KTH Electrical Engineering

Design and Implementation of Resource-Aware Wireless Networked Control Systems

JOSÉ ARAÚJO

Licentiate Thesis
Stockholm, Sweden 2011

KTH School of Electrical Engineering
Automatic Control Lab
SE-100 44 Stockholm
SWEDEN

2011:065
1653-5146
978-91-7501-122-6

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i reglerteknik fredagen den 14 oktober 2011, klockan 10:15 i sal L1, Kungliga Tekniska högskolan, Drottning Kristinas väg 30, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Networked control over wireless sensor and actuator systems is of growing importance in many application domains. Energy and communication bandwidth are scarce resources in such systems. Despite that feedback control might only be needed occasionally, sensor and actuator communications are often periodic and with high frequency in today's implementations. In this thesis, resource-constrained wireless networked control systems with an adaptive sampling period are considered.

Our first contribution is a system architecture for aperiodic wireless networked control. As the underlying data transmission is performed over a shared wireless network, we identify scheduling policies and medium access controls that allow for an efficient implementation of sensor communication. We experimentally validate three proposed mechanisms and show that best performance is obtained by a hybrid scheme, combining the advantages of event- and self-triggered control as well as the possibilities provided by contention-based and contention-free medium access control.

In the second contribution, we propose an event-triggered PI controller for wireless process control systems. A novel triggering mechanism which decides the transmission instants based on an estimate of the control signal is proposed. It addresses some side-effects that have been discovered in previous PI proposals, which trigger on the state of the process. Through simulations we demonstrate that the new PI controller provides setpoint tracking and disturbance rejection close to a periodic PI controller, while reducing the required network resources.

The third contribution proposes a co-design of feedback controllers and wireless medium access. The co-design is formulated as a constrained optimization problem, whereby the objective function is the energy consumption of the network and the constraints are the packet loss probability and delay, which are derived from the performance requirements of the control systems. The design framework is illustrated in a numerical example.

Acknowledgements

Many people have greatly contributed to make the work presented in this thesis possible, for which they deserve a huge thanks. First of all, I would like express my gratitude to my supervisor Karl H. Johansson for giving me the opportunity to do what I do everyday for the last two and half years. You have shown me what research is all about, and I really love what I do! Thank you for the support and inspiration, and for being so enthusiastic about everything we achieve. I am thrilled about the time ahead! I would also like to thank my co-supervisor Henrik Sandberg for always being there to help me when I needed the most.

I am grateful to all the co-authors and collaborators of my research. It is a great privilege to be able to work side-by-side with such talented people. Thanks to Faisal Altaf, Adolfo Anta, Yassine Ariba, João Faria, Alf Isaksson, Erik Henriksson, Aitor Hernandez, Vikram Krishnamurthy, Manuel Mazo Jr., Pangun Park, Alex Wang, Jim Weimer, Paulo Tabuada and Ubaldo Tiberi. All the practical implementations of the work would have not been possible without Faisal Altaf, David Andreu, João Faria and Aitor Hernandez. Thank you for always saying that “nothing is impossible” and doing such an amazing work and showing such commitment. A special thanks also to Alf Isaksson and Thiemo Voigt for being my reference group.

I would also like to take the opportunity to thank André Teixeira for all support, inspiring discussions, motivation and most of all, your friendship. Thanks also to my office mates, Assad Alam, Phoebus Chen, Burak Demirel, Farhad Farokhi, António Gonga Euhanna Ghadimi, Per Hägg, Christian Larsson, Piergiuseppe Di Marco, Pangun Park, Chithrupa Ramesh, Iman Shames, Christopher Sturk, Jim Weimer, and Zhenhua Zou for all the support and fun moments throughout these years. I would also like to thank all the professors and lab administrators at the Automatic Control Lab for creating such a great environment. It is fantastic to be part of this group.

Much of my work has also been influenced by the experienced gained through the supervision of our research engineers and master students. Thanks to Alireza, Faisal, João, Aitor, Aziz, Oriol, Florian, Kui, Tim, Fabien and Fahad. A big thanks to the students involved in the summer projects, Daniel, Emil, Fredik, Isak, Jens, Jonas W., Jonas L., and Waldemar, for making the last summer such an enriching experience.

I would also like to acknowledge the support of my family and friends. In particular, I would like to express my gratitude to my parents and my brother for all their support throughout my life. Special thanks to Tomé, Nuno, Melo, Viana, Bokas, Ricardo, Tocha, Diana e Green for keeping the Gang alive.

Most of all I would like to thank all the support, love and incredible life that my girlfriend Nina has given to me. Thank you for being mine. I am sorry for all the crazy working hours lately. Your comments on the thesis were greatly appreciated.

This work has been financially supported by the EU FP6 project FeedNetBack, the Swedish Research Council, the Swedish Strategic Research Foundation and the Swedish Governmental Agency for Innovation Systems, through the projects NECS and WiComPi. Their support is greatly acknowledged.

Contents

Chapter 1

Introduction

In the last several decades we have seen great advances in computation, communication and control. The proliferation of tiny devices capable of performing computation, communication, sensing and actuation has provided the means to create many intelligent complex systems. These systems are expected to enable more efficient use of the global energy, reduce CO_2 emissions, enhance transportation systems and improve industrial productivity. They place numerous challenges since they are often to be deployed in a distributed manner, perform asynchronous decisions, possibly be mobile, and transmit information over an unreliable packet-based network. Additionally, each device is supposed to live for several years on the same battery pack. All these issues bring new requirements and a need for new tools for modeling, design and analysis of systems where an integration of computation, communication and control is necessary. Research in networked control systems (NCSs) has been recently providing some tools to deal with these design challenges. In this thesis we contribute to the theory and practice for NCS design.

1.1 Motivating Examples

To motivate the methods developed in the thesis, we present two applications: industrial process control and building automation.

1.1.1 Industrial Process Control

In industrial process control, the integration of wireless networks and control is an enabler of a more flexible and easy to maintain system, with increased productivity. Wire elimination in hazardous locations is also a key aspect. From an economic point of view, adding more sensing through wireless communication and substituting the wired links in control systems, correspond to large cost savings. 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 a hot rolling mill in the steel industry. By using a higher number

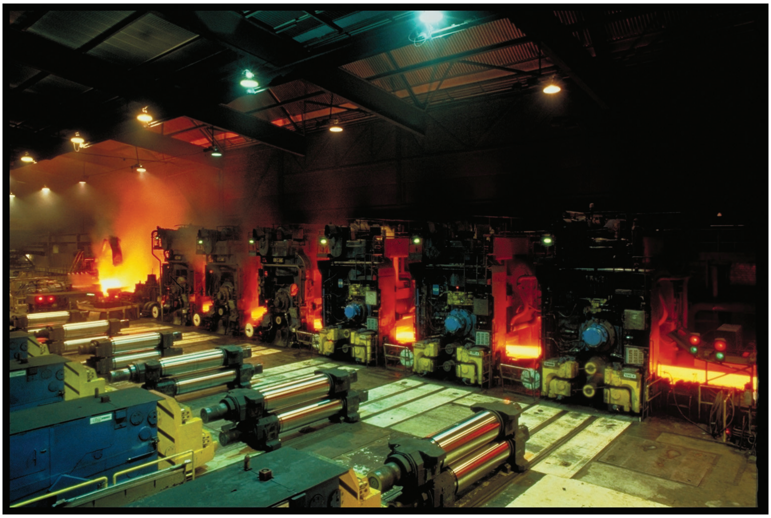


Figure 1.1: A hot rolling mill composed of a sequence of rolling stands yielding the desired steel bar section as the bar moves through the mill. (Courtesy of ABB)

of sensors and designing a more flexible control system design, higher productivity and higher product quality can be achieved. Several wireless system solutions have been proposed for the process industry and are commercially available, e.g., WirelessHART (HART Communication Foundation, 2007), ISA100 (International Society of Automation, 2010) and SmartMesh Industrial (Dust Networks, 2011). All these solutions combine platform-specific hardware and protocols, and use the low-power IEEE 802.15.4 standard as the physical layer.

In monitoring applications of wireless sensor networks, there is the need for high reliability so sensed data is always received by the data logging unit. In the case of control applications, there is the need for high reliability but also for timely packet delivery, where latency should be kept as low as possible. If the control system is not designed to cope with the wireless network imperfections, control performance may be degraded or may even cause full system stop. Battery lifetime of wireless devices is an important aspect for real deployments. It is therefore required that efficient control methods are designed to cope with the wireless medium, and that the wireless network design is optimized for the control requirements, while reducing energy consumption of the network nodes as much as possible.

1.1.2 Building Automation

Studies indicate that residential, office and commercial buildings account for nearly 40% and 47% of the U.S. and U.K. energy consumption (U.S. Department of Energy, 2008; UK Department of Trade and Industry, 2011), respectively. Heating,

Figure 1.2: The Sino-Italian Ecological and Energy-Efficient Building at Tsinghua University in Beijing, China, was designed to maximize both passive and active solar efficiency, and contains advanced HVAC control systems. (http://www.ecofriend.com)

ventilation and cooling (HVAC) is known to be the largest contributor, accounting for 43% of U.S. residential energy consumption. Some of the issues being targeted by research projects are energy-efficient HVAC systems, high-performance lighting systems, systemic approach for retrofitting existing buildings and demonstration of nearly zero energy buildings. Studies discuss how to save up to 50% of energy consumption in hospitals (Bonnema, 2010), large-scale offices (Leach, 2010) and commercial buildings (Deru et al., 2011). Figure 1.2 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. An important component of future building automation is low-cost wireless sensor and actuator networks for monitoring temperature, humidity and CO_2 levels, (Kim et al., 2009) as well as occupancy (Lu et al., 2010; Erickson et al., 2011). The data gathered by the wireless devices is essential for an energy efficient HVAC system and user comfort. Battery life span of the wireless nodes is of major concern due to battery replacement costs. Since many sensors may be deployed in a building, network bandwidth utilization must be limited to avoid traffic communication congestion. Therefore, efficient algorithms must be devised both for control and communications for high performance HVAC systems.

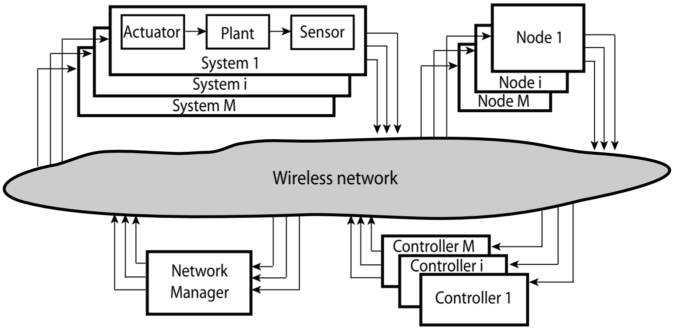


Figure 1.3: Typical architecture of a wireless networked control system. Several sensors are responsible of monitoring the state of the plant, and transmit measurements to controller units. The controllers are designed to compute appropriate control actions to be sent to actuators, connected to the plant. Several other external nodes could also be deployed and share the same wireless network. A network manager is responsible of performing configuration of the communication properties of each wireless device.

1.2 Problem Formulation

In this thesis, we focus on the problem of controlling wireless NCSs with ensured closed-loop performance guarantees and efficient resource usage. An example of such a wireless NCS is depicted in Figure 1.3, where the communications of several control systems and other nodes are coordinated by a *network manager*. The control systems are composed by a *plant*, several *sensors*, and *actuators* as well as *controllers*. The sensors take measurements and transmit them to a controller, which computes control commands, and transmits them to the actuators. Several external *nodes* could share the same wireless network, creating additional network traffic.

We address the following problems for wireless NCSs:

1) What is a suitable control architecture for wireless control systems?

2) How can an aperiodic PI controller be implemented over a resource-constrained wireless network?

3) How can the feedback controller and the wireless medium access be jointly optimized?

1.3 Thesis Outline and Contributions

The outline of the thesis is given below, together with references to the related publications.

Chapter 2: Background

An introduction to the literature on NCS theory and aperiodic sampling methods for control is given, followed by an overview of wireless medium access control schemes.

Chapter 3: System Architecture for Aperiodic Networked Control

Novel mechanisms for performing aperiodic sampling for NCSs are presented. Each mechanism is composed of an aperiodic sampling scheme and controller design, a scheduling policy and a suitable wireless network configuration. The design of such architectures aims at achieving an efficient resource usage of the NCS, while guaranteeing desired levels of control performance.

This work was performed in collaboration with the authors of the following paper and D. Andreu.

- J. Araujo, A. Anta, M. Mazo Jr., J. Faria, A. Hernandez, P. Tabuada, K. H. Johansson, "Self-Triggered Control for Wireless Sensor and Actuator Networks", in Proceedings 7th IEEE International Conference on Distributed Computing in Sensor Systems, Barcelona, Spain 2011.

Chapter 4: Experimental Evaluation of Aperiodic Networked Control Systems

In this chapter, we present the experimental evaluation of the aperiodic architectures proposed in the previous chapter. Parts of this work were reported in the paper above.

Chapter 5: Event-triggered PI Control

This chapter deals with the design of event-triggered PI controllers for industrial process control. The introduction of an aperiodic sampling scheme for PI controllers poses new challenges with respect to the achievable control performance, for which we provide a detailed analysis. Suitable PI controllers and aperiodic sampling schemes are proposed and evaluated through simulations.

This work was performed in collaboration with U. Tiberi, E. Henriksson, A. Isaksson, H. Sandberg and K. H. Johansson.

Chapter 6: Wireless Network and Control Co-Design

We present a framework for the joint design of the wireless network and control for NCSs, while minimizing the energy consumption of the wireless network. An analysis on the effects of the wireless network in the control system under packet losses and delays is performed, in which we are able to define the achievable control performance, under the aforementioned network imperfections. A numerical example illustrates the proposed co-design framework.

- P. Park, J. Araujo and K. H. Johansson, “Wireless Networked Control System Co-Design”, Proceedings of IEEE International Conference in Networking Sensing and Control, Delft, The Netherlands 2011. (Best Paper Award Finalist)

Chapter 7: Conclusions

A summary of the thesis contents and directions of future work are presented.

Other publications

The following publications are not covered in this thesis, but contain material that has influenced the work here presented:

- M. Larsson, J. Lindberg, J. Lycke, K. Hansson, A. Khakulov, E. Ringh, F. Svensson, I. Tjernberg, A. Alam, J. Araujo, F. Farokhi, E. Gadhimi, A. Teixeira, D. V. Dimarogonas, K. H. Johansson, “Toward an Indoor Testbed for Mobile Networked Control Systems”, submitted to the First Workshop on Research, Development and Education on Unmanned Aerial Systems, Seville, Spain 2011.

- J. Weimer, J. Araujo, A. Hernandez, K. H. Johansson, “Periodic Constraint-based Control using Dynamic Wireless Sensor Scheduling”, to appear IEEE Conference on Decision and Control, Orlando, USA 2011.

- F. Altaf, J. Araujo, A. Hernandez, H. Sandberg and K. H. Johansson, “Wireless Event-Triggered Controller for a 3D Tower Crane Lab Process”, Proceedings of IEEE Mediterranean Control Conference, Corfu Island, Greece 2011.

- A. Hernandez, J. Faria, J. Araujo, P. Park, H. Sandberg, K. H. Johansson, “Inverted Pendulum Control over an IEEE 802.15.4 Wireless Sensor and Actuator Network” (Demo Paper), in Proceedings of the European Wireless Sensor Networks, Bonn, Germany 2011.

- J. Araujo, Y. Ariba, P. Park and H. Sandberg K. H. Johansson, “Control over a Hybrid MAC Wireless Network”, Proceedings of the IEEE SmartGridComm, Gaithersburg, USA 2010. Presented also at Reglermote, Lund, Sweden 2010.

- J. Araujo, H. Sandberg, K. H. Johansson, “Experimental Validation of a Localization System Based on a Heterogeneous Sensor Network”, in Proceedings of IEEE 7th Asian Control Conference, Hong Kong, China 2009.

- A. Wang, J. Araujo and V. Krishnamurthy, “Syntactic Inference For Highway Traffic Analysis”, in Proceedings of the 12th International Conference on Information Fusion, Seattle, USA 2009.

Chapter 2

Background

Feedback control of dynamical systems plays an important role in many applications domains, such as process control, aerospace and automotive control. Since the 1930s, research in feedback control went from analog components, to digital control with the introduction of digital computers in the 1950s. Process control has been an important driver of research in control technology, introducing the first distributed control system in 1975 with Honeywell's TDC 2000 (Samad et al., 2007). Such a control system had spatially distributed components, connected over a local network, with sensing, computation and actuation capabilities. They increased productivity and reduced installation costs, close to millions of dollars, for large installations (Samad et al., 2007). In the 1980s, the automotive industry began the development of a communication network that would introduce feedback control over computer networks in passenger cars. The Control Area Network (CAN) was created, which is regarded as the first protocol specially designed for control. Guarantees of safe operation and real-time control were achieved, while connecting hundreds of devices inside a single vehicle. This is seen as the start of the emerging area of Networked Control Systems (NCS) (Baillieul and Antsaklis, 2007). Through the last decades, these systems have been successfully implemented in a variety of industrial applications, improving flexibility, economic performance and increasing safety (CSS, 2011). With the introduction of wireless communications in the early 2000s, new application domains became possible, such as robotics and intelligent machines, intelligent transportation systems and smart control of the power grid (Murray et al., 2003). However, several challenging problems arise when performing feedback control over a wireless communication medium. This motivated the current research of NCSs of performing control with communication constraints, such as limited bandwidth, delay, loss of information and energy constraints (Antsaklis and Baillieul, 2007), representing characteristics less critical or inexistent in wired communications. During this past decade, many researchers have investigated how to design controllers that guarantee suitable levels of performance when the communication network is wireless. Others have developed techniques to improved wireless protocols to be used specially for feedback control. Despite of this progress, wireless NCSs are not widely used in practice, partly because of the unreliability of the wireless medium, safety concerns and energy constraints of wireless devices. These issues motivate the work developed in this thesis.

In the rest of this chapter, we give a brief overview of previous work in control over wireless networks and aperiodic sampling for control. We also introduce the details on Medium Access Control (MAC) protocols that will be used in later chapters.

2.1 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 among feedback control systems and other applications with different requirements. The wireless network is then a common resource, which cannot be disregarded when designing the control system. Additionally, wireless devices are often battery powered, which impose computation and communication constraints of the NCS design. We now present an overview of the relevant research that addresses these issues.

In (Hespanha et al., 2007), 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 the aforementioned communication constraints.

Design methods on how to achieve high performance of control systems through a communication network have been recently proposed. The existing approaches can be mainly grouped in two categories: design of the control algorithm and design of the communication protocol. Research has targeted the design of controllers and estimators that are adaptive and robust to the communication faults: packet dropouts (Schenato et al., 2007; Yu et al., 2004), packet delays (Nilsson, 1998; Henriksson and Cervin, 2005), and data rate limitations (Nair et al., 2007). More recently, Heemels and co-workers (Heemels et al., 2010) further analyze the issues of variable sampling, delays and packet losses and find tighter bounds on the stability properties of such systems, using a hybrid systems approach. Similar approaches to analyze several network protocols are considered in (Tabbara and Nesic, 2008). Using a stochastic hybrid systems framework, Antunes and collaborators (Antunes et al., 2010, 2011, 2012) have established tighter bounds on the stability and performance of the NCS. Similar results are also presented in (Donkers et al., 2010). All these contributions deal with simplistic network abstractions where network optimization is disregarded. Communication protocols and their parameters are designed in order to achieve a given control performance. In (Henriksson and Cervin, 2005; Cervin et al., 2010), the authors present a scheduling policy to minimize a linear quadratic (LQ) cost under computational delays. They also experimentally validate their method. In (Liu and Goldsmith, 2004), the authors propose an adaptive tuning scheme for parameters of the link layer, MAC layer and sampling period.

The authors consider simulations of the wireless network. Communication protocols are designed mainly to achieve high reliability and high energy efficiency for various applications of WSNs and not specifically for control applications (Al-Karaki and Kamal, 2004; Bachir et al., 2010). In each of these approaches, either control system parameters or communication system parameters are able to be tuned by the system designers to obtain desired control and communication performances. In (Demirel et al., 2011), the authors propose to jointly design scheduling policies and controllers over a multi-hop network. However, no energy consumption optimizations are considered. In Chapter 6, we propose a framework to deal with this problem, where the system designer is able to jointly tune parameters of the control and communication system, optimizing the energy consumption of the network, and guaranteeing a desired control performance.

An extensive set of tools and techniques has been developed to reduce power consumption in wireless sensor networks. Unfortunately, the situation is much less favorable for wireless sensor and actuator networks, which are a vital part of 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 of the physical systems being controlled (Ploennigs et al., 2010). In particular, most efforts of the network control systems community have been conducted under the assumption of periodic sampling and actuation (Antsaklis and Baillieul, 2007), which, in general, may require data rates not practical to apply in a wireless system. To address these issues, aperiodic sampling techniques for NCSs were proposed, which we review next.

2.2 Aperiodic Sampling for Control

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 denoted as 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 events that can either be controlled, or occur naturally (Cassandras and Lafortune, 2008). The concept of event-triggered control was recently reconsidered by (Åström and Bernhardsson, 1999; Årzén, 1999; Åström and Bernhardsson, 2002). 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 certain triggering condition is violated, which is continuously monitored. In this way, communication transmission may be reduced, and thereby providing an extension of the battery life span of network nodes.

These techniques have been developed not only for control, but also for estimation, e.g., (Rabi, 2006; Cogill et al., 2007; Rabi et al., 2008; Li and Lemmon, 2011; Shi et al., 2011). In recent years, many researchers have proposed several types of event-triggered implementations for both linear and non-linear systems. These can be divided into deadband control (Otanez et al., 2002; Yook et al., 2002; Hirche et al., 2005; Heemels et al., 2008; Ploennigs et al., 2010), event-based control of linear stochastic systems (Henningsson et al., 2008; Rabi et al., 2008; Rabi and Johansson, 2008, 2009; Molin and Hirche, 2010) and Lyapunov approaches (Tabuada and Wang, 2006; Tabuada, 2007; Wang and Lemmon, 2008b; Mazo Jr. and Tabuada, 2008, 2009; Wang and Lemmon, 2011a; Heemels et al., 2011). Several studies have considered distributed event-triggered control (Mazo Jr. and Tabuada, 2010; Wang and Lemmon, 2011a) and event-triggered control for multi-agent systems (Dimarogonas and Frazzoli, 2009; Dimarogonas and Johansson, 2009; Seyboth et al., 2011). In (Garcia and Antsaklis, 2011a,b; Lehmann, 2011), the authors propose an architecture to perform event-triggered control when the model of the plant is unknown or uncertain.

