(k)-J_e^\star|$ and $|J_c(k)-J_c^\star|$, the errors in the model-matching constraint $\|H^e\eta(k)-w^e\|$ and $\|H^c\eta(k)-w^c\|$ and the maximum eigenvalues of $A_e = A - L\tilde{T}(k)\Gamma_y C$ and $A_c = A - B\Gamma_u\tilde{K}(k)$. The results are depicted in Figure 7.5. As it can be seen, the distributed method asymptotically achieves the optimal cost and guarantees the model-matching constraint. Moreover, the state estimation error dynamics is unstable for the first 2 steps, i.e., $\lambda_{\max}(A_e(k)) > 0,\ k = \{1, 2\}$, while the closed-loop dynamics are unstable for only the first step since $\lambda_{\max}(A_c(k)) > 0,\ k = \{1\}$. Applying Theorem 7.4.2 from Section 7.4, we achieve that A_e is stable after $\bar{k} = 53$ steps and A_c is stable after $\bar{k} = 8$ steps. Since Lemma 7.6 used in Theorem 7.4.2, provides a conservative stability guarantee, the obtained $\bar{k}$ is expected to be conservative. The distributed sensor reconfiguration takes 15 steps to converge to $|J_e(k) - J_e^\star| < 10^{-3}$ and $\|H^e\eta(k)-w^e\| < 10^{-1}$. Similarly, the distributed actuator reconfiguration takes approximately 16 steps to converge.

The time-responses of the distributed sensor and actuator reconfiguration under the faults in Figure 7.4 are depicted in Figure 7.6 where are shown the state trajectories, the estimation error and the control input values. In Figure 7.6 we depict the case where the sensor and actuator detection, isolation and reconfiguration is assumed to take place instantaneously (solid line) and the case where the sensor and actuator detection and isolation is instantaneous but the reconfiguration is performed in real-time (solid-star line). In the latter case, each step of the reconfiguration is set to take 1 s to run, which includes both computation and communication time. Such large time is selected so one can analyse the impact of a slow real-time reconfiguration in the system dynamics. However, in practice, the computation and communication times can be greatly reduced. This case aims at

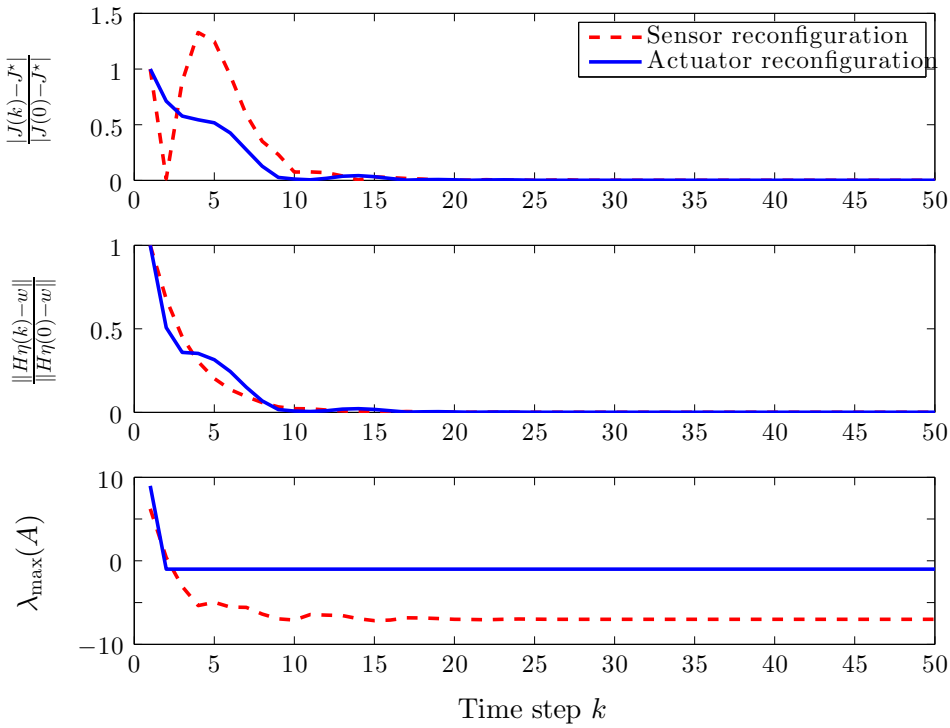


Figure 7.5: Performance of the distributed sensor and actuator reconfiguration method for the networks depicted in Figure 7.4.

demonstrating the impact of applying the reconfigured output, before the reconfiguration algorithm has converged to a stable region, which takes at least, $\tau = 3\,\mathrm{s}$ for the sensor reconfiguration and $\tau = 1\,\mathrm{s}$ for the actuator reconfiguration. Additionally, we depict the case where reconfiguration does not take place (dashed line). The sensor faults occur at time $t = 10\,\mathrm{s}$ and the actuator faults at $t = 300\,\mathrm{s}$. As it can be seen, the system performance greatly deteriorates when reconfiguration is not performed. When the actuator reconfiguration is not instantaneous, a slight loss of performance (maximum deviation of 0.1°C) occurs in the first 1 s, but is recovered afterwards.

7.6 Experimental setup

An experimental evaluation of the distributed reconfiguration method was conducted on the room heating testbed, described in Chapter 2.8 and depicted in Figure 2.8. We now describe the control system setup, the wireless network and the

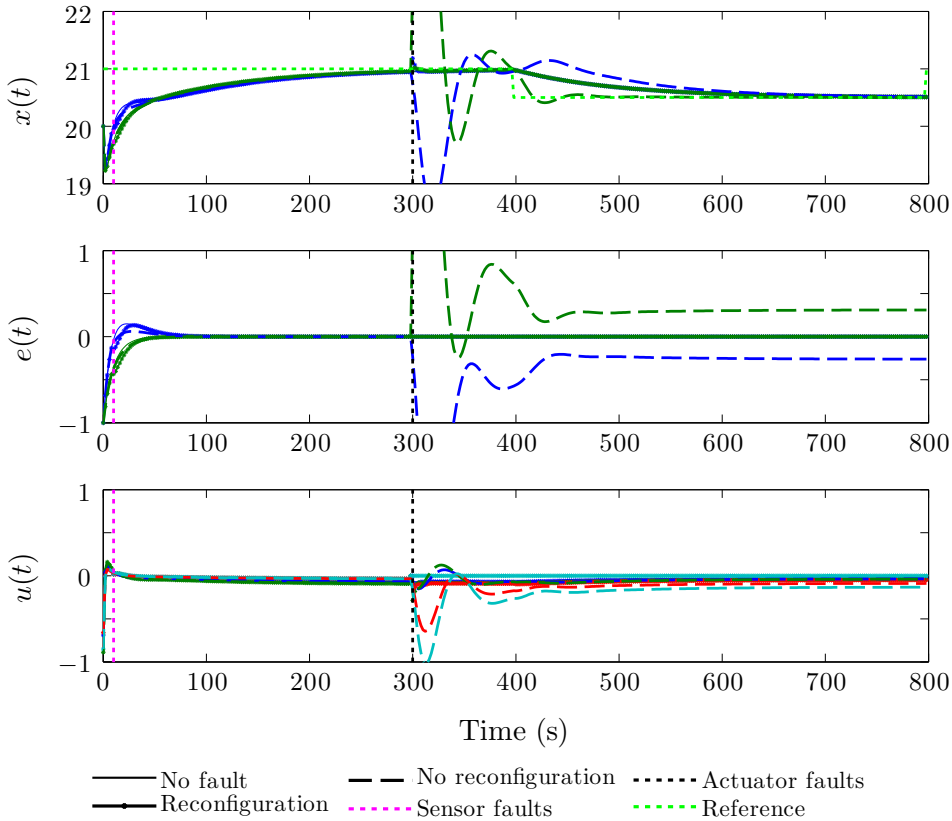


Figure 7.6: Time-response of the state and estimation error trajectories and control input for the distributed sensor and actuator reconfiguration in Figure 7.4. Reference value to be tracked is depicted by the black dotted line. Sensor faults occur at time $t = 10\,\mathrm{s}$ and actuator faults at $t = 300\,\mathrm{s}$. Instantaneous reconfiguration (solid), real-time reconfiguration (solid-star) and no reconfiguration (dashed), are compared.

implementation of the distributed reconfiguration algorithm in this scenario.

7.6.1 Control system

In this setup, the wireless temperature sensor transmits the temperature value to the wireless nodes of each heater A, B and C every $T_s = 2\,\mathrm{s}$. Hence, the sensor message acts as a synchronization message to all the heaters.

As in Chapters 5 and 6, the state of the system is augmented with an integral state so that piecewise constant references can be tracked. Hence, the complete state

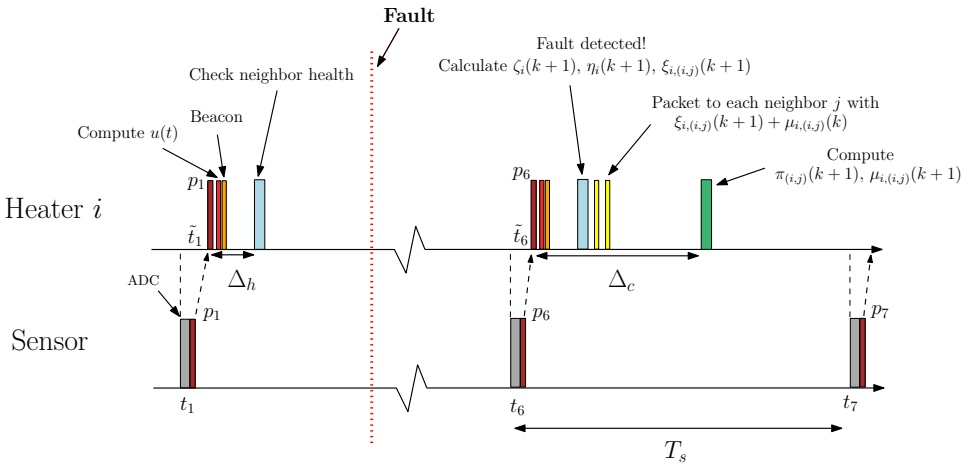


Figure 7.7: Sensor and heater operation in the room heating testbed under distributed reconfiguration.

becomes $x(t) = \begin{bmatrix} \tilde{x}(t) & x_c(t) \end{bmatrix}^\top$, where $\tilde{x}(t)$ is the temperature in the room and given by (2.1) and the integral is approximated by the discrete-time computation at each heater every T_s s as

$$x_c(k+1) = x_c(k) + T_s \left(r(k) - \tilde{x}(k) \right),$$

and $r(k)$ is the reference value. Even though the measurement transmission and the computation of the integral state is performed in a discrete-time manner, if T_s is selected to be much smaller than the temperature dynamics, they provide a good approximation of the continuous-time model. The full LQR controller gain K is then designed by utilizing the continuous-time linear time-invariant model identified in Chapter 2.8 and the cost function parameters are set to

$$Q = \mathrm{diag}(0.1, 0.001), \;\; R = \mathrm{diag}(550, 250, 500).$$

The controller gain K_i is implemented at each heater. The reference value is set to change between 24°C and 22°C, every 30 min and this value is transmitted to the heaters. Additionally, the reconfiguration Algorithm 7.1 is implemented on the wireless node of each heater.

The complete operation procedure of the sensor and a each heater is shown in Figure 7.7. After the temperature measurement is received at heater i, the wireless device computes $u(t)$. The latest $u(t)$ value is used as the duty-cycle of the PWM signal, which is refreshed every 2 min.

7.6.2 Wireless network

All wireless devices utilize the CSMA/CA MAC included in TinyOS. Moreover, as the nodes are all located in the same room, a fully connected network is formed between the wireless actuators.

7.6.3 Fault detection and distributed reconfiguration algorithms

The fault detection and distributed reconfiguration algorithm operation in the wireless heater is illustrated in Figure 7.7. After the sensor message is received by each wireless heater, a beacon is broadcasted by each healthy heater. Therefore, we assume that a faulty heater does not have the capability to transmit the beacon. After $\Delta_h = 0.5\,\text{s}$, each heater verifies if any neighbor has failed, by analyzing the beacons received. In order to avoid false fault detections, as the detection algorithm is based on the reception of a wireless beacon, a neighbor node j is set as faulty if a total of 5 consecutive beacons from node j do not arrive at node i. If a fault is detected by node i, node i sets a fault flag to 1 in its beacon, so that it can inform its neighbors that a fault has occurred. If a fault then occurs, as illustrated in Figure 7.7, each heater calculates $\zeta_i(k+1)$ and $\eta_i(k+1)$, as well as $\xi_{i,(i,j)}(k+1)$ for each neighbor j. Then, a packet is transmitted to each neighbor j, with its content being the value $\xi_{i,(i,j)}(k+1) + \mu_{i,(i,j)}(k)$, to be used in the calculation of $\pi_{(i,j)}$. The calculation of $\pi_{(i,j)}$ and $\mu_{i,(i,j)}$, takes place at $\tilde{t}_k + \Delta_c$, where $\Delta_c = 1.5\,\text{s}$. The condition $\|\eta_i(k) - \eta_i(k+1)\| \leq 10^{-3}$ is evaluated for each element of vector η_i each node i. If the condition is verified, node i sets the fault flag to 0 to be transmitted in the future beacons. The distributed reconfiguration algorithm is stopped at node i if the fault flag at node i is 0 and if all neighbors j have set their fault flag to 0.

7.7 Experimental results

We now provide the experimental results for the case when heater B fails. We evaluate the case when no reconfiguration takes place and the case when the system runs the distributed reconfiguration algorithm.

The time-response of the system with and without reconfiguration is depicted in Figure 7.8 and the analysis of specific parameters of the system are shown in Figure 7.9. Heater B (in red for both scenarios) becomes faulty at $t = 270\,\text{s}$ for the case when reconfiguration takes place, while it becomes faulty at $t = 442\,\text{s}$ when no reconfiguration occurs. The fault detection took 11 s to be detected and the reconfiguration algorithm took 7 steps for $\|H^c\eta(k) - w^c\| \leq 10^{-4}$. As the algorithm is run at the same frequency as the sensor measurement updates every 2 s, the reconfiguration took 14 s to converge. For the system with no reconfiguration, the temperature dropped by slightly more than 1 °C after the fault occurs, while in the system with reconfiguration no change occurs. The single difference is that the variability of the temperature dynamics around the set-point is decreased as 2 heaters are used instead of 3. Additionally, in the case of no reconfiguration,

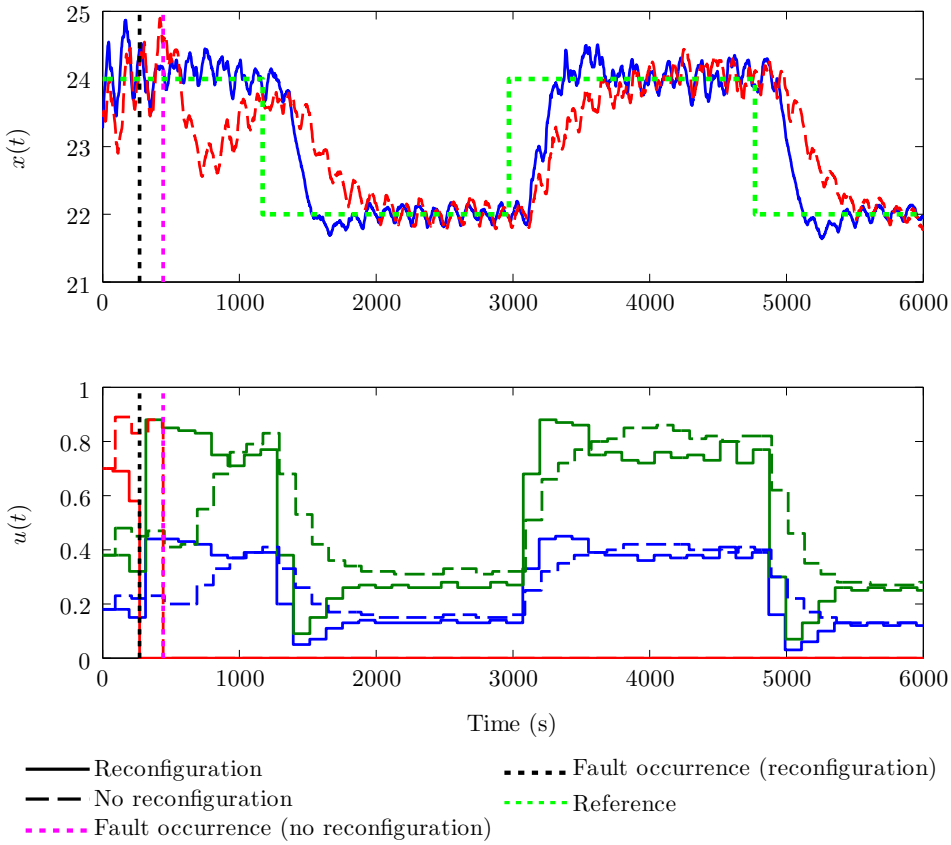


Figure 7.8: Time-response of the room heating experiment with and without reconfiguration. The temperature dynamics $x(t)$ and control input values $u(t)$ for each of the heaters.

the closed-loop dynamics change, becoming significantly slower and increasing the settling time by approximately 400 s. This is confirmed by Figure 7.9, where we see that, when no reconfiguration occurs, the value of $\lambda_{\max}(A_c)$ is modified. However, when reconfiguration takes place, this value only suffers a deviation for the first 25 s (11 s of fault detection plus 14 s of reconfiguration), recovering to the pre-fault value afterwards. Note that the model-matching constraint is not guaranteed during the same period, as depicted in Figure 7.9. Additionally, applying Theorem 7.4.2 from Section 7.4, we achieve that A_c is stable after $k \geq \bar{k} = 6$ steps. However, in the experiment, A_c is stable for all $k > 0$.

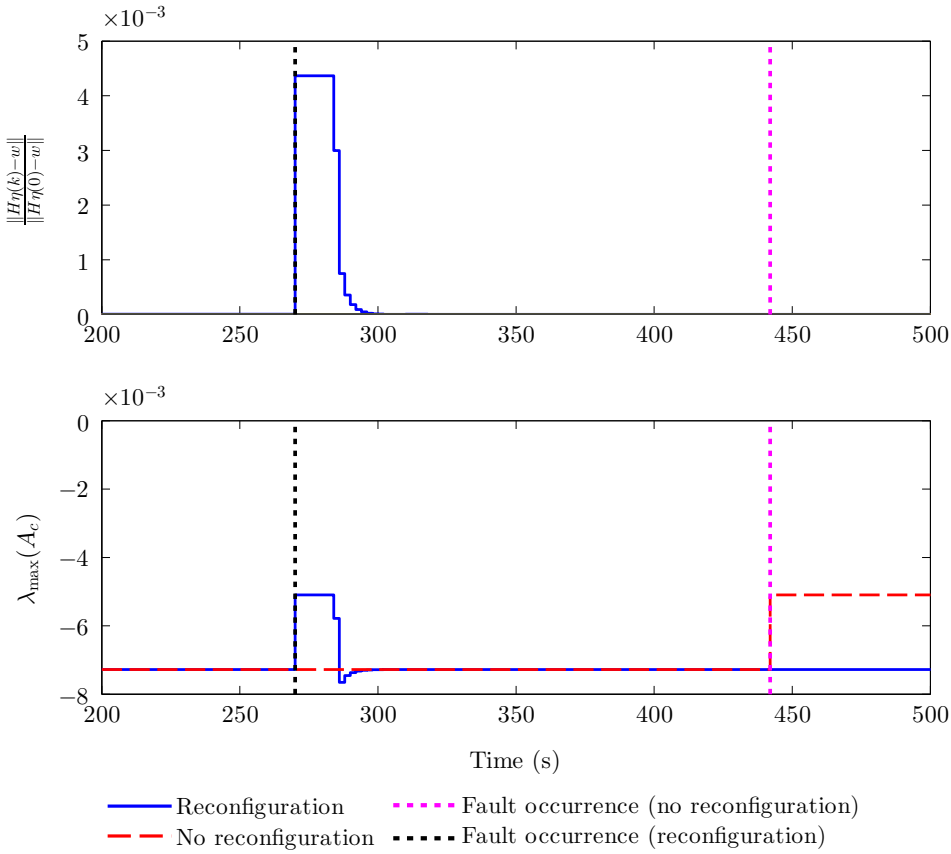


Figure 7.9: Analysis on the model-matching condition with reconfiguration and $\lambda_{\max}(A_c)$ with and without reconfiguration.