Self-triggered control was introduced in (Velasco et al., 2003) as another approach for aperiodic control. In this case, the next triggering time is computed at the sensor node and is based on the current measurement and the plant's model. This can be seen as an emulation of the event-triggered scheme described above. There is no need for continuously monitoring a triggering condition, but instead sensor nodes can be turned off between sampling instants. Several implementations of this triggering technique have been proposed for linear and nonlinear plants (Wang and Lemmon, 2008a; Lemmon et al., 2007; Anta and Tabuada, 2008; Mazo Jr. and Tabuada, 2008; Wang and Lemmon, 2009b; Mazo Jr. et al., 2009; Anta and Tabuada, 2009a; Mazo Jr. et al., 2010; Anta and Tabuada, 2009b, 2010a).

During the last year, the problem of minimum-attention control introduced in (Brockett, 1997) has been revisited by (Anta and Tabuada, 2010b; Donkers et al., 2011; Wang and Lemmon, 2011b) as well as anytime control (Fontanelli et al., 2008; Gupta, 2009), aiming at maximizing the inter-sampling time when performing closed-loop control over networks.

Although research attention has been devoted to the development of aperiodic sampling techniques for control systems, the interaction with the wireless network has not been fully addressed. Analysis of event-triggered control with packet losses and delays has been presented in (Molin and Hirche, 2009; Blind and Allgöwer, 2011a,b; Ramesh et al., 2011b,a; Lehmann and Lunze, 2011) for simple wireless network abstractions. The system-level design of self-triggered controllers over a wireless network has been addressed in (Tiberi et al., 2010) for a single control loop, and later extended to multiple loops (Tiberi et al., 2011). In Chapters 3 and 4 we propose and experimentally evaluate a system architecture for aperiodic networked control based on aperiodic control techniques such as event-triggered and self-triggered control. The proposed architecture guarantees closed-loop control performance while minimizing energy consumption of the NCS and providing an efficient use of the network bandwidth.

In order to achieve efficient resource usage, the wireless network characteristics must be well understood. Particularly, the mechanisms used by the nodes to communicate over the wireless channel, may have a large impact on the performance of the NCS. In the following section, we introduce the details of wireless networks with respect to the MAC mechanisms, which are the base of the architectures developed in this thesis.

2.3 MAC

In this section, we present multiple access schemes and MAC protocols that are being considered when performing control over wireless networks. Since the wireless medium cannot support multiple 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 Logical Link Control (LLC) and MAC are sub-layers of the Data Link Layer of the OSI network model, as depicted in Figure 2.1. The MAC protocols are commonly classified as contention-based or conflict free protocols. We describe these in detail below.

2.3.1 Contention-free MAC

Contention-free protocols ensure that a transmission is always successful in the MAC, when the physical medium does not cause any losses. This is achieved by allocating the channel to 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 code:

- 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. In the case of dynamic scheduling, information must be exchanged regularly between the central scheduler and the network nodes. In real applications of wireless sensor networks time synchronization is hard to keep. Therefore, synchronization messages are required to be exchange between the coordinator and the devices.

2.3.2 Contention-based MAC

In contention-based MAC protocols, nodes compete for the medium where simulations transmissions may occur. The common mechanism to handle channel collisions

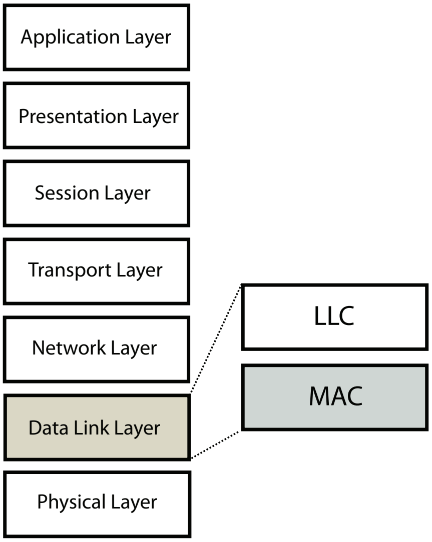


Figure 2.1: The LLC and MAC are sub-layers of the Data Link Layer of the OSI network model. The multiple access protocol is implemented in the MAC sub-layer.

is the Carrier Sense Multiple Access (CSMA) scheme. A transmitting node tries to detect the presence of an encoded signal from any 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 know as a p-persistent CSMA scheme. A CSMA scheme with collision avoidance (CSMA/CA) is also available, where if the channel is found busy at a transmission instant, the node delays the transmission of a message by a random amount of time.

2.3.3 Hybrid MAC

Hybrid MACs, with both contention-free and contention-based schemes. They allow the possibility to obtain a 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, 2006) which we describe in the following section.

2.3.4 IEEE 802.15.4 MAC Protocol

The standardization of low data rate and low power wireless networks is an ongoing process and there is not yet any widely accepted complete protocol stack, particularly for control (Willig, 2008). The IEEE 802.15.4 protocol (IEEE, 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,

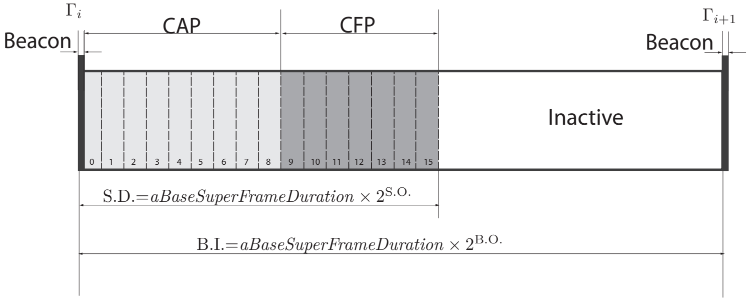

Γ_i
Beacon
CAP
CFP
Γ_{i+1}
Beacon
Inactive
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
S.D.=*aBaseSuperFrameDuration* × $2^{\text{S.O.}}$
B.I.=*aBaseSuperFrameDuration* × $2^{\text{B.O.}}$

Figure 2.2: 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 in order to save battery.

2008). These standards rely on a completely centralized TDMA and fixed scheduling approach for mesh networks, where dynamical schedule changes, required in aperiodic sensor sampling has not been taken into account.

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 (PAN) coordinator*, is responsible for synchronizing and configuring all the nodes in the network. The synchronization and configuration messages take 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 satisfies

$$\text{B.I.}{=}\,\textit{aBaseSuperFrameDuration} \times 2^{\text{B.O.}}$$

symbols, with $0 \leq$ B.O. ≤ 14, and where B.O. is the *Beacon Order* and *aBaseSuperFrameDuration* is defined by the protocol and specifies the shortest superframe duration, corresponding to S.O.= 0. The B.I. is further divided in *active* and *inactive* periods, as shown in Figure 2.2. 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

$$\text{S.D.}{=}\,\textit{aBaseSuperFrameDuration} \times 2^{\text{S.O.}}$$

symbols, with $0 \leq$ S.O. ≤ 14, and where S.O. is the *Superframe Order. aBaseSlotDuration* specifies the shortest slot duration, corresponding to S.O.= 0. The *active period* is further split into a Contention Access Period (CAP) and a Collision Free Period (CFP). During the CAP, the MAC scheme is CSMA/CA where the nodes in the network sense if the channel is busy before transmitting a message. If that is the case, they randomly backoff until succeeding or dropping the message if a maximum number of retransmissions is achieved. The CAP period is used by nodes to send *best effort* messages where packet drops can happen due to collision or channel congestion. On the other hand, the CFP is intended to provide real-time guaranteed service, by allocating *Guaranteed Time Slots* (GTS) to the nodes using it, in a Time Division Multiple Access (TDMA) scheme. Since during the CFP there are no packet losses due to collisions or channel congestion, this mechanism is an attractive feature for time-sensitive wireless applications, as is the case of real-time control of several plants over a wireless network. The total number of GTS slots is limited to 7 in the current standard. Additionally, during the active period an acknowledgement mechanism is present: nodes receive a short acknowledgement packet after each transmission indicating that its packet was received. Let us denote ΔCAP and ΔCFP as the values of the CAP and CFP duration, respectively. The scheduling of the GTS is done by the PAN coordinator in a *first-come-first-served* (FCFS) request-based scheme. At each CAP the nodes requiring a GTS, send a request to the PAN coordinator which will allocate the slot to the node if there are available GTSs. Since this standard is designed for low-power applications, an *inactive period* is defined in the end of the active period so the network nodes and the PAN coordinator enter a low-power mode and save energy. After this period, all the nodes leave the low-power mode in order to receive the beacon message.

We now present an overview of the details of the CSMA/CA mechanism of the CAP, as this will be used during the thesis.

CSMA/CA mechanism of CAP

Consider a node trying to transmit a data packet during CAP. In slotted CSMA/CA of IEEE 802.15.4, first the MAC initializes four variables, i.e., the number of backoffs (NB=0), contention window (CW=2), backoff exponent (BE=*macMinBE*) and retransmission times (RT=0). Then the MAC delays for a random number of complete backoff periods in the range $[0, 2^{\mathrm{BE}} - 1]$ units. If the number of backoff periods is greater than the remaining number of backoff periods in the CAP, the MAC sublayer pauses the backoff countdown at the end of the CAP and resumes it at the start of the CAP in the next superframe. Otherwise the MAC sublayer counts its backoff delay. When the backoff period is zero, the node needs to perform the first clear channel assessment (CCA). The MAC proceeds if the remaining CSMA/CA algorithm steps (i.e., two CCAs), the frame transmission, and any ACK can be completed before the end of the CAP. If the MAC sublayer cannot proceed, it waits until the start of the CAP in the next superframe and apply a further random backoff delay in the range $[0, 2^{\mathrm{BE}} - 1]$ units before evaluating whether it can proceed again. Otherwise the MAC proceeds the CCA in the current superframe. If two consecutive CCAs are idle, then the node commences the packet transmission. If either of the CCA fails due to busy channel, the MAC sublayer increases the value of both NB and BE by one, up to a maximum value *macMaxCSMABackoffs* and *macMaxBE*, respectively. Hence, the values of NB and BE depend on the number of CCA failures of a packet. Once BE reaches *macMaxBE*, it remains at the value *macMaxBE* until it is reset. If NB exceeds *macMaxCSMABackoffs*, then the packet is discarded due to channel access failure. Otherwise, the CSMA/CA algorithm generates a random number of complete backoff periods and repeats the process. Here, the variable *macMaxCSMABackoffs* represents the maximum number of times the CSMA/CA algorithm is required to backoff. If channel access is successful, the node starts transmitting packets and waits for an ACK. The reception of the corresponding ACK is interpreted as successful packet transmission. If the node fails to receive ACK due to collision or ACK timeout, the variable RT is increased by one up to *macMaxFrameRetries*. If RT is less than *macMaxFrameRetries*, the MAC sublayer initializes two variables CW=0, BE=*macMinBE* and follows the CSMA/CA mechanism to re-access the channel. Otherwise the packet is discarded due to the retry limit. Note that the default MAC parameters are $macMinBE = 3, macMaxBE = 5, macMaxCSMABackoffs = 4, macMaxFrameRetries = 3$. See (IEEE, 2006) for further details.

2.4 Summary

In this chapter we have discussed the background in wireless NCSs. In particular, we gave an overview of the research on control over wireless networks where control and communication issues are addressed. Several researchers have addressed each problem separately, but a recent trend on considering the joint design of control and the wireless communications is emerging. The design of aperiodic controllers for wireless NCSs is a promising research topic which aims at providing a more efficient use of the network resources. Additionally, efficient resource usage can also be obtained by properly designing the MAC scheme. A summary of the available MAC schemes and the IEEE 802.15.4 MAC protocol for low-power wireless networks was presented. In the following chapters, we propose architectures that provide control-loop performance guarantees and efficient resource usage for wireless NCSs.

Chapter 3

System Architecture for Aperiodic Networked Control

In this chapter, we propose an architecture for aperiodic sampled wireless NCSs. The architecture is based on event-triggered and self-triggered control. As the underlying data transmission is performed over a shared wireless network, we provide scheduling policies and MAC designs that allow for an efficient implementation.

The rest of the chapter is organized as follows. We introduce the system architecture in Section 3.1. Section 3.2 describes the proposed MAC design. Section 3.3 proposes an event-based sensor communication mechanism and Section 3.4 introduces a predictive sensor communication mechanism. A hybrid sensor communication mechanism which joins the previously proposed mechanisms is presented in Section 3.5. Finally, Section 3.6 summarizes the chapter.

3.1 System Architecture

The problem we aim to solve is that of performing control over wireless networks. Given a control loop that is closed over a wireless network, we are interested in solutions that guarantee stability and performance of the control system while minimizing energy consumption of the wireless sensor and actuator nodes involved and network bandwidth usage. A wireless NCS is depicted in Figure 3.1, where several control systems and other nodes are coordinated and scheduled by a *Network Manager*. Each control system is composed by a *plant*, several *sensors* and *actuators* as well as a *controller*.

We consider the problem of controlling an individual plant:

$$\dot{x}(t) = f(x(t), u(t)) \tag{3.1}$$

in which $x(t) \in \mathbb{R}^n$ denotes the state of the plant, and $u(t) \in \mathbb{R}^m$ the control signal where $u(t) = k(x(t))$, is the control law.

A central issue is the scheduling of transmissions of measurements from the sensors to the controller, and from the controlle to the actuation nodes. In the case

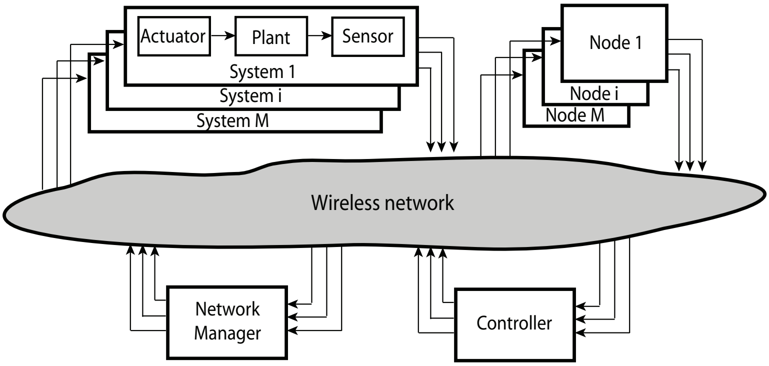


Figure 3.1: Typical architecture of a wireless networked control system. Several sensors are responsible of measuring the plant dynamics, and transmit measurements of the state dynamics to controllers. The controllers are designed to compute appropriate control actions to be sent to actuators, connected to the plant. Several other external nodes can also be deployed and share the same wireless network.

of wireless networks an even more fundamental question is that of deciding which mechanism is to be used to decide the instants of time at which these measurements should be transmitted. We refer to the instants at which transmissions occur as *update times*, to remark the fact that these measurements are actually being used to update the actuation signals at the actuators.

We propose three mechanisms for performing the aperiodic implementation of the update times: *Event-Based Sensor Communications*, *Predictive Sensor Communication* and *Hybrid Sensor Communications*. Each mechanism will define an aperiodic control scheme and a scheduling approach, for a specific wireless MAC design. The underlying idea in each of these mechanisms is to introduce a feedback loop to decide the update times, and thus linking control and communication.

The aperiodic scheme used in the event-based mechanism is based on continuously supervising a performance index, normally a function of the plant dynamics, and only updating the control action when this index violates a pre-defined performance specification. In this work we will focus on how to implement the performance index introduced in (Mazo Jr. et al., 2009) since this implementation allows the event-based sensor communication to assure that a certain control stability performance, in terms of a Lyapunov function, is guaranteed. This technique is commonly known as event-triggered control.

Predictive sensor communication has the goal of reducing the requirement of continuous supervision of a performance index. Instead, a prediction of its evolution is made from the most recent measurement. Through this prediction, this mechanism can provide, right after a measurement, when the next sensor communication should take place. Having access to the next update time, before it happens, allows a tighter scheduling of the communications, allowing for a dynamical allocation of the network bandwidth based on the physical system state. On the other hand, predictive implementations run unsupervised for the time in between updates, which makes them slower to respond to disturbances than event-based implementations. In order to be able to compare the performance achieved using both approaches, we will use the same performance index (Mazo Jr. et al., 2009) as for the event-based implementation. This aperiodic scheme is commonly known as self-triggered control.

The hybrid sensor communication mechanism joins the aperiodic control schemes of the event-based and predictive mechanisms. By doing so, we aim at providing dynamic network bandwidth utilization by relying on a prediction of the required update times, while reducing disturbance rejection response times, by continuously monitoring the state of the plant.

In order to allow timely, reliable, and energy efficient communication, a suitable MAC scheme must be proposed, which we introduce in the following section.

3.2 IEEE 802.15.4 MAC

As presented in Chapter 2, Section 2.3.4, several protocols have been proposed for the process control industry. These standards rely on a completely centralized and fixed TDMA scheduling. However, dynamic scheduling is known to perform a more efficient usage of resources, since resources are allocated depending on the current requirements, and not on worst case conditions. Dynamic scheduling is part of the IEEE 802.15.4 standard MAC and a main interest of the IEEE 802.15.4e task group which works on MAC enhancements for process control and factory automation (IEEE, 2011). Its configuration flexibility allows for adjusting network performance, such as communication throughput, reliability and latency, and energy efficiency, as investigated in (Park, 2011). Therefore, we see the current IEEE 802.15.4 standard MAC as a suitable starting point for the specification of a MAC suitable for our aperiodic control mechanisms.

We are interested in implementing the proposed mechanisms in an heterogeneous WNCSs, composed of wireless sensors and actuators that communicate over a IEEE 802.15.4 network with a controller. We also envision that other devices besides those involved in the control loops are present, sharing the same network resources. Our main contribution comes in the form of suggested modifications and an implementation of the standard MAC layer, to allow the implementation of aperiodic sampling schemes under each mechanism. Also, the MAC design should minimize the energy consumption of the wireless nodes.

As depicted in Figure 3.1, a network manager is used to coordinate all the network operations, and configure the sensor, controller and actuator nodes in the network. This network manager is able to make decisions in an autonomous way, or it can be enabled for user configurations. The unit will have different behaviors depending on which mechanism is selected. With respect to network topology,

Table 3.1: Current consumption (mA) of a Telos wireless platform for different microcontroller (μC) and radio operation modes.

Mode	Description	Measure Current
1	μC active, Radio Tx	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

we assume that all the deployed nodes connect directly to the network manager, in a star network topology. When deploying large scale networks, our framework would still hold in the case of having several interconnected network managers, in a clustered network topology.

In order to design an energy efficient MAC, one has to understand how energy is spent in each wireless node in the network. Sensors and actuators are assumed to be battery powered, and so their life span should be maximized. We now present MAC design guidelines based on the energy consumption of a real wireless node.

3.2.1 MAC Properties

We are interested to perform aperiodic implementations of control systems minimizing the energy consumption of the wireless nodes. To achieve this goal, we suitably design certain MAC parameters. We start by presenting the energy consumption characteristics of a real wireless node, which motivate the MAC design. For details on the the IEEE 802.15.4 MAC see Chapter 2, Section 2.3.4.

The power consumption of the widely used wireless sensor platforms Telos (Polastre et al., 2005) is given in Table 3.1 (Prayati et al., 2010). These platforms are equipped with the microcontroller (μC) TI MSP430F1611 and the CC2420 radio, IEEE 802.15.4 compliant. The table shows the amount of current needed in different modes by the wireless platform according to radio and μC usage. These values show that it is highly power demanding to send and receive messages, and that receiving and listening requires more current than sending. Moreover, mode 1 and 2 are 10 times more expensive than mode 3, and approximately 550 times more expensive than mode 4.

Within the proposed NCS, network transmissions/receptions take place as follows. Sensors transmit information to controllers, and controllers transmit information to actuators. The network configuration is performed periodically by the network manager with a period equal to the B.I.. In our study, we assume that both the controller and network manager are connected to an infinite power source. The controller is thought as a device with large computation capabilities, receiving information of several sensors and transmitting control inputs to several actuators. The IEEE 802.15.4 standard also suggests that the network manager should not be

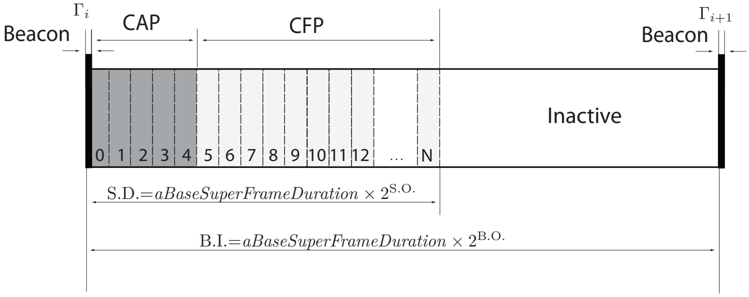

Γ_i
Beacon
CAP
CFP
Γ_{i+1}
Beacon
Inactive
0 1 2 3 4 5 6 7 8 9 10 11 12 ... N
S.D.=*aBaseSuperFrameDuration* × $2^{\text{S.O.}}$
B.I.=*aBaseSuperFrameDuration* × $2^{\text{B.O.}}$

Figure 3.2: Superframe structure of IEEE 802.15.4. 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 Contention Access Period (CAP) and Collision-Free Period (CFP). In the inactive mode, the nodes go to a low-power mode in order to save battery.

battery powered (IEEE, 2006), since it must manage the network at all times.

Naturally, to save large amounts of energy the nodes should be in mode 4 for most of the time, which is achieved by enlarging the B.I., as well as reducing the amount of listening time and receptions/transmissions (mode 1 and 2). Moreover, the S.D. should be set the smallest possible, such that the slot length is minimized. This would allow the controller and actuator nodes to be in mode 2 the shortest amount of time since listening lasts for a complete slot interval. Mode 4 should be enforced since there is a huge benefit of setting the μC to idle, instead of active (mode 3). Note that if no transmissions/receptions of messages take place between nodes, they are in mode 4 at all times, besides the moment when the beacon is received (mode 1). We denote this energy consumption as the *energy baseline.*

3.2.2 MAC Modifications

In order to apply the IEEE 802.15.4 MAC protocol for control applications, we would like to obtain the flexibility of transmitting information over TDMA, CSMA/CA as well as over a hybrid MAC, combining TDMA and CSMA/CA. These features are provided by the IEEE 802.15.4 MAC but have particular constraints that may limit the implementation of the proposed aperiodic mechanisms. Thus, we propose modifications to the standard. An implementation of these modifications is available in (KTH Wireless NCS Testbed, 2011) and the full implementation details are reported in (Hernandez, 2011). We remark that these modifications do not modify the overall structure of the IEEE 802.15.4 MAC protocol presented in Section 2.3.4, but make slight changes on the maximum allowed number of superframe slots, and the Guaranteed Time Slots (GTSs) scheduling methodology.

Limitation on the number of superframe slots

One restriction imposed by the IEEE 802.15.4 MAC protocol is that the GTS slots at the CFP is lower or equal than 7, and the total number of slots assigned in the Superframe (CAP and CFP) must be lower than 16. If a full TDMA MAC would be needed for a given control application, the CAP and Inactive period would be removed, which would render a TDMA schedule with very large slot times since time slot duration is defined as $timeSlotDuration = \text{B.I.}/aNumSuperframeSlots$ in this. Thus, increasing the superframe "discretization" step. This may not be desired since it introduces a large delay between two consecutive transmissions. Therefore, we see a motivation to increase the number of GTS slots given by the standard. We propose a modification of the total number of superframe slots to 32, which can be either CAP or CFP slots, or both. In order to increase the number of slots in the superframe, we increase the *aNumSuperframeSlots* MAC parameter. Recall *aNumSuperframeSlots*=16 in the original MAC, as presented in Chapter 2. By increasing this value, we modify the B.I. and S.D. parameters as well. The following equations show these modifications:

$$
\begin{aligned}
\text{S.D.} &= aBaseSlotDuration \times aNumSuperframeSlots \times 2^{\text{S.O.}}\\
B.I. &= aBaseSlotDuration \times aNumSuperframeSlots \times 2^{\text{B.O.}}
\end{aligned}
$$

where the slot duration is constant for a given S.O.. It is important to note the modification of the B.I. and S.D. to the current standard, where now their intervals are proportional to the *aNumSuperframeSlots*. The modified superframe structure is presented in Figure 3.2. For more details on GTS modification see (Hernandez, 2011).

GTS scheduling

A difference between our MAC and the specification in the IEEE 802.15.4 MAC standard also lies in the GTS allocation mechanism. We propose the use of a scheduling mechanism at the network manager for the assignment of GTS slots instead of the *first-come-first-served* (FCFS) request-based scheme in the IEEE 802.15.4 MAC specification. The network manager is responsible for, according to the selected scheduling algorithm, decide which GTS is assigned to which specific node in the next superframe. At each superframe, all the network nodes have the radios ON to receive the beacon message from the network manager. This beacon contains the information necessary to configure the nodes that have an allocated GTS and inform them of the B.I and S.D. values. If a node has a GTS allocated, then it will transmit/receive during that time and otherwise it will enter in the low-power mode during the B.I..

3.3 Event-Based Sensor Communication

In this section we introduce the event-based sensor communication mechanism. A joint design of the aperiodic sampling technique, scheduling policy and MAC choice is performed, such that control perform guarantees are achieved.

The proposed mechanism is presented in Figure 3.3, and is composed by the plant, sensors and actuators. Each sensor is composed by a Zero-Order-Hold (ZOH) which samples a the state of the plant, and is interfaced by an event generator. The event generator is responsible for implementing the schemes proposed in this section, in order to decide the update times.

For simplicity, we focus our exposition on physical systems that can be described by linear differential equations, although similar ideas can be developed for systems described by nonlinear differential equations (Anta and Tabuada, 2010c). We consider linear control systems of the form:

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m \tag{3.2}$$

where A and B are matrices of appropriate dimensions. A controller $u = Kx$ is usually designed to render the system asymptotically stable and u remains constant between two consecutive control updates. Hence there exists a Lyapunov function of the form $V = x^T Px$ satisfying

$$\dot{V} = \frac{\partial V}{\partial x}(A + BK)x = -x^T Qx \tag{3.3}$$

where Q is a positive definite matrix. The Lyapunov function V can be seen as a *certificate of stability*, since according to equation (3.3) V is always decreasing, but also of *performance* since (3.3) also ensures that the rate of decrease is at least $x^T Qx$. As mentioned in the previous section, the input u cannot be updated continuously but only at discrete time instants t_k, whenever an actuator message is received. Traditionally, control-related messages are exchanged periodically, that is, $t_{k+1} - t_k = T$ for all $i \in \mathbb{N}_0$. The period T is chosen in order to guarantee stability and desired performance under all possible operating conditions. This approach represents a conservative solution to the message scheduling problem since T is selected based on a worst-case scenario. In this mechanism we let the event generator decide the next update time for the sensor and actuator messages.

Since the input u is kept constant between updates, the evolution of the Lyapunov function V for the implementation is now given by:

$$\dot{V} = \frac{\partial V}{\partial x}(Ax + BKx(t_k)) \tag{3.4}$$

As the evolution of V determines the behavior of the system, we specify the desired performance index for the implementation by means of a function S upper bounding the evolution of V:

$$V(t) \leq S(t). \tag{3.5}$$

The previous inequality guarantees that V decreases at least as fast as S does. In that sense we can regard S as defining the control performance that the self-triggered implementation will guarantee. Among other options, a possible choice for S is the Lyapunov function:

$$S(t) = x_s(t)^T P x_s(t) \tag{3.6}$$

for the hybrid system:

$$\begin{aligned} \dot{x}_s(t) &= A_s x_s(t) \qquad t \in [t_k, t_{k+1}[\qquad (3.7) \\ x_s(t_k) &:= x(t_k) \qquad (3.8) \end{aligned}$$

where A_s is a Hurwitz matrix satisfying the following Lyapunov equation $A_s^T P + PA_s = -R$. The matrix R has to be chosen so that $Q - R$ is positive definite, in order to guarantee a minimum inter-transmission time for the control messages (see (Mazo Jr. et al., 2010) for details). Therefore, we can design $R = \sigma Q$, where $\sigma \in]0,1[$ and defines the speed of decay of S, with respect to the rate of decay of V. By lowering σ, the decay rate of S is decreased and approaches the rate of V. In other words, the Lyapunov function of the implemented system (3.2) decays at least as fast as the Lyapunov function of the reference system (3.7).

Inequality (3.5) can be enforced by closing the loop whenever:

$$V(t) = S(t) \tag{3.9}$$

Indeed, notice that at every time instant t_k we have $\dot{V} = -\mathrm{x}^T(t_k)Q\mathrm{x}(t_k) < \dot{S} = -\mathrm{x}^T(t_k)R\mathrm{x}(t_k)$ (since $Q - R$ is positive definite), and therefore $V(t) < S(t)$ for $t \in \ [t_k, t_{k+1}[$. Equality (3.9) implicitly defines a sequence of time instants t_k at which the input needs to be updated.

Under this paradigm, the update times for the system are not know apriori. It is the sensors that on-the-fly decide when it is time to update the controller with fresh measurements. While event-based implementations are certainly robust to disturbances, as there is a continuous supervision of the state of the plant, and reduce the amount of measurements that nodes need to transmit, it has also a couple of 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 capable of liberating the bandwidth not used by an event-based control loop.

3.3.1 Static Scheduling of GTSs

The goal of event-based implementations is to be able to reduce communication between sensors and controllers while providing performance guarantees. In order to provide such performance guarantees the most restringing requirement is that the communications infrastructure should be able to provide a bounded delay between

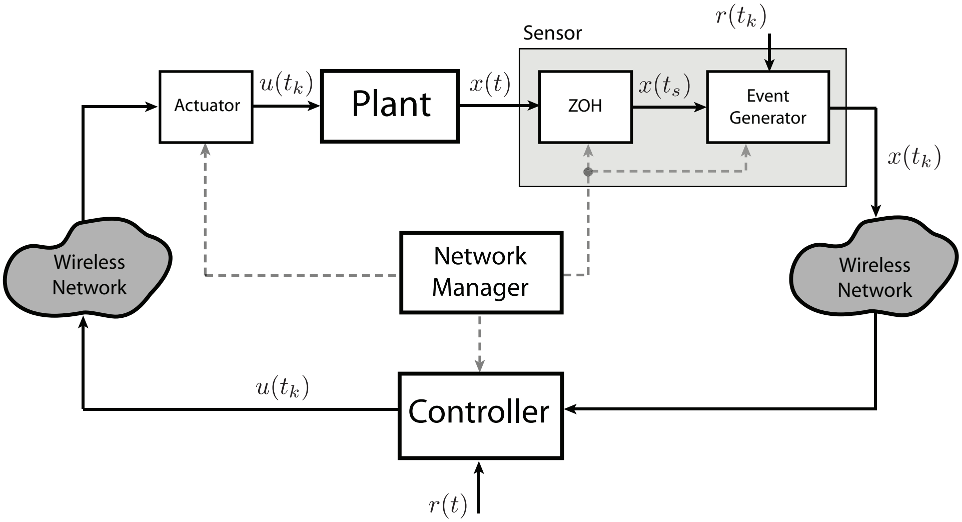


Figure 3.3: Architecture of event-based sensor communication mechanisms for wireless sensor and actuator networks. The event generator is responsible for deciding the update times of the control loop, by transmitting the state of the plant $x(t_k)$, to the controller. A network manager is responsible for distributing a fixed transmission and reception schedule to sensors, controllers and actuators. Beacons are transmitted by the Network Manager to keep all the network nodes synchronized.

the instant at which a measurement is taken, as dictated by the triggering condition, and that a measurement is received at the controller. In order to attain such a guarantee, especially in a wireless network, some form of scheduling needs to take place. We propose the use of a TDMA scheme in which the GTSs are reserved for each event-triggered loop in the network. Then it is the sensors' task to decide based on their triggering condition if they make use of the assigned slots or not. For each plant, two slots are allocated for sensor to controller and controller to actuation messages. This TDMA scheme is performed in GTSs during the CFP of each superframe. This superframe main contain also a CAP and inactive period.

Let us denote the minimum inter-transmission time for both the sensor and actuator messages as t_{min} for all the possible plant conditions. The worst-case inter-transmission times need to be considered in the choice of the $B.I.$ length and the number of GTSs to allocate the transmissions for each control loop. In fact, the inter-slot times must be lower than the minimum inter-transmission time t_{min} so that no more than one transmission would occur between each allocated sensor slot. Let us define the inter-transmission time of sensor messages as $\tau_k = t_k - t_{k-1}$. The minimum inter-transmission time for both the sensor and the actuator messages is given by $t_{min} = \min_k \tau_k$. However, since the transmission times t_k depend on state of the plant and are not known in advance, we need to select t_{min} as the worst inter-transmission time over all possible initial conditions x_0 in the operating region Ω:

$$t_{min} = \min_{x_0} g(x_0, S), \qquad x_0 \in \Omega, \tag{3.10}$$

where $g(x_0, S)$ represents the triggering scheme in (3.9). This time always guarantees a certain level of performance and stability of the closed loop system as defined by $S(t)$. This time can be computed using Lemma 4.1 in (Mazo Jr. et al., 2010). To guarantee that (3.10) holds, (3.6) must be updated when the control input u is applied to the plant.

It is important to notice that an event-based mechanism introduces a delay between events and transmission of measurements, as a sensor might have to wait from the triggering of an event until its assigned slot to transmit. Nevertheless, this delay is bounded and can be adjusted by design, through the assignment of more or less slots to a given event-triggered loop. As with other communication or computation induced delays, this delay is accommodated through the use of more restrictive triggering conditions, as we present next.

3.3.2 Delay Compensation

In event-based implementations there is the need to compensate for the possible delay induced by not having access to the channel right at the moment at which the event is generated, which we denote by channel access delay $\bar{\Delta}$, or due to transmission and computation delays, which we denote by $\tilde{\Delta}$. Moreover, to guarantee (3.10), we must update (3.6) taking the possible delays into account. Note that the channel access delay is fixed and known, and we assume that the computation and communication delays are also fixed.

In order to solve this issue, we propose to check the condition $V(t) \leq S(t)$ ahead of time, by predicting the value of V and S some amount of time in advance so that we can guarantee that a measurement will be sent before the condition is violated. Since the network access is defined by a schedule, and a sensor cannot access the channel if it does not have an assigned slot, we will only check condition $V(t) \leq S(t)$, at each sensor's slot time. This approach has a slight predictive flavor, and requires that the sensing nodes compute the control locally. A known overall delay $\Delta = \bar{\Delta} + \tilde{\Delta}$, characterizing transmission, communication and channel access delay, can be compensated for in the following way. Let $A_\Delta = e^{A\Delta}$ and $B_\Delta = \int_0^\Delta e^{A(\Delta - r)} B dr$, then from a sample acquired by the sensor at time $t_s = t_k + \tau$, where t_k is the time of the last sample and τ is the time passed since the last sample, one can estimate:

$$\hat{x}(t_s + \Delta) = A_\Delta x(t_s) + B_\Delta u(t_k), \tag{3.11}$$

where $u(t_k) = Kx(t_k + \tilde{\Delta})$ is the current value of the input being applied to the system, and $\hat{x}(t_k + \tilde{\Delta})$ represents the estimate of the state of the plant acquired at time t_k by the sensor, plus the computation and transmission delay $\tilde{\Delta}$, until it reached the actuator. The triggering rule instead of verifying when $V(t_s) \leq S(t_s)$ is violated, as in the delay free case, is now modified as:

$$V(t_s + \Delta) \leq S(t_s + \Delta). \tag{3.12}$$

If the condition is violated, the estimate $\hat{x}(t_s + \Delta)$ is sent to the controller, which computes $u(t_s) = K\hat{x}(t_s + \Delta)$, and transmits this value to the actuator, which will apply the control input to the plant at time $t_s + \Delta$. Additionally, the sensor node must update x_s at time $t = t_s + \tilde{\Delta}$ so (3.10) is guaranteed. In this way, we guarantee that $V(t) > S(t)$ never occurs and that the minimum inter-transmission time t_{min} is given by (3.10).

In the proposed approach, we rely on the model of the plant, and sensors must compute the control input values to estimate (3.11) and check the triggering condition (3.12). Sensor nodes must then be able to perform such computations. Moreover, there is not a need to require a separate controller unit to compute the control inputs based on the sensor measurements, and sensor nodes could communicate directly with the actuators. However, in wireless NCSs deployments over large areas, the distance between sensors and actuators may be large, and where the controller node is required to relay information between the two.

Another possible approach to deal with delays would be to check a different triggering condition, e.g., $V(t) \leq \lambda S(t)$, where λ is designed according to the overall delays. This solution would be more conservative than the one presented above, thus generating more triggerings. However, the benefits are that such approach can be used in a distributed manner, where each node takes decisions based on their local information. Additionally, no computation of the control input is performed at the sensor.

3.4 Predictive Sensor Communication

We introduce the predictive sensor communication as an emulation of the event-based implementation presented in Section 3.3. As we mentioned before, we follow the methodology presented in (Mazo Jr. et al., 2009) for designing the aperiodic sampling scheme. Figure 3.4 shows the proposed mechanism for the implementation of predictive sensor communication. In this case, there is a feedback between sensor, controllers and actuators with the network manager. We assume that a single controller node is used for performing closed-loop control of all the plants in the network.

Predictive implementations *identify* the time instants t_k at which (3.9) is satisfied, taking into account the plant model given by (3.2), the last measurement of the state of the system $x(t_k)$ and the performance specification in (3.6). The prediction of the time between two consecutive updates is embodied in the function:

$$\tau_k = t_{k+1} - t_k = g(x(t_k), S). \tag{3.13}$$

There exists several methods in the literature to compute such a function g (Wang and Lemmon, 2009a; Anta and Tabuada, 2010c; Mazo Jr. et al., 2010). We focus on the technique developed in (Mazo Jr. et al., 2009) as proposed in Section 3.3, although similar analyses can be carried out for the other available techniques, including those for nonlinear systems. (Mazo Jr. et al., 2010).

Notice that t_{k+1} represents the time at which the input needs to be updated, therefore both a sensor message and an actuator message need to be delivered between t_k and t_{k+1}. Since (3.13) defines the sequence of inter-transmission times, a scheduling analysis can be carried out beforehand to guarantee the schedulability of the control-related messages.

In the proposed predictive mechanism, the controller will be responsible for computing the value τ_k for all the plants in the network. After all values of τ_k are achieved, the controller implements the scheduling algorithms that are proposed in Section 3.4.1, and transmits this information to the Network Manager. This node then informs all the sensor, controller and actuator nodes of the message transmission/reception slots. We remark that each sensor node may also compute τ_k, and transmit this information to the centralized unit which performs the scheduling of each τ_k.

Due to the prediction nature of this technique, no channel access delays are present, as in the case of the event-based implementation. However, in the case of communication and computation delays, the same approach presented in Section 3.3.2 can be used.

We now introduce the scheduling method, and the required schedulability analysis that guarantees a feasible predictive implementation of several control systems on a shared wireless network.

3.4.1 EDF Scheduling

As proposed in Section 3.2 the network manager is responsible for the scheduling of GTS slots for the nodes in the network and configuration of the wireless network. 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, introduced in the seminal paper by (Liu and Layland, 1973), which is know to be optimal for time-constrained schedules (Buttazzo, 2005). For the explanation of the algorithm we assume the general MAC protocol structure with a CAP and CFP, where CSMA/CA and TDMA algorithms are implemented, respectively, as introduced in Section 2.3.4. Note that the modifications proposed in Section 3.2.2 do not change the overall structure of the modified IEEE 802.15.4 MAC protocol presented in Section 2.3.4.

Recall that ΔCAP and ΔCFP are the values of the CAP and CFP duration, respectively. The design of the GTS scheduling should take into account the following issues:

1. There are two types of messages: hard messages with high priority and hard deadlines, and soft messages with lower priority. Given this characteristic, even if a soft message has an earlier deadline than a hard message, the sched-

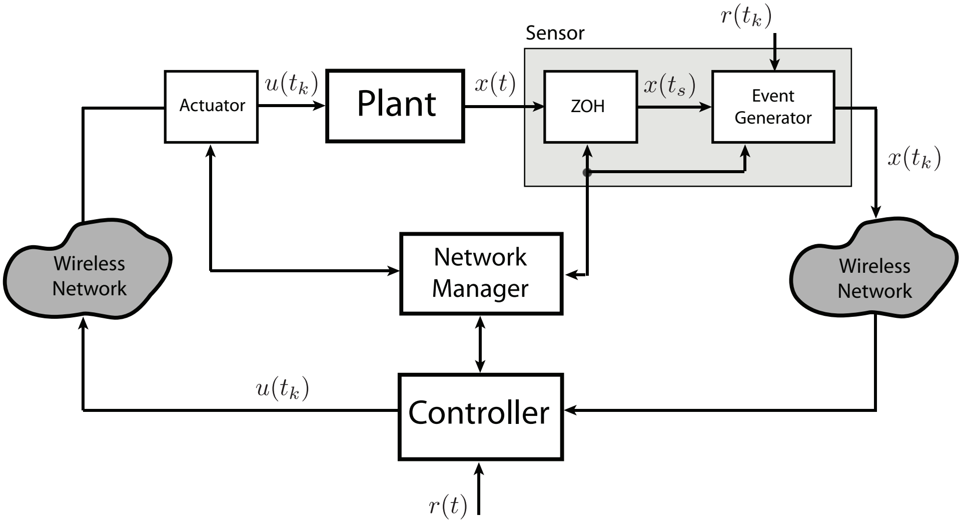


Figure 3.4: Architecture of predictive sensor communication mechanisms for wireless sensor and actuator networks. Note the bi-directional communication between the network manager and sensor, controller and actuators. In this case, the network manager dynamically schedules each node in the network at every superframe, according to the algorithm proposed in Section 3.4.1.

uler node should schedule the hard messages first, followed by soft messages. 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 (3.13), when $t_{k+1} \in [\Gamma_k + \Delta\text{CAP}, \Gamma_{k+1} + \Delta\text{CAP}]$, where Γ_k denotes the beggining time of the $k-$th superframe. 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. Γ_k denotes the superframe i start time.

3. The triggering times t_{k+1} need to be adjusted to new values $\hat{t}_{k+1} \leq t_{k+1}$ if $\hat{t}_{k+1} \leq \Gamma_{i+1} + \Delta\text{CAP}$ in order to fit the triggering time inside the GTS, where $\hat{t}_k$ defines the adjusted triggering time. Recall that no scheduling should be made during the CAP or inactive period.

This last condition is illustrated in Figure 3.5, 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.

Next we introduce the required schedulability analysis in order to prove that an EDF schedule is possible for the predictive sensor communication over wireless networks.

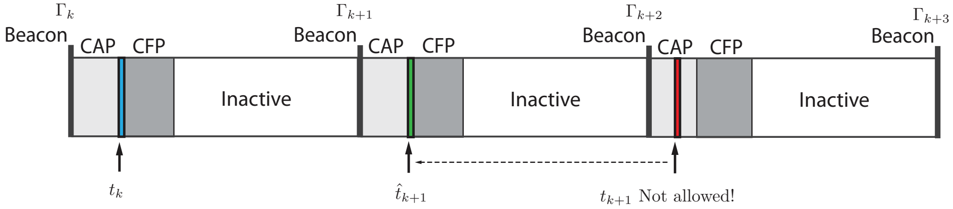


Figure 3.5: EDF Scheduling over a wireless IEEE 802.15.4-based MAC. Here we describe the scheduling of one of the nodes in the network using the self-triggered control scheme. The required triggering time t_{k+1} is not allowed since it does not occur during a CFP. An adjustment of this time to $\hat{t}_{k+1}$ is made in order to allocate t_{k+1} inside a GTS and guarantee (3.5) under EDF scheduling.

3.4.2 Offline Schedulability Analysis

In this section we analyze the schedulability of a set of hard messages under the modified IEEE 802.15.4 MAC. 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 (see previous section).

Each message can be characterized by a triple (t_{min}, C, d), where t_{min} 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 t_{min}). 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 control node.

The inter-transmission times of a pair sensor-actuator message is defined by equation (3.13). Since offline the scheduler is not aware of the evolution of the state, worst-case inter-transmission times t_{min}, need to be considered in the offline schedulability analysis.

Indeed, as in the case of periodic or event-based communications, 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 communication policy modifies these requirements in 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):

$$\sum_{i=1}^{n+2p} \frac{C_i}{t_{min\,i}} \quad \leq \quad 1 \tag{3.14}$$

$$\sum_{i=1}^{n+2p} \left\lceil \frac{t-d_i-\varphi_i}{t_{min\,i}} \right\rceil^+ C_i + C_m \quad \leq \quad t, \quad \forall t \in S \tag{3.15}$$

where the set S is defined as:

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

$$S_i = \left\{ d_i + m \cdot t_{min\,i} : m = 0, 1, \ldots, \left\lfloor \frac{t_{max} - d_i - \varphi_i}{t_{min\,i}} \right\rfloor \right\},$$

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

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

The previous set of equations assume that messages can be transmitted at any time. However, under the modified IEEE 802.15.4, hard messages are not transmitted during the CAP (since success guarantees cannot be provided in CSMA/CA networks) and during the inactive period (to save energy). We model this property by means of two dummy tasks with periods $t_{min\,k}$ equal to the superframe duration S.D., and deadlines equal the inactive period (B.I. - S.D.) and ΔCAP respectively. Moreover, the dummy task modeling the inactive period should have an offset equal to S.D.. In this way, equations (3.14) and (3.15) can be used to analyze the schedulability under IEEE 802.15.4, where now n represents the number of hard messages plus these two dummy messages. For other several related scheduling issues we refer the interested reader to (Anta and Tabuada, 2009a).