7.8 Summary

In this work, we developed a distributed reconfiguration method for networked control systems under sensor and actuator faults. The proposed approach guarantees a model-matching reconfiguration while minimizing the steady-state estimation error covariance and a linear-quadratic control cost. The distributed reconfiguration method is guaranteed to achieve the same solution as the centralized reconfiguration, while only requiring local communication between healthy sensors and actuators. The approach is illustrated through a numerical example and evaluated in a room heating testbed using heaters with wireless and computation capabilities. A fault detection mechanism and a distributed reconfiguration operation procedure were devised and implemented for the wireless NCS setup. The results show that

the distributed reconfiguration algorithm allows the wireless actuators to quickly detect and recover from faults, and achieve the reconfiguration goals.

Chapter 8

A tool for implementation and validation of wireless networked control systems

The integration of wireless communication in networked control systems (NCSs) such as in process automation, building automation and intelligent transportation systems, poses many challenges and has become an area of extensive research (Samad et al., 2007; Åkerberg et al., 2011; Kim and Kumar, 2012). Wireless NCSs are complex systems characterized by complex interactions between sensors, controllers, actuators, the communication medium and the physical processes being controlled. The correct design, implementation, and validation of NCSs is recognized to be an extremely hard task, requiring deep knowledge of communication, computation and control (Kim and Kumar, 2012; Poovendran et al., 2012). Simulators are an important tool for evaluating various wireless NCS designs and for developing a thorough understanding of the interactions among the numerous components in wireless NCSs. In this chapter, we introduce GISOO (**G**raphical **I**ntegration of **S**imulink and C**OO**JA), a co-simulator designed to provide realistic simulations of all the components of a wireless NCS. GISOO integrates Simulink (MathWorks, 2014) and COOJA (Osterlind et al., 2006). Simulink is one of the most widely used tools to design and study control systems by control engineers, while COOJA is a comprehensive wireless sensor network (WSN) simulator that provides simultaneous cross-level simulation at application, operating system and machine code level in a single framework. COOJA is specifically designed to simulate and emulate WSNs using realistic wireless radio models (Eriksson et al., 2009). Additionally, the COOJA simulator is flexible in the sense that all levels of the system can be modified or replaced, e.g, sensor node platforms, operating system software, radio transceivers, and radio propagation models. Moreover, the simulator is able to emulate wireless devices in both Contiki OS (Dunkels et al., 2004) and TinyOS (Levis et al., 2004), which are the most widely used operating systems for wireless sensor networks.

GISOO supports a general wireless NCS architecture as discussed in Chapter 1 and depicted in Figure 1.9. The wireless NCS consists of several wireless sensors and

actuators co-located with a single or multiple physical systems, communicating with a set of controllers through a wireless network. In this architecture the controllers may be centralized or distributed and can be implemented on wireless devices or dedicated computers. The wireless network uses the IEEE 802.15.4 standard as the physical layer and is open for any specification of the other communication layers. Particularly, MAC and routing protocols may be designed by the user or available protocols may be used off-the-shelf. Network transmissions may be scheduled by a centralized networked manager or decided locally by each individual node in the network.

The virtual testbed that we propose has the following features:

a) runs real embedded software including the full wireless communication stack, combining medium-access control, routing and application layer, emulating the interactions between the device's hardware and software,

b) allows embedded wireless communication code to be emulated without any changes, so that the same code that has been evaluated in simulations can be executed directly on the target platform,

c) provides full flexibility in the implementation of the NCS architecture, where algorithms may be implemented directly in wireless devices or in Simulink,

d) allows a comprehensive analysis and validation of the interactions of communication, computation and control components of the NCS and,

e) supports various widely adopted wireless platforms in both TinyOS and Contiki operating systems through COOJA.

The software, manual and examples for the co-simulator are available online (GISOO, 2014). As of the 24$^{\text{th}}$ of September 2014, the website has been visited[1] 1245 times by 461 users in 60 different countries, since it was launched in October 2013.

The rest of the chapter is organized as follows. In Section 8.1 we present GISOO's architecture, including details on the method implemented to allow synchronization and the interaction between Simulink and COOJA. Section 8.2 discusses GISOO's main features and tools. In Section 8.3 several examples illustrate how GISOO can be applied study different wireless NCS scenarios. Finally, Section 8.4 concludes the chapter.

8.1 Architecture

In this section, we discuss the co-simulator architecture. The two main components of the co-simulator are the wireless devices and the physical system model. The

[1] Statistics obtained from Google Analytics

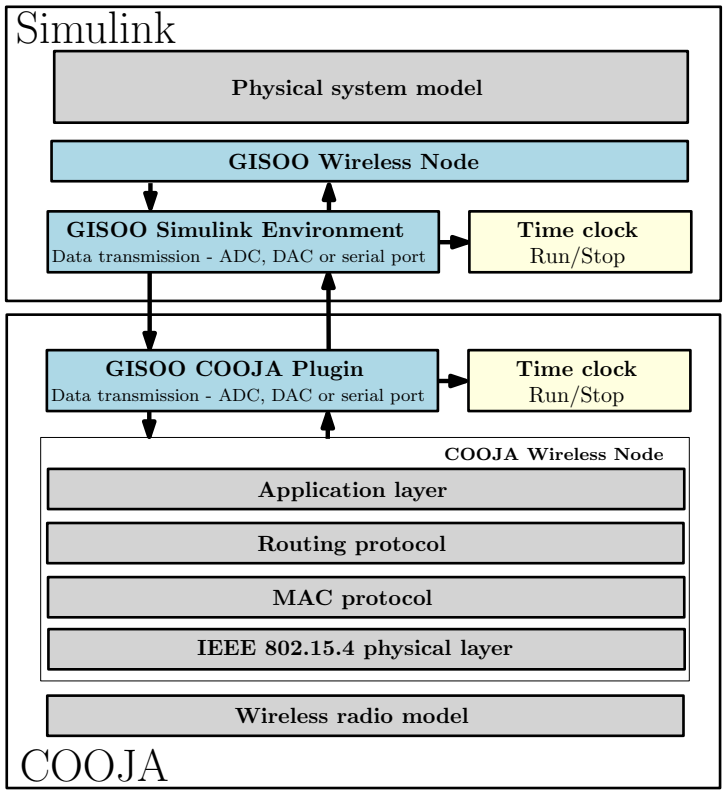


Figure 8.1: The GISOO architecture.

wireless devices are emulated in COOJA, and can communicated with other wireless devices through the radio environment in COOJA, and the physical system is modelled in Simulink. Our work has focused on developing a COOJA plugin and a Simulink environment to combine these two softwares. The GISOO architecture is illustrated in Figure 8.1. The interaction between the wireless devices and the physical world is performed through the available communication interfaces and input/output (I/O) ports of the wireless device. The architecture of the wireless devices considered is discussed in Chapter 2.8 and illustrated in Figure 2.2(b). The main design goal of the co-simulator is to enable the embedded wireless software to be used in both GISOO and in a real experimental scenario without requiring any modifications. Therefore, the issuing of the Analog-to-Digital converter (ADC), Digital-to-Analog converter (DAC), the serial port (UART/SPI/I2C) and/or other I/O ports in the wireless device to interact with the physical system, must be handled by the co-simulator. This is achieved by the GISOO COOJA Plugin (GCP) implemented in COOJA which monitors any calls made by the native ADC, DAC,

serial port or I/O ports in the wireless devices. Whenever these functions are called, the GCP is responsible for exchanging data with the GISOO Wireless Nodes (GWN) through the GISOO Simulink Environment (GSE), located in Simulink. This allows for data in Simulink to be sent to the wireless device in COOJA, and data from the wireless device to be sent to Simulink.

The synchronization between the two softwares is of major importance. Synchronization is maintained by a stop-and-run mechanism where the GCP in COOJA controls the time clock. This is a similar mechanism to the one implemented in the PiccSIM toolchain (Björkbom et al., 2011). The operation of the simulator under this synchronization mechanism is as follows. The communication over ADC, DAC or other communication ports is caught by the GCP in COOJA, which generates an event with a time stamp, that is transmitted to the GWN at the GSE in Simulink. The COOJA simulation is suspended until the computation in Simulink has finished. Simulink then uses the time stamp of the event to synchronize its clock, simulate the physical system until the time of this event and perform the requested action. The action can be to either read, e.g., the ADC port, or to write, e.g., the DAC port. If a read action is issued, the requested data from Simulink is returned through the GWN and received by the GCP in COOJA. When a write action is issued, an acknowledgement is returned from Simulink to the GCP in COOJA. After this is complete, the COOJA simulation is resumed.

In the co-simulator, the controller algorithm for the specific control system may be implemented in a wireless node but also in Simulink, using appropriate Simulink blocks. In the former case, after the computation of a new control input, the controller node transmits this data to a wireless actuator node through the wireless network. In the latter scenario, the sensor data is communicated to the Simulink controller block by a wireless base station/controller node through the serial communication bus. This scenario occurs in practice when a controller is implemented in dedicated computer which interfaces the wireless network through a wireless basestation connected to the computer's USB port. Such an experimental setup is used in Chapter 4. The advantage of allowing for controller design in Simulink is that changes in the control algorithm do not require a modification and recompilation of the wireless devices code. This significantly increases the speed of evaluation of new controller designs.

8.2 Features and tools

We now present some of the main features and tools that are available in the co-simulator.

8.2.1 Wireless channel models

Various radio medium models are currently available in COOJA. While new models can be easily added, existing models may also be modified. The following models are available in the current version of COOJA:

- Unit disk graph medium (UDGM) - constant loss: a simple model where the transmission range of a node is modelled as a disk, where all nodes inside the disk receive packets while nodes outside the disk cannot receive packets from the transmitting node. The maximum transmission range of each node can be specified. Moreover, interference occurs for transmissions taking place at the same time, and interfering packets are lost.

- Unit disk graph medium (UDGM) - distance loss: this model is an extension of the previous model, but where the packet reception depends on a success probability parameter which is specified by the user, and the log-distance path loss model (Goldsmith, 2005) is used.

- Directed graph radio medium (DGRM): in this model, asymmetric per-link packet reception success probability can be specified. Moreover, propagation delays can be defined per link and the loss is modelled by the log-distance path loss model.

- Multi-path ray-tracer medium (MRM): this is a Signal-to-Noise Ratio (SNR) model which utilizes the ray tracing technique modelled in 2D, where specified objects are attenuators (Goldsmith, 2005). Environment maps may be imported and used in the simulation. Given a 2D environment map, refractions, reflections and diffractions are calculated and define the range of the transmitted signal. Additional parameters such as the transmitted power, packet lengths and antenna gains can be specified. If a signal arriving at a node is stronger than the sum of all other signals at the node by at least the SNR ratio, the node can properly receive the signal.

The above models are typical models utilized in WSN simulations (Goldsmith, 2005), being also part of TOSSIM (Levis et al., 2003). In (Halkes and Langendoen, 2010) the authors propose an experimental validation of the UDGM - constant loss model and the MRM model in an office building, under different MAC protocols. The MRM model is shown to give results close to the real-world results with respect to packet loss and transmission delay. However, the simple UDGM significantly deviated from the experimental results obtained. In (Strübe et al., 2014), the authors evaluated the UDGM - distance loss model in an office building as well, giving results close to reality. A new radio model was developed and implemented in (Zinonos et al., 2012), which validates the radio model with real radio data from an oil refinery in Portugal.

8.2.2 Communication protocols and devices

Several MAC and routing protocols are currently available in Contiki and TinyOS. We highlight the IEEE 802.15.4 MAC which is the current MAC standard for WSNs (IEEE 802.15.4, 2006; Hauer, 2009; Hernandez and Park, 2011), the routing protocol for low power networks (RPL) proposed by the internet engineering

task force (IETF) (Winter et al., 2012; Ko et al., 2011b) and the Collection Tree Protocol (CTP) (Gnawali et al., 2009). Additionally, many of the protocols surveyed in Chapter 2 are implemented in these operating systems, e.g., the Priority-MAC and a simplified implementation of WirelessHART (Shen et al., 2014, 2013) and tournament-based MACs (Ramesh et al., 2014; Pereira et al., 2007; Christmann et al., 2014). Moreover, several wireless sensor platforms, besides the Tmote sky/Telosb platform utilized in this thesis, are supported by both Contiki and TinyOS (Contiki, 2014; Levis et al., 2004).

8.2.3 Additional features

The co-simulator allows for the usage of the standard features, toolboxes and plugins provided by Simulink and COOJA. Particularly, through COOJA, one has access to many debugging features such as break points, watches, logging and single stepping. Additionally, full access to the wireless transmissions is provided, along with printf logging from nodes, and device statistics such as energy consumption and reliability. Other features are available in GISOO which are suitable for the implementation and validation of wireless NCSs. GISOO is able to perform co-simulations with Truetime (Cervin et al., 2003) which is beneficial for users implementing wireless NCSs together with wired NCSs and other control setups. Additionally, integration with real hardware is possible through Simulink or COOJA. In Section 8.3.5 we demonstrate a GISOO hardware-in-the-loop experiment where a real double tank system is controlled over a multi-hop network emulated in COOJA.

8.3 Examples

In this section, we provide an extensive set of examples which validate the platform, and demonstrate GISOO's applicability to implement and validate wireless NCSs.

We start by describing the wireless NCS setup which is composed of the physical system which we aim to control, the wireless network and its devices. Afterwards, we compare the control and communication performance obtained when controlling a double tank system (Åström and Lundh, 1992), which was introduced in Section 2.8 of Chapter 2, in both reality and in GISOO, using a simple network setup. Thereafter, we present a large-scale simulation of a closed-loop control experiment of ten double tank systems controlled over a multi-hop relay network with sixteen relay nodes. Using this setup, we demonstrate how we can use GISOO to analyze the impact of the data transmission policy, interference and faults on the end-to-end delay and packet losses in the network.

A hardware-in-the-loop experiment is proposed, where a real double tank experimental setup is controlled over a multi-hop wireless network simulated in GISOO. After, we then perform an essential evaluation of the simulation time of the GISOO simulator when performing the proposed experiments.

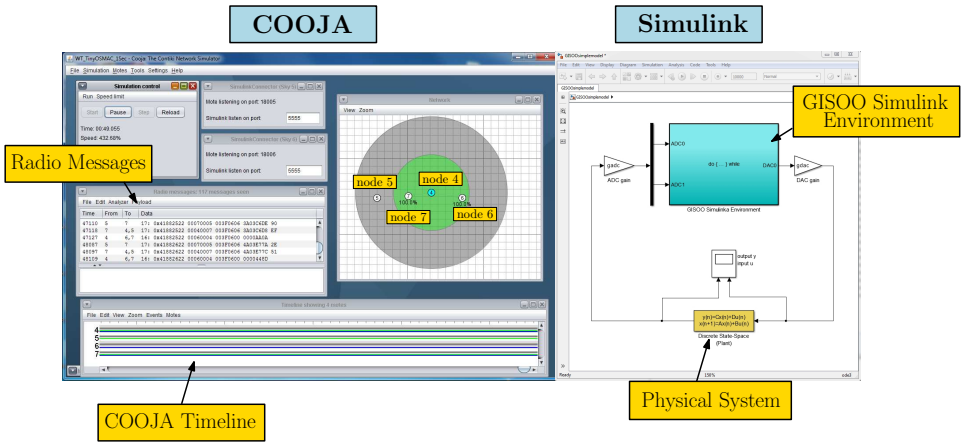


Figure 8.2: GISOO environment, integrating COOJA (left) and Simulink (right) for the closed-loop control of a single double tank system. Four wireless nodes are used in this experiment, a sensor (node 5), relay (node 7), controller (node 4) and actuator (node 6). The green region denotes the transmission range and the grey region the interference range of node 4.

8.3.1 Wireless NCS setup

We now present the control system setup, the wireless devices and communication protocols used in the examples.

Control system

Recall the double tank system setup introduced in Section 2.8 of Chapter 2 and illustrated in Figure 2.3. The control objective is to track a given water level of the lower tank L_2 by adjusting the motor voltage V_p accordingly. Tracking of constant reference signals $r(t)$ can be achieved by using a state-feedback controller with integral action as was performed in Chapters 5 and 6. In this way, the control input $u(k)$ is given by

$$\begin{aligned} u(k) &= K_p x(k) + K_i x_c(k), \\ x_c(k+1) &= x_c(k) + T\left(r(t) - Cx(k)\right), \end{aligned}$$

where $x(k) = \begin{bmatrix} L_1 & L_2 \end{bmatrix}^T$, $x_c(k)$ is the integral state, $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$, and (K_p, K_i) the controller gains. Moreover, T is the sampling and transmission period for the sensor nodes, to be specified below.

Wireless network

The wireless devices used are the Telos platform introduced in Section 2.8 of Chapter 2. We use the TinyOS original MAC, BoX-MAC (Moss and Levis, 2008) which is a CSMA/CA MAC with a maximum of 3 retransmissions in case of transmission failures, and for the routing layer, we selected CTP (Gnawali et al., 2009). Later in the hardware-in-the-loop experiment we utilize the RPL routing protocol (Winter et al., 2012).

The wireless channel model utilized in all the experiments below is the UDGM with constant loss. This simple model was used since it suitably captures the typical network imperfections and we focused on demonstrating the general GISOO capabilities.

8.3.2 GISOO validation

We start by performing a comparison between a real closed-loop control experiment on the double tank system and a closed-loop experiment simulation in GISOO, using the same wireless code.

In this setup, four wireless devices are used which are the sensor, relay, controller and actuator. The GISOO environment for this simulation is presented in Figure 8.2. The wireless sensor node (node 5) periodically samples its ADC every 1 s to acquire the tank level values, and transmits these values to an intermediate relay node (node 7). This node forwards the data packet to the wireless controller node (node 4), which computes the control action according to (8.1) and communicates the actuation input value to the wireless actuator (node 6). The wireless actuator applies the required voltage to the double tank pump through the DAC. The reference r is set to be initially 8 cm, and change to 5 cm after 100 s.

The tank levels, control input and end-to-end delay for this experiment are shown in Figure 8.3. The end-to-end delay is calculated as the time between the ADC sampling at the sensor, until a value is requested to be set at the actuator's DAC, in both GISOO and in the real experiment. From Figure 8.3, we verify that the behavior of the linear double tank model in GISOO follows the real nonlinear double tank system behavior with a minor deviation, and set-point tracking on the lower tank is achieved. The average difference between the end-to-end delay of both experiments is only 1.27 ms. This is caused by the fact that the ADC reading and DAC writing action can be detected immediately in GISOO, while in a real experiment *printf* commands must be used to time-stamp the ADC reading and DAC writing actions.

8.3.3 Multiple processes and multi-hop network

We now evaluate different communication metrics in various typical scenarios in large-scale wireless NCSs. From now on, all evaluations are performed in the co-simulator. We deploy ten double tank systems where its sensors communicate with a

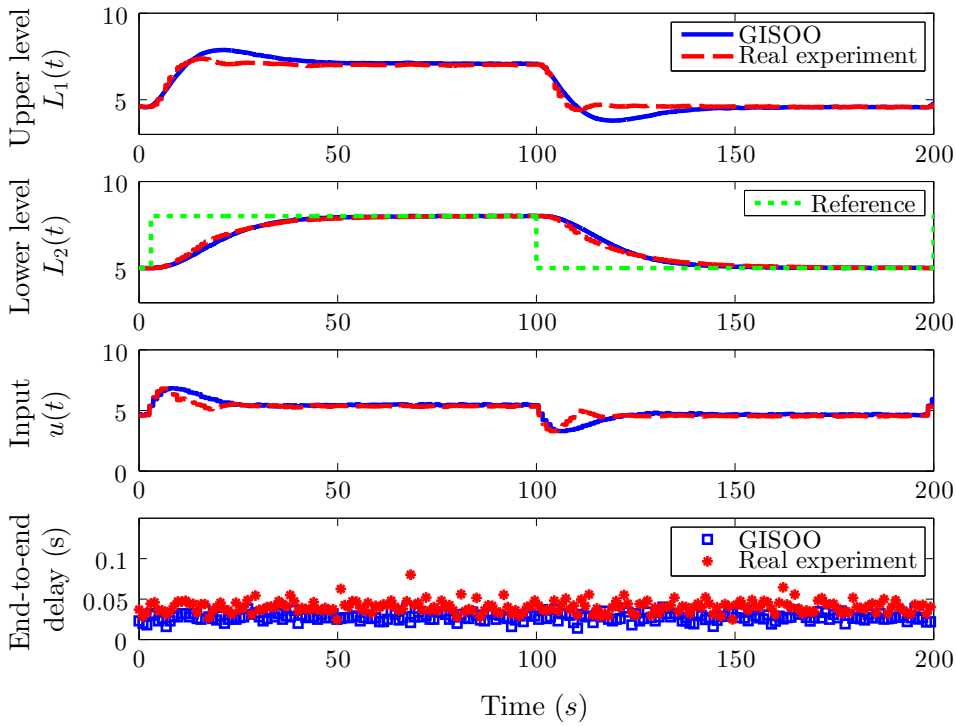


Figure 8.3: Double tank levels $L_1(t)$ and $L_2(t)$ and control input value $u(t)$ for a real experiment and the GISOO simulation. The last plot depicts the end-to-end delay (s) for both experiments.

single wireless controller node through a sixteen wireless relay multi-hop network. Firstly, the case when sensors transmit periodically in the network is evaluated, followed by the analysis when the sensor transmission is event-based. Later, we exploit the impact of interference and faults in the wireless NCS.

Periodic sensor transmissions

This scenario is depicted in Figure 8.4. Each process has a single sensor and a single actuator. The sensor nodes for each of the ten processes are labeled from 1 to 10, while their respective actuators are labeled from 21 to 30. The controller is node 41, while the relay nodes are labeled from 42 to 58. The results for the end-to-end delay and packet loss for this experiment are shown in Figure 8.5(a), for sensor nodes 1, 2, 5 and 8, which are located at different hop levels in the network. The end-to-end delay for a sensor node is calculated as the time between an ADC reading at the respective sensor, until the DAC writing by the corresponding

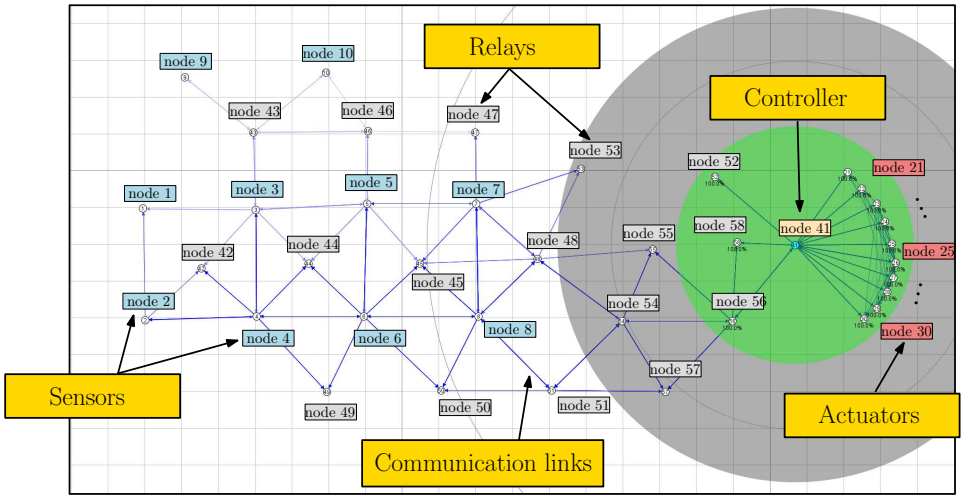


Figure 8.4: Topology for the control of multiple double tank systems over a mesh network with sixteen relay nodes. Sensors nodes (blue) are nodes 1 to 10, actuator nodes (red) are nodes 21 to 30, controller (yellow) is node 41 and the relay nodes (grey) range from 42 to 58. The transmission (grey circle) and interference range (green cricle) of all nodes is set as the controller node 41 in the figure.

actuator. The sampling and transmission period of the sensor nodes is set to 1 s. As is evident from the plots, the distance to the controller greatly affects the end-to-end delay. The delay varies considerably due to the interference generated by the high density of nodes transmitting data and routing messages in the network. However, this interference is not enough to create packet losses as the total number of retransmissions is never reached. This can be explained by the large transmission period of the sensors.

Event-based sensor transmissions

In order to reduce the complexity of the analysis, we utilize the simple event-based strategy proposed in (Årzén, 1999). The event condition which decides if the sensor node transmits the most recent tank levels is given by

$$|L_2(t) - L_2(t_s)| > e_{\text{lim}} \quad \text{OR} \quad t_{\text{act}} \geq t_{\text{lim}},$$

where $L_2(t_s)$ is the last transmitted lower tank level, e_{lim} the threshold value, t_{act} the time elapsed since the last transmission and t_{lim} is the maximum allowed inter-transmission time. The thresholds are set to $e_{\text{lim}} = 0.2$ and $t_{\text{lim}} = 10\,\text{s}$, which generated 75 aperiodic transmissions on 150 s of simulation time, while achieving set-point tracking. The end-to-end delay results are depicted in Figure 8.5(b). The

delay tends to vary considerably, and an increase in the average delay of the nodes 1 and 2 occurs. This is due to the fact that a CSMA/CA MAC is used, where nodes sporadically compete for channel access.

8.3.4 Impact of interference and faults

The same scenario with multiple processes controlled over a multi-hop network is evaluated under 1) network interference and 2) network failure, where a set of four relay nodes become faulty.

Interference

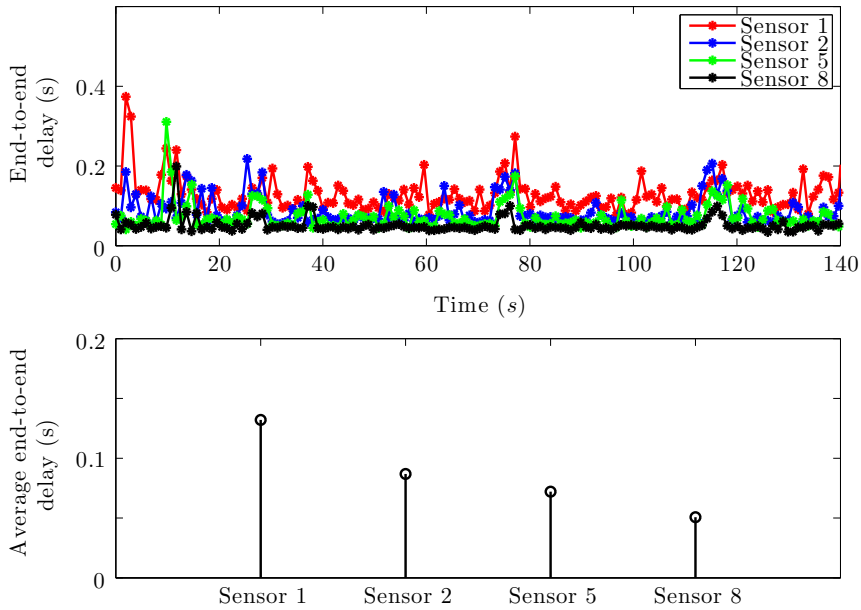
Interference is inflicted by an external node 59, inserted in the region close to the controller node 41. This device starts transmitting packets periodically every 20 ms with a 48 bytes size, from time $t = 40$ s. The end-to-end delay and packet loss statistics are shown in Figure 8.6(a). As expected, the interference increases the average end-to-end delay and the delay variability. This, however, does not cause any packet losses in the network.

Network faults

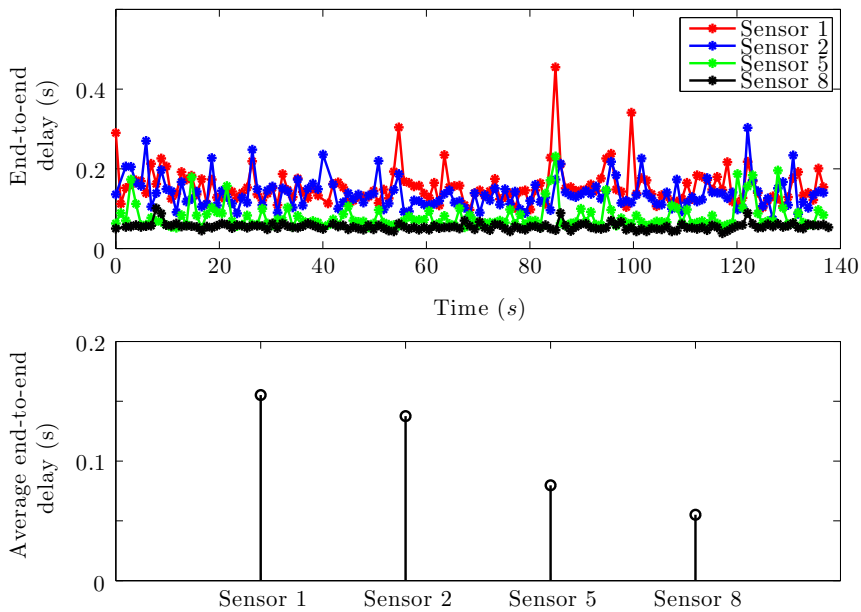
Removal of relay nodes 50, 54, 55 and 58 which are close to the controller node 41 takes place at time $t = 40$ s. Figure 8.6(b) depicts the end-to-end delay and packet loss statistics of the node fault experiment. Because of this failure, there is an end-to-end delay peak that occurs due to the fact that the faulty nodes were communicating the data from the selected sensors. The two packet losses, transmitted by sensor 2 and 8, occurred since the relay nodes which were in their routing path failed, not allowing those packets to successfully arrive to the controller. Since the network topology offers many redundant paths, the nodes are able to quickly, and in a distributed manner using the CTP routing mechanism, reselect a new neighbor from their neighbor list to forward the sensor messages.

8.3.5 Hardware-in-the-loop experiment

A hardware-in-the-loop multi-hop control experiment of the double tank system was conducted as illustrated in Figure 8.7. For this experiment, a real double tank system was connected through the NI PCI-6221 DAQ board to a computer running Simulink ("Process Simulink"). This Simulink interface is responsible for reading the double tank sensors (analog inputs) and to send the actuation signals to the pump motor (analog outputs). The sensor measurements are then transmitted via UDP every 5 ms to another computer which runs GISOO ("GISOO Simulink"). The sensor data received via UDP is then sampled periodically by the sensor node in COOJA (node 1) every 1 s. This data is further transmitted over the multi-hop network to the controller/actuator node (node 10), which computes the actuation signal to be applied to the double tank system. The actuation data is set at the

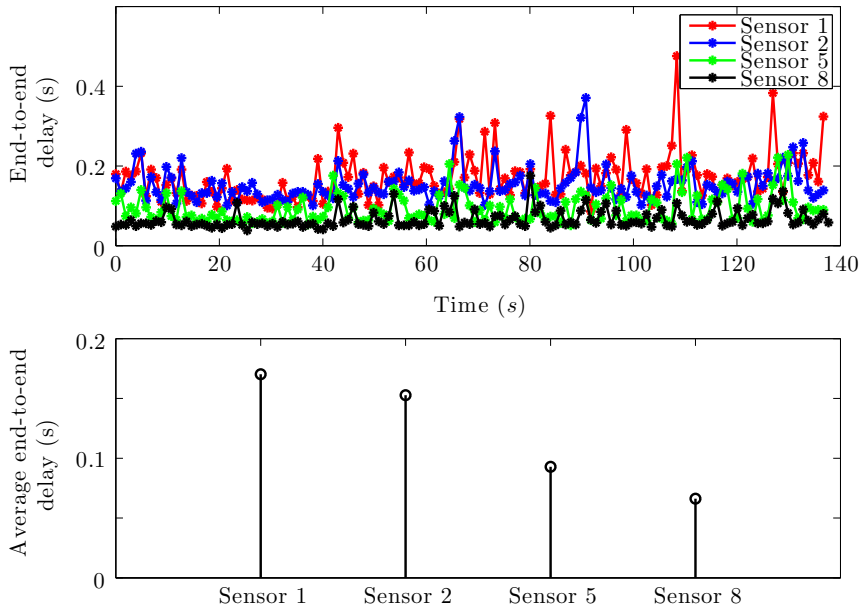


(a) Periodic transmission of sensor data.

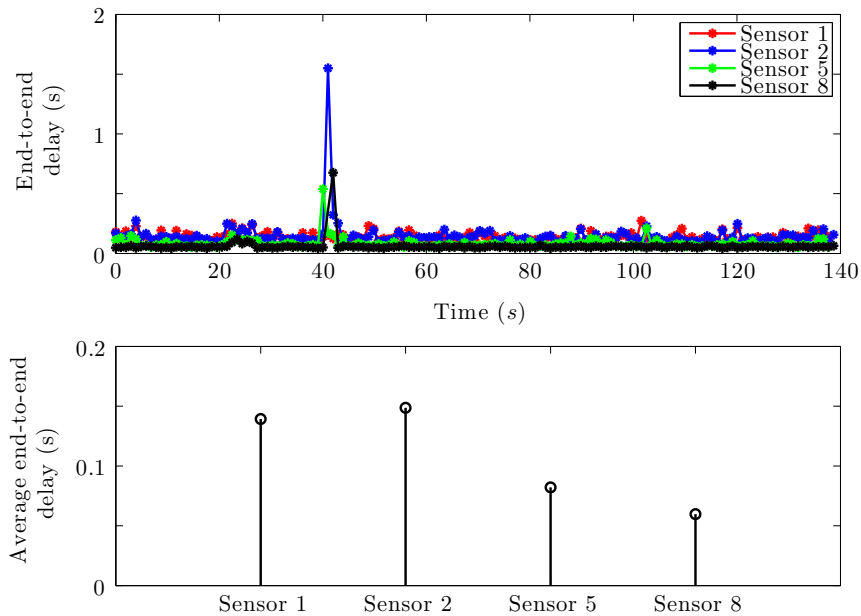


(b) Event-based transmission of sensor data.

Figure 8.5: Evaluation of the end-to-end delay for the closed-loop control of ten double tank processes through a large mesh network of sixteen relays. Results for nodes 1, 2, 5 and 8 located at different hop levels in the network. No packet loss occurs in the cases evaluated.



(a) Interference in the region close to the controller node starting at $t = 40\,\text{s}$. No packet loss occurs.



(b) Fault and removal of four relay nodes close to the controller node at $t = 40\,\text{s}$. One packet loss occurs in the data transmitted by sensor 2 and another in the data transmitted by sensor 8.

Figure 8.6: Evaluation under network interference and node faults.

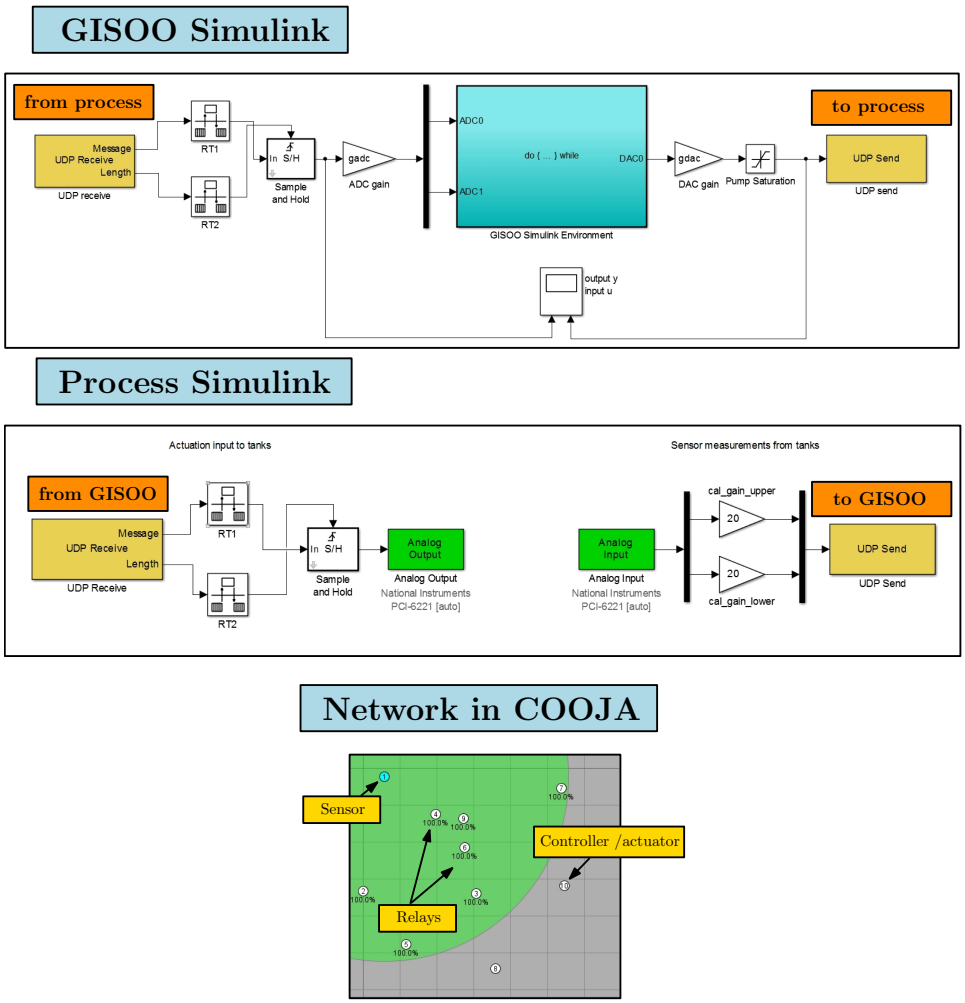


Figure 8.7: Hardware-in-the-loop closed-loop control experiment with GISOO. A sensor node (node 1) samples a real double tank system and transmits the sensor measurements to the controller/actuator node (node 10) over a multi-hop network. A UDP communication interface is setup in GISOO simulink allowing the reading of sensor measurements and application of control input values from/to the process simulink which connects to the real double tank system.