As mentioned before, the schedulability analysis has to be based on the worst-case inter-transmission time t_{min} as defined in (3.10) since the initial condition is in general not known in advance, or disturbances might steer the system to this worst-case condition. Nevertheless, the inherent dynamic nature of the predictive implementation allows the scheduler to reallocate resources in an online manner. Different 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. We recall that a drawback of this mechanism when compared to periodic or event-based communications is that it is less robust to disturbances since sensor nodes may be sleeping while a disturbance affects the plant. This is a motivation for the proposed hybrid sensor communication mechanism in the following section.

3.5 Hybrid Sensor Communication

We have presented both event-based and predictive communication mechanisms for control of linear systems, where control performance is guaranteed through a well defined Lyapunov performance index. Event-based implementations are very robust, but require a fixed network schedule of all the system components, where network bandwidth is used in a conservative manner, such that a required performance level is guaranteed. On the other hand, we also propose the use of a predictive implementation, where a dynamic scheduling of network resources is performed, allowing for a more efficient use of the network bandwidth. The drawback of this scheme lies in the fact that if disturbances occur during inter-sampling times, they are not detected and compensated by the control system. Naturally, one may wonder, if an interconnection of both mechanisms would possibly join its merits and reduce its drawbacks. We now propose such a mechanism.

According to the modified IEEE 802.15.4 MAC protocol proposed, there is the possibility to utilize the network in both CSMA/CA during the CAP, and TDMA during the CFP time in GTS. One natural choice for interconnecting both mechanisms is to define the usage of event-based communication during the CAP and the predictive mechanism during the CFP. In this case, the predictive implementation would guarantee the required control performance as specified by the performance index, and during the CAP, an event-based implementation would increase the system's robustness to external disturbances. This mechanism is defined as *Hybrid Sensor Communication*. Since the transmission of event-based messages are subject to contention, which cause a variable delivery delay, no canceling of GTS slots allocated by the predictive scheme is made. Note that even if a sensor node receives an acknowledgement of successfull transmission of a sensing message, no guarantees that the control input message will successfully arrive at the actuator in the same superframe. However, we are able to guarantee that no packet is dropped by the contention mechanism if the parameters of the CSMA/CA MAC are well design. This is the price to pay for an increased robustness againts disturbances. We must recall that no guarantees are provided that the disturbances will be rejected completely, since sensor and actuator messages may be subject to long delays for high network traffic conditions. In any case, the control performance can always be guaranteed by the predictive implementation.

3.6 Summary

In this chapter we have proposed aperiodic implementations of control systems that are specially designed for WNCSs. Each mechanism is achieved by jointly designing the aperiodic sampling technique, the wireless MAC protocol and a scheduling algorithm, that together guarantee a required control performance while obtaining an efficient usage of the network resources. The use of TDMA MAC schemes are preferred over CSMA/CA, such that real-time guarantees of control updates are achieved. In order to implement these mechanisms, we also proposed the modification of the current IEEE 802.15.4 MAC protocol, so a higher flexibility would be achieved when implementing the proposed aperiodic mechanisms.

In order to implement an event-based mechanism, channel access delays must be taken into account since each wireless node is not able to transmit a message whenever an event occurs. This mechanism also requires a static scheduling of network resources which is inefficient when compared to a predictive mechanism. The predictive approach allows the possibility of dynamically allocating the network bandwidth based on the physical system state.

A tradeoff between disturbance is rejection performance and efficiency of network resource utilization is observed regarding the event-based and predictive mechanisms. Therefore, depending on how frequent disturbances occur in the system, and how important they are, one of these schemes would be preferred.

By observing that the event-based and predictive mechanism could complement eachother, we propose a novel hybrid sensor communication mechanism. In this case, we make use of a hybrid MAC, where a TDMA MAC is used by a predictive mechanism, guaranteeing a desired control performance level, while a CSMA/CA MAC can be used to perform control updates, generated by local events occuring at the sensor nodes. This mechanism has the potential of achiving robustness against disturbances as the event-based mechanism, while allowing for a more efficient network bandwith usage as obtained in the predictive mechanism.

All the proposed mechanisms will be implemented and evaluated in an experimental setup in Chapter 4.

Chapter 4

Experimental Evaluation of Aperiodic Networked Control Systems

In order to evaluate the performance of the aperiodic mechanisms presented in Chapter 3, we built a lab process with a wireless network shared by two control loops and several independent sensor nodes transmitting soft messages, with no hard deadlines. The control loops are regulating two double-tank processes from Quanser (Quanser, 2011), where the tanks are collocated with the sensors and actuators and communicate wirelessly with a controller node. The soft messages are monitoring messages with temperature, humidity and light values measured inside the room. Figure 4.1 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 which communicate with the controller. The controller node is also the network manager in our setup. A scheduler node is added to the NCS and connected to the network manager. This unit performs scheduling computations for each mechanism, reducing the computation load of the network manager. The NCS will be used to evaluate the control performance, energy efficiency and network bandwidth utilization of each of the proposed mechanisms.

The rest of the chapter is organized as follows. We introduce the communication network of the network control system, and the MAC design for each mechanism in Section 4.1. The double-tank system and the state-feedback controller are presented in Section 4.2. Section 4.3 contains the experiments and the results obtained for each aperiodic mechanism, with respect to control performance, energy and network bandwidth utilization. Finally, Section 4.4 summarizes the experimental results.

4.1 Communication network

The wireless sensor platform chosen for this experiment is the Telos platform (Polastre et al., 2005). These nodes are equipped with a 250kbps 2.4 GHz Chipcon CC2420 IEEE 802.15.4 compliant radio and on-board sensors. Furthermore, this node has integrated Analog-to-Digital (ADC) and Digital-to-Analog (DAC) con-

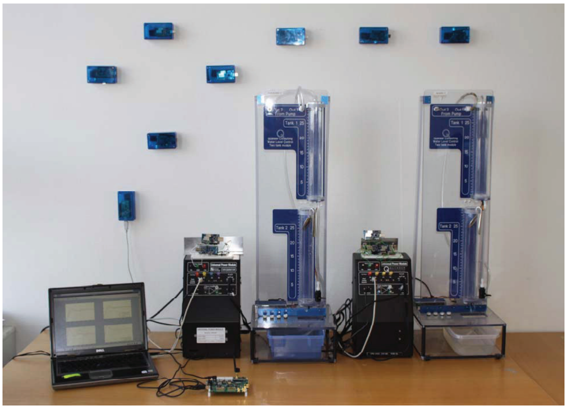


Figure 4.1: Lab process with a wireless network. Two double-tank systems share the wireless network with eight wireless nodes performing monitoring tasks.

verters allowing the nodes to be use as sensor and actuator nodes. The operating system used is TinyOS (Levis et al., 2004).

We use the modified IEEE 802.15.4 MAC protocol as proposed in Chapter 3. The standard IEEE 802.15.4 MAC protocol has been partially implemented in TinyOS in TKN15.4 (Hauer, 2009) and validated in the Telos platform. An extension of TKN15.4 to include the CFP and the GTS mechanism has been performed in (Hernandez, 2010) for the same platform. The implementation of the protocol used in our setup is based on (Hernandez, 2010) with the modification presented in Chapter 3, and detailed in (Hernandez, 2011). The code for these experiments is available at (KTH Wireless NCS Testbed, 2011).

Additionally, we added the sniffer node CC2420 Development Kit from Texas Instruments IEEE 802.15.4 compliant, that allows for debugging and visualization of all the packets transmitted in the network. With this node, we are able to properly evaluate our experimental setup and confirm the correct CAP and CFP intervals, GTS scheduling, acknowledgments and beacon messages.

4.1.1 MAC Parameters

The NCS is composed by eleven wireless nodes, in which, two are wireless sensors and two wireless actuators, for the two control loops, eight are monitoring nodes and one is the network manager. For performing a control action, a sensing transmission from sensor to controller, and an actuation transmission from controller to actuator takes place. Therefore, since the control loop nodes must communicate during the CFP in a GTS, the total number of GTSs should always be larger or equal than four. In the following, we discuss the step for the three communication mechanisms:

Event-Based Sensor Communication

In this case, all the wireless nodes will be scheduled by the network manager during the CFP in the GTSs. The schedule is assigned at the beginning of the experiment and kept static. The MAC communication structure is defined by one CAP slot and ten CFP slots as the active period, followed by an inactive period where all the network nodes are in energy mode 4. The CAP slot is selected such that the wireless node scheduled in the first GTS has time to perform computation or sensing tasks after receiving the beacon message and transmitting a message. We used a S.D.= 323.1ms and a B.I.= 646.3ms. The inactive period as the same total length as the S.D.. Each of the 11 slots has a duration of 29.4ms.

Predictive Sensor Communication

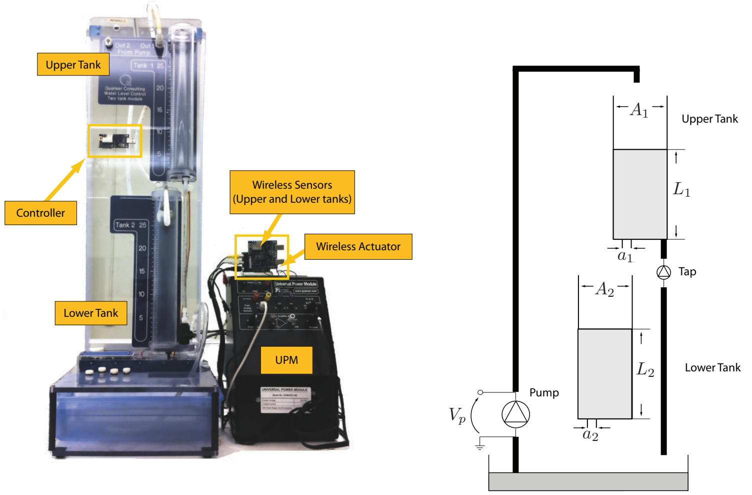
The same MAC structure and parameter choice of the event-based sampling is used in this scheme. The difference comes on how the GTSs are scheduled by the network manager. In this case, a new schedule is disseminated through the network in the beacon message in every superframe.

Hybrid Sensor Communication

For the hybrid mechanism, while the predictive implementation dynamically schedules GTSs for sensing and actuation in the control loop, the sensor nodes of the control loop are able to communicate during the CAP. The allocation of the GTS is performed dynamically by by the network manager, as in the predictive mechanism. Likewise, we also allow the monitoring nodes to communicated over the CAP. The MAC communication structure is defined by 7 slots for the CAP and 4 slots for the CFP, followed by an inactive period where all the network nodes are in energy mode 4. The same S.D., B.I., inactive period and slot duration is used for the hybrid mechanism. Even though the soft messages are of low-priority, for logging purposes, no data should be lost. Therefore, we set the following CAP parameters for the CSMA/CA mechanism, as presented in Chapter 2: $macMinBE = 3$, $macMaxBE = 5$, $macMaxCSMABackoffs = 4$ and $macMaxFrameRetries = 3$.

4.2 Double-Tank System

The double-tank system developed by (Åström and Lundh, 1992) and commercialized by Quanser (Quanser, 2011), consists of a pump, a water basin and two tanks of uniform cross sections. Figure 4.2(a) depicts the experimental apparatus, and Figure 4.2(b) a diagram of the physical system. The liquid in the lower tank flows to the water basin. A pump is responsible for pumping water from the water basin to the upper tank, which flows to the lower tank. The holes in each of the tanks have the same diameter. The sensing of the water levels L_i is performed by pres-



(a) double-tank system with wireless sensing, control and actuation.

(b) double-tank system diagram

Figure 4.2: Double-tank system setup. Wireless sensor nodes are responsible for sensing the water levels in both tanks, compute the control input and actuate the water tank pump.

sure sensors placed under each tank. The ratio between the sensor measurement and water level is given by $L_i = K_s \cdot V_{out}$, where $K_s = 6.25$ cm/V, meaning that a pressure value of 1 PSI is equivalent to V_{out}=4.8V and $L_i = 30$cm. The sensor signals are connected to the Quanser Universal Power Module (UPM) (Quanser, 2011). One wireless sensor node interfaces the sensing channels of the UPM with an ADC, in order to sample the pressure sensor values for both tanks. The plant actuation is made through the DAC of the wireless actuator node, connected to an amplification circuit that will convert the output voltage from $[0, 2.5]$ V to $[0, 15]$ V. This voltage is fed to the UPM which actuates in the pump motor.

The nonlinear equations of motion of the double-tank system can be described as:

$$\begin{aligned} \dot{L}_1 &= -\frac{a_1}{A_1}\sqrt{2gL_1} + \frac{K_p}{A_1}V_p \\ \dot{L}_2 &= \frac{a_1}{A_2}\sqrt{2gL_1} - \frac{a_2}{A_2}\sqrt{2gL_2}, \end{aligned} \tag{4.1}$$

where, a_i is the outflow diameter of upper and lower tanks, A_i is the diameter of the upper and lower tanks, g is the gravitational acceleration in cm/s^2, V_p is the voltage applied to the pump motor, K_p is the pump motor constant, and L_i is the height of the water in both upper and lower tanks.

Linearizing both equations around an operating point L_{10}, L_{20} we obtain:

$$\begin{aligned}\Delta\dot{L}_1 =& -\frac{a_1}{A_1}\sqrt{\frac{g}{2L_{10}}}\Delta L_1 + \frac{K_p}{A_1}\Delta V_p \\ \Delta\dot{L}_2 =& \frac{a_1}{A_2}\sqrt{\frac{g}{2L_{10}}}\Delta L_1 - \frac{a_2}{A_2}\sqrt{\frac{g}{2L_{20}}}\Delta L_2,\end{aligned} \tag{4.2}$$

where $\Delta L_1 = L_1 - L_{10}$, $\Delta L_2 = L_2 - L_{20}$ and $\Delta V_p = V_p - V_{p0}$ represent the incremental values of the state and the input with respect to the operating point. In equilibrium, the value of the control input is $V_{p0} = a_1\frac{\sqrt{2gL_{10}}}{K_p}$ and $L_{10} = \frac{{a_2}^2}{{a_1}^2}L_{20}$. To achieve the correct values for the parameters of the tanks, we perform the parameter identification before running the experiments.

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 the feedforward tracking method (Glad and Ljung, 2000), with the control input defined as,

$$u(t) = Kx(t) + Mr(t), \tag{4.3}$$

where 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 implementations proposed in Chapter 3, 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:

$$\begin{aligned}\Delta\dot{x} &= A\Delta x + B\Delta u, \\ \Delta u &= K\Delta x,\end{aligned} \tag{4.4}$$

where $\Delta x(t) = x(t) - \bar{r}$ and $\Delta u(t) = u(t) - M\bar{r}$, which achieves $\lim_{t\to+\infty}\|\Delta x(t)\| = 0$.

For the double-tank system, we denote $x = [\Delta L_1, \Delta L_2]$, and the shifted state as $\Delta x = [\Delta L_1 - \Delta L_{1\text{ref}}, \Delta L_2 - \Delta L_{2\text{ref}}]$, where $\Delta L_{i\text{ref}}$, for all $i = 1, 2$, is the steady-state value of upper and lower tank. 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 system in steady-state, i.e. $\Delta\dot{x} = 0$.

4.3 Experiments

The experiment details and the obtained results for the NCS under different aperiodic mechanisms are presented and discussed below. Firstly, we provide analysis of the control performance and energy efficiency of the control loops, and secondly we analyze the network bandwidth utilization when the control loops share the network with monitoring nodes.

4.3.1 Control Loops

The goal of the double-tank experiment is to control the level of the lower tank to a desired setpoint. An initial phase takes place to bring the water level to a value of 5 cm, and at the start of the experiment a command is issued to raise the level of water to 10 cm. Each wireless sensor node will sample the pressure sensors 10 ms before transmitting the sensor value message to the controller. Following the reception of the sensor message, the controller should compute the control input $u(t)$ and transmit it to the actuator node. The reference value $r(t)$ must be known both at the sensor and controller sides in the event-based and hybrid implementations such that the node is able to compute the triggering rule. We define that the user sets this value at the controller, which is transmitted to the sensor node in the beacon message. Recall that the beacon is transmitted periodically and so, reference changes can only be performed at a rate below the beacon message. In the predictive case, only the controller must know this value.

An input disturbance of magnitude 1 V is applied to the pump actuation at time $t = 200\text{B.I.} = 130$ s. In this case, a 1 V value is added to the control input being applied to the pump. This shows 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 $450\text{B.I.} = 220$ s.

In order to guarantee robustness of the predictive scheme with respect to disturbances (Mazo Jr. et al., 2010), an upper bound on the inter-sampling times t_i needs to be imposed. We fix that bound in 10 s. The performance function $S(t)$ in (3.6) is identical in all the mechanisms, where we define $R = 0.1Q$ and Q is selected as the identity matrix. We compute the minimum inter-transmission time for this system (as defined in (3.10)) using Lemma 4.1 in (Mazo Jr. et al., 2010), which, for this physical system, gives a minimum time of 1 s. Hence, the inter-transmission times τ_i for the control-related messages will be in the range $[1, 10]$ s. As mentioned in Sections 3.4.1 and 3.3.1 these times will be adjusted to $\hat{\tau}_i$ to be allocate at a GTS. Since the B.I. is defined to be approximately 0.64 s, then $\hat{\tau}_i$ is in the range of $[0.64, 9.6]$ s.

A periodic implementation of the control loops is implemented for comparison purposes. The sampling period of the periodic implementation is set to $t_{min} = 0.64$ s, given by (3.10), since stability has to be guaranteed under all possible conditions. In this case, the sensor and actuator nodes communicate with the controller at every B.I., and have a fixed GTS schedule.

The analysis of energy efficiency of the control loop nodes is performed with respect to the battery life expectation. For the battery lifetime expectation calculation, we sum the total current consumption of the wireless sensor over the 220 s period of reference tracking and repeat it until we consume the 2900 mAh of battery capacity.

4.3.2 Monitoring Nodes

The soft messages are transmitted by the eight monitoring nodes to the network manager. These nodes transmit 8 bytes of data in each packet. For the evaluation of the network bandwidth utilization in Section 4.3.4, we define two message deadline types for the soft messages, which represent different traffic patterns that could be found in real NCS.

Slow traffic: Slow periodic transmission deadlines, with period $T_m = 5\text{B.I.} = 3.3$ s, through the whole experiment.

Bursty traffic: Fast traffic during 25B.I. = 16 s, starting at $t = \{0, 120, 200\}$ s and slow traffic during the rest of experiment.

The start deadline of each of the eight nodes is randomized within the CFP for the event-based and predictive mechanism, and randomized within the CAP for the hybrid mechanism.

4.3.3 Control Performance and Energy Efficiency

The control performance and energy efficiency is evaluated for one of the double-tank system, considering each mechanism. 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 water level (see Appendix A.1 for the definition), before and after the occurrence of the disturbance, as well as the overall IAE. The number of updates transmitted by the sensor to the controller and the respective battery consumption of the node characterize the energy efficiency of the mechanism.

Event-Based Sensor Communication

In this scheme, the sensor node of each double-tank performs the implementation provided in Section 3.3. Figure 4.3 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 implementations track the reference signal with similar behavior. The inter-transmission times τ_i vary between 1.3 s and 9.6 s, leading to a significantly lower number of transmissions than the periodic scheme, but still achieving similar transient response and disturbance rejection. Table 4.1 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 outperforms the periodic implementation. This results is due to the fact that at start, the water flow rate is high, and no adjustment is made until the second transmission arrives at the

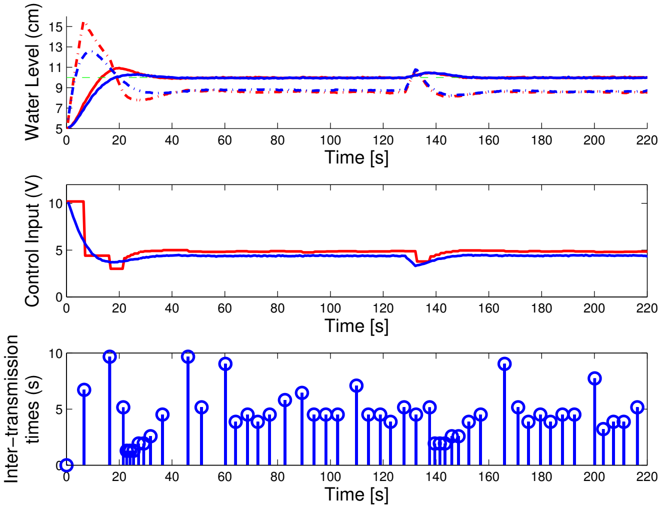


Figure 4.3: Experimental results for event-based sensor communication. The upper plot depicts the evolution of the water tank level and middle plot the control input values for event-based mechanism (red) and a periodic controller (blue). Upper (dash-dotted line) and lower (solid line) water levels are presented. The rise time for event-based scheme is faster than the periodic. Short inter-transmission times (lower plot) are observed when the water level approaches the setpoint, and when the disturbance is rejected. Approximately the same disturbance rejection performance is observed. The sampling period for the periodic control is 0.64 s.

controller. A faster rise time is achieved, followed by a fine adjustment of the water level when closer to the reference. Even though the number of transmissions of the event-based scheme is only 14.1% of the periodic, the battery lifetime increase is of 54.4% and not 700% as it could be expected, if only the number of transmissions would consume battery. This difference originates from the fact that the wireless nodes still need to turn on the radio and μC to receive the beacon message at each B.I., and spend energy during the inactive period. Therefore, high reductions in the number of transmissions do not imply the same ratio of energy savings. This motivates the enlargement of the B.I. in order to increase battery lifetime, which for this dynamical system and controller design, was not possible to achieve in order to guarantee stability, as defined by (3.10).