DAC port of the emulated node in COOJA and then sampled and transmitted by the GISOO Simulink to the Process Simulink via UDP. When the actuation data arrives at the Process Simulink, the actuation voltage is set through the NI

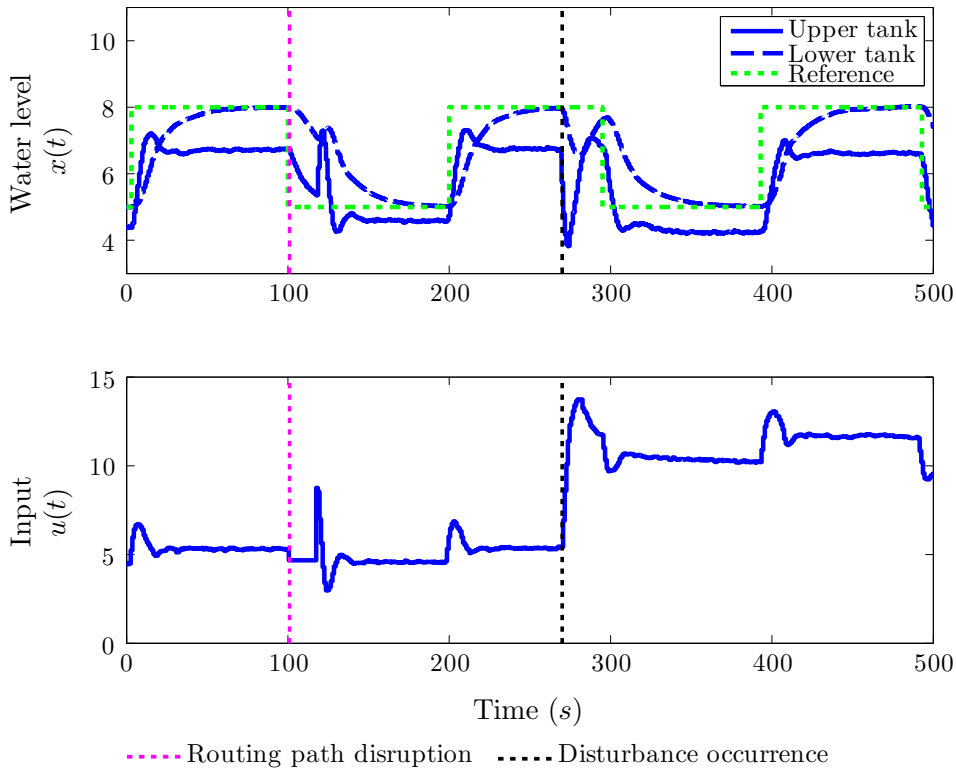


Figure 8.8: Time-response for the hardware-in-the-loop closed-loop control experiment with GISOO. Tracking of a reference signal (green) at the lower tank. A disruption of the routing path occurs at time $t = 100\,\text{s}$, and a disturbance affects the system from time $t = 270\,\text{s}$, when the upper tank tap is opened.

PCI-6221 DAQ board.

The time-response of this experiment is shown in Figure 8.8 where a squared reference signal (green dashed line) is tracked by the lower tank (blue dashed line). The end-to-end delay, number of packet losses and the relay ID is depicted in Figure 8.9. The communication between the sensor and the controller/actuator is initially performed through node 4. A routing path disruption occurs at $t = 100\,\text{s}$, as node 4 has a failure and becomes unavailable. The routing protocol then rebuilds a new routing path, taking 18 s to do so, and node 9 becomes the new relay node. As a consequence 18 packets are lost during the re-building phase. Additionally, the upper-tank disturbance tap is opened at $t = 270\,\text{s}$ causing a disturbance in the system. The controller promptly recovers from the disturbance.

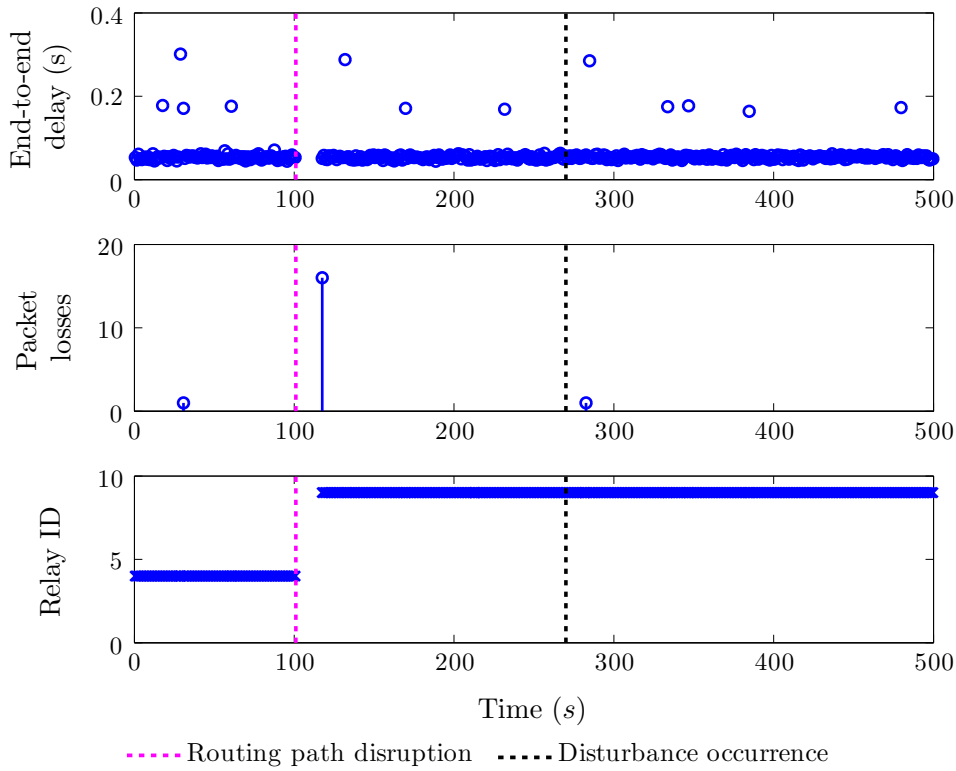


Figure 8.9: Communication details for the hardware-in-the-loop closed-loop control experiment with GISOO. The end-to-end delay between two consecutive sensor packet deliveries is depicted, together with the number of packet losses since the last successfully received packet.

8.3.6 Simulation speed

Results on the total time required to run several of the above experiments in GISOO are presented in Table 8.1. The simulations were performed on an Intel Core i7-2760QM with 2.4 GHz, running with Windows 7. As it can be seen, the total simulation time increases greatly with the total number of wireless devices that exchange data with Simulink. Also, the addition of wireless nodes that do not interact with Simulink have a very small impact on the simulation time.

8.4 Summary

In this chapter, we have presented and evaluated an implementation and validation co-simulator for wireless NCSs. By providing the integration between Simulink

Table 8.1: Real simulation time of a 150 s wireless NCS simulation in GISOO

Scenario	Time (sec)
1 plant (controller in node) - 1 relay	30
1 plant (controller in PC) - 1 relay	158
1 plant (controller in node) - 16 relays	36
10 plants (controller in node)- 16 relays - no interference or faults	278
10 plants (controller in node) - 16 relays - faults	268

and COOJA, we are able to model and simulate physical systems, evaluate control algorithms and communication protocols, while using the actual software running on the wireless devices. Hence, this tool allows for the understanding of the interactions between all these wireless NCS components, in a single platform. Several examples provide an insight on the capabilities and performance of the simulator.

Chapter 9

Conclusions and future work

In this chapter, we present a brief summary of the contributions of this thesis, and some directions for future research.

9.1 Conclusions

In this thesis, we focused on the problem of design, implementation and validation of wireless networked control systems (NCSs). Particularly, we considered sampling and communication strategies that achieve efficient resource usage with guaranteed closed-loop performance. The resources we considered were wireless network bandwidth and energy. Additionally, we devised strategies dealing with out-of-order and delayed communications, as well as sensor and actuator faults. The implementation and validation was performed on both experimental testbeds and on a newly developed wireless NCS co-simulator.

Resource-aware aperiodic control

In Chapters 3, 4 and 5, we explored aperiodic sampling strategies and compared them to traditional periodic control.

Chapters 3 and 4 presented triggering conditions and communication strategies for aperiodic control over a wireless network. In Chapter 3, we show how to design triggering conditions in an efficient manner. Moreover, we proposed an algorithm to tune the event-triggered condition to allow for larger sampling intervals. In Chapter 4 we devised and implemented three communication mechanisms for aperiodic control, which we denoted event-based, predictive and hybrid. For each mechanism, we proposed suitable scheduling methods and a MAC scheme. The mechanisms were experimentally implemented and compared with respect to closed-loop performance, energy efficiency and network bandwidth usage. The results showed that all the proposed schemes achieved closed-loop performances similar to the ones provided by a typical periodic implementation, while increasing battery lifetime between 40% to 60%. We demonstrated that an event-based mechanism is suitable

for applications where high control performances are required and the network is used solely for the control system. On the other hand, a predictive mechanism has a clear advantage if the wireless network is shared among other nodes. However, this mechanism has the drawback of being less robust to disturbances affecting the closed-loop system. The hybrid communication mechanism joined the benefits of the two mechanisms, as it exhibited high control performances, robustness to disturbances and an efficient network bandwidth utilization.

Chapter 5 presented an event-based sampling-rate selection strategy. Performance guarantees were given with respect to a quadratic cost. The strategy was shown to reduce the sampling rate compared to periodic control, while achieving the same cost. Additionally, it gave improved control performance when compared to other recently proposed aperiodic techniques. Similarly to Chapter 4, we proposed scheduling and MAC schemes to show how the proposed controller can be implemented in a wireless NCS, where multiple control loops share the same wireless network. Finally, an experimental validation of the devised strategy was performed on two double tank systems.

Compensation for out-of-order communications and time-varying delays

In Chapter 6, we considered the design and implementation of a minimax controller which compensates for out-of-order communications and time-varying delays. Such network effects may be caused from malicious behavior of relay nodes in the network, by network congestion or device malfunctions. We assumed that only sensor communication is affected by the delays, and that communication is periodic. We devised a linear temporal coding mechanism for the minimax controller, in which the sensor combines the current measurement with previously transmitted measurements, thus transmitting the same amount of data at each step. Numerical results show that in case of delayed or lost information, estimation performance can be improved by using the proposed method. An experimental evaluation of the minimax controller was performed in a multi-hop wireless NCS, where a malicious relay node mounts an attack at the routing layer level. By exploiting a vulnerability of the state-of-the-art routing protocol for low-power and lossy wireless networks, the relay node forces the packets to be forwarded, inflicting a variable, but bounded, delay in each packet. Experimental results demonstrated the benefits of the proposed compensator to allow the closed-loop system to be resilient to the network conditions imposed by the malicious relay node.

Distributed reconfiguration

In Chapter 7, we studied distributed reconfiguration for sensor and actuator faults in an NCS. The reconfiguration objectives were set to achieve model-matching while minimizing the steady-state estimation error covariance and a linear quadratic control cost. The system was assumed to have enough redundancy to enable model-

matching. Moreover, it was imposed that sensor and actuator nodes must only exchange information with their local neighbors. An efficient distributed reconfiguration strategy was devised. It was numerically and experimentally evaluated on room temperature control experiments. Specifically, we built a testbed comprising of three wireless heaters which were set to regulate the room temperature in a distributed manner. When a neighboring heater failed, the remaining wireless heaters were able to detect the fault and cooperate to compensate for the loss. Specifically, no deterioration in temperature tracking were obtained.

A novel co-simulator

A co-simulator was presented in Chapter 8 and it was shown how it can be used to study and validate wireless NCSs. By combining Simulink and the wireless network simulator COOJA, the co-simulator allows for the integration of plant dynamics with realistic wireless network models and the actual software running on embedded devices. Hence, the co-simulator helps in the understanding of the effects that the wireless code and the operating system have on the complete system. As the standard protocols from both IEEE and IETF are currently available for the operating systems and wireless devices supported by COOJA, our co-simulator allows for the validation of control strategies using state-of-the-art protocols in realistic wireless network scenarios. The methods developed in Chapters 5 and 6 were validated using the co-simulator.

9.2 Future work

There are many directions to extend the work presented in this thesis. Some suggestions for future work are given below.

Application to large-scale systems

The design and implementation of the strategies developed in Chapters 3, 4 and 5, considered systems with relatively small size and where a network manager is able to manage the whole network. Future work can be devoted to the extension of these methods to consider multi-hop networks, using distributed routing protocols such as the IETF's RPL and the IEEE 802.15.4e. We remark that these protocols are not yet capable of efficiently implement multi-hop communications between the controllers and actuators. Hence, there is an opportunity to contribute to their design.

In large-scale systems, the use of a TDMA MAC may impose many design constraints. Hence, an interesting question is what performance guarantees can be achieved by the methods we proposed if a CSMA/CA MAC would be used. Would the methods be required to adapt to the current conditions of the network? A promising MAC and transmission scheme was devised in (Ramesh et al., 2014), where devices in the network compute, in a distributed manner, a priority index

based on their current state and compete for channel access by resorting to tournaments. The application and adaptation of the methods proposed in this thesis to such scheme would be of interest.

Compensation of network imperfections

Strategies compensating for network imperfections have been studied in Chapter 6. These strategies were designed to be implemented at both the sensor and controller, and they assumed the controller and actuator were co-located. A natural extension of these strategies would be to adapt them to systems with distributed sensors or aggregator nodes, as the network proposed in Chapter 7. Furthermore, the introduction of an actuation link poses additional challenges to the compensation strategies. The approach must be adapted and also consider the future modifications to the IEEE and IETF protocols, which will specifically address multi-hop communication between controllers and actuators.

Co-simulator extensions

Several improvements of the GISOO co-simulator could be developed. GISOO is affected by scalability issues. The simulation of a large-scale wireless network is performed reasonably fast in COOJA. However, when many sensor and actuator nodes are interfaced in Simulink and COOJA at the same time, the simulation time is large, as discussed in Chapter 8. By optimizing the implementation of this interaction, we believe that much faster simulations can be obtained. Additionally, we would like to improve the visualization tools in COOJA. Particularly, we intend to provide real-time plots with data that is available at each device, so that one can analyze the performance of the implemented algorithms in individual devices.

Chapter A

Appendix to chapter 3

A.1 Some closed-form expressions

A.1.1 Calculation of the triggering time

For the diagonalized linear system, the solution of the $i-th$ state linear differential equation at a given time t_e is given by:

$$x(t_e) = e^{at_e}x(t_0) + \frac{v}{a}\left(e^{at_e} - 1\right) = e^{at_e}\left(x(t_0) + \frac{v}{a}\right) - \frac{v}{a}, \tag{A.1}$$

where $v = [Bu]_i$ is the input signal contribution $i-th$ state and $a = A_{ii}$.

The triggering condition for each state x_i is defined as

$$G_i(x_i) - \theta_i = 0, \tag{A.2}$$

as discussed in Section 3.3.1. Consequently, through (A.2) we achieve:

$$\bar{a}x(t_e)^2 + \bar{b}x(t_e) + \bar{c} = 0. \tag{A.3}$$

Roots q_1, q_2 of the quadratic equation are the solutions for $x(t_e)$.

Equalizing by inserting (A.1) in the two solutions of (A.3), we arrive to the following expression for the triggering times:

$$t_e = \frac{1}{a}\log\left(\frac{q_1 + \frac{v}{a}}{x(t_k) + \frac{v}{a}}\right) \bigvee \quad t_e = \frac{1}{a}\log\left(\frac{q_2 + \frac{v}{a}}{x(t_k) + \frac{v}{a}}\right)$$

A.1.2 Finding $\bar{\delta}$

The value $\bar{\delta}$ is given by:

$$\bar{\delta} = \max_{t\in[t_k, T(\theta)]} G_i(t) - \theta_i. \tag{A.4}$$

In order to evaluate (A.4) we must investigate the solution of $\frac{\mathrm{d}(G_i(t)-\theta_i)}{\mathrm{d}t} = \frac{\mathrm{d}G_i(t)}{\mathrm{d}t} = 0$. Substituting (A.1) in $G_i(t)$ and calculating the derivative we get

$$\frac{\mathrm{d}G_i(t_e)}{dt} = 2e^{at_e}\bar{a}\left(x(t_0)+\frac{v}{a}\right)^2 + \left(x(t_0)+\frac{v}{a}\right)\left(\bar{b}-2\bar{a}\frac{v}{a}\right) \tag{A.5}$$

We can now solve (A.5) to find the time at which there exists a maximum/minimum of the gap $G_i(t)$, which has the following solution

$$t_e^* = \frac{1}{a}\log\frac{2\bar{a}\frac{v}{a}-\bar{b}}{2\bar{a}\left(x(t_0)+\frac{v}{a}\right)}$$

Since there only exists one maximum/minimum of the $G_i(t)$, the solution to $\bar{\delta}$ is then computed as:

$$\bar{\delta} = \max\left\{G_i(t)-\theta_i : t = \{t_k, t_e^*, T(\theta)\}\right\}$$

A.2 Solving optimization problem (3.13)

In this section we show how one can efficiently solve the optimization problem (3.13) for the δ, which we rewrite here:

$$\begin{array}{ll} \underset{\delta_1,\ldots,\delta_n}{\text{minimize}} & \max_{i=1,\ldots,n}(c_i+\delta_i) \\ \text{subject to} & \delta_i \le \overline{\delta_i} \text{ for all } i = 1,\ldots,n \\ & \sum_{i=1}^n \delta_i = 0 \end{array} \tag{A.6}$$

We will assume that $\overline{\delta_i} \ge 0$ in this section, which is true in our case (cf. equation (A.4)). Note that the problem can be written as a linear program (LP) and so can be solved by any LP solver. One can however use a simple bisection algorithm to solve this problem more efficiently without resorting to LP packages.

The algorithm relies on the observation that given a real number X, there is a simple explicit method to decide whether the optimal value of the problem is less than or equal to X. This decision problem can be written as follows:

Given $X \in \mathbb{R}$, does there exist $\delta_1,\ldots,\delta_n$ such that:

(1) $\max\limits_{i=1,\ldots,n}(c_i+\delta_i) \le X$

(2) $\delta_i \le \overline{\delta_i}$ for all $i = 1,\ldots,n$

(3) $\sum\limits_{i=1}^n \delta_i = 0$

This decision problem can be solved as follows: Let $I_> = \{i \mid c_i \ge X\}$ and $I_< = \{i \mid c_i < X\}$. If δ is a feasible solution to the decision problem above, then from

conditions (1) and (2) we necessarily have, for $i \in I_>$, $\delta_i \leq X - c_i \leq 0$, and for $i \in I_<$, we will have $0 \leq \delta_i < \min(X - c_i, \overline{\delta_i})$. Such vector δ satisfying (1), (2) and (3) exists if, and only if:

$$\sum_{i \in I_>} (c_i - X) \geq \sum_{i \in I_<} \min(X - c_i, \overline{\delta_i}).$$

Moreover, if this condition is true and $s = \sum_{i \in I_<} \min(X - c_i, \overline{\delta_i}) - \sum_{i \in I_>} |X - c_i| > 0$, then the vector δ defined by:

$$\delta_i = \begin{cases} X - c_i - s/|I_>| & \text{if } i \in I_> \\ \min(X - c_i, \overline{\delta_i}) & \text{otherwise} \end{cases}$$

will satisfy conditions (1), (2) and (3) above.

The complete bisection algorithm to solve the optimization problem (A.6) is given in Algorithm A.1.

Algorithm A.1 Bisection algorithm to solve the optimization problem (A.6). The algorithm reaches an accuracy of $2^{-\texttt{numIterDelta}}$ after `numIterDelta` iterations.

Input: $c, \overline{\delta} \in \mathbb{R}^n$, `numiterDelta` ($\approx 15$)
Output: $\delta^* := \arg\min\{\max_i(c_i + \delta_i) : \delta_i \leq \overline{\delta}_i, \sum_i \delta_i = 0\}$
$X_1 \leftarrow \min\{c_i, i = 1, \ldots, n\}$
$X_2 \leftarrow \max\{c_i, i = 1, \ldots, n\}$
while $k \leq$ `numIterDelta` **do**
  $X \leftarrow (X_1 + X_2)/2$
  $I_> \leftarrow \{i \in \{1, \ldots, n\} : c_i \geq X\}$
  $I_< \leftarrow \{i \in \{1, \ldots, n\} : c_i < X\}$
  if $\sum_{i \in I_>} (c_i - X) \geq \sum_{i \in I_<} \min(X - c_i, \overline{\delta_i})$ **then**
    $X_1 \leftarrow X$
  else
    $X_2 \leftarrow X$
  end if
  $k \leftarrow k + 1$
end while
$\delta^* \leftarrow \min(X - c, \overline{\delta})$

Chapter B

Appendix to chapter 5

B.1 Proof of Lemma 5.1

Let us denote by $n_N^- = \left(\left\lfloor \frac{d_\kappa + \delta_d}{\delta_N} \right\rfloor + 1 \right) \delta_N$ the nominal sampling instant closest (from below) to the the minimum disturbance occurrence time $k = d_\kappa + \delta_d$. In the same manner, let us denote by $n_N^+ = \left(\left\lfloor \frac{d_\kappa}{\delta_N} \right\rfloor + 1 \right) \delta_N$ the nominal sampling instant closest (from above) to the disturbance occurrence at time d_κ.