Predictive Sensor Communication

The implementation follows the definitions presented in Section 3.4. We define two different ranges of deadlines for the soft and the hard messages, to avoid having soft messages blocking hard messages. The scheduler computes the deadlines for the

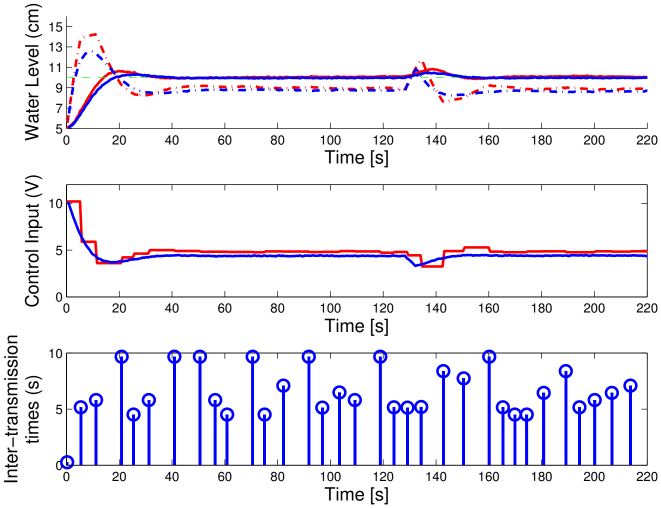


Figure 4.4: Experimental results for predictive sensor communication. The same signals are shown as in Figure 4.3. Similar transient responses to the event-based case are obtained. However, disturbance take a longer time to be rejected since the sensor node is not active at the disturbance input time.

sensor and actuator messages according to (3.13). After the schedule is computed, it is transmitted to the network manager which configures all the nodes in the network in the following beacon message. The scheduling phase takes place during the inactive period.

Figure 4.4 shows the time response and inter-transmission times of one double-tank system for predictive and periodic mechanisms. As in the previous case, both control implementations track the reference signal with similar behavior. The inter-transmission times τ_i vary between 4.48 s and 9.6 s, leading to a much lower number of transmissions than in the periodic scheme, and lower than the event-based mechanism. Table 4.1 presents the results. The IAE analysis show that the predictive scheme outperforms the periodic and event-based scheme during the transient, since its IAE is lower, but has a much worse performance when rejecting the disturbance. This occurs due to the fact that the sensor node is only active at transmission times, and not at every superframe, like the event-based or periodic schemes. The predictive scheme allows for an even higher reduction of the number of transmissions. However, since the event-based mechanism and predictive share the same performance criterion, both are expected to have the same behavior, in the absence of disturbances. This can be explained by the fact that noise affects the real plant. In fact, since the event-based scheme verifies the triggering condition at every B.I., any noise that may occur at those times, may have a higher effect in the triggering rule computed by each of the mechanisms.

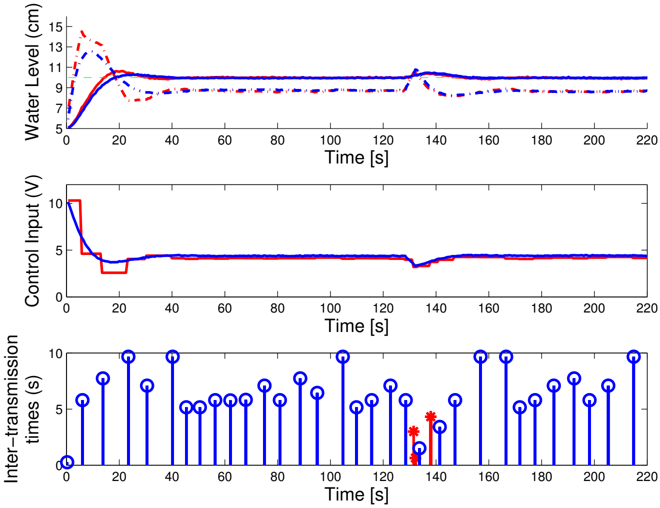


Figure 4.5: Experimental results for hybrid sensor communication. The same signals are shown as in Figure 4.3. Similar transient responses to the event-based case are obtained. Benefits arise when using event-based sensor communications, since disturbance rejection is improved, while achieving the same control quality during transient as the predictive scheme. The red star marker represents an event-based transmission and the blue round marker represents a predictive transmission.

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 arguments as above, 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 Sensor Communication

The implementation follows the definitions presented in Section 3.5. The sensor nodes compute locally the condition 3.5, and are able to transmit during the CAP if the condition is violated, in an event-based fashion. We choose to implement the case when the triggering rule is computed during the superframe at all times. We denote this by Hybrid^{CS} since a continuous supervision of the triggering rule is performed. Note that this mechanism differs from the one implemented in the event-based mechanism since in that case, a Δ step-ahead prediction is performed once every superframe at the GTS, and no continuous supervision is performed.

Additionally, the predictive mechanism schedules the GTS to be used by the control loops, with real-time guarantees. Low-priority monitoring nodes also transmit during the experiment, with a bursty traffic pattern. Since the communication

Table 4.1: 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	IAE$^{[0,130]}$	IAE$^{[130,220]}$	IAE$^{[0,220]}$	N_{updates}	Battery life (days)
Event-based	70.78	12.48	83.22	49	982.57
Predictive	67.96	28.73	96.63	34	1010.18
Hybrid$^{\text{CS}}$	67.05	16.51	83.52	36	63.66
Hybrid$^{\Delta}$	67.05	16.51	83.52	36	905.35
Periodic	73.08	15.42	88.43	347	636.81

during the CAP is performed with a CSMA/CA mechanism, no real-time guarantees can be provided for the events generated by the sensors. Moreover, with the defined MAC parameters for the CSMA/CA mechanism in Section 4.1.1, no packet is dropped by the sensor since the traffic is not high enough to reach the maximum number of retransmissions.

Figure 4.5 depicts the time response and inter-transmission times of one double-tank system for the hybrid and periodic mechanisms. As in previous cases, the control implementations track the reference signal with similar behavior. The inter-transmission times τ_i are depicted for the case in which the transmission was generated by the event-based or the predictive mechanism. As seen in the figure, only predictive transmissions take place during transient, and event-based transmissions occur during the disturbance rejection phase. Table 4.1 depicts the results for this mechanism as Hybrid^{CS}. The IAE during the transient is kept close to the predictive scheme, as expected. The benefits of using the hybrid scheme become clear when the disturbance occurs. In this case, event-based transmissions occur, rejecting the disturbance faster than with the periodic scheme.

As observed in the battery life results, there is a large difference between the Hybrid$^{\text{CS}}$ implementation and the results observed until now with the other mechanisms. This results from the fact that if the μC is kept computing for all times (mode 3 in Table 3.1, Chapter 3), no large savings are performed. In this way, the baseline energy consumption is high, since the node never sleeps (mode 4). For comparison purposes, we also show the case in which the hybrid scheme would be implemented using the Δ step-ahead prediction as in the event-based mechanism, in Table 4.1. We denote it by Hybrid$^{\Delta}$. In this case, we assume that the sensor node, at the beacon reception time, computes if condition 3.5 will be true during the current superframe. If the condition is verified, the sensor node attempts to transmit during the CAP. In this case, the battery life increases to the same levels as the other aperiodic schemes, since mode 4 is used most of the time, and the node is set to sleep if no transmission takes place.

The number of transmissions of the hybrid scheme is 10.1% of the periodic and lower than the event-based scheme, but the battery lifetime stays below the one achieved using the event-based scheme. This comes from the energy required to check condition 3.5 ate every superframe, and the event-based attempts, when rejecting the disturbances.

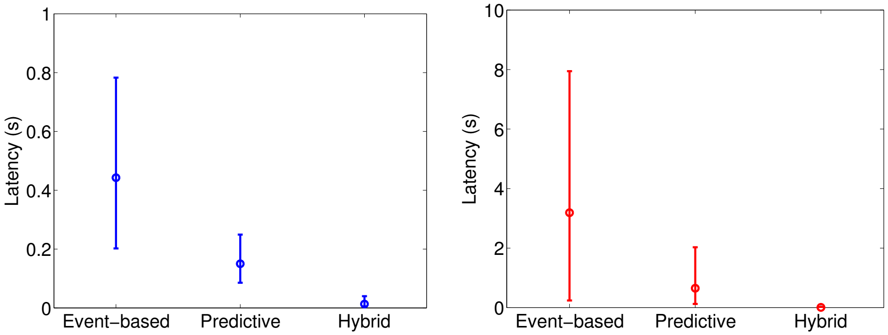
4.3.4 Network Bandwidth Utilization

The network bandwidth utilization is characterized by how well the network is shared among the wireless nodes. In the experiments shown, the control loops had real-time deadlines that were achieved by allocating the transmissions in the GTSs. The remaining GTSs were shared by the low-priority monitoring nodes, in the event-based and predictive mechanisms, while these nodes used the CAP to transmit messages in the hybrid mechanism. We analyze the latency experienced by these nodes in each of the mechanisms. By latency we mean the time between the transmission deadline expired, and the moment where an acknowledgment was received for that particular message. The deadlines are the same as the transmission times defined in Section 4.3.2.

Figure 4.6 depicts the latency analysis of the monitoring wireless nodes, with respect to the traffic patterns. For each mechanism, the 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 also increases for all schemes, with a higher impact in the event-based mechanism, since it is based on a static scheduling mechanism. In this case, the queue of soft messages is large, since only 6 slots of the superframe are available to be shared among 8 monitoring nodes. We remark that the periodic scheme has the same results of 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 best network bandwidth utilization since its latency is low, when comparing to the other schemes, and its latency stays always between $[1, 40]$ ms. Each node is able to transmit messages during the CAP, where several other nodes may attempt to transmit. However, no GTS scheduling queueing will occur in this case. Naturally, there is an advantage to allow monitoring nodes to transmit soft messages during the the CAP, instead of being allocated in GTSs.

4.4 Discussion

Experiments have been performed to evaluate the aperiodic mechanisms proposed in Chapter 3. All the mechanisms achieve setpoint tracking and disturbance rejection, with closed-loop control performances close to the ones obtained with a traditional periodic paradigm. These benefits can also be seen both in terms of energy savings and network bandwidth utilization. As expected, the predictive mechanism is slower when responding to disturbances, and so, by introducing the hybrid mechanism,



(a) Latency for Slow Traffic

(b) Latency for Bursty Traffic

Figure 4.6: 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). Note the scale difference in both traffic patterns. By increasing the generation of messages, the latency also increases for all schemes, with a higher impact in the event-based mechanism, since it is based on a static scheduling mechanism. The latency of the predictive mechanism is kept at low levels, since the GTS scheduling for control loops is dynamically adjusted as function of the control requirements. The worst case delay observed for the hybrid mechanism under slow traffic in 40 ms for all traffic conditions.

better control performances are achieved. Despite providing a lower battery lifetime expectation, the hybrid scheme provides an efficient network bandwidth utilization by making use of the CSMA/CA mechanism of the IEEE 802.15.4-based MAC. Therefore, there always exists a trade-off between control performance together with efficient network bandwidth utilization and the energy consumption of the wireless nodes in an NCS.

Chapter 5

Event-triggered PI Control

Since its creation in 1910, the proportional-integral-derivative (PID) controller has been applied to solve many control problems. Even though many controller choices are currently available, PID controllers are still by far the most widely used form of feedback control. In process industry it is know that 90% of the control loops are regulated by PID controllers (Åström and Hägglund, 2011). Most control loops are Proportional-Integral (PI) controllers since the derivative part is usually not used in practice (Åström and Hägglund, 2011). The implementation of PI controllers has always been performed by assuming that sensing, computation and actuation is performed periodically. With the introduction of networked control, the development of new PI tuning methods and controller structures was required. This originated from the fact that the network introduced large communication delays and loss of information. See (Eriksson, 2008) for an overview of design methods of PID controllers for NCSs. As already mentioned in Chapter 2, there is a great interest of introducing wireless communications in industrial process control. The main challenge of wireless NCSs in this application is that of guaranteeing suitable control performance while achieving the largest life span of the wireless devices.

As presented in Chapter 3, aperiodic sampling techniques for control, efficiently use the NCS resources, such as energy consumption of the wireless devices and network bandwidth. This is achieved by exchanging information between sensor, controllers and actuators only when relevant information is available. However, these techniques have not yet been implemented in real systems.

The steady-state performance required by most industrial processes has never been a concern for control engineers. Currently, PI controllers are employed periodically, achieving zero steady-errors errors, while rejecting any possible disturbances affecting the control loop. When introducing a new paradigm of aperiodic sampling, such as event-triggered control, these characteristics appear to be no longer valid. Particularly, the implementation of improper sampling techniques and controller structures may give rise to fast limit cycles of the process output, as observed by (Årzén, 1999; Cervin and Åström, 2007; Durand and Marchand, 2009; Vasyutynskyy and Kabitzsch, 2007). A common conclusion of previous studies has been that a large integral component of the controller, due to long time-intervals between control updates, appear to be the cause of fast limits cycles. Another issue arising from the implementation of event-triggered PI controllers is that a sticking effect may occur. This effect is characterized by the absence of events, even when the plant output is far from the desired setpoint. Previous works have proposed methods to enhance event-triggered PI controllers performance. However, no analysis of long-term steady state conditions were proposed. In this chapter, we propose solutions to these issues which allow for the introduction of event-triggered PI controllers in wireless industrial process control.

This chapter is organized as follows. We start by describing the problem we aim at solving in Section 5.1. In Section 5.2 we introduce the potential issues of implementing event-triggered PI controllers. An overview of aperiodic sampling methods is given in Section 5.3, followed by new event generation proposals. PI controller design and analysis is given in Section 5.4. The developed ideas are illustrated in simulation studies presented in Section 5.5. Section 5.6 gives an overview of the obtained results and concludes the chapter.

5.1 Problem Formulation

The design of event-triggered PI controllers for industrial process control is addressed in this chapter. Figure 5.1 depicts the event-triggered control architecture. Events are generated by an *event generator*, which decides when measurements should be communicated from the *sensor* to the *PI controller*. The PI controller is updated upon the arrival of new sensor measurements, and a new control input is computed. The control input is sent to the actuator, performing feedback control of the *plant*. The condition implemented at the event generator, such that if violated a transmission occurs, is denoted by *triggering rule*. The actuation is implemented in a Zero-Order-Hold (ZOH) fashion, the control input signal is constant between the arrival of sensor measurements.

Consider a linear time-invariant continuous and first-order stable plant in state-space form,

$$\dot{x}_p(t) = ax_p(t) + bu(t), \quad a < 0 \tag{5.1}$$

$$y(t) = x_p(t), \tag{5.2}$$

where $x_p(t) \in \mathbb{R}$ is the plant state, $u(t) \in \mathbb{R}$ is the control signal and $y(t)$ is the plant output. Let us denote the error $e(t) = r(t) - y(t) = r(t) - x_p(t)$, which describes the difference between the reference to be tracked and the current output. An ideal continuous-time PI controller calculates the control signal $u(t)$ as,

$$u(t) = K_p\Big(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\Big), \tag{5.3}$$

where K_p (proportional gain) and T_i (integration time) are the controller tuning parameters. Often, the PI controller is also expressed in the equivalent form

$$u(t) = K_p e(t) + K_i \int_0^t e(s)ds, \tag{5.4}$$

where $K_i = \frac{K_p}{T_i}$ and is denoted by the integral gain. Currently, PI controllers are implemented in a discrete-time form in digital devices. A sampled implementation of the PI controller, using Euler's approximation for the integral term, is given by:

$$x_c(t_{k+1}) = x_c(t_k) + \Delta T e(t_k) \tag{5.5}$$

$$u(t_k) = K_p e(t_k) + K_i x_c(t_k), \tag{5.6}$$

where $\Delta T = t_k - t_{k-1}$ is the inter-sampling time, t_k is the current sampling time, $x_c(t_k)$ is the integral state of the controller at time k and $e(t_k)$ is the output error at time t_k.

In this chapter, we aim at finding an event-triggered PI controller which:

1. Achieve similar control performances of a periodically sampled PI controller, when performing reference tracking and disturbance rejection and
2. Generate a low amount of sensor communications, while guaranteeing suitable control performances,

where control performance must be evaluated not only during the transient, but in steady-state conditions.

Usually in industrial process control systems, the model of the real plant is not available and simple first-order approximations are available, as noted in (Åström and Hägglund, 2006). We develop the event-triggered architecture to be model-independent. By model-independent we mean that event generators should not use the model in the triggering rule, and that model-based (feedforward) control should not be employed. In the current periodic control architectures, the reference signal is selected at the controller unit and proper reference tracking is performed. In the current event-triggered architecture, we would like to keep this constraint, where the reference signal is not transmitted to each event generator, but is available only at the controller. This restriction would avoid communication between controllers and sensors.

Before designing event generator mechanisms and PI controller structures we summarize the potential issues of event-triggered PI controller reported in the literature (Årzén, 1999; Cervin and Åström, 2007; Durand and Marchand, 2009; Vasyutynskyy and Kabitzsch, 2007).

5.2 Potential Issues with Event-Triggered PI control

When performing sensing and actuation in an event-triggered fashion, the information is transmitted aperiodically, possibly with quantized measurements with

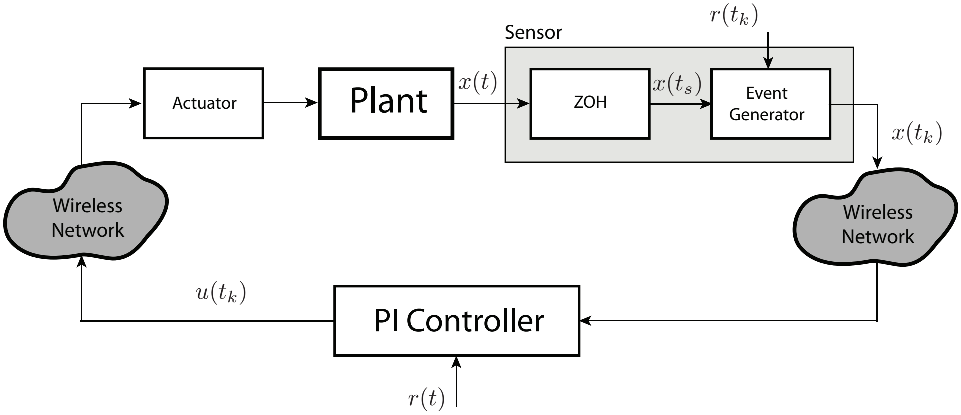
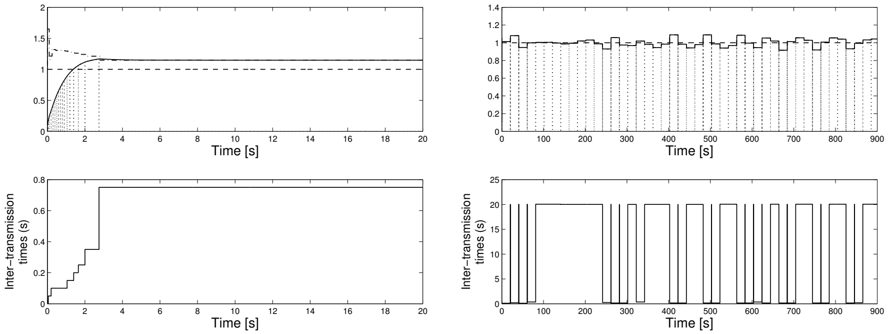


Figure 5.1: System Architecture for event-triggered control composed by sensors, actuators, controllers and event generators. A decision on the moments of transmission of information between components in the closed loop is managed by event-generators. Each component in the control loop may be perform synchronous or asynchronous actions.

large quantization steps. These two properties are known to generate fast limit cycles and large oscillations, which may have a large impact in the control performance (Vasyutynskyy and Kabitzsch, 2007; Cervin and Åström, 2007; Ploennigs et al., 2010). Another issue with event-triggered PI control is that output of the plant may "stick" and no new transmissions are sent to the controller by the event generator(Vasyutynskyy and Kabitzsch, 2007). This can happen regardless of how far the plant output is from the reference signal.

5.2.1 Sticking Effect

The sticking effect is characterized by the absence of new sensor transmissions to the controller, when the plant output, $y(t)$ is not equal to the reference signal. This issue may arise when using PI controllers in ZOH fashion a triggering rule with, e.g., output deadband sampling at the event generator, which is proposed in (Åström and Bernhardsson, 1999; Årzén, 1999; Vasyutynskyy and Kabitzsch, 2007). Figure 5.2(a) illustrates the sticking effect, when deadband sampling is applied together with a sampled PI controller (5.5) and (5.6), with aperiodic updates. This triggering rule consists on transmitting a new measurement when $|y(t) - y(t_k)| > \varepsilon$, where $\varepsilon = 0.05$. The control objective is to track a reference signal $r(t) = 1$. The output error remains at $e = -0.25$ after the last sample is taken at $t = 3.6$ s. No more events are generated since the control input value $u = 0.78$ achieved an output value $y = 1.25$ which is inside a deadband, no more deadband crossings occur. As observed in (Vasyutynskyy and Kabitzsch, 2007), the sticking effect happens in



(a) Sticking effect

(b) Large oscillations

Figure 5.2: The sticking effect and large oscillations occurring in event-triggered PI control implementations. Plant output (solid), control input (dash-dotted), samples (vertical dotted line), and reference signal (dashed). Figure 5.2(a) depicts the time-response and intertransmission times given by a sampled PI controller with aperiodic updates governed by deadband sampling. The sticking effect occurs at $t = 3.6$ s and no more events are generated. Figure 5.2(b) shows the effect of the addition of a time-out function in the event-generator, with time-out value of 20 s. A poorly designed event-triggered PI controller generates large oscillations of the plant output.

the moments when both output and control input become with opposite derivative signs, as in the peak of an overshoot.

The authors in (Årzén, 1999; Vasyutynskyy and Kabitzsch, 2007) suggest to introduce a time-out at the event generator to avoid this issue. Whenever a maximum period of time has passed, a new control update is performed. This feature may cancel the benefit of using event-triggered sampling, since sensors will eventually transmit periodically. If the period is selected too large, large oscillations may occur together with as large oscillations as we observe next. One way to avoid this issue is to take the reference signal into account at the event generator and design a triggering rule based on how far the output $y(t)$ is from the reference value $r(t)$. Therefore, it may not be possible to require the reference signal to be only available at the controller when performing event-triggered PI control. In Section 5.3 we propose suitable event generator designs to overcome the sticking effect.

5.2.2 Limit Cycles and Large Oscillations

Limit cycles and large undamped oscillations may occur in event-triggered PI control. These effects occur due to the following facts 1) the measurements received at the controller have large quantization step sizes, where not enough information is available to calculate the controller input to drive the output of the plant to the desired reference, and 2) the integral contribution of the state, may destabilize the system for large inter-transmission times. An example of this issue is depicted in Figure 5.2(b), where large oscillations are observed. The time-out method described earlier is used with time-out value of 20 s. This method is applied together with the classic PI controller (5.5), (5.6) for a scalar plant with $a = -1$, $b = 1$. Since a long time can occur between two transmissions, the integral state of the controller (5.5) may be too large, generating strong actuation inputs. For smaller time-out values this behavior could be reduced, but more transmissions would occur.

This issue introduces undesirable output behavior and a high number of message transmissions. Therefore, it is important that a suitable structure of the PI controller is designed, overcoming these drawbacks. We propose such controllers in Section 5.4.

5.3 Event Generator Design

In this section, we survey triggering schemes previously proposed in the literature and provide an analysis on their applicability to the event-triggered PI architecture. The event generator must be able to avoid the sticking effect and steady-state errors, while generating a low number of sensor transmission.

5.3.1 Previous Proposals

Several event-triggering methods have been proposed in the literature in the past decade. We survey all relevant triggering methods, and divide them according to requirement of the event generator to know the reference value, or if it can be only available at the controller. Moreover, we are also concerned if a specific triggering method is based on the plant model or not.

We denote $x_\Delta(t) = x(t_k) - x(t)$, where $x(t_k)$ is the last sampled value of the state, and $x(t)$ is the continuous-time state, $y_\Delta(t) = y(t_k) - y(t)$, where $y(t_k)$ is the last sampled value of the output, and $y(t)$ is the current output value.

No reference available at event generator

The following approaches do not require the knowledge of the the reference signal $r(t)$ at the event generator.

Trigger on $|y_\Delta(t)| \geq \varepsilon$ - This is the Lebesque sampling, introduce in (Åström and Bernhardsson, 1999, 2002), also referred as deadband sampling in e.g. (Otanez et al., 2002; Vasyutynskyy and Kabitzsch, 2007; Ploennigs et al., 2010). The main issue with this sampling method is that the sticking effect may occur as described in Section 5.2.1.

Trigger on $|\int_{t_k}^{t} y_\Delta(t)| \geq \varepsilon$ - This method is denoted as Integral Deadband (ID) sampling in (Ploennigs et al., 2010). This method is proposed to avoid the sticking effect. However, if a disturbance would take place after a sample is taken that would drive $y(t) = y(t_k)$ constant, then the system would still experience the sticking effect.

Both of these techniques are not able to eliminate the stickiness effect and must require additional conditions at the event generator. The authors propose the use of a time-out, which have the drawbacks observed in Section 5.2.1.

Reference available at event generator

These proposals assume that a reference signal $r(t)$ can be transmitted from the controller to the event generator.

Trigger on $|e(t)| = \varepsilon$ and if $e(t) > \varepsilon$ sample periodically - This case was proposed in (Heemels et al., 2008) and guarantees ultimate boundedness of the state $x(t)$. The benefits of event-triggered control are not maximized in this case, since the transient response will always be performed periodically.

Trigger on $\frac{|x_{\Delta}(t)|}{|x(t)-x_{ss}|} \geq \varepsilon$, $0 < \varepsilon < 1$ - This method was proposed in (Tabuada, 2007) and guarantees input-to-state stability of the plant, where ϵ is a design parameter. The model of the plant is required so x_{ss} can be calculated, where $x_{ss} = \lim_{t\to\infty} x(t)$ denotes the steady-state value of the plant, i.e., the equilibrium point. See (Altaf et al., 2011) for details on the application of this triggering rule for PI controllers. In (Garcia and Antsaklis, 2011b), the authors propose an extension of this technique to deal with model uncertainties.

Trigger on $V(t) \geq S(t)$ - This triggering rule was proposed in (Mazo Jr. et al., 2009, 2010) and is the basis of the aperiodic architecture in Chapter 3. The model of the plant is also required to be known at the event generator for designing V and S.

Trigger on $|x_{CT}(t) - x(t)| \geq \varepsilon$ - This method was proposed by (Lunze and Lehmann, 2010) and is defined as state-feedback event-based control. An event-based PI controller is proposed in (Lehmann, 2011) based on this triggering scheme. The state $x_{CT}(t)$ denotes the continuous-time value of the state given by the model, instead of comparing $x(t)$ to the last transmitted value $x(t_k)$, as seen in all the other triggering methods. Methods to deal with model uncertainties for this scheme are proposed in (Lehmann, 2011; Garcia and Antsaklis, 2011a).

As it can be seen, all but the triggering rule proposed by (Heemels et al., 2008), requires the full knowledge of the model, or at least an uncertain version of it, to be implemented. However, all methods required the reference signal to be known at the event generator. Since our motivation is not to use the plant model, we cannot design the event generator following these approaches. However, the methods proposed in (Garcia and Antsaklis, 2011a) appear to be very suitable for performing closed-loop system identification with event-triggered control. In all these cases, the sticking issue does not occur if the integral state of the controller is defined as part of the state $x(t)$, used in the triggering rule. See (Altaf et al., 2011) for an example of this approach.

5.3.2 New Proposals

In order to achieve suitable event generator designs for event-triggered PI control, we re-configure some triggering rules previously propose and propose a novel triggering method based on the control input signal.

Integral Deadband

To remove the sticking problem one can either implement a time-out mechanism at the event generator or allow the reference signal to be available at the event generator. Using an ID triggering rule may lead to the sticking effect only in the case of external disturbances affecting the plant. This characteristic motivated the design of the following rules:

Trigger on $|z_\Delta(t)| \geq \varepsilon_{ND}$ - This method is denoted as the *Norm Deadband* (ND), where we define $z_\Delta(t) = z(t_k) - z(t)$, and $z(t) = [x(t) \ x_c(t)]^T$, where $x_c(t)$ is the integral state of the controller (5.5). This value must be calculated at the event generator. This technique resembles the one proposed in (Tabuada, 2007), without the scaling of $z_\Delta(t)$ by $(z(t) - z_{ss})$. Note that without the scaling $z(t) - z_{ss})$, the knowledge of the plant model is no longer required. The drawback of this scheme is that taking Eucledian norm of the the extended state vector, one only accounts for the interaction of both states and not each independently. This may lead to conservative triggering of events by the event generator.

Trigger on $|x_\Delta(t)| \geq \varepsilon$ **and** $|x_{c\Delta}(t)| \geq \varepsilon_c$ - We denote this method as *State and Integrator Deadband* (SID), where $x_{c\Delta}(t) = x_c(t_k) - x_c(t)$. In this case we must define ε and ε_c separately which allows for increased design flexibility.

Although these triggering schemes require the availability of the reference value at the event generator, their design is not based on the plant model. Higher benefits with respect to a reduction of the number of sensor transmissions is expected these types of triggering rules, instead of sampling according to a maximum inter-transmission time as in the time-out scheme. Naturally, by imposing a maximum value for the inter-transmission times one is reducing the benefits of event-triggered control. The stickiness effect is avoided by designing the triggering rule based on the integral state $x_c(t)$ (5.5). The integral state will never stop increasing or decreasing while $y(t) \neq r(t)$, thus it generates events until $y(t) = r(t)$. Note that all the above schemes are based on either the state of the plant, the output error or the state of the controller. We now propose a different triggering rule, based on the control input signal, calculated at the event generator.

Control Input Sampling

If one has to drive the output of the system into a given reference value, and the system is already moving towards the reference, would it be necessary to take more samples? In this case, the input signal may have small variations during transient, and if a new control action would be issued, its effect could be almost negligible. Therefore, it appears to be unnecessary to generate an event based on the output $y(t)$ or the plant state $x(t)$ without taking the control input into account. We propose a triggering rule based on the control input, since we believe an event should only occur when a meaningful variation on control action is required. In this way, we propose the following triggering scheme,

$$|u(t_k) - u(t)| \geq \varepsilon, \quad t \geq t_k, \tag{5.7}$$

where $u(t_k)$ is the last control input signal sent to the plant actuator, and $u(t)$ is the computed control signal of the PI controller at the event generator. The triggering procedure is as follows. A ZOH samples the signal $x(t)$ at a fast rate $\Delta\bar{T}$, i.e., $t_{s+1} - t_s = \Delta\bar{T}$, and sends this value to the event generator. The controller at the event generator, calculates the integral state and control input value according to (5.5) and (5.6). Then, an event may be generated if (5.7) is violated. The underlying sensor sampling period $\Delta\bar{T} << \Delta T$, where ΔT is the inter-transmission time of sensor measurements. When an event is generated, either the control input value $u(t_k)$ at the triggering instant or the current plant state $x(t_k)$ could be transmitted to the controller. By transmitting $u(t_k)$, the control input value will be a quantized version, with quantization step ε, of the sampled PI controller (5.6). The control input being applied to the plant is then represented by:

$$u(t_k) = u(t_{k-1}) + \varepsilon. \tag{5.8}$$

Therefore, the control input will be constrained by the quantization step ε, where this value specifies how far the control input will be from the desired steady-state control input u_{ss} that achieves $y(t) = r(t)$, in steady-state. The smaller the ε the better performance is achieved, while more events will by generated. One drawback of this controller is that a limit cycle will always exist if the control deadband $u(t_k) \neq u_{ss}$ for all t_k. Only if a deadband crossing point is placed exactly at the u_{ss} value, one achieves zero steady-state errors. If that case does not occur, the controller will always be characterized by limit cycles in steady-state, where the control input will vary around the steady-state control input u_{ss} value. The frequency of the limit cycles will be given by the PI parameters combined with the deadband value ε. Note that by governing the transmission of sensors measurements by a triggering rule based on a PI controller value $u(t)$, no stickiness effect occurs, since if the PI controller contains an integral state. Therefore, $u(t)$ will be increasing or decreasing while $y(t) \neq r(t)$.

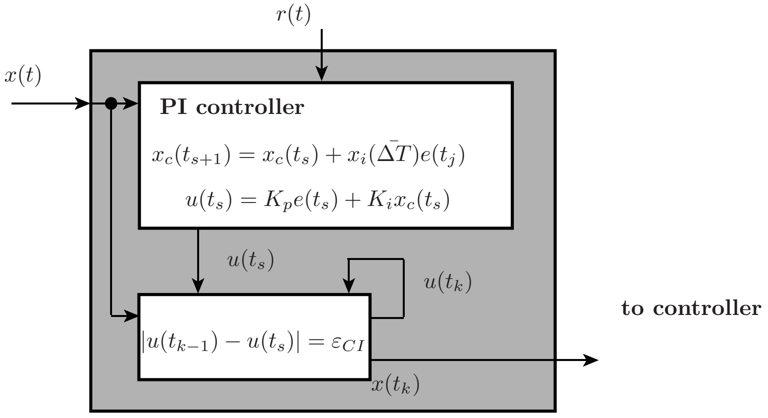

$r(t)$
$x(t)$
PI controller
$x_c(t_{s+1}) = x_c(t_s) + x_i(\bar{\Delta T})e(t_j)$
$u(t_s) = K_p e(t_s) + K_i x_c(t_s)$
$u(t_s)$
$u(t_k)$
to controller
$|u(t_{k-1}) - u(t_s)| = \varepsilon_{CI}$
$x(t_k)$

Figure 5.3: Event generator design for control input sampling. In this case, a PI controller is implemented at the event generator. An event is generated when the difference between the calculated control input $u(t)$ and the current control input value $u(t_{k-1})$ differ by a quantity ε_{CI}.

A solution to the limit cycles issue is to switch off the event generator when the plant output $y(t)$ is close enough to the reference value. In that case, a zero steady-state error is achieved, but can be small if ε is properly designed. Another solution can be to generate events based on control input $u(t)$, but instead of applying the control input value at event times $u(t_k)$ to the plant, we let the PI controller decide which value to apply. The main idea is to decouple the PI controller for which the event-generator is designed, from the one that is responsible for the actuation of the plant, and implemented at the controller. Different types of controllers at the event generator and at the controller can be implemented to achieve suitable performances. The controller at the event generator runs periodically with period $\Delta\bar{T}$, and the control algorithm at the controller, is updated by the reception of new sensor messages, occurring with inter-transmission time $\Delta T >> \Delta\bar{T}$. Therefore, suitable controllers to reduce or eliminate limit cycles can then be designed at the controller. We define this triggering rule as *Control Input* sampling (CI). Figure 5.3 depicts the triggering mechanism.

The triggering rule we present is solely based on the control input value computed at the event generator. However, we believe that other types of triggering rules could eliminate the sticking effect and generate events strictly when necessary. We are currently looking at finding other triggering rules which would represent the real need of a control update to be performed, as an optimal triggering rule.

5.4 PI Controller Implementations

Suitable PI controller structures must be devised to overcome the limit cycles and large oscillations observed in Section 5.2. We start by presenting a formal structure for representing generic PI controllers, which are instrumental for the interpretation of the PI controller behavior when implemented in an aperiodic fashion. Furthermore, we propose the use of an enhanced PI controller based on a controller structure that has been proposed in (Song et al., 2006), for wireless industrial process control, and provide simple analysis on the stability of such controller.

5.4.1 Sampled PI Controller

Recall that the sampled implementation of a PI controller is represented by (5.5) and (5.6). Let us now introduce a new state variable $x_i(t)$ which we denote the *integral weight*. Rewriting (5.5) and introducing $x_i(t_k)$, we obtain

$$x_c(t_{k+1}) = x_c(t_k) + x_i(\Delta T)e(t_k) \tag{5.9}$$

which together with the control input $u(t_k)$ (5.6), we denote as the *generic PI structure*. In order for (5.9) to represent the periodic implementation of a sampled PI controller as in (5.5), $x_i(t_k)$ is defined as,

$$x_i(\Delta T) = t_{k+1} - t_k = \Delta T, \tag{5.10}$$

where the integral weight $x_i(\Delta T)$ is reset to zero every time an event is generated and a sampling occurs.

It is clear that as the time from the last sample increases, the integral weight $x_i(\Delta T)$ grows linearly with ΔT, leading to high integral state values and strong control actions. This fact is responsible for generating limit cycles and oscillations in steady-state when sampled PI controllers are implemented aperiodically, as seen in Section 5.2.2 and reported in (Vasyutynskyy and Kabitzsch, 2007). An interpretation of this situation is that the PI controller does not yield the closed-loop system stable for such large inter-transmission times. In that case, whenever the controller is updated, a destabilizing control input is applied, generating short inter-transmission times and suitable control inputs. This cycle will continue to occur and no asymptotic convergence to the reference value is possible to be achieved. However, this effect may be eliminated by properly designing $x_i(\Delta T)$ as we present next.

5.4.2 Modified PIDPLUS Controller

A new PID controller denoted as PIDPLUS has been proposed in (Song et al., 2006) to cope with packet losses and delays when PI controllers are implemented over networks. The main idea of this PID structure is to scale the integral and derivative terms as a function of the inter-sampling time, ΔT. Since our study deals with PI controllers, we disregard the derivative part of the PIDPLUS. We will first present the structure as it appears in (Song et al., 2006), and show that it is possible to transform it to the generic PI structure as in (5.9).

From (Song et al., 2006), the PI controller is defined as,

$$F(t_{k+1}) = F(t_k) + \Big[u(t_k) - F(t_k)\Big]\Big(1 - e^{-\frac{\Delta T}{Ti}}\Big) \tag{5.11}$$

$$u(t_k) = K_p e(t_k) + F(t_k) \tag{5.12}$$

where $F(t_k)$ is denoted as a "filter" by the authors, and replaces the integral term of the sampled PI controller. Although presented with a special structure, the PIDPLUS can be expressed in a generic PI controller form. By inserting (5.12) in (5.11), and denoting $F(t_k)$ as $x_c(t_k)$, we obtain

$$x_i(\Delta T) = \frac{K_p}{K_i}(1 - e^{-\frac{K_i}{K_p}\Delta T}). \tag{5.13}$$

Recall that $x_i(\Delta T)$ is reset to zero at each sampling event. This is the PIDPLUS controller represented in the generic PI structure form.

We observe that the benefit of using PIDPLUS comes from the fact that $x_i(\Delta T)$ is always bounded by $\frac{Kp}{Ki}$ as the inter-transmission time ΔT increases, instead of growing linearly with ΔT as the periodically sampled implementation. However, for small ΔT, both controllers are the same. This is proved by performing a Taylor's expansion of the term $e^{-\frac{K_i}{K_p}\Delta T}$ in (5.13) and so,

$$x_i(\Delta T) = \frac{K_p}{K_i}\Big(1 - 1 + \frac{K_i}{K_p}\Delta T + O(\Delta T^2)\Big) \tag{5.14}$$

$$= \Delta T + O(\Delta T^2), \quad \Delta T \to 0 \tag{5.15}$$

The integral weight $x_i(\Delta T)$ updating the integral state $x_c(t_k)$ (5.9) for large inter-transmission times is small resulting in softer control inputs. Since K_i multiplies by $x_c(t_k)$ in (5.6), the update of $x_c(t_k)$ is governed by $K_p(1 - e^{-\frac{Ki}{Kp}\Delta T})e(t_k)$, since K_i is canceled in (5.13). Therefore, the parameter K_p will be of major importance for determining the behavior of the controller response, according to the inter-transmission time ΔT.

The PIDPLUS has the goal of reducing limit cycles when in steady-state and achieving suitable control performances. Our intuition is that the integral update is the main cause of aggressive control actions, and therefore, the integral weight $x_i(t_k)$ must be properly tuned.

Normally, in the case of first order systems, fast limit cycles occur and are characterized by a fast switching between a sequential and finite set of control input values, around the steady-state control input u_{ss}. This occurs since the implemented controller, receiving measurements in an aperiodic fashion given by the limit cycle, is not able to converge to u_{ss} as in a periodic implementation.

In order to smooth the frequency of the limit cycle and any large output oscillations, we propose the use of a control input smoothing filter in the form,

$$\bar{u}(t_k) = \alpha u(t_k) + (1-\alpha)\bar{u}(t_{k-1}), \tag{5.16}$$

where $\bar{u}(t_k)$ represents the filtered value of the control input, which is sent to the actuator. The parameter α must be tuned appropriately to obtain the desired smoothing of output oscillations. We remark that this filter is in discrete-time form, and α does not depend on the inter-sampling time.

In this section, we have proposed a generic PI controller structure wish provide an insight on the behavior of PI controllers when implemented in an aperiodic fashion with large inter-transmissions times. Moreover, we have presented the PIDPLUS controller which may be suitable for event-triggered PI control, if properly tuned. Additionally, we proposed the modification of the PIDPLUS with the addition of a smoothing filter to its output. We now provide simple stability analysis of the presented PI controllers in order to gain analytical intuition to its behavior.

5.4.3 Stability Analysis

No evaluation of the PIDPLUS in an event-triggered setting has been made in the current literature. However, analysis on the performance under packet losses and delays of the PIDPLUS against a PI and PID controllers have been proposed by (Kaltiokallio et al., 2010; Ungan, 2010).

Both the periodically sampled PI controller and the PIDPLUS introduced earlier are modelled by discrete-time dynamics and have aperiodic updates of the integral weight $x_i(t_k)$ and the integral state of the controller $x_c(t_k)$. Additionally, the plant is described by a continuous-time differential equation. Therefore, the closed-loop system is described by a hybrid system with both continuous and discrete-time dynamics, where its analysis are not trivial. To the best of our knowledge, stability analysis of such hybrid dynamical system with aperiodic updates has not yet been proposed in the current literature. However, we are able to perform analysis of the stability of the closed-loop system, when the inter-transmission time ΔT is constant, by using analysis tools from sampled data systems theory (Åström and Wittenmark, 1990).

We start by analyzing the stability of a plant controlled by the periodically sampled PI controller and the standard PIDPLUS. Afterwards, we provide an analysis on the modified PIDPLUS. Consider the discrete-time process given by sampling (5.1) with a ZOH with sampling interval ΔT as,

$$x_p(t_{k+1}) = a_d x_p(t_k) + b_d u(t_k) \tag{5.17}$$

$$y(t_k) = x_p(t_k), \tag{5.18}$$

where $a_d = e^{a\Delta T}$ and $b_d = \int_0^{\Delta T} e^{as} ds b = \frac{b}{a}(e^{a\Delta T} - 1)$. Assume that a control action (5.6) is applied to the process, and the integral state is given by (5.9), which we had defined as the generic PI structure. By substituting (5.9) in (5.6), and using it in (5.1), we achieve the closed-loop system given by,

$$\bar{x}(t_{k+1}) = \underbrace{\begin{pmatrix} e^{a\Delta T} + K_p \frac{b}{a}(e^{a\Delta T} - 1) & K_i \frac{b}{a}(e^{a\Delta T} - 1) \\ x_i(\Delta T) & 1 \end{pmatrix}}_{H(\Delta T)} \bar{x}(t_k), \tag{5.19}$$

where $\bar{x}(t_k) = [x_p(t_k)\ x_c(t_k)]^T$. We now derive the stability of the closed-loop systems (5.19) using the Jury stability criterion (Åström and Wittenmark, 1990).

Proposition 5.4.1. *The characteristic discrete-time polynomial of system* (5.19) *can be represented by:*

$$P(z) = z^2 - Tr(H(\Delta T)) + det(H(\Delta T)), \tag{5.20}$$

where, $Tr(\cdot)$ represents the Trace and $det(\cdot)$ the determinant of the matrix $H(\Delta T)$. The necessary conditions for stability are:

1. $\left| det\Big(H(\Delta T)\Big) \right| < 1$
2. $1 - Tr\Big(H(\Delta T)) + det(H(\Delta T)\Big) > 0$
3. $1 + Tr\Big(H(\Delta T)) + det(H(\Delta T)\Big) > 0$

By solving condition 1 we achieve,

$$\left| e^{a\Delta T} + \frac{b}{a}(-K_i x_i(\Delta T) - K_p)(e^{a\Delta T} - 1) \right| < 1,$$

which for stable processes, i.e, $a < 0$, condition 1 is given by,

$$0 < \frac{b}{a}(K_i x_i(\Delta T) + K_p) < 2. \tag{5.21}$$

Condition 2 can be simplified as,

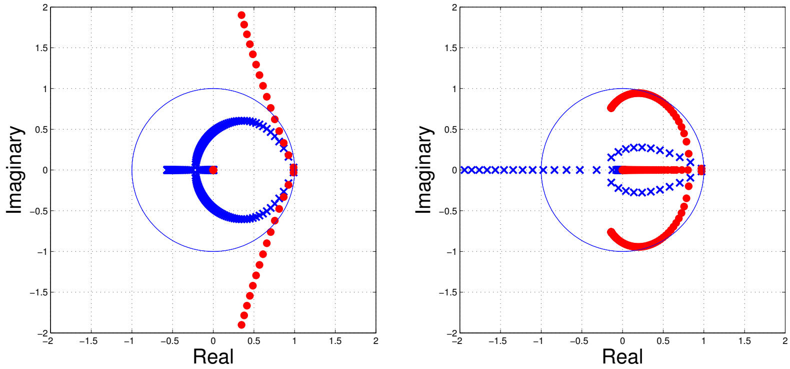
$$K_i x_i(\Delta T) b > 0, \tag{5.22}$$

while condition 3 can be represented by,

$$2(1 + e^{a\Delta T}) + \frac{b}{a}(K_i x_i(\Delta T) + 2K_p)(e^{a\Delta T} - 1) > 0, \tag{5.23}$$

which for the cases,

- $\frac{b}{a}(K_i x_i(\Delta T) + 2K_p) < 0$, *condition 3 is verified for all values of a and for*
- $\frac{b}{a}(K_i x_i(\Delta T) + 2K_p) > 0$, *condition 3 is verified if condition 1 is true.*



(a) Stability region of periodically sampled PI and the original PIDPLUS controller

(b) Stability region of an original PIDPLUS controller and the modified PIDPLUS.