Consider the nominal cost-to-go $J_N^{[n_N^-, \infty)}$ at time $k = n_N^-$. Such value is a valid upper bound to the cost-to-go at time $k = d_{\kappa+1}$, before the occurrence of the disturbance, i.e.,

$$J_N^{(d_{\kappa+1}, \infty)} \leq J_N^{[n_N^-, \infty)}.$$

This holds since the cost-to-go is a decreasing function with time, when the closed-loop system is stable. Given this fact, it is also true that the cost-to-go at time $k = n_N^-$ is larger than the cost-to-go at time $k = d_\kappa + \delta_d$. Furthermore, since $\delta_d \geq d_{\kappa+1} - d_\kappa$, the cost-to-go at time $k = d_\kappa + \delta_d$, is larger than the cost-to-go at time $k = d_{\kappa+1}$. Equality holds when $k = n_N^- = d_\kappa + \delta_d$, i.e., the nominal sample is synchronous with the disturbance instant. The above reasoning is illustrated in Figure B.1.

We now focus on finding the value of ϵ for the cost-to-go $J_N^{[n_N^-, \infty)}$. This cost is given by,

$$J_N^{[n_N^-, \infty)} = x(n_N^-)^T P_N x(n_N^-), \tag{B.1}$$

since a nominal sampling occurs at time $k = n_N^-$. The state $x(n_N^-)$ can be written as a function of $x(d_\kappa)$ as follows. If the disturbance occurring at d_κ was not synchronous with a nominal sample, i.e., $\text{rem}(d_\kappa, \delta_N) \neq 0$, the state $x(k)$ in the interval $k \in [d_\kappa, n_N^- - 1]$, is given by

$$x(k) = \tilde{\Phi}(k) x(d_\kappa) + \tilde{\Gamma}(k) \bar{u},$$

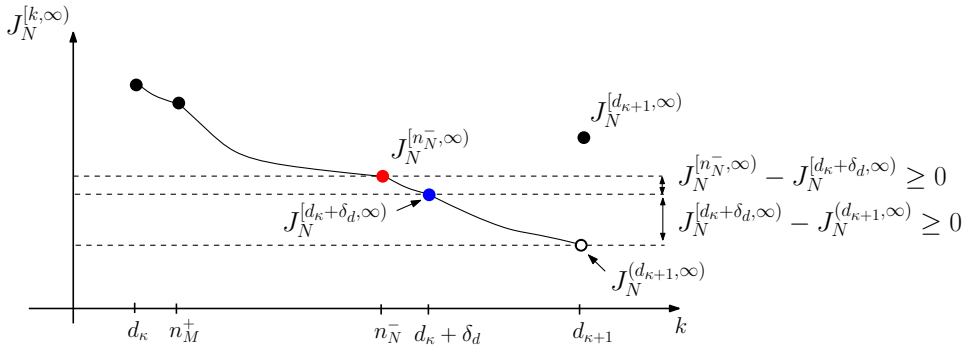


Figure B.1: Illustration of the cost-to-go $J_N^{[k,\infty)}$.

where $\bar{u}$ is the control input being applied from the last nominal sampling instant and at the time $k = d_\kappa$, and $\tilde{\Phi}(l) = \Phi^l$ and $\tilde{\Gamma}(l) = \sum_{j=0}^{l-1} \Phi^j \Gamma$. Let $\tilde{\Phi}_N(l) = (\Phi_N - \Gamma_N K_N)^{\frac{l}{\delta_N}}$. Hence, the cost-to-go (B.1) can be represented as

$$J_N^{[n_N^-,\infty)} = \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) V(n_N^-, d_\kappa) Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix},$$

where

$$V(n_N^-, d_\kappa) = \begin{bmatrix} \bar{\Phi}_N^T P_N \bar{\Phi}_N & \bar{\Phi}_N^T P_N \bar{\Gamma}_N \\ \bar{\Gamma}_N^T P_N \bar{\Phi}_N & \bar{\Gamma}_N^T P_N \bar{\Gamma}_N \end{bmatrix} \tag{B.2}$$

with $\bar{\Phi}_N = \tilde{\Phi}_N(n_N^- - n_N^+)\tilde{\Phi}(n_N^+ - d_\kappa)$ and $\bar{\Gamma}_N = \tilde{\Phi}_N(n_N^- - n_N^+)\tilde{\Gamma}(n_N^+ - d_\kappa)\bar{u}$. Additionally, the matrix $Y(d_\kappa)$ is given by

$$Y(d_\kappa) = \begin{cases} \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} & \text{if } \operatorname{rem}(d_\kappa, \delta_N) = 0, \\ \begin{bmatrix} I_n & 0 \\ 0 & I_m \end{bmatrix} & \text{otherwise,} \end{cases} \tag{B.3}$$

for the case when the nominal sample and disturbance instant are synchronous, i.e., $\operatorname{rem}(d_\kappa, \delta_N) = 0$ or otherwise. From (Horn and Johnson, 2012), the following bound can be found for the cost-to-go $J_N^{[n_N^-,\infty)}$:

$$\begin{aligned} J_N^{[n_N^-,\infty)} &= \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) V(n_N^-, d_\kappa) Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix} \\ &\leq \lambda_{max}\left(V(n_N^-, d_\kappa)\right) \left(\begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix}^T Y(d_\kappa) \begin{bmatrix} x(d_\kappa) \\ 1 \end{bmatrix} \right) \end{aligned}$$

Thus, one can define $\epsilon = \lambda_{max}\left(V(n_N^-, d_\kappa)\right)$. Consequently, the value of ϵ does not depend on the value of the disturbance but only on the length of the disturbance interval δ_d. Due to this fact and without loss of generality, one can set $d_\kappa = 0$. Hence, we use the simpler notation $V(n_N^-) = V(n_N^-, d_\kappa)$ and ϵ becomes $\epsilon = \lambda_{max}\left(V(n_N^-)\right)$ as (5.19). This concludes the proof.

B.2 Derivation of $J_N^{[l,\infty)}$

The infinite-horizon cost-to-go for the nominal controller from time $k = l$, $J_N^{[l,\infty)}$ is defined as

$$J_N^{[l,\infty)} = \begin{cases} x(l)^T P_N x(l) & \text{if } \mathrm{rem}(l, \delta_N) = 0, \\ \hat{J}_N^{[l,\infty)} & \text{otherwise,} \end{cases} \tag{B.4}$$

depending if a nominal transmission occurs at time $k = l$ or not, respectively. Moreover, the cost $\hat{J}_N^{[l,\infty)}$ is given by

$$\hat{J}_N^{[l,\infty)} = \hat{J}_N^{[l,n_N^+-1]} + x(n_N^+)^T P_N x(n_N^+), \tag{B.5}$$

where n_N^+ is the closest sampling instant (from above) of the nominal controller to the time $k = l$, and given by $n_N^+ = \left(\left\lfloor \frac{l}{\delta_N} \right\rfloor + 1\right)\delta_N$. Notice that $\hat{J}_N^{[l,n_N^+-1]}$ is given by (5.11) iterated from time $k \in [l, n_N^+ - 1]$ where the control input is the input being applied since the last nominal sampling instant, which we denote by $\bar{u}$. Thus, $\hat{J}_N^{[l,n_N^+-1]}$ is given by

$$\hat{J}_N^{[l,n_N^+-1]} = \sum_{i=0}^{n_N^+ + 1 - l} x(i)^T Q x(i) + 2x(i)^T N \bar{u} + \bar{u}^T R \bar{u}. \tag{B.6}$$

We can then substitute (B.6) in (B.5), and rewrite (B.5) as a function of $x(l)$. Following these steps, we arrive to cost

$$\hat{J}_N^{[l,\infty)} = \begin{bmatrix} x(l) \\ 1 \end{bmatrix}^T \begin{bmatrix} P_N + F_N & E_N \\ E_N^T & G_N \end{bmatrix} \begin{bmatrix} x(l) \\ 1 \end{bmatrix},$$

where

$$F_N = \tilde{\Phi}(n_N^+ - l)^T P_N \tilde{\Phi}(n_N^+ - l) + \sum_{i=0}^{n_N^+ + 1 - l} \tilde{\Phi}(i)^T Q \tilde{\Phi}(i)$$

$$E_N = \left(\tilde{\Phi}(n_N^+ - l)^T P_N \tilde{\Gamma}(n_N^+ - l) + \sum_{i=0}^{n_N^+ + 1 - l} \tilde{\Phi}(i)^T Q \tilde{\Gamma}(i) + \tilde{\Phi}(i)^T N \right) \bar{u}$$

$$G_N = \bar{u}^T \left(\tilde{\Gamma}(n_N^+ - l)^T P_N \tilde{\Gamma}(n_N^+ - l) \sum_{i=0}^{n_N^+ + 1 - l} \tilde{\Gamma}(i)^T Q \tilde{\Gamma}(i) + 2\tilde{\Gamma}(i) N + R \right) \bar{u}$$

where we recall that $\tilde{\Phi}(l) = \Phi^l$ and $\tilde{\Gamma}(l) = \sum_{j=0}^{l-1} \Phi^j \Gamma$. Finally, we can rewrite (B.4) as

$$J_N^{[l,\infty)} = \begin{bmatrix} x(l) \\ 1 \end{bmatrix}^T U_N(l) \begin{bmatrix} x(l) \\ 1 \end{bmatrix}, \tag{B.7}$$

where

$$U_N(l) = \begin{cases} \begin{bmatrix} P_N & 0 \\ 0 & 0 \end{bmatrix} & \text{if } \operatorname{rem}(l, \delta_N) = 0. \\ \begin{bmatrix} P_N & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} F_N & E_N^T \\ E_N & G_N \end{bmatrix} & \text{otherwise,} \end{cases}$$

Additionally, notice that since $\begin{bmatrix} F_N & E_N^T \\ E_N & G_N \end{bmatrix} \succ 0$, a valid bound for $U_N(l)$ is

$$U_N(l) \succeq \begin{bmatrix} P_N & 0 \\ 0 & 0 \end{bmatrix},$$

where equality holds if $\operatorname{rem}(l, \delta_N) = 0$, i.e., the nominal sampling is synchronous with the time instant $k = l$.

B.3 Proof of Lemma 5.2

Recall the previous derivation of $J_N^{[l,\infty)}$ in Section B.2 for $l = d_\kappa$. The cost-to-go at time $k = d_\kappa$, as shown in (B.5), is given by

$$J_N^{[d_\kappa,\infty)} = \hat{J}_N^{[d_\kappa, n_N^+ - 1]} + x(n_N^+)^T P_N x(n_N^+) = \hat{J}_N^{[d_\kappa, n_N^+ - 1]} + J_N^{[n_N^+,\infty)}$$

Hence, the cost-to-go is the lowest if $\operatorname{rem}(d_\kappa, \delta_N) = 0$, since then $\hat{J}_N^{[d_\kappa, n_N^+ - 1]} = 0$. Therefore, we can write

$$J_N^{[d_\kappa,\infty)} = J_N^{[n_N^+,\infty)} = x(n_N^+)^T P_N x(n_N^+).$$

This concludes the proof.

B.4 Derivation of $J_S^{[l,\infty)}$

The infinite-horizon cost-to-go for the down-sampled controller for $q(k) = q_S$ from time $k = l$, $J_S^{[l,\infty)}$ is defined as

$$J_S^{[l,\infty)} = \begin{cases} x(l)^T P_S x(l) & \text{if } \operatorname{rem}(l - t_s^-, \delta_S) = 0, \\ \hat{J}_S^{[l,\infty)} & \text{otherwise,} \end{cases} \tag{B.8}$$

depending if a slow transmission occurs at time $k = l$ or not, respectively. The cost $\hat{J}_S^{[l,\infty)}$ is given by

$$\hat{J}_S^{[l,\infty)} = \hat{J}_S^{[l,n_S^+-1]} + x(n_S^+)^T P_S x(n_S^+), \tag{B.9}$$

where n_S^+ is the closest sampling instant (from above) of the down-sampled controller from time $k = l$, and given by $n_S^+ = t_s^- + \left(\left\lfloor \frac{l - t_s^-}{\delta_S} \right\rfloor + 1 \right) \delta_S$. Notice that $\hat{J}_S^{[l,n_S^+-1]}$ is given by (5.11) iterated from time $k \in [l, n_S^+ - 1]$ where the control input is the last computed control input, which we denote by $\bar{u}$. If no switching occurred between time $k = [n_S^+ - \delta_S, l]$, then $\bar{u} = u(n_S^+ - \delta_S)$, otherwise $\bar{u} = u(t_s^-)$. Thus, $\hat{J}_S^{[l,n_S^+-1]}$ is given by

$$\hat{J}_S^{[l,n_S^+-1]} = \sum_{i=0}^{n_S^+ + 1 - l} x(i)^T Q x(i) + 2x(i)^T N \bar{u} + \bar{u}^T R \bar{u}. \tag{B.10}$$

We can then substitute (B.10) in (B.9) and rewriting (B.9) as a function of $x(l)$ arriving to

$$\hat{J}_S^{[l,\infty)} = \begin{bmatrix} x(l) \\ 1 \end{bmatrix}^T \begin{bmatrix} F_S & E_S \\ E_S^T & G_S \end{bmatrix} \begin{bmatrix} x(l) \\ 1 \end{bmatrix},$$

where

$$F_S = \tilde{\Phi}(n_S^+ - l)^T P_S \tilde{\Phi}(n_S^+ - l) + \sum_{i=0}^{n_S^+ + 1 - l} \tilde{\Phi}(i)^T Q \tilde{\Phi}(i)$$

$$E_S = \left(\tilde{\Phi}(n_S^+ - l)^T P_S \tilde{\Gamma}(n_S^+ - l) + \sum_{i=0}^{n_S^+ + 1 - l} \tilde{\Phi}(i)^T Q \tilde{\Gamma}(i) + \tilde{\Phi}(i)^T N \right) \bar{u}$$

$$G_S = \bar{u}^T \left(\tilde{\Gamma}(n_S^+ - l)^T P_S \tilde{\Gamma}(n_S^+ - l) \sum_{i=0}^{n_S^+ + 1 - l} \tilde{\Gamma}(i)^T Q \tilde{\Gamma}(i) + 2\tilde{\Gamma}(i) N + R \right) \bar{u}$$

where we recall that $\tilde{\Phi}(l) = \Phi^l$ and $\tilde{\Gamma}(l) = \sum_{j=0}^{l-1} \Phi^j \Gamma$. Finally, we can rewrite (B.8) as

$$J_S^{[l,\infty)} = \begin{bmatrix} x(l) \\ 1 \end{bmatrix}^T U_S(l) \begin{bmatrix} x(l) \\ 1 \end{bmatrix}, \tag{B.11}$$

where

$$U_S(l) = \begin{cases} \begin{bmatrix} P_S & 0 \\ 0 & 0 \end{bmatrix} & \text{if } \mathrm{rem}(l - t_s^-, \delta_S) = 0, \\ \begin{bmatrix} P_S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} F_S & E_S^T \\ E_S & G_S \end{bmatrix} & \text{otherwise.} \end{cases}$$

In the same way as performed in the nominal case, notice that $\begin{bmatrix} F_S & E_S^T \\ E_S & G_S \end{bmatrix} \succ 0$. Hence, a valid bound for $U_S(l)$ is

$$U_S(l) \succeq \begin{bmatrix} P_S & 0 \\ 0 & 0 \end{bmatrix},$$

where equality holds if $\mathrm{rem}(l, \delta_N) = 0$, i.e., the slow sampling is synchronous with the time instant $k = l$. Moreover, with respect to the cost-to-go, due to (B.9) and the fact that $\hat{J}_S^{[l, n_N^+ - 1]} \geq 0$, it holds that

$$J_S^{[l,\infty)} \geq J_S^{[n_S^+,\infty)} = x(n_S^+)^T P_S x(n_S^+).$$

B.5 Expression of the equilibrium solution to ξ

In steady-state conditions, the value of $F_s \xi_{eq} = r$ and $\xi(k+1) = \xi(k) = \xi_{eq}$ for a fixed constant disturbance $w(k) = w$ and disturbance $r(k) = r$. Writing these two conditions by using (5.29) and solving for ξ_{eq} and feedback gain K_δ, one arrives to the solution

$$\xi_{eq} = -(\Phi_\xi + \Gamma_\xi K_\delta F_s - I)^\dagger (D_\xi r + D_\xi w).$$

The value u_{eq} then follows as $u_{eq} = K_\delta F_s \xi_{eq}$.

B.6 Expression of $\Upsilon_\delta(l)$

A closed-form solution can be obtained for $\tilde{\xi}(k+l)$ by iterating the system dynamics (5.30), which give $\tilde{\xi}(k+l) = \Upsilon_\delta(l)\tilde{\xi}(k)$, where

$$\Upsilon_\delta(l) = \tilde{\Psi}(n)\tilde{\Psi}^m(\delta), \tag{B.12}$$

with

$$n = \mathrm{rem}(l, \delta),$$
$$m = \left\lfloor \frac{l}{\delta} \right\rfloor.$$

and $\tilde{\Psi}(l) = \tilde{\Phi}(n) + \tilde{\Gamma}(n)$, $\tilde{\Phi}(n) = \Phi^n$ and $\tilde{\Gamma}(n) = \sum_{j=0}^{n-1} \Phi^j \Gamma K_\delta F_s$.

Chapter C

Appendix to chapter 6

C.1 Physical systems for numerical evaluation

Stable first-order system

The discrete-time linear time-invariant model of the stable first-order system is

$$A = 0.9,\ B = 1,\ C = 1,\ D = \begin{bmatrix} 0.2 & 0 \end{bmatrix},\ E = \begin{bmatrix} 0 & 1 \end{bmatrix},\ Q = 1.$$

Unstable first-order system

The discrete-time linear time-invariant model of the unstable first-order system is

$$A = 1.1,\ B = 1,\ C = 1,\ D = \begin{bmatrix} 0.2 & 0 \end{bmatrix},\ E = \begin{bmatrix} 0 & 1 \end{bmatrix},\ Q = 1.$$

Stable second-order system: Double tank

As the second-order system we use the double tank system model. The continuous-time linear time-invariant model is

$$A = \begin{bmatrix} -0.08 & 0 \\ 0.08 & -0.06 \end{bmatrix},\quad B = \begin{bmatrix} 0.15 \\ 0 \end{bmatrix},\ C = \begin{bmatrix} 0 & 1 \end{bmatrix},\ D = \begin{bmatrix} 0.2I_2 & 0 \end{bmatrix},\ E = \begin{bmatrix} 0 & I_2 \end{bmatrix}.$$

The system is sampled with a period of 0.1 s. Additionally, $Q = I$.

Unstable second-order system

The continuous-time linear time-invariant model of the unstable second-order system has parameters

$$A = \begin{bmatrix} 1.1 & -0.8 \\ 0.6 & -1.4 \end{bmatrix},\quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} 0 & 1 \end{bmatrix},\ D = \begin{bmatrix} 0.2I_2 & 0 \end{bmatrix},\ E = \begin{bmatrix} 0 & I_2 \end{bmatrix}.$$

The system is sampled with a period of 0.5 s. Additionally, $Q = I$.

Second-order system: Double integrator

The continuous-time linear time-invariant model of the double integrator

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 0.7I_2 & 0 \end{bmatrix}, \; E = \begin{bmatrix} 0 & I_2 \end{bmatrix}.$$

The system is sampled with a period of 0.3 s. Additionally, $Q = I_2$.