Figure 5.4: Stability regions for a first-order plant with $a = -0.1$, $b = 0.1$ for discretization steps $\Delta T \in [0.1, 100]$. In Figure 5.4(a) is depicted the stability region for the periodically sampled PI controller (●) and the PIDPLUS (×) controller, with adjusted K_p parameter.
Figure 5.4(b) shows the stability region of the standard PIDPLUS using a non adjusted K_p (×), and the modified PIDPLUS with filter parameter $\alpha = 0.4$ (●).

Conditions (5.21), (5.22) and (5.23) can then be verified for different choices of parameter $x_i(\Delta)$ and parameters K_p and K_i, in order to guarantee stability of the closed-loop system. With a periodically sampled PI controller the conditions are not verified for $\Delta T \leq \frac{2a}{K_i b} - K_p/K_i$. Independently of the tuning of K_p and K_i there is always a sampling period for which the closed-loop system is unstable. However, for the PIDPLUS controller, the designed $x_i(\Delta)$ can be made stable for all sampling periods ΔT, if the parameter K_p is properly tuned. Intuitively, we expect the PIDPLUS to achieve higher control performances than a periodically sampled PI controller, since even for large inter-transmissions times the PIDPLUS renders the closed-loop system.

Figure 5.4(a) depicts the eigenvalues of the closed-loop system matrix $H(\Delta T)$, for a first order plant with $a = -0.1$, $b = 0.1$, for different inter-sampling times steps $\Delta T \in [0.1, 100]$ for the periodically sampled PI controller (●) and the PIDPLUS (×), with an adjusted K_p parameter to guarantee the stability conditions. Note that for stability, $H(\Delta T)$ must be Schur, i.e., the eigenvalues should be inside the unit circle (blue circle).

When applying the modified PIDPLUS proposed in Section 5.4.2, with control action $\bar{u}(t_k)$, the closed-loop system becomes,

$$\bar{z}(t_{k+1}) = \underbrace{\begin{pmatrix} e^{a\Delta T} + \bar{K}\alpha & \bar{K}\alpha & \bar{K}(1-\alpha) \\ x_i(\Delta T) & 1 & 0 \\ \alpha K_p & \alpha K_i & (1-\alpha) \end{pmatrix}}_{\bar{H}} \bar{z}(t_k), \tag{5.24}$$

where $\bar{z}(t_k) = [x_p(t_k)\ x_c(t_k)\ \bar{u}(t_{k-1})]^T$ and $\bar{K} = K_p \frac{b}{a}(e^{a\Delta T} - 1)$. Now considering the limit case, we achieve the closed-loop system matrix, We skip the presentation of stability conditions for the modified PIDPLUS, which we can instead compute numerically. For stable systems, it is possible to find suitable α values, that combined with the controller parameters, achieve $\bar{H}$ Schur. In this way, the closed-loop system will be asymptotically stable for any sampling period in the same way as the original PIDPLUS. Figure 5.4(b) depicts the eigenvalues of the closed-loop system matrix $\bar{H}$, for a first-order plant with $a = -1$, $b = 1$, for different sampling periods $\Delta T \in [0.1, 100]$, with a PIDPLUS with a non adjusted K_p parameter (×) and the modified PIDPLUS with $\alpha = 0.4$ (●). Note that the system is stabilized for any sampling period ΔT. However, the robustness margin of the modified PIDPLUS controller is decreased for this choice of α, when compared to the original PIDPLUS from Figure 5.4(a), since the eigenvalues are closer to the unit circle.

Even though simple stability analysis are achieved for the periodic case, analyzing the situation where ΔT is time-varying, is in the scope of our future work. However, these results give us good indications on 1) how to tune the parameters of the PIDPLUS controller and 2) how to tune the α parameter of the output filter. Moreover, we can clearly observe why a periodically sampled PI controller should not be used for event-triggered control.

5.5 Simulation Studies

We now evaluate the performance of event-triggered PI controllers. We are particularly interested in evaluating the performance of the proposed schemes against the classic periodic implementation, under different control and event-generator policies. Moreover, we evaluate event-triggered PI controllers in first-order plus time-delay (FOTD) models which approximate real plants. These models are commonly known as KLT models. See Appendix A.1 for more details in the KLT model. The selected models are a) balanced lag-delay, b) delay dominated and c) lag dominated processes found in industry (Åström and Hägglund, 2006). The AMIGO tuning rules for PI controllers is used in each of the examples used. See Appendix A.1 for a summary of this tuning method.

We start by performing an analysis on the control performance with respect to the IAE and the number of samples ($N_{samples}$) for the selected event-generator schemes, considering the three types of KLT processes. After this, an evaluation of the transient and steady-state performance considering a combination of different controllers and event-generators is presented for a lag-dominated KLT process. We only consider the analysis for the lag-dominated process since it is the one with the higher demand of a properly designed event-triggered PI controller. This is due to the fact that for this type of processes, the proportional gain is very high, which has a high influence on the stability of the closed-loop system, as presented in Section 5.4.3. The transient performance of a control loop can be characterized according to the rise-time T_r, the settling-time T_s and the overshoot M (Åström and Hägglund, 2006). For steady-state condition analysis we look at the number of samples taken by the sensor ($N_{samples}$), which give a clear indicator if there exists limit cycles and large oscillations in the system. Moreover, the IAE parameter can be seen as a general indicator for both transient and steady-state performance.

The processes to be evaluated are presented next, and are part of the "test batch" introduced in (Åström and Hägglund, 2006) to achieve the AMIGO tuning rules. The KLT approximations of each process were also taken from (Åström and Hägglund, 2006).

In this setup, a step reference change occurs at start time $t = 0$, where $r(t) = 1$ and $y(0) = x_p(0) = 0$. A load step disturbance $d(t) = 0.2$ is introduce at $t = 500$ s. The simulation has a duration of 900 s.

Balanced lag-delay

We consider the system with transfer function:

$$P(s) = \frac{1}{(s+1)^4}, \tag{5.25}$$

which the KLT approximation is represented by $K = 1$, $L = 1.42$ and $T = 2.90$. This example has the characteristics of a balanced lag-delay process.

The PI controller parameters using the AMIGO tuning rules for this process is $K_p = 0.414$ and $K_i = 0.16$.

Delay dominated

As an example of a delay dominated process we introduce the system with transfer function:

$$P(s) = \frac{1}{(1+0.05s)^4} e^{-s}, \tag{5.26}$$

which the KLT approximation is $K = 1$, $L = 1.01$ and $T = 0.0932$.

The AMIGO tuning rules for designing the PI controller for this process give $K_p = 0.175$ and $K_i = 0.486$.

Lag dominated

An example of a lag dominated can be expressed by the following transfer function:

$$P(s) = \frac{1}{(s+1)(1+0.1s)(1+0.01s)(1+0.001s)}, \tag{5.27}$$

which is represented with $K = 1$, $L = 0.075$ and $T = 1.04$ for the KLT approximation.

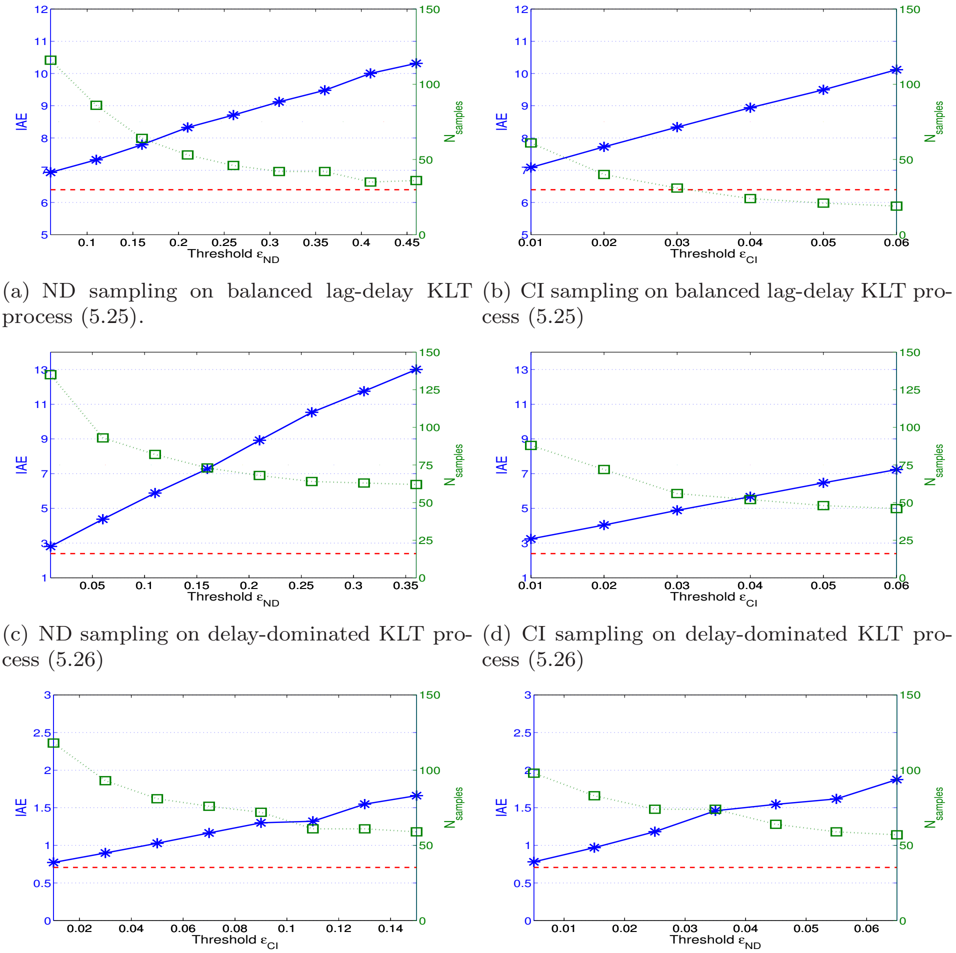
The PI parameters given by the AMIGO tuning rules is $K_p = 4.13$ and $K_i = 7.66$.

5.5.1 Event Generator Analysis

We now provide a comparative analysis of the most suitable policies for event generation. As stated earlier in Section 5.3.2, simple absolute output deadband (AOD) is not suitable for control purposes since the sticking effect may take place. Other methods were proposed to avoid these issues, and that not require the knowledge of the plant model to be implemented. The analysis will be performed with respect to the Norm Deadband (ND) and the Control Input Deadband (CI). We do not evaluate the State and Integrator Deadband (SID) scheme, since no advantages over ND were observed.

In this case, both ND and CI are implemented together with an original PIDPLUS controller, and applied to the balanced lag-delay (5.25), delay-dominated (5.26) and lag-dominated (5.27) KLT plants. The parameter K_p is adjusted to guarantee the stability conditions for each of the processes.

The CI is implemented with the PIDPLUS controller for the control input calculation at the event generator. Moreover, the controller actuation in the plant is performed using a PIDPLUS, receiving sensor measurements from the event generator, in order to avoid limit cycles. No disturbance was applied in this case. Figure 5.5 shows the comparison between the performance of the ND and CI event generation schemes, with respect to IAE and $N_{samples}$, for different threshold values and for all three types of KLT processes. For the case of balanced lag-delay and delay dominated processes, it is clear that the CI sampling scheme outperforms the ND scheme, since to achieve the same IAE, CI requires a much lower number of samples. The CI scheme is more efficient with respect to control performance and number of samples than the ND for these types of processes. This is achieved since less samples are taken during the transient in the CI scheme since the decision is based upon the control input. The dashed line in the figure depicts the IAE obtained with the classic periodic implementation with period $T = 0.05$s, using the same PIDPLUS controller. For an experiment of 900 s a periodic scheme generates 18000 samples. As it can be seen, the event-triggered implementation has an IAE value very close to the periodic implementation, and using less than 1% of the samples. Implementing the periodic scheme with period $T = 9$ s in order to generate the same number of samples of the event-triggered case, the closed-loop system performance worsens and an IAE=32.93 is obtained, which is approximately 3 times higher than with the event-triggered scheme. Moreover, the settling time of the periodic scheme is approximately 100 times higher. Likewise, due to large inter-transmission times, this implementation would perform a slow disturbance rejection. When considering the lag-dominated process, the results point to slightly different conclusions with respect to performance of the event generator. For this type of process, both ND



(a) ND sampling on balanced lag-delay KLT process (5.25).

(b) CI sampling on balanced lag-delay KLT process (5.25)

(c) ND sampling on delay-dominated KLT process (5.26)

(d) CI sampling on delay-dominated KLT process (5.26)

(e) ND sampling on lag-dominated KLT process (5.27)

(f) CI sampling on lag-dominated KLT process (5.27)

Figure 5.5: Comparison between ND and CI sampling for balanced lag-delay (5.25), delay-dominated (5.26) and lag-dominated (5.27) KLT plants with the PIDPLUS controller. The dashed line represents the IAE value of a periodic implementation of the PIDPLUS with period $\Delta T = 0.05$, which is seen as the lower bound in performance, but generating 18000 sensor transmissions. No disturbances is applied in this case and only reference tracking is performed.

and CI generate approximately the same number of samples, obtaining similar IAE results. In fact, when trying to achieve control performances closer to the periodic implementation, the ND requires less samples. However, for less performance requirements with respect to IAE, there is a slight advantage of the CI scheme over the ND. When analyzing other type of lag-dominated processes, we verified that for larger values of delay, while still maintaining $L << T$, CI sampling becomes increasingly better than ND. This fact is not well understood and is scope of our current research. In any case, the same conclusions apply to motivate the benefits of performing event-triggered control over periodic control for lag-dominated processes.

By performing these type of experiments, one is able to pick a threshold that gives the best trade-off between the control performance and number of samples, which must be decided by the control engineer. We remark that no noise was introduced in the system. In the presence of noise, having small thresholds would increase the number of samples generated, and so, a careful choice of the threshold should be made. A closer analysis with respect to the transient and steady state performance will be shown next.

5.5.2 Transient and Steady-State Analysis

Here we present an analysis of the combination of PI controllers and event generators according to transient and steady-state performance indicators. We start by evaluating the PI designs presented in Section 5.4 when implemented periodically. In this way we analyze the benefits and drawbacks of each case, without the influence of the event generation design. Afterwards, we analyze the PI designs when applied together with the event generators proposed in Section 5.3. The disturbance is applied during each of evaluated cases.

Table 5.1 depicts the evaluation of the classic PI controller and the original PIDPLUS and modified PIDPLUS controllers with periodic ($\Delta T = 0.05$s) sampling and actuation, for the lag-dominated process. We denote the modified PIDPLUS as MPIDPLUS in the results. We analyze the case where the classic AMIGO tuning method is used, followed by the AMIGO tuning with adjusted K_p parameter. As it can be seen, the use of the AMIGO tuning gives good responses and the lowest IAE, as expected. When adjusting the K_p parameter, so the closed-loop system is stable under any sampling period, one can see that the IAE is degraded with an increased rising time, settling time and overshoot. If instead of decreasing K_p, the modified PIDPLUS (5.16) is introduced, the performance degradation is much lower since the controller gains can still be kept high, while the parameter α works as a smoothing filter of the control input.

One natural question when implementing event-triggered controllers is about the total number of samples that are generated during the transient and in steady-state conditions. Naturally, one would like to reduce or eliminate completely any event generations during steady-state since the process output is very close to the desired reference. In order to evaluate this, we propose the comparison of the usage of the Classic PI and the PIDPLUS controller using ND sampling, with constant threshold $\varepsilon_{ND} = 0.02$. We split the number of samples generated during transient

Table 5.1: Performance analysis of the classic PI, PIDPLUS and modified PIDPLUS (MPIDPLUS) controllers with periodic ($\Delta T = 0.05$s) sampling and actuation, for the lag-dominated process. The controllers are evaluated under AMIGO tuning rules, with and without the adjustment of the K_p parameter. The classic PI is also evaluated with the application of filter (5.16).

Controller	Sampling	K_p	K_i	α	ΔT	ε_{CI}	ε_{ND}	T_r	T_s	M	IAE	$N_{samples}$
Classic PI	Periodic	4.13	7.67	1.00	0.05	-	-	0.21	1.10	26.96	0.33	18000
PIDPLUS	Periodic	4.13	7.67	1.00	0.05	-	-	0.21	1.12	25.10	0.33	18000
Classic PI	Periodic	1.90	7.67	1.00	0.05	-	-	0.38	4.53	44.62	0.84	18000
PIDPLUS	Periodic	1.90	7.67	1.00	0.05	-	-	0.39	3.74	39.47	0.77	18000
Classic PI	Periodic	4.13	7.67	0.70	0.05	-	-	0.23	1.41	33.14	0.38	18000
MPIDPLUS	Periodic	4.13	7.67	0.70	0.05	-	-	0.24	1.06	31.11	0.36	18000

i.e. $t \in [0, 100]$s, and the overall number of generated samples for $t \in [0, 900]$s. The results presented in Table 5.2 show that the classic PI controller is not suited for event-triggered control. In this case, limit cycles occur, generating 1754 samples during the total experiment and 199 during transient. By applying the PIDPLUS controller with the classic AMIGO tuning rules the limit cycles are reduced, and very close transient performance to the periodic implementation, with respect to T_r, T_s and M are obtained. However, limit cycles still occur since 799 samples are generated throughout the experiment. This happens since the K_p parameter given by the AMIGO tuning rules does not satisfy the stability conditions in Proposition 5.4.1. If either an adjustment of the controller parameters to create a stable closed-loop system is performed, or a modified PIDPLUS is introduced, the limit cycles are eliminated and a drastic reduction of the number of samples is obtained. A higher penalty on the transient response is seen when adjusting the controller gain K_p, than by introducing the modified PIDPLUS, as it was seen in the periodic implementations.

We now compare the performance of the event-triggered PIDPLUS and modified PIDPLUS controllers when using ND and CI as the event generator policies, evaluated with AMIGO and AMIGO tuning with adjusted K_p parameter. A threshold of $\varepsilon_{ND} = 0.02$ and $\varepsilon_{CI} = 0.03$ are chosen since they gave approximately the same IAE performance, as presented in Figure 5.5.

Table 5.3 presents the obtained results. It is clear that by using the classic AMIGO tuning rules, limit cycles with high frequency occur during steady-state with both the overall IAE and the number of samples generated being very high. However, better transient responses with respect to T_r, T_s and M are obtained, for both CI and ND sampling schemes, with CI having a lower IAE than ND. When the K_p parameter is adjusted, limit cycles disappear, and it can still be seen a clear advantage on using CI sampling. The modified PIDPLUS is evaluated for each sampling method, with $\alpha = 0.4$ for CI sampling and $\alpha = 0.7$ for ND sampling. In this case, the IAE is improved for both sampling schemes, where the number of

Table 5.2: Comparison between classic PI, PIDPLUS and modified PIDPLUS (MPIDPLUS) controllers for ND sampling with respect to the number of samples during transient and steady-state conditions.

Controller	K_p	K_i	α	ε_{ND}	T_r	T_s	M	IAE	$N^{[0,100]}_{samples}$	$N^{[0,900]}_{samples}$
Classic PI	4.13	7.67	1.00	0.02	0.21	1.34	26.97	13.59	199	1754
PIDPLUS	4.13	7.67	1.00	0.02	0.22	1.14	24.35	5.80	91	799
PIDPLUS	1.90	7.67	1.00	0.02	0.41	7.82	38.97	1.19	70	92
MPIDPLUS	4.13	7.67	0.70	0.02	0.26	2.60	30.90	0.73	40	54

Table 5.3: Performance analysis of the classic PI, PIDPLUS and modified PIDPLUS (MPIDPLUS) controllers with event-triggered sampling and actuation, for the lag-dominated process. The controllers are evaluated under AMIGO tuning rules, with and without the adjustment of the K_p parameter.

Controller	Sampling	K_p	K_i	α	ΔT	ε_{CI}	ε_{ND}	T_r	T_s	M	IAE	$N_{samples}$
PIDPLUS	CI	4.13	7.67	1.00	-	0.03	-	0.21	1.11	25.04	2.51	818
PIDPLUS	ND	4.13	7.67	1.00	-	-	0.02	0.22	1.14	24.35	5.80	799
PIDPLUS	CI	1.90	7.67	1.00	-	0.03	-	0.39	4.93	39.27	1.09	128
PIDPLUS	ND	1.90	7.67	1.00	-	-	0.02	0.41	7.82	38.97	1.55	125
MPIDPLUS	CI	4.13	7.67	0.40	-	0.03	-	0.28	2.51	42.04	0.79	114
MPIDPLUS	ND	4.13	7.67	0.70	-	-	0.02	0.26	2.60	30.90	1.43	125

samples is only decrease for the CI case.

5.6 Discussion

In this chapter, we presented a study of the potential issues of performing event-triggered and proposed suitable solutions to implement event-triggered PI controllers for first-order, stable systems. By using the proposed methods, the control performance of an event-triggered implementation is kept very close to a classic periodic PI controller implementation, under various conditions. Moreover, we have shown that there exists a tradeoff between the achievable control performance and number of events generated. If battery lifetime and network bandwidth usage is of major concern, then a penalty on control performance must occur. On the other hand, increased performance can always be obtained by allowing tighter triggering rules. However, we must not that very close performance is achieved by an event-triggered PI controller when compared to a classic PI controller implementation, where a much lower number of events are generated.

With respect to the proposed event generation methods, we proposed three solutions which eliminate the sticking issue. By applying a deadband on the control input value appears to deliver the best tradeoff between control performance and events generated. However, further investigations of the optimal triggering rule are required.

We also elect both PIDPLUS and the modified PIDPLUS as suitable controller choices for event-triggered PI control. By properly designing the integral weight and adjusting the proportional gain of the controller, the PIDPLUS controller is able to guarantee stability of the closed-loop system for any sampling periods, removing limit cycles and large oscillations in steady-state. The modified PIDPLUS provided similar benefits, but significantly increased closed-loop transient performances when compared to the original PIDPLUS. We aim at investigating the stability properties under aperiodic sampling and propose new tuning methods for both PI controllers in our future work.

Chapter 6

Wireless Network and Control Co-Design

The tradeoff between tractability and accuracy of the analytical model of a wireless network is important in order to hide the system complexity through a suitable abstraction without losing critical aspects of the network. Furthermore, WSNs require an energy-efficient operation due to the limited battery power of each sensor node. When performing control over wireless networks, the system designer has to take into account all these details, and both control and communication characteristics should be addressed. As an example, if one aims at achieving better control performances, a faster sampling rate should be used for performing control of a given system. However, requiring the use of higher bandwidth may increase network congestion, introduce loss of information, and higher transmission delay variance.

We propose a different framework to deal with this problem, where the system designer is able to jointly tune parameters of the control and communication system, optimizing the energy consumption of the network, and guaranteeing a desired control performance.

In this work, we present two original contributions:

1) We investigate the wireless network effects on the control performance.

2) We propose a co-design approach to meet the required control performance while minimizing the energy consumption of the network.

The key issue addressed here is how to derive the explicit relation between the performance of the control systems and the characteristics of a wireless network. Furthermore, the well-defined design procedure is required in order to achieve high performances in NCSs.

The outline of the chapter is as follows. Section 6.1 defines the considered problem of control over a wireless network. In Section 6.2, we describe the IEEE 802.15.4 network model. The design of the control is presented in Section 6.3. In Section 6.4, we discuss the proposed co-design approach. In Section 6.4.1, we illustrate it through numerical examples. Section 6.6 discusses the results achieved of the chapter.

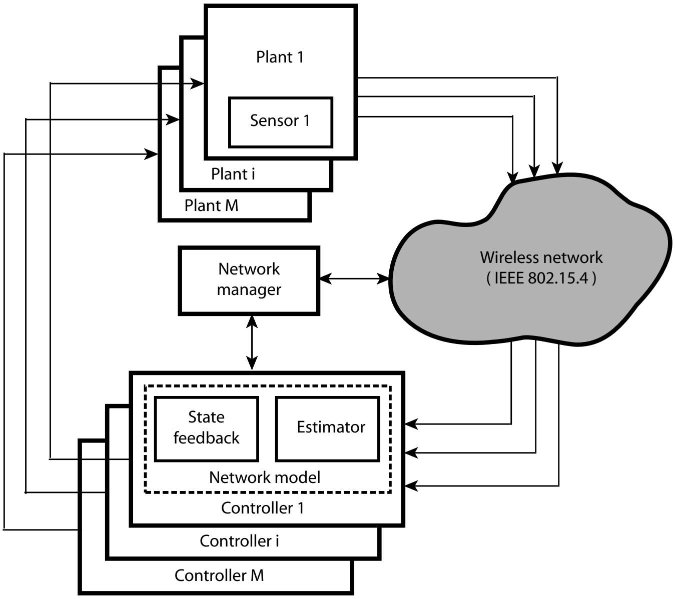


Figure 6.1: Overview of the networked control system setup. M plants need to be controlled by M controllers. The wireless network closes the loop from the sensor nodes to the controllers.

6.1 Problem Formulation

The problem considered is depicted in Figure 6.1, where multiple plants are controlled over a WSN using the IEEE 802.15.4 protocol. All M plants contend to transmit sensor measurements to the controller over a wireless network which induces packet losses and varying delays. We assume that a sensor node is attached to each plant. Many practical NCSs have several sensing channels and the controllers are collocated with the actuators, as in heat, ventilation and air-conditioning (HVAC) control systems (Arampatzis et al., 2005). A contention-based IEEE 802.15.4 MAC protocol, using a Carrier Sense Multiple Access with Collision Avoidance (CS-MA/CA) scheme, is used to determine which sensor node accesses the wireless channel. Throughout this chapter we consider control applications where nodes asynchronously generate packets when a timer expiries. When a node sends a packet successfully or discards a packet, it stays in an idle period for $\overline{h}$ seconds without generating packets. The data packet transmission is successful if an acknowledgement (ACK) packet is received. We assume that the controller commands are always received by the actuator reliably.

We consider a plant i given by a linear stochastic differential equation

$$dx(t) = Ax(t)dt + Bu(t)dt + dw(t) \tag{6.1}$$

where $x(t) \in \mathbb{R}^n$ is the plant state and $u(t) \in \mathbb{R}^m$ is the control signal. The process disturbance $w(t) \in \mathbb{R}^n$ has a mean value of zero and uncorrelated increments, with incremental covariance $R_w dt$. We neglect the plant index i to simplify notation. Let us consider the sampling of the plant with time-varying sampling period $h_k = t_{k+1} - t_k$ and delay τ_k. The sampling period is $h_k = \overline{h} + \tau_k$ where the idle period $\overline{h}$ is constant and the random delay is τ_k, which is bounded $\tau_k \leq \tau_{\max}$. We assume that the random sequences $\{\tau_k\}$ and $\{h_k\}$ are bounded, $0 < \tau_k < h_k$ and $0 < h_{\min} \leq h_k \leq h_{\max}$. In addition, they are stochastic, independent, and have known distributions. Notice that the networked induced delay τ_k is less than h_k and allows the packets to arrive at the controller in the correct order. By considering zero-order-hold, a time-varying discrete-time system is obtained

$$\begin{aligned} x_{k+1} &= \Phi_k x_k + \Gamma_0^k u_k + \Gamma_1^k u_{k-1} + w_k \\ y_k &= C x_k + v_k \end{aligned} \tag{6.2}$$

where $\Phi_k = e^{Ah_k}$, $\Gamma_0^k = \int_0^{h_k - \tau_k} e^{As} ds B$, $\Gamma_1^k = \int_{h_k - \tau_k}^{h_k} e^{As} ds B$, and v_k is a discrete-time white Gaussian noise with zero mean and variance R_v. The parameter k is the discrete time index. The initial state x_0 is white Gaussian with mean $\bar{x}_0$ and covariance P_0.

Packet loss is first modelled as a random process whose parameters are related to the behavior of the network. The measurement at the controller side is given by

$$\hat{y}_k = \begin{cases} Cx_k + v_k\,, & \gamma_k = 1\,, \\ \emptyset\,, & \gamma_k = 0\,, \end{cases} \tag{6.3}$$

where γ_k is a Bernoulli random variable with $\Pr(\gamma_k = 1) = 1 - p$ and p is the packet loss probability. The null measurement value at the controller $\hat{y}_k) = \emptyset$, represents that no measurement was received at time k.

By considering both the packet loss and delay induced by a wireless network, we introduce an augmented discrete-time state variable $z_k = \begin{pmatrix} x_k & u_{k-1} \end{pmatrix}^T$ in order to analyze the system. The augmented state-space is

$$\begin{aligned} z_{k+1} &= \Phi_d z_k + \Gamma_d u_k + w_k \\ \hat{y}_k &= \gamma_k y_k \end{aligned} \tag{6.4}$$

where $\Phi_d = \begin{pmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{pmatrix}$, $\Gamma_d = \begin{pmatrix} \Gamma_0 \\ I \end{pmatrix}$ and $C_d = \begin{pmatrix} C & 0 \end{pmatrix}$.

In Figure 6.1, a network manager block is introduced. The network manager requires an analytical model of the packet loss and delay, which is then used to design the estimator and controller that compensates for the packet loss and delay induced by the wireless network. The network manager formulates a constrained optimization problem where the objective function, denoted by E_{tot}, is the total energy consumption of the wireless network and the constraint is the requirement of control performance. Hence, the constrained optimization problem of the control system is

$$\min_{h,\mathbf{V}} \quad E_{tot}(h, \mathbf{V}, \Delta) \tag{6.5a}$$

$$\text{s.t.} \quad J(h, p(h, \mathbf{V}, \Delta), \tau(h, \mathbf{V}, \Delta)) \leq J_{req}\,. \tag{6.5b}$$

The decision variable h is the sampling period and $\mathbf{V}$ are the protocol parameters of the network. Δ includes the parameters of the network setup such as a network topology, length of packet, and number of nodes. J is the control cost, J_{req} is the required maximum control cost. We remark that the packet loss probability and delay of the network is also a function of the sampling period h, protocol parameters $\mathbf{V}$ and parameters of the network setup Δ. Thus, the sampling period h affects the performance of both wireless network and control system. In (6.5b), the decision variables are feasible if they satisfy a given control cost J_{req}.

6.2 Wireless Communication Model

In this section, we introduce the effective analytical model of packet loss probability and delay of the wireless network imposed by the IEEE 802.15.4 protocol which was originally derived in (Park et al., 2009). An overview of the CSMA/CA MAC was presented in Chapter 2, Section 2.3.4. Here we present an overview of the analytical model of the wireless network.

A precise and effective analytical model of the slotted CSMA/CA of the IEEE 802.15.4 standard was proposed in (Park et al., 2009). It is modelled through a Markov chain taking into account retry limits, ACKs, unsaturated traffic load, and the parameters of the network setup such as a length of packet and number of nodes. Figure 6.2 illustrates the Markov chain. Let $s(t), c(t)$ and $r(t)$ be the stochastic process representing the backoff stage, the state of the backoff counter and the state of retransmission counter at time t, respectively, experienced by a node to transmit a packet. By assuming that nodes start sensing independently, the stationary probability μ that the node attempts a first carrier sensing in a randomly chosen slot time is constant and independent of the other nodes. It follows that (s, c, r) results in a three dimensional Markov chain with the time unit *aUnitBackoffPeriod* (corresponding to 0.32 ms). The probability μ that a node attempts CCA$_1$ and the busy probabilities α (busy CCA$_1$) and β (busy CCA$_2$) are derived by solving the state transition probabilities associated with the Markov chain model. The expressions of μ, α, and β are computed by solving a system of non-linear equations. The precise model gives us the objective function, energy consumption (6.5a), and the packet loss probability and delay in a numerical form. Note that the proto-

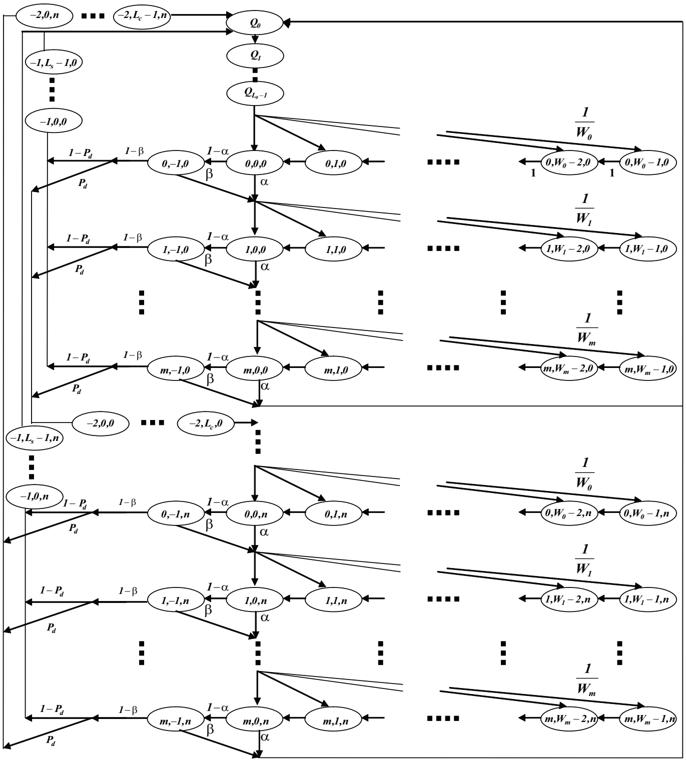


Figure 6.2: Markov chain model proposed in (Park et al., 2009) for the CSMA/CA algorithm of the IEEE 802.15.4 standard.

col parameters $\mathbf{V}$ of the decision variables are the MAC parameters (*macMinBE*, *macMaxCSMABackoffs*, *macMaxFrameRetries*).

6.3 Design of Estimator and Controller

In this section, we investigate how the packet loss probability and delay of the network affect the control performance. We discuss the design of an optimal feedback controller and present a control cost to analyze the NCSs described in Section 6.1. We first introduce our performance indicator as a control cost function, which is an explicit function of the sampling period h, packet loss probability p, and delay τ of the network. Then, we design the estimator and controller under packet losses and delays in Section 6.3.1 and 6.3.2, respectively. This is achieved by extending the results on optimal stochastic estimation and control under packet losses in (Schenato et al., 2007) with delays in (Nilsson, 1998).

Let us first define the information set under the packet loss and network induced delay as follows

$$\mathcal{I}_k = \{\mathbf{y}^k, \boldsymbol{\gamma}^k\} \tag{6.6}$$

where $\mathbf{y}^k = (y_k, y_{k-1}, \ldots, y_1)$ and $\boldsymbol{\gamma}^k = (\gamma_k, \gamma_{k-1}, \ldots, \gamma_1)$.

Consider the control cost function as follows

$$\begin{aligned} J_N(\boldsymbol{u}^{N-1}, \bar{z}_0, P_0) &= \mathbf{E}[z_N^T W_N z_N \\ &+ \sum_{k=0}^{N-1} (z_k^T W_k z_k + 2 z_k^T N_k u_k + u_k^T U_k u_k)], \end{aligned} \tag{6.7}$$

where $\bar{z}_0 = \begin{pmatrix} \bar{x}_0 & 0 \end{pmatrix}^T$.

6.3.1 Estimator Design

The estimator design is based on arguments similar to the standard Kalman filtering. Let us define the following variables

$$\begin{aligned} \hat{z}_{k|k} &= \begin{pmatrix} \mathbf{E}[x_k|\mathcal{I}_k] & u_{k-1} \end{pmatrix}^T \\ e_{k|k} &= z_k - \hat{z}_{k|k} \\ P_{k|k} &= \mathbf{E}[e_{k|k} e_{k|k}^T | \mathcal{I}_k]. \end{aligned}$$

The innovation step is given by

$$\begin{aligned} \hat{z}_{k+1|k} &= \Phi_d \mathbf{E}[z_k|\mathcal{I}_k] + \Gamma_d u_k = \Phi_d \hat{z}_{k|k} + \Gamma_d u_k && (6.8) \\ e_{k+1|k} &= z_{k+1} - \hat{z}_{k+1|k} = \Phi_d e_{k|k} + w_k \\ P_{k+1|k} &= \mathbf{E}[e_{k+1|k} e_{k+1|k}^T | \mathcal{I}_k] = \Phi_d P_{k|k} \Phi_d^T + R_w && (6.9) \end{aligned}$$

where w_k and $\mathcal{I}_k$ are independent and u_k is a deterministic function of $\mathcal{I}_k$. The correction step is given by

$$\begin{aligned} \hat{z}_{k+1|k+1} &= \hat{z}_{k+1|k} + \gamma_{k+1} K_{k+1} (y_{k+1} - C_d \hat{z}_{k+1|k}) && (6.10) \\ e_{k+1|k+1} &= z_{k+1} - \hat{z}_{k+1|k} = \Phi_d e_{k|k} + w_k \\ K_{k+1} &= P_{k+1|k} C_d^T (C_d P_{k+1|k} C_d^T + R_v)^{-1} \\ P_{k+1|k+1} &= P_{k+1|k} - \gamma_{k+1} K_{k+1} C_d P_{k+1|k} && (6.11) \end{aligned}$$

where we apply the standard derivation for the Kalman filter.

6.3.2 Controller Design

We introduce the feedback control law and present the finite and infinite horizon control cost functions. The cost function given by (6.7) can be expressed as

$$\begin{aligned} J_N^* =& V_0(x_0) = \bar{z}_0^T S_0 \bar{z}_0 + \mathrm{Tr}(S_0 P_0) + \sum_{k=0}^{N-1} (\mathrm{Tr}((\Phi_d^T S_{k+1} \\ & \times \Phi_d + W_k - S_k)\mathbf{E}_\gamma[P_{k|k}]) + \mathrm{Tr}(S_{k+1} R_w)) \end{aligned} \tag{6.12}$$

where Tr denotes the trace of a square matrix. $\mathbf{E}_\gamma[\cdot]$ is the expectation with respect to the arrival sequence $\{\gamma_k\}$. The control input that minimizes the cost function of (6.7) is

$$u_k = -(\Gamma_d^T S_{k+1} \Gamma_d + U_k)^{-1} \Gamma_d^T S_{k+1} \Phi_d \hat{z}_{k|k} = -L_k \hat{z}_{k|k}. \tag{6.13}$$

It is not possible to compute the exact value of these matrices $\mathbf{E}_\gamma[P_{k|k}]$ as shown in (Sinopoli et al., 2004), but bounds on their values can be achieved.

The expected value $\mathbf{E}_\gamma[P_{k|k}]$ is bounded by

$$\tilde{P}_{k|k} \leq \mathbf{E}_\gamma[P_{k|k}] \leq \hat{P}_{k|k}, \qquad \forall k \geq 0$$

where the matrices $\tilde{P}_{k|k}$ and $\hat{P}_{k|k}$ can be found in (Schenato et al., 2007). Then, it is possible to derive the bound of control cost given in (6.12). In the next section, we use two deterministic sequences $J_N^{\min}$ and $J_N^{\max}$, which bound the expected minimum cost as follows

$$\frac{1}{N} J_N^{\min} \leq \frac{1}{N} J_N^* \leq \frac{1}{N} J_N^{\max}, \tag{6.14}$$

and the two sequences converge to the following values:

$$\begin{aligned} J_\infty^{\max} =& \mathrm{Tr}((\Phi_d^T S_\infty \Phi_d + W_k - S_\infty)(\overline{P}_\infty - (1-p)\overline{P}_\infty C_d^T \\ & \times (C_d \overline{P}_\infty C_d^T + R_v)^{-1} C_d \overline{P}_\infty)) + \mathrm{Tr}(S_\infty R_w) \end{aligned} \tag{6.15}$$

$$J_\infty^{\min} = p\mathrm{Tr}((\Phi_d^T S_\infty \Phi_d + W_k - S_\infty)\underline{P}_\infty) + \mathrm{Tr}(S_\infty R_w) \tag{6.16}$$

where,

$$\begin{aligned} \overline{P}_\infty =& \Phi_d \overline{P}_\infty \Phi_d^T + R_w - (1-p)\Phi_d \overline{P}_\infty C_d^T \\ & \times (C_d \overline{P}_\infty C_d^T + R_v)^{-1} C_d \overline{P}_\infty \Phi_d^T \\ \underline{P}_\infty =& p\Phi_d \underline{P}_\infty \Phi_d^T + R_w \,. \end{aligned}$$

We remark that Eqs. (6.15) and (6.16) are explicit functions of the sampling period h, packet loss probability p, and delay τ. The finite horizon cost and the cost bounds of the infinite horizon case will be used as the performance indicators in Section 6.4.1.

It is also possible to derive the stability bounds of the packet loss probability as:

$$\frac{1}{\prod_i |\lambda_i^u(\Phi_d)|^2} \leq p_c \leq \frac{1}{\max_i |\lambda_i^u(\Phi_d)|^2} \tag{6.17}$$

where $\lambda_i^u(\Phi_d)$ are the unstable eigenvalues of A, see (Schenato et al., 2007).

6.4 Co-Design Framework

In this section, we first show the achievable control performance by taking into account realistic simulation results. Then, we propose a co-design approach of the wireless NCSs.

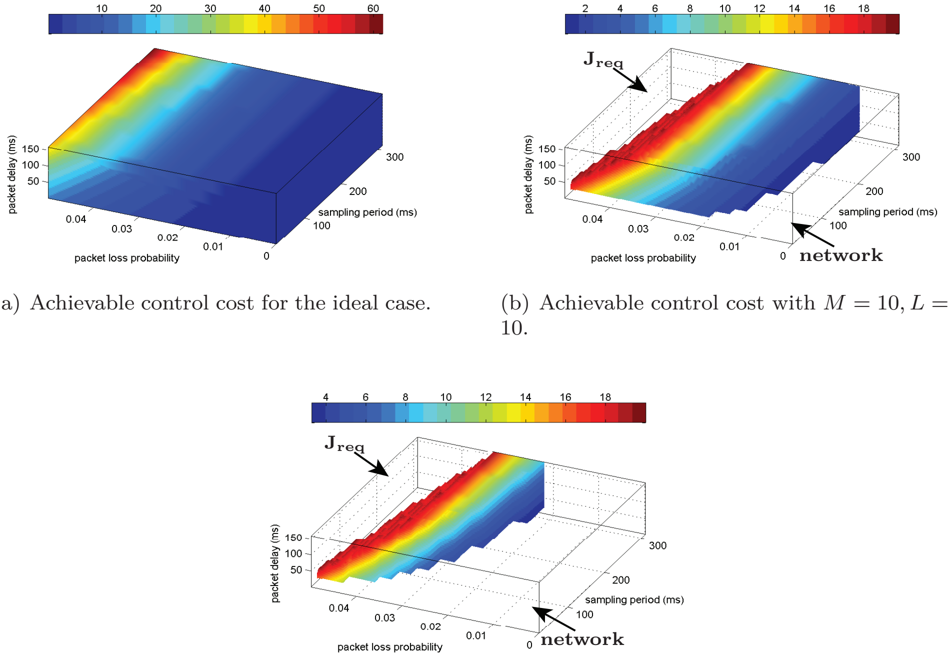
6.4.1 Effects of Wireless Network

In this section, we discuss the fundamental issues of joint communication design and control design for wireless NCSs. The control cost (6.15) is considered as a performance indicator of the control system as described in Section 6.3.2. As an example we consider an unstable second-order plant in the form of (6.1) with

$$A = \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix}, \; B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; P_0 = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; U = 0.01, \; N = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \; R_w = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; R_v = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}.$$

Figure 6.3 shows the achievable control cost over different sampling periods, packet loss probabilities, packet delays with the ideal case and the realistic wireless networks for the different number of nodes $M = 10, 20$. Note that the different colors show the achievable control cost. Figure 6.3(a) depict the ideal case where longer sampling periods increase the control cost. Furthermore, we observe that packet losses at a higher sampling period are more critical than packet losses at a lower sampling period, indicating that we are sampling in a conservative way. In a similar fashion, we derive the effects of packet delay on the control cost. Figures 6.3(b) and 6.3(c) depict the achievable region for $M = 10$ and 20 nodes, respectively. Note that we set the required cost $J_{req} = 20$. A point is achievable if it satisfies a given required cost, packet loss probability and delay for each sampling period. The achievable region is the set of all achievable points. It is natural that as the control requirement becomes strict, the infeasibility region increases, which is greatly affected by the wireless network performance. Observe in Figures 6.3(b) and 6.3(c) that the packet loss probability $p \leq 0.09$ is not achievable when the sampling period



(a) Achievable control cost for the ideal case.

(b) Achievable control cost with $M = 10, L = 10$.

(c) Achievable control cost with $M = 20, L = 10$.

Figure 6.3: Achievable control cost over different sampling periods