Chapter D

Appendix to chapter 7

D.1 Proof of Lemma 7.1

The first constraint in (7.12) is the model-matching constraint which is derived as follows. Following Section 7.1.2, model-matching is guaranteed if the closed-loop matrix before fault is the same as after the fault, i.e.,

$$A - L\tilde{T}\Gamma_y C = A - LTC = A_e.$$

Moreover, the objective function and last constraint follow are given as follows. The objective function J_e in (7.7) is equivalent to $J_e = \mathrm{tr}(\tilde{\Sigma})$, where $\tilde{\Sigma}$ is steady-state covariance of the estimation error after a fault and defined as $\tilde{\Sigma} = \lim_{t\to\infty} \mathbf{E}\{\tilde{e}(t)\tilde{e}(t)^\top\}$. Additionally $\tilde{\Sigma}$ under any given estimator gain L is given by the following Lyapunov equation (see (Åström, 1970) for details),

$$\begin{aligned}(A - LTC)\tilde{\Sigma} + \tilde{\Sigma}(A - LTC)^\top + \\ +W + L\tilde{T}\Gamma_y V \Gamma_y{}^\top \tilde{T}^\top L^\top = 0.\end{aligned}$$

The solution of the above Lyapunov equation, can also be expressed as $\tilde{\Sigma} = \int_0^\infty e^{A_e t}(W + L\tilde{T}\Gamma_y V \Gamma_y \tilde{T}^\top L^\top) e^{A_e^\top t}\,\mathrm{d}t$. By noticing that the term $W + L\tilde{T}\Gamma_y V \Gamma_y \tilde{T}^\top L^\top$ is independent of time, one can arrive to the following equivalence of the cost $J_e = \mathrm{tr}(\tilde{\Sigma}) = \mathrm{tr}\left((W + L\tilde{T}\Gamma_y V \Gamma_y \tilde{T}^\top L^\top)\int_0^\infty e^{A_e^\top t} e^{A_e t}\,\mathrm{d}t\right)$. By denoting $Z_e = \int_0^\infty e^{A_e^\top t} e^{A_e t}\,\mathrm{d}t$ and noticing that Z_e is the solution to the Lyapunov equation $A_e^\top Z_e + Z_e A_e + I = 0$, the proof is concluded.

D.2 Proof of Lemma 7.3

In order to prove Lemma 7.3, we rewrite the sensor and actuator reconfiguration problems (7.12) and (7.14) as quadratic optimization problems with equality constraints.

D.2.1 Distributed sensor reconfiguration

Lemma D.1. *Define* $\tilde{T} = \begin{bmatrix}\eta_1 \cdots \eta_p\end{bmatrix}$, $\eta_i \in \mathbb{R}^s$ *and let* $H^e{}_i \in \mathbb{R}^{n^2 \times s}$ *for* $i = 1, \ldots, p$. *The optimization problem* (7.12) *can be rewritten as*

$$\begin{aligned} \min_{\eta_1,\ldots,\eta_p} \quad & \sum_{i=1}^{p} [\Gamma_y]_{ii} V_{ii} \|\eta_i\|^2 \\ s.t. \quad & \sum_{i=1}^{p} H^e{}_i \eta_i = \omega^e \end{aligned}$$

where $H^e \triangleq \begin{bmatrix} H^e{}_1 \ldots H^e{}_p \end{bmatrix} = \left((C^\top \Gamma_y^\top) \otimes L \right)$ *and* $\omega^e = vec\,(LTC)$.

Proof. Recall that the cost J_e in (7.6) is given by $J_e = \mathrm{tr}(\tilde{\Sigma}) = \mathrm{tr}((W + L\tilde{T}\Gamma_y V \Gamma_y T^\top L^\top) Z_e)$ as derived in (7.12). As shown in Proposition 7.1, the optimal solution is independent of the constant terms W and $L^\top Z_e L$, which can be replaced with 0 and I, respectively. Since V and Γ are diagonal, one can write the new objective function as $\mathrm{tr}(\tilde{T}\Gamma_y V \Gamma_y \tilde{T}^\top) = \mathrm{tr}(\sum_{i=1}^{p} [\Gamma_y]_{ii} V_{ii} \eta_i \eta_i^\top) = \sum_{i=1}^{p} [\Gamma_y]_{ii} V_{ii} \|\eta_i\|^2$. The model-matching constraint follows directly by applying the vectorization operation. □

D.2.2 Distributed actuator reconfiguration

We now formulate the distributed actuator reconfiguration problem which follows from rewriting the centralized actuator reconfiguration problem from Lemma 7.2.

Lemma D.2. *Define* $\tilde{K} = \begin{bmatrix} \eta_1^\top & \cdots & \eta_m^\top \end{bmatrix}^\top$, *the column vector* $\eta_i \in \mathbb{R}^n$, P_r *as the permutation matrix for which* $P_r vec\left(\tilde{K}\right) = [\eta_1 \ldots \eta_m]^\top$, *and let* $H^c{}_i \in \mathbb{R}^{n^2 \times n}$ *for* $i = 1, \ldots, m$. *The optimization problem* (7.14) *can be rewritten as*

$$\begin{aligned} \min_{\eta_1,\ldots,\eta_m} \quad & \sum_{i=1}^{m} [\Gamma_u]_{ii} R_{ii} \|\eta_i\|^2 \\ s.t. \quad & \sum_{i=1}^{m} H^c{}_i \eta_i = \omega^c vec\,(BK) \end{aligned}$$

where $H^c \triangleq [H^c{}_1 \ldots H^c{}_m] = (I \otimes B\Gamma_u)\, P_r^{-1}$ *and* $\omega^c = vec\,(BK)$.

Proof. The proof follows the derivations from Lemma D.1 and is therefore omitted. □

D.3 Proof of Lemma 7.4

Consider the optimization problem (7.15). Using the Lagrange multiplier ζ associated with the equality constraint, the optimal solution may be computed by solving

$$\max_{\zeta} \min_{\eta_1, \ldots, \eta_l} \sum_{i=1}^{l} \Big(f_i(\eta_i) - \zeta^\top H_i \eta_i \Big) + \zeta^\top \omega.$$

Introducing the local variables $\zeta_1, \ldots, \zeta_l$ and $\omega_1, \ldots, \omega_l$ satisfying $\sum_{i=1}^{l} \omega_i = \omega$ and imposing the constraint $\zeta_i = \zeta_j$ for all $i \neq j$ yields

$$\begin{aligned} \max_{\zeta_1, \ldots, \zeta_l} \quad & \sum_{i=1}^{l} \min_{\eta_i} \Big(f_i(\eta_i) - \zeta_i^\top H_i \eta_i \Big) + \zeta_i^\top \omega_i \\ \text{subject to} \quad & \zeta_i = \zeta_j, \quad \forall i, j,\ i \neq j. \end{aligned}$$

Each inner optimization problem with respect to η_i can be rewritten in terms of the convex conjugate function, i.e., $-f_i^\star(\zeta) = \min_{\eta_i} f_i(\eta_i) - \zeta^\top \eta$. Introducing the convex conjugate function in the objective function results in the following

$$\begin{aligned} \min_{\{\zeta_i\}, \{\pi_{(i,j)}\}} \quad & \sum_{i=1}^{l} \Big(f_i^\star(H_i^\top \zeta_i) - \omega_i^\top \zeta_i \Big) \\ \text{subject to} \quad & \zeta_i = \pi_{(i,j)}, \quad \forall i \in \mathcal{V},\ j \in \mathcal{N}_i, \end{aligned}$$

one can compute the value of the convex conjugate function as follows $f_i^\star(H_i^\top \zeta_i) = \frac{1}{4} \zeta_i^\top H_i S_i^{-1} H_i^\top \zeta_i$. Substituting this value in the objective function of the dual problem, gives (7.17) in Lemma 7.4.

D.4 Proof of Lemma 7.5

The computations are performed locally, since by construction only the neighbors of the faulty node j are involved in the computations. The coefficient ν_i indicates how much i compensates for the contribution of the faulty node j before the fault. Moreover, having $\bar{\omega}_i + \nu_i \bar{\omega}_j,\ i \in \mathcal{J}$ and $\sum_{i \in \mathcal{J}} \nu_i = 1$ ensures that $\sum_{i \in \mathcal{V}} \bar{\omega}_i = \omega$. Hence, each healthy node i in the neighborhood of the faulty node must solely exchange and agree on the set of parameters ν_i.

Bibliography

J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *Proceedings of the 9th IEEE International Conference on Industrial Informatics*, 2011.

I. Akyildiz and I. Kasimoglu. Wireless sensor and actuator networks: research challenges. *Ad hoc networks*, 2(4):351–367, 2004.

S. Al-Areqi, D. Gorges, S. Reimann, and S. Liu. Event-based control and scheduling codesign of networked embedded control systems. In *Proceedings of the American Control Conference*, pages 5299–5304, 2013.

J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE Transactions on*, 11(6):6–28, 2004.

J. Almeida, C. Silvestre, and A. Pascoal. Self-triggered output feedback control of linear plants in the presence of unknown disturbances. *Automatic Control, IEEE Transactions on*, 2014. to appear.

F. Altaf, J. Araújo, A. Hernandez, H. Sandberg, and K. H. Johansson. Wireless event-triggered controller for a 3d tower crane lab process. In *Proceedings of the 19th Mediterranean Conference on Control Automation*, pages 994–1001, 2011.

S. Amin, A. A. Cardenas, and S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In R. Majumdar and P. Tabuada, editors, *Hybrid Systems: Computation and Control*, volume 5469 of *Lecture Notes in Computer Science*, pages 31–45. Springer Berlin Heidelberg, 2009.

B. Aminian, J. Araújo, M. Johansson, and K. H. Johansson. GISOO: a virtual testbed for wireless cyber-physical systems. In *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 5588–5593, 2013.

A. Anta and P. Tabuada. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. *Proceedings of the 30th IEEE Real-Time Systems Symposium*, pages 3–12, 2009.

A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *Automatic Control, IEEE Transactions on*, 55(9):2030 –2042, 2010a.

A. Anta and P. Tabuada. On the minimum attention and anytime attention problems for nonlinear systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 3234–3239, 2010b.

P. Antsaklis. Special issue on hybrid systems: theory and applications a brief introduction to the theory and applications of hybrid systems. *Proceedings of the IEEE*, 88(7):879 –887, 2000.

P. Antsaklis and J. Baillieul. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1), 2007.

D. Antunes. Event-triggered control under poisson events: The role of sporadicity. In *Proceedings of the IFAC Workshop on Distributed Estimation and Control of Networked Systems*, volume 4, pages 269–276, 2013.

D. Antunes and W. Heemels. Rollout event-triggered control: Beyond periodic control performance. *Automatic Control, IEEE Transactions on*, to appear, 2014.

J. Araújo, H. Sandberg, and K. H. Johansson. Experimental validation of a localization system based on a heterogeneous sensor network. In *Proceedings of the 7th Asian Control Conference*, pages 465–470, 2009.

J. Araújo, Y. Ariba, P. Park, H. Sandberg, and K. H. Johansson. Control over a hybrid mac wireless network. In *Proceedings of the First IEEE International Conference on Smart Grid Communications*, pages 197 –202, 2010.

J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson. Self-triggered control over wireless sensor and actuator networks. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops*, pages 1–9, 2011.

J. Araújo, H. Fawzi, M. Mazo Jr., P. Tabuada, and K. H. Johansson. An improved self-triggered implementation for linear controllers. In *Proceedings of the IFAC Workshop on Distributed Estimation and Control of Networked Systems*, volume 3, pages 37–42, 2012.

J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson. System architectures, protocols and algorithms for aperiodic wireless control systems. *Industrial Informatics, IEEE Transactions on*, 10(1):175–184, 2014.

J. Araújo, A. Teixeira, E. Henriksson, and K. H. Johansson. A down-sampled controller to reduce network usage with guaranteed closed-loop performance. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014. to appear.

K.-E. Årzén. A simple event-based pid controller. In *Proceedings of the 14th IFAC World Congress*, volume 18, pages 423–428, 1999.

K. Åström and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, 2002.

K. J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, 1970. Republished by Dover Publications, 2006.

K. J. Aström. Event based control. In *Analysis and design of nonlinear control systems*, pages 127–147. Springer, 2008.

K. J. Åström and B. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of the 14th IFAC World congress*, volume 11, pages 301–306, 1999.

K. J. Åström and T. Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.

K. J. Åström and P. Kumar. Control: A perspective. *Automatica*, 50(1):3 – 43, 2014.

K. J. Åström and M. Lundh. Lund control program combines theory with hands-on experience. *Control Systems, IEEE*, 12(3):22 –30, 1992.

K. J. Åström and B. Wittenmark. *Computer controlled systems*. Prentice Hall Englewood Cliffs, NJ, 1990.

A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. MAC essentials for wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 12(2):222–248, 2010.

Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: exact solution. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3):769 – 777, 2002.

J. d. J. Barradas Berglind, T. Gommans, and M. Heemels. Self-triggered mpc for constrained linear systems and quadratic costs. In *Proceedings of the IFAC Nonlinear Model Predictive Control*, 2012.

T. Başar and P. Bernhard. *H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008.

J. Bendtsen, K. Trangbaek, and J. Stoustrup. Plug-and-play control - modifying control systems online. *Control Systems Technology, IEEE Transactions on*, 21 (1):79–93, 2013.

D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1 and 2. Athena Scientific Belmont, 1995.

M. Beschi, S. Dormido, J. Sánchez, and A. Visioli. Two degree-of-freedom design for a send-on-delta sampling pi control strategy. *Control Engineering Practice*, 30:55–66, 2014.

M. Björkbom, S. Nethi, L. M. Eriksson, and R. Jäntti. Wireless control system design and co-simulation. *Control Engineering Practice*, 19(9):1075–1086, 2011.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Engineering online library. Springer-Verlag, 2nd edition, 2006.

T. Blevins. Pid advances in industrial control. In *Proceedings of the IFAC Conference on Advances in PID Control, Brescia, Italy*, 2012.

T. Blevins, M. Nixon, and M. Zielinski. Addressing control applications using wireless devices, 2012.

T. Blevins, M. Nixon, and W. Wojsznis. Pid control using wireless measurements. In *Proceedings of the American Control Conference*, pages 790–795, 2014.

R. Blind and F. Allgower. Analysis of networked event-based control with a shared communication medium: Part i-pure aloha. In *Proceedings of the IFAC World Congress*, 2011.

R. Blind, S. Uhlich, B. Yang, and F. Allgöwer. Robustification and optimization of a kalman filter with measurement loss using linear precoding. In *Proceedings of the American Control Conference*, pages 2222–2227, 2009.

S. Bodenburg and J. Lunze. Plug-and-play control - theory and implementation. In *Proceedings of the 11th IEEE International Conference on Industrial Informatics*, pages 165–170, 2013.

A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli. System level design for clustered wireless sensor networks. *Industrial Informatics, IEEE Transactions on*, 3(3):202–214, 2007.

D. Borgers and W. Heemels. Event-separation properties of event-triggered control systems. *Automatic Control, IEEE Transactions on*, 2014.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

M. S. Branicky, V. Liberatore, and S. M. Phillips. Networked control system co-simulation for co-design. In *Proceedings of the American Control Conference*, volume 4, pages 3341–3346, 2003.

R. Brockett. Minimum attention control. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 3, 1997.

N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks*, 2007.

L. G. Bushnell. Networks and control. *Control Systems, IEEE*, 21(1):22–23, 2001.

G. Buttazzo. *Hard real-time computing systems: predictable scheduling algorithms and applications*. Springer-Verlag New York Inc, 2005.

A. Camacho, P. Martí, M. Velasco, C. Lozoya, R. Villa, J. Fuertes, and E. Griful. Self-triggered networked control systems: An experimental case study. In *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pages 123–128, 2010.

J. Campelo, F. Rodriguez, A. Rubio, R. Ors, P. Gil, L. Lemus, J. Busquets, J. Albaladejo, and J. Serrano. Distributed industrial control systems: a fault-tolerant architecture. *Microprocessors and Microsystems*, 23(2):103 – 112, 1999.

N. Cardoso de Castro, D. Quevedo, F. Garin, and C. Canudas de Wit. Smart energy-aware sensors for event-based control. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 7224–7229, 2012.

C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, second edition, 2008.

C. G. Cassandras. The event-driven paradigm for control, communication and optimization. *Journal of Control and Decision*, 1(1):3–17, 2014.

A. Cervin. *Integrated Control and Real-Time Scheduling*. PhD thesis, Department of Automatic Control, Lund University, Sweden, 2003.

A. Cervin and J. Eker. The control server: a computational model for real-time control tasks. In *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, pages 113–120, 2003.

A. Cervin and T. Henningsson. Scheduling of event-triggered controllers on a shared network. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3601 –3606, 2008.

A. Cervin, J. Eker, B. Bernhardsson, and K.-E. Årzén. Feedback–feedforward scheduling of control tasks. *Real-Time Systems*, 23(1-2):25–53, 2002.

A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. *Control Systems, IEEE*, 23(3):16–30, 2003.

A. Cervin, M. Velasco, P. Marti, and A. Camacho. Optimal on-line sampling period assignment: Theory and experiments. *Control Systems Technology, IEEE Transactions on*, 2010.

J. Chacón, J. Sánchez, A. Visioli, L. Yebra, and S. Dormido. Characterization of limit cycles for self-regulating and integral processes with PI control and send-on-delta sampling. *Journal of Process Control*, 23(6):826 – 838, 2013.

S. Chen, G. Tao, and S. Joshi. On matching conditions for adaptive state tracking control of systems with actuator failures. *Automatic Control, IEEE Transactions on*, 47(3):473–478, 2002.

T. Chen and B. A. Francis. *Optimal sampled-data control systems*. Springer Verlag, 1995.

D. Christmann, R. Gotzhein, S. Siegmund, and F. Wirth. Realization of try-once-discard in wireless multihop networks. *Industrial Informatics, IEEE Transactions on*, 10(1):17–26, 2014.

B. D. Ciubotaru and M. Staroswiecki. Extension of modified pseudo-inverse method with generalized linear quadratic stabilization. In *Proceedings of the American Control Conference*, pages 6222–6224, 2010.

R. Cogill. Event-based control using quadratic approximate value functions. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 5883 –5888, 2009.

Contiki. Contiki and COOJA wiki, 2014. URL `https://github.com/contiki-os/contiki/wiki`.

M. Cunguara, T. Mendes Oliveira e Silva, and P. Bacelar Reis Pedreiras. On the application of block transmissions for improving control over lossy networks. In *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation*, pages 1–8, 2013.

J. Dattorro. *Convex Optimization and Euclidean Distance Geometr*. Meboo Publishing, USA, 2005.

C. De Persis and P. Frasca. Robust self-triggered coordination with ternary controllers. *Automatic Control, IEEE Transactions on*, 58(12):3024–3038, 2013.

C. De Persis, R. Sailer, and F. Wirth. On a small-gain approach to distributed event-triggered control. *arXiv preprint arXiv:1010.6148*, 2010.

B. Demirel, Z. Zou, P. Soldati, and M. Johansson. Modular design of jointly optimal controllers and forwarding policies for wireless control. *Automatic Control, IEEE Transactions on*, 2014. to appear.

P. Di Marco, P. Park, C. Fischione, and K. H. Johansson. Trend: A timely, reliable, energy-efficient and dynamic wsn protocol for control applications. In *Proceedings of the IEEE International Conference on Communications (ICC)*, 2010.

D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson. Distributed event-triggered control for multi-agent systems. *Automatic Control, IEEE Transactions on*, 57 (5):1291–1297, 2012.

S. Ding. *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer-Verlag Berlin Heidelberg, 2008.

M. Donkers and W. Heemels. Output-based event-triggered control with guaranteed $l_\infty - gain$ and improved and decentralised event-triggering. *Automatic Control, IEEE Transactions on*, 57(6):1362– 1376, 2012.

M. Donkers, P. Tabuada, and W. Heemels. Minimum attention control for linear systems. *Discrete Event Dynamic Systems*, 24(2):199–218, 2014.

P. Dorato, V. Cerone, and C. Abdallah. *Linear-quadratic control: an introduction*. Simon & Schuster, 1994.

R. Dorf, M. Farren, and C. Phillips. Adaptive sampling frequency for sampled-data control systems. *Automatic Control, IRE Transactions on*, 7(1):38–47, 1962.

A. Dunkels, B. Gronvall, and T. Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462, 2004.

S. Durand and N. Marchand. Further Results on Event-Based PID Controller. In *Proceedings of the European Control Conference* , pages 1979–1984, 2009.

A. Dvir, T. Holczer, and L. Buttyan. Vera-version number and rank authentication in rpl. In *Proceedings of the 8th IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 709–714, 2011.

J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming heterogeneity-the ptolemy approach. *Proceedings of the IEEE*, 91(1):127–144, 2003.

N. Elia and S. K. Mitter. Stabilization of linear systems with limited information. *Automatic Control, IEEE Transactions on*, 46(9):1384–1400, 2001.

A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. Event-based model predictive control for the cooperation of distributed agents. In *Proceedings of the American Control Conference*, pages 6473–6478, 2012.

V. Erickson, M. Carreira-Perpinan, and A. Cerpa. Observe: Occupancy-based system for efficient reduction of hvac energy. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks*, pages 258 –269, 2011.

J. Eriksson, F. Österlind, N. Finne, N. Tsiftes, A. Dunkels, T. Voigt, R. Sauter, and P. J. Marrón. Cooja/mspsim: interoperability testing for wireless sensor networks. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, page 27, 2009.

H. Everett III. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, 11(3):399–417, 1963.

E. Eyisi, J. Bai, D. Riley, J. Weng, W. Yan, Y. Xue, X. Koutsoukos, and J. Sztipanovits. Ncswt: An integrated modeling and simulation tool for networked control systems. *Simulation Modelling Practice and Theory*, 27:90–111, 2012.

L. Fabietti. Control of hvac systems via explicit and implicit mpc: an experimental case study. Master's thesis, KTH, Automatic Control, 2014.

Y. Fan, G. Feng, Y. Wang, and C. Song. Distributed event-triggered control of multi-agent systems with combinational measurements. *Automatica*, 49(2):671–675, 2013.

P. Ferrari, A. Flammini, and E. Sisinni. Development of a co-simulation tool for wirelesshart networks. In *Sensors*, pages 553–557. Springer, 2014.

H. R. Feyzmahdavian, A. Gattami, and M. Johansson. Distributed output-feedback lqg control with delayed information sharing. In *Proceedings of the IFAC Workshop on Distributed Estimation and Control in Networked Systems*, volume 3, pages 192–197, 2012.

J. Fischer, M. Dolgov, and U. Hanebeck. On stability of sequence-based lqg control. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 6627–6633, 2013.

C. Fiter, L. Hetel, W. Perruquetti, and J.-P. Richard. A state dependent sampling for linear state feedback. *Automatica*, 48(8):1860–1867, 2012.

C. Fiter, H. Omran, L. Hetel, and J.-P. Richard. Tutorial on arbitrary and state-dependent sampling. In *Proceedings of the European Control Conference*, pages 1440–1445, 2014.

E. Fridman. *Introduction to Time-Delay Systems*. Springer International Publishing, 2014.

O. Gaddour and A. Koubâa. Rpl in a nutshell: A survey. *Computer Networks*, 2012.

Z. Gao and P. J. Antsaklis. Stability of the pseudo-inverse method for reconfigurable control systems. *International Journal of Control*, 53(3):717–729, 1991.

E. Garcia and P. Antsaklis. Model-based event-triggered control with time-varying network delays. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 1650–1655, 2011.

E. Garcia and P. J. Antsaklis. Model-based event-triggered control for systems with quantization and time-varying network delays. *Automatic Control, IEEE Transactions on*, 58(2):422–434, 2013.

E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer. Decentralised event-triggered cooperative control with limited communication. *International Journal of Control*, 86(9):1479–1488, 2013.

E. Garcia, P. J. Antsaklis, and L. A. Montestruque. *Model-Based Control of Networked Systems*. Systems & Control: Foundations & Applications. Springer International Publishing, 2014.

E. Ghadimi, A. Teixeira, M. Rabbat, and M. Johansson. The ADMM algorithm for distributed averaging: Convergence rates and optimal parameter selection. In *Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers*, 2014.

GISOO. KTH - GISOO a virtual testbed for wireless cyber-physical systems., 2014. URL `https://code.google.com/p/kth-gisoo/`.

O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2009.

A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.

T. Gommans, W. Heemels, N. Bauer, and N. v. d. Wouw. Compensation-based control for lossy communication networks. *International Journal of Control*, 86 (10):1880–1897, 2013.

T. Gommans, D. Antunes, M. Donkers, P. Tabuada, and W. Heemels. Self-triggered linear quadratic control. *Automatica*, to appear, 2014.

M. Grant, S. Boyd, and Y. Ye. CVX: Matlab software for disciplined convex programming, 2008.

L. Grüne, S. Hirche, O. Junge, P. Koltai, D. Lehmann, J. Lunze, A. Molin, R. Sailer, M. Sigurani, C. Stöker, and F. Wirth. Event-based control. In J. Lunze, editor, *Control Theory of Digitally Networked Dynamic Systems*, pages 169–261. Springer International Publishing, 2014.

K. Gu, J. Chen, and V. L. Kharitonov. *Stability of time-delay systems*. Springer, 2003.

V. Gupta, B. Hassibi, and R. M. Murray. Optimal LQG control across packet-dropping links. *Systems & Control Letters*, 56(6):439 – 446, 2007.

V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi. Data transmission over networks for estimation and control. *Automatic Control, IEEE Transactions on*, 54(8):1807 –1819, 2009.

G. P. Halkes and K. Langendoen. Experimental evaluation of simulation abstractions for wireless sensor network MAC protocols. *EURASIP Journal on Wireless Communications and Networking*, 2010:24, 2010.

S. Han, X. Zhu, K. Aloysius, M. Nixon, T. Blevins, and D. Chen. Control over wirelesshart network. In *Proceedings of the 36th Conference on IEEE Industrial Electronics Society*, pages 2114–2119, 2010.

S. Han, X. Zhu, A. Mok, D. Chen, and M. Nixon. Reliable and real-time communication in industrial wireless mesh networks. In *Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 3–12, 2011.

O. Härkegård and S. T. Glad. Resolving actuator redundancy: optimal control vs. control allocation. *Automatica*, 41(1):137 – 144, 2005.

HART Communication Foundation. *WirelessHART Data Sheet*, 2007. Datasheet.

J. Hauer. TKN15.4: An IEEE 802.15.4 MAC Implementation for TinyOS 2. Technical Report TKN-08-003, Telecommunication Networks Group, Technical Univ. Berlin, 2009.

L. He, D. Han, X. Wang, and L. Shi. Optimal linear state estimation over a packet-dropping network using linear temporal coding. *Automatica*, 49(4):1075–1082, 2013.

W. Heemels, J. Sandee, and P. van den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, pages 81(4), 571–590, 2008.

W. Heemels, A. Teel, N. van de Wouw, and D. Nesicand. Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *Automatic Control, IEEE Transactions on*, 55(8):1781 –1796, 2010.

W. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3270–3285, 2012.

W. Heemels, M. Donkers, and A. R. Teel. Periodic event-triggered control for linear systems. *Automatic Control, IEEE Transactions on*, 58(4):847–861, 2013.

D. Henriksson and A. Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4469–4474, 2005.

E. Henriksson. *Predictive Control for Wireless Networked Systems in Process Industry*. PhD thesis, KTH, Automatic Control, ACCESS Linnaeus Centre, 2014.

E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. Self-triggered model predictive control for network scheduling and control. In *Proceedings of the IFAC Advanced Control of Chemical Processes*, 2012.

J. Heo, J. Hong, and Y. Cho. Earq: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks. *Industrial Informatics, IEEE Transactions on*, 5(1):3 –11, 2009.

A. Hernandez. Modification of the IEEE 802.15.4 implementation extended gts implementation. Technical report, KTH Royal Institute of Technology, 2011. URL `http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x-contrib/kth/index.html`.

A. Hernandez and P. Park. Ieee 802.15. 4 implementation based on tkn15. 4 using tinyos. Technical report, Technical report, KTH Electrical Engineering, Stockholm, 2011.

A. Hernandez, J. Faria, J. Araújo, P. Park, H. Sandberg, and K. H. Johansson. Inverted pendulum control over an IEEE 802.15.4 wireless sensor and actuator network. In *Proceedings of the European Conference on Wireless Sensor Networks*, 2011.

J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

H. Hirano, M. Mukai, T. Azuma, and M. Fujita. Optimal control of discrete-time linear systems with network-induced varying delay. In *Proceedings of the American Control Conference*, pages 1419 – 1424 vol. 2, 2005.

G. Hoblos, M. Staroswiecki, and A. Aitouche. Optimal design of fault tolerant sensor networks. In *Proceedings of the IEEE International Conference on Control Applications*, pages 467–472, 2000.

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

D. Hristu-Varsakelis and W. S. Levine. *Handbook of networked and embedded control systems*. Springer, 2005.

I. Hwang, S. Kim, Y. Kim, and C. Seah. A survey of fault detection, isolation, and reconfiguration methods. *Control Systems Technology, IEEE Transactions on*, 18(3):636–653, 2010.

IEEE 802.15.4. *IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, 2006. URL `http://www.ieee802.org/15/pub/TG4.html`.

IEEE 802.15.4e. *IEEE 802.15 task group 4e: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, 2012. URL `http://www.ieee802.org/15/pub/TG4e.html`.

International Society of Automation. ISA-SP100 wireless systems for automation website. `http://www.isa.org/isa100`, 2010.

R. Isermann. Model-based fault-detection and diagnosis - status and applications. *Annual Reviews in Control*, 29(1):71 – 85, 2005.

H. Ishii and B. A. Francis. *Limited Data Rate in Control Systems with Networks*. Springer-Verlag New York, Inc., 2002.

X. Z. Jin and G. H. Yang. Distributed fault-tolerant control systems design against actuator faults and faulty interconnection links: An adaptive method. In *Proceedings of the American Control Conference*, pages 2910–2915, 2009.

B. Johansson. *On Distributed Optimization in Networked Systems*. PhD thesis, KTH, Automatic Control, 2008.

S. M. Joshi and P. Patre. Direct model reference adaptive control with actuator failures and sensor bias. *Journal of Guidance, Control, and Dynamics*, 37(1): 312–317, 2013.

C. Kambhampati, R. Patton, and F. Uppal. Reconfiguration in networked control systems. In *Proceedings of the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, page 126, 2007.

A. Khan, J. Araújo, P. di Marco, D. Lehmann, E. Henriksson, H. Sandberg, and K. H. Johansson. Design and implementation of multi-hop wireless protocols for process control applications. *Industrial Informatics, Transactions on*, 2014. submitted.

P. Khargonekar. State-space h_∞ control theory and the lqg problem. In A. C. Antoulas, editor, *Mathematical System Theory*, pages 159–176. Springer Berlin Heidelberg, 1991.

G. A. Kiener, D. Lehmann, and K. H. Johansson. Actuator saturation and anti-windup compensation in event-triggered control. *Discrete Event Dynamic Systems*, 24(2):173–197, 2014.

O. Kilkki and M. Bjorkbom. Optimization of control transmissions by event-driven model prediction. In *Proceedings of the 39th Conference of the IEEE Industrial Electronics Society*, pages 5668–5673, 2013.

K.-D. Kim and P. R. Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, 2012.

Y. Kim, T. Schmid, M. B. Srivastava, and Y. Wang. Challenges in resource monitoring for residential spaces. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 1–6, 2009.

J. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, and A. Terzis. Evaluating the performance of rpl and 6lowpan in tinyos. In *Proceedings of the Workshop on Extending the Internet to Low Power and Lossy Networks*, 2011a.

J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, A. Terzis, A. Dunkels, and D. Culler. Contikirpl and tinyrpl: Happy together. In *Proceedings of the Workshop on Extending the Internet to Low Power and Lossy Networks*, 2011b.

K. Kobayashi and K. Hiraishi. Self-triggered model predictive control with delay compensation for networked control systems. In *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society*, pages 3200–3205, Oct 2012.

T. Kohtamaki, M. Pohjola, J. Brand, and L. M. Eriksson. Piccsim toolchain-design, simulation and automatic implementation of wireless networked control systems. In *Proceedings of the International Conference on Networking, Sensing and Control*, pages 49–54, 2009.

KTH. KTH HVAC Testbed, 2014. URL `http://hvac.ee.kth.se/`.

A. Lamperski and J. C. Doyle. On the structure of state-feedback lqg controllers for distributed systems with communication delays. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 6901–6906, 2011.

M. Larsson, J. Lindberg, J. Lycke, K. Hansson, A. Khakulov, E. Ringh, F. Svensson, I. Tjernberg, A. Alam, J. Araújo, F. Farokhi, E. Ghadimi, A. Teixeira, D. V. Dimarogonas, and K. H. Johansson. Towards an indoor testbed for mobile networked control systems. In *Proceedings of the First Workshop on Research, Development and Education on Unmanned Aerial Systems*, 2011.

A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo. 6lowpan: a study on qos security threats and countermeasures using intrusion detection system approach. *International Journal of Communication Systems*, 25(9):1189–1212, 2012.

J. H. Lee, W. H. Kwon, and J.-W. Lee. Quadratic stability and stabilization of linear systems with frobenius norm-bounded uncertainties. *Automatic Control, IEEE Transactions on*, 41(3):453–456, 1996.

D. Lehmann and J. Lunze. Extension and experimental evaluation of an event-based state-feedback approach. *Control Engineering Practice*, 19(2):101 – 112, 2011.

D. Lehmann and J. Lunze. Event-based control with communication delays and packet losses. *International Journal of Control*, 85(5):563–577, 2012.

D. Lehmann, J. Lunze, and K. H. Johansson. Comparison between sampled-data control, deadband control and model-based event-triggered control. In *Proceedings of the 4th Conference on Analysis and Design of Hybrid Systems. Eindhoven, Netherlands*, 2012.

M. Lemmon. Event-triggered feedback in control, estimation, and optimization. In A. Bemporad, M. Heemels, and M. Johansson, editors, *Networked Control Systems*, volume 406 of *Lecture Notes in Control and Information Sciences*, pages 293–358. Springer Berlin / Heidelberg, 2011.

P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and scalable simulation of entire tinyos applications. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, pages 126–137, 2003.

P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al. TinyOS: An operating system for wireless sensor networks. *Ambient Intelligence*, 2004.

B. Li, Z. Sun, K. Mechitov, G. Hackmann, C. Lu, S. Dyke, G. Agha, and B. F. Spencer Jr. Realistic case studies of wireless structural control. In *Proceedings of the International Conference on Cyber-Physical Systems*, 2013.

L. Li, X. Wang, and M. Lemmon. Stabilizing bit-rates in quantized event triggered control systems. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 245–254, 2012.

B. Lincoln and B. Bernhardsson. Optimal control over networks with long random delays. In *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, 2000.

B. Lincoln and B. Bernhardsson. LQR optimization of linear system switching. *Automatic Control, IEEE Transactions on*, 47(10):1701–1705, 2002.

K. Liu, E. Fridman, and L. Hetel. Networked control systems: A time-delay approach. In *Proceedings of the European Control Conference*, pages 1434–1439, 2014.

X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 3, pages 2823–2829, 2004.

C. Lozoya, P. Martí, M. Velasco, J. M. Fuertes, and E. X. Martin. Resource and performance trade-offs in real-time embedded control systems. *Real-Time Systems*, 49(3):267–307, 2013.

J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, pages 211–224, 2010.

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

R. Luck and A. Ray. Experimental verification of a delay compensation algorithm for integrated communication and control systems. *International Journal of Control*, 59(6):1357–1372, 1994.

J. Lunze. *Control Theory of Digitally Networked Dynamic Systems*. Springer, 2014.

J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

J. Lunze and J. H. Richter. Reconfigurable fault-tolerant control: a tutorial introduction. *European Journal of Control*, 14(5):359–386, 2008.

J. Lunze and T. Steffen. Control reconfiguration after actuator failures using disturbance decoupling methods. *Automatic Control, IEEE Transactions on*, 51(10): 1590–1601, 2006.

J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. S. Sastry. Dynamical properties of hybrid automata. *Automatic Control, IEEE Transactions on*, 48 (1):2–17, 2003.

J. Maciejowski. Reconfigurable control using constrained optimization. In *Proceeding of European Control Conference*, pages 107–130, 1997.

N. Marchand, S. Durand, and J. F. G. Castellanos. A general formula for event-based stabilization of nonlinear systems. *Automatic Control, IEEE Transactions on*, 58(5):1332–1337, 2013.

P. Martí, C. Lin, S. Brandt, M. Velasco, and J. Fuertes. Optimal state feedback based resource allocation for resource-constrained control tasks. In *Proceedings of the 23rd IEEE Real-Time Systems Symposium*, 2004.

MathWorks. Simulink - Simulation and Model-Based Design, 2014. URL `http://www.mathworks.se/products/simulink/`.

N. Matni and J. Doyle. Optimal distributed lqg state feedback with varying communication delay. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 5890–5896, 2013.

S. E. Mattsson, H. Elmqvist, and M. Otter. Physical system modeling with modelica. *Control Engineering Practice*, 6(4):501–510, 1998.

M. Mazo and P. Tabuada. On event-triggered and self-triggered control over sensor/actuator networks. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 435–440, 2008.

M. Mazo Jr. and P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *Automatic Control, IEEE Transactions on*, 56(10): 2456 –2461, 2011.

M. Mazo Jr., A. Anta, and P. Tabuada. On self-triggered control for linear systems: Guarantees and complexity. In *Proceedings of the European Control Conference*, 2009.

M. Mazo Jr., A. Anta, and P. Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46(8):1310–1314, 2010.

X. Meng and T. Chen. Optimal sampling and performance comparison of periodic and event based impulse control. *Automatic Control, IEEE Transactions on*, 57 (12):3252–3259, 2012.

A. R. Mesquita, J. P. Hespanha, and G. N. Nair. Redundant data transmission in control/estimation over lossy networks. *Automatica*, 48(8):1612–1620, 2012.

S. H. M.H. Mamduhi, A. Molin. Event-based scheduling of multi-loop stochastic systems over shared communication channels. In *Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems*, pages 266–273, 2014.

J. Mirkovic and T. Benzel. Teaching cybersecurity with deterlab. *Security & Privacy, IEEE*, 10(1):73–76, 2012.

M. Miskowicz. Send-on-delta concept: an event-based data reporting strategy. *sensors*, 6(1):49–63, 2006.

M. Moayedi, Y. K. Foo, and Y. C. Soh. Networked lqg control over unreliable channels. *International Journal of Robust and Nonlinear Control*, 23(2):167–189, 2013.

A. Molin and S. Hirche. Structural characterization of optimal event-based controllers for linear stochastic systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

A. Molin and S. Hirche. On the optimality of certainty equivalence for event-triggered control systems. *Automatic Control, IEEE Transactions on*, 58(2): 470–474, 2013.

A. Molin and S. Hirche. Price-based adaptive scheduling in multi-loop control systems with resource constraints. *Automatic Control, IEEE Transactions on*, PP(99):1–1, 2014.

J. Moon and T. Başar. Control over lossy networks: A dynamic game approach. In *Proceedings of the American Control Conference*, pages 5367–5372, 2014.

R. Moraes, F. Vasques, P. Portugal, and J. Fonseca. Vtp-csma: A virtual token passing approach for real-time communication in IEEE 802.11 wireless networks. *Industrial Informatics, IEEE Transactions on*, 3(3):215–224, 2007.

D. Moss and P. Levis. Box-macs: Exploiting physical and link layer boundaries in low-power networking. Technical report, Technical Report SING-08-00, Stanford University, 2008.

G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, 95(1):108–137, 2007.

A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 55(4):922–938, 2010.

Nest Labs. Nest protect carbon monoxide. field study: Results from. november 2013 to may 2014. White paper, Nest Labs, 2014.

J. Nilsson. *Real-Time control systems with delays*. PhD thesis, Lund Institute of Technology, 1998. Ph.D. thesis.

C. Nowzari and J. Cortés. Self-triggered coordination of robotic networks for optimal deployment. *Automatica*, 48(6):1077–1087, 2012.

F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *Proceedings of the 31st IEEE Conference on Local Computer Networks*, pages 641–648, 2006.

M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler. Standardized protocol stack for the internet of (important) things. *Communications Surveys Tutorials, IEEE*, 15(3):1389–1406, 2013.

A. Parisio, M. Molinari, D. Varagnolo, and K. Johansson. A scenario-based predictive control approach to building hvac management systems. In *Proceedings of the IEEE International Conference on Automation Science and Engineering*, pages 428–435, 2013.

P. Park, J. Araújo, and K. H. Johansson. Wireless networked control system co-design. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pages 486 –491, 2011.

G. Pattarello, L. Wei, A. Ebadat, B. Wahlberg, and K. H. Johansson. The KTH open testbed for smart hvac control. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 34:1–34:2, 2013.

R. J. Patton. Fault-tolerant control systems: The 1997 situation. In *Proceedings of the IFAC symposium on fault detection supervision and safety for technical processes*, volume 3, pages 1033–1054, 1997.

R. J. Patton, C. Kambhampati, A. Casavola, P. Zhang, S. Ding, and D. Sauter. A generic strategy for fault-tolerance in control systems distributed over a network. *European journal of control*, 13(2):280–296, 2007.

N. Pereira, B. Andersson, and E. Tovar. Widom: A dominance protocol for wireless medium access. *Industrial Informatics, IEEE Transactions on*, 3(2):120–130, 2007.

K. Pister and L. Doherty. TSMP: time synchronized mesh protocol. In *Proceedings of the IASTED International Symposium*, volume 635, pages 391–398, 2008.

J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch. Comparative study of energy-efficient sampling approaches for wireless control networks. *Industrial Informatics, IEEE Transactions on*, 6(3):416–424, 2010.

N. J. Ploplys, P. A. Kawka, and A. G. Alleyne. Closed-loop control over wireless networks. *Control Systems, IEEE*, 24(3):58 – 71, 2004.

J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, 2005.

R. Poovendran, K. Sampigethaya, S. K. S. Gupta, I. Lee, K. V. Prasad, D. Corman, and J. Paunicka. Special issue on cyber - physical systems. *Proceedings of the IEEE*, 100(1):6 –12, 2012.

R. Postoyan, A. Anta, W. Heemels, P. Tabuada, and D. Nesic. Periodic event-triggered control for nonlinear systems. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 7397–7402, 2013.

A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos. A modeling approach on the telosb wsn platform power consumption. *Journal of Systems and Software*, 83(8):1355–1363, 2010.

Quanser. *Coupled Water Tanks*, 2014. URL `http://www.quanser.com/english/downloads/products/Specialty/CoupledTanks_PIS_031708.pdf`.

D. E. Quevedo and V. Gupta. Sequence-based anytime control. *Automatic Control, IEEE Transactions on*, 58(2):377–390, 2013.

D. E. Quevedo and D. Nesic. Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts. *Automatic Control, IEEE Transactions on*, 56(2):370–375, 2011.

M. Rabi and K. H. Johansson. Event-triggered strategies for industrial control over wireless networks. In *Proceedings of the International Conference on Wireless Internet*, pages 1–7, 2008.

M. Rabi and K. H. Johansson. Scheduling packets for event-triggered control. In *Proceedings of 10th European Control Conference*, pages 3779 – 3784, 2009.

C. Ramesh. *State-based Channel Access for a Network of Control Systems*. PhD thesis, KTH, Automatic Control, ACCESS Linnaeus Centre, 2014.

C. Ramesh, H. Sandberg, and K. H. Johansson. On the dual effect in state-based scheduling of networked control systems. In *Proceedings of the American Control Conference*, 2011.

C. Ramesh, H. Sandberg, and K. H. Johansson. Design of state-based schedulers for a network of control loops. *Automatic Control, IEEE Transactions on*, 58(8): 1962–1975, 2013.

C. Ramesh, D. Jenkins, J. Araujo, H. Sandberg, and K. H. Johansson. State-based priorities for tournaments in wireless networked control systems. *submitted to journal publication*, 2014.

J. H. Richter, W. P. M. H. Heemels, N. van de Wouw, and J. Lunze. Reconfigurable control of piecewise affine systems with actuator and sensor faults: Stability and tracking. *Automatica*, 47(4):678–691, 2011.

S. Riverso, M. Farina, and G. Ferrari-Trecate. Plug-and-play decentralized model predictive control for linear systems. *Automatic Control, IEEE Transactions on*, 58(10):2608–2614, 2013.

C. Robinson and P. Kumar. Sending the most recent observation is not optimal in networked control: Linear temporal coding and towards the design of a control specific transport protocol. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 334–339, 2007.

R. Rom and M. Sidi. *Multiple access protocols: performance and analysis*. Springer-Verlag New York, Inc., 1990.

C. Rozell and D. Johnson. Power scheduling for wireless sensor and actuator networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, pages 470–478, 2007.

Á. Ruiz, J. E. Jiménez, J. Sánchez, and S. Dormido. A practical tuning methodology for event-based pi control. *Journal of Process Control*, 24(1):278–295, 2014.

A. Saifullah, Y. Xu, C. Lu, and Y. Chen. Real-time scheduling for WirelessHART networks. *Proceedings of the 31st IEEE Real-Time Systems Symposium*, 2010.

A. Saifullah, C. Wu, P. B. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen. Near optimal rate selection for wireless control systems. In *Proceedings of the 18th IEEE Real Time and Embedded Technology and Applications Symposium*, pages 231–240, 2012.

T. Samad and A. Annaswamy. The impact of control technology. *IEEE Control Systems Society*, 2011.

T. Samad, P. McLaughlin, and J. Lu. System architecture for process automation: Review and trends. *Journal of Process Control*, 17(3):191 – 201, 2007.

S. Samii, P. Eles, Z. Peng, P. Tabuada, and A. Cervin. Dynamic scheduling and control-quality optimization of self-triggered control applications. In *Proceedings of the 31st IEEE Real-Time Systems Symposium*, pages 95–104, 2010.

J. Sánchez, A. Visioli, and S. Dormido. A two-degree-of-freedom pi controller based on events. *Journal of Process Control*, 21(4):639 – 651, 2011.

L. Schenato. Optimal estimation in networked control systems subject to random delay and packet drop. *Automatic Control, IEEE Transactions on*, 53(5):1311–1317, 2008.

L. Schenato, B. Sinopoli, M. Franceschetti, K. Poola, and S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1): 163–187, 2007.

P. Seiler and R. Sengupta. An H_∞ approach to networked control. *Automatic Control, IEEE Transactions on*, 50(3):356 – 364, 2005.

M. M. Seron, J. A. De Doná, and J. H. Richter. Integrated sensor and actuator fault-tolerant control. *International Journal of Control*, 86(4):689–708, 2013.

G. Seyboth, D. V. Dimarogonas, and K. H. Johansson. Control of multi-agent systems via event-based communication. In *Proceedings of the 18th IFAC World Congress*, 2011.

W. Shen, T. Zhang, and M. Gidlund. Joint routing and mac for critical traffic in industrial wireless sensor and actuator networks. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, pages 1–6, 2013.

W. Shen, T. Zhang, F. Barac, and M. Gidlund. Prioritymac: A priority-enhanced mac protocol for critical traffic in industrial wireless sensor and actuator networks. *Industrial Informatics, IEEE Transactions on*, 10(1):824–835, 2014.

L. Shi, Y. Yuan, and J. Chen. Finite horizon lqr control with limited controller-system communication. *Automatic Control, IEEE Transactions on*, 58(7):1835–1841, 2013.

N. Z. Shor, K. C. Kiwiel, and A. Ruszcaynski. *Minimization methods for non-differentiable functions*. Springer-Verlag New York, Inc., 1985.

Y. Shoukry, J. Araújo, P. Tabuada, M. Srivastava, and K. H. Johansson. Minimax control for cyber-physical systems under network packet scheduling attacks. In *Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*, pages 93–100, 2013.

J. Sijs and M. Lazar. Event based state estimation with time synchronous updates. *Automatic Control, IEEE Transactions on*, 57(10):2650–2655, 2012.

J. Silvo, L. M. Eriksson, M. Bjorkbom, and S. Nethi. Ultra-reliable and real-time communication in local wireless applications. In *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society*, pages 5611–5616, 2013.

K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson. Scheduling smart home appliances using mixed integer linear programming. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 5144–5149, 2011.

M. Souza, G. S. Deaecto, J. C. Geromel, and J. Daafouz. Self-triggered linear quadratic networked control. *Optimal Control Applications and Methods*, 35(5): 524–538, 2014.

M. Staroswiecki. On fault tolerant estimation in sensor networks. In *Proceedings of the European Control Conference*, 2003.

M. Staroswiecki. Intelligent sensors: a functional view. *Industrial Informatics, IEEE Transactions on*, 1(4):238–249, 2005a.

M. Staroswiecki. Fault tolerant control : The pseudo-inverse method revisited. In *Proceedings of the 16th Triennial World Congress*, 2005b.

M. Staroswiecki and D. Berdjag. A general fault tolerant linear quadratic control strategy under actuator outages. *International Journal of Systems Science*, 41 (8):971–985, 2010.

M. Staroswiecki and F. Cazaurang. Fault recovery by nominal trajectory tracking. In *Proceedings of the American Control Conference*, pages 1070–1075, 2008.

M. Staroswiecki, H. Yang, and B. Jiang. Progressive accommodation of parametric faults in linear quadratic control. *Automatica*, 43(12):2070 – 2076, 2007.

C. Stöcker, D. Vey, and J. Lunze. Decentralized event-based control: Stability analysis and experimental evaluation. *Nonlinear Analysis: Hybrid Systems*, 10: 141–155, 2013.

M. Strübe, F. Lukas, B. Li, and R. Kapitza. Drysim: Simulation-aided deployment-specific tailoring of mote-class wsn software. In *Proceedings of the The 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2014.

J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

T. Suia, K. You, M. Fu, and D. Marelli. Stability of mmse state estimators over lossy networks using linear coding. *Automatica*, 2014. to appear.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *Automatic Control, IEEE Transactions on*, 52(9):1680–1685, 2007.

P. Tallapragada and N. Chopra. On event triggered tracking for nonlinear systems. *Automatic Control, IEEE Transactions on*, 58(9):2343–2348, 2013.

P. Tallapragada and J. Cortés. Event-triggered stabilization of linear systems under bounded bit rates. *arXiv preprint arXiv:1405.6196*, 2014.

P. L. Tang and C. de Silva. Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *Control Systems Technology, IEEE Transactions on*, 14(4):707 – 718, 2006a.

P. L. Tang and C. de Silva. Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *Control Systems Technology, IEEE Transactions on*, 14(4):707 – 718, 2006b.

G. Tao, S. Joshi, and X. Ma. Adaptive state feedback and tracking control of systems with actuator failures. *Automatic Control, IEEE Transactions on*, 46 (1):78–95, 2001.

S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. *Automatic Control, IEEE Transactions on*, 49(9):1549 – 1561, 2004.

A. Teixeira, J. Araújo, H. Sandberg, and K. H. Johansson. Distributed actuator reconfiguration in networked control systems. In *Proceedings of the IFAC Workshop on Distributed Estimation and Control in Networked Systems*, volume 4, pages 61–68, 2013.

P. Teixeira, D. Dimarogonas, K. Johansson, and J. Sousa. Multi-agent coordination with event-based communication. In *Proceedings of the American Control Conference*, pages 824–829, 2010.

U. Tiberi, J. Araujo, and K. H. Johansson. On event-based PI control of first-order processes. In *Proceedings of the IFAC Conference on Advances in PID control*, 2012a.

U. Tiberi, C.-F. Lindberg, and A. J. Isaksson. Dead-band self-triggered pi control for processes with dead-time. In *Proceedings of the IFAC Conference on Advances in PID Control*, 2012b.

U. Tiberi, C. Fischione, K. H. Johansson, and M. D. Di Benedetto. Energy-efficient sampling of networked control systems over ieee 802.15. 4 wireless networks. *Automatica*, 49(3):712–724, 2013.

Y. Tipsuwan and M.-Y. Chow. Control methodologies in networked control systems. *Control engineering practice*, 11(10):1099–1111, 2003.

S. Trimpe and R. D'Andrea. An experimental demonstration of a distributed and event-based state estimation algorithm. In *Proceedings of the 18th IFAC World Congress*, 2011.

S. Trimpe and R. D'Andrea. Event-based state estimation with variance-based triggering. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 6583–6590, 2012.

UK Department of Trade and Industry. *DTI, Energy Trends 2005*, 2011. URL `http://webarchive.nationalarchives.gov.uk/+/http://www.berr.gov.uk/files/file10735.pdf`.

U.S. Department of Energy. *Energy Efficiency Trends in Residential and Commercial Buildings*, 2008. URL `http://apps1.eere.energy.gov/buildings/publications/pdfs/corporate/bt_stateindustry.pdf`.

O. Valle, C. Montez, P. Portugal, F. Vasques, and D. Costa. Expansion of the available use classes in ieee 802.15.4 networks for usage in the industrial environment. In *Proceedings of the International Symposium on Wireless Communication Systems*, pages 131–135, 2012.

P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen. Event-based model predictive control for networked control systems. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 567–572, 2009.

V. Vasyutynskyy and K. Kabitzsch. Simple pid control algorithm adapted to deadband sampling. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, pages 932 –940, 2007.

M. Velasco, J. Fuertes, and P. Martí. The self triggered task model for real-time control systems. In *Proceedings of the 24th IEEE Real-Time Systems Symposium (work in progress)*, pages 67–70, 2003.

M. Velasco, P. Martí, and E. Bini. On lyapunov sampling for event-driven controllers. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 6238–6243, 2009.

P. Voulgaris and S. Jiang. Failure-robust distributed controller architectures. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 1, pages 513–518 Vol.1, 2004.

G. Walsh and H. Ye. Scheduling of networked control systems. *Control Systems, IEEE*, 21(1):57–65, 2001.

G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *Control Systems Technology, IEEE Transactions on*, pages 2876–2880, 1999.

X. Wang and M. Lemmon. Self-triggered feedback control systems with finite-gain l2 stability. *Automatic Control, IEEE Transactions on*, 45:452–467, 2009.

X. Wang and M. Lemmon. Event-triggering in distributed networked control systems. *Automatic Control, IEEE Transactions on*, 56(3):586 –601, 2011.

WCPS. Wireless cyber-physical simulator, 2013. URL `http://wsn.cse.wustl.edu/`.

J. Weimer, J. Araújo, A. Hernandez, and K. H. Johansson. Periodic constraint-based control using dynamic wireless sensor scheduling. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 4789–4796, 2011.

J. Weimer, S. A. Ahmadi, J. Araújo, F. M. Mele, D. Papale, I. Shames, H. Sandberg, and K. H. Johansson. Active actuator fault detection and diagnostics in hvac systems. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 107–114, 2012a.

J. Weimer, J. Araújo, and K. H. Johansson. Distributed event-triggered estimation in networked systems. In *Proceedings of the 4th IFAC conference on Analysis and Design of Hybrid Systems*, pages 178–185, 2012b.

J. Weimer, J. Araújo, M. Amoozadeh, S. Ahmadi, H. Sandberg, and K. Johansson. Parameter-invariant actuator fault diagnostics in cyber-physical systems with application to building automation. In D. C. Tarraf, editor, *Control of Cyber-Physical Systems*, volume 449 of *Lecture Notes in Control and Information Sciences*, pages 179–196. Springer International Publishing, 2013.

M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolie. Design of a low-latency, high-reliability wireless communication system for control applications. In *Proceedings of the IEEE International Conference on Communications*, pages 3829–3835, 2014.

A. Westenberger, B. Duraisamy, M. Munz, M. Muntzinger, M. Fritzsche, and K. Dietmayer. Impact of out-of-sequence measurements on the joint integrated probabilistic data association filter for vehicle safety systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 438 –443, 2012.

A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *Industrial Informatics, IEEE Transactions on*, 4(2):102 –124, 2008.

T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. Rpl: Ipv6 routing protocol for low-power and lossy networks. *Network Working Group Internet Drafts*, 2012.

J. Wu, Q.-S. Jia, K. H. Johansson, and L. Shi. Event-based sensor data scheduling: Trade-off between communication rate and estimation quality. *Automatic Control, IEEE Transactions on*, 58(4):1041–1046, 2013.

N. Wu, K. Zhou, and G. Salomon. Control reconfigurability of linear time-invariant systems. *Automatica*, 36(11):1767 – 1771, 2000.

Y. Xu and J. P. Hespanha. Optimal communication logics in networked control systems. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 4, pages 3527 – 3532 Vol.4, 2004.

I. Yang, D. Kim, and D. Lee. Fault-tolerant control strategy based on control allocation using smart actuators. In *Proceedings of the Conference on Control and Fault-Tolerant Systems*, pages 377–381, 2010.

T. Yang. Networked control system: a brief survey. *Control Theory and Applications, IEE Proceedings*, 153(4):403 – 412, 2006.

J. Yépez, M. Velasco, P. Marti, E. Martin, and J. M. Fuertes. One-step finite horizon boundary with varying control gain for event-driven networked control systems. In *Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society*, pages 2606–2611. IEEE, 2011.

J. Yook, D. Tilbury, and N. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *Control Systems Technology, IEEE Transactions on*, 10(4):503 –518, 2002.

H. Yu and P. J. Antsaklis. Event-triggered output feedback control for networked control systems using passivity: Achieving stability in the presence of communication delays and signal quantization. *Automatica*, 49(1):30 – 38, 2013.

J. Yu and L.-C. Fu. A new compensation framework for LQ control over lossy networks. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 6610–6614, 2013.

M. Yu, L. Wang, G. Xie, and T. Chu. Stabilization of networked control systems with data packet droupout via switched system approach. In *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*, pages 362–367, 2004.

D. Yue, Q.-L. Han, and C. Peng. State feedback controller design of networked control systems. In *Proceedings of the IEEE International Conference on Control Applications*, volume 1, pages 242–247, 2004.

P. Zand, E. Mathews, P. Havinga, S. Stojanovski, E. Sisinni, and P. Ferrari. Implementation of wirelesshart in the ns-2 simulator and validation of its correctness. *Sensors*, 14(5):8633–8668, 2014.

K. Zhang, X. Li, and Y. Zhu. Optimal update with out-of-sequence measurements. *Signal Processing, IEEE Transactions on*, 53(6):1992 – 2004, 2005.

L. Zhang and D. Hristu-Varsakelis. Communication and control co-design for networked control systems. *Automatica*, 42(6):953 – 958, 2006.

W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *Control Systems, IEEE*, 21(1):84 –99, 2001.

Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229 – 252, 2008.

Q. Zheng and K. Shin. On the ability of establishing real-time channels in point-to-point packet-switched networks. *Communications, IEEE Transactions on*, 42 (234):1096–1105, 1994.

M. Zhong and C. Cassandras. Asynchronous distributed optimization with event-driven communication. *Automatic Control, IEEE Transactions on*, 55(12):2735–2750, 2010.

T. Zhong, M. Zhan, Z. Peng, and W. Hong. Industrial wireless communication protocol WIA-PA and its interoperation with foundation fieldbus. In *Proceedings of the International Conference on Computer Design and Applications*, volume 4, pages 370–374, 2010.

Z. Zinonos, V. Vassiliou, and T. Christofides. Radio propagation in industrial wireless sensor network environments: From testbed to simulation evaluation. In *Proceedings of the 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, pages 125–132. ACM, 2012.

Z. Zou, B. Demirel, and M. Johansson. Minimum-energy packet forwarding policies for guaranteed lqg performance in wireless control systems. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3341–3346, 2012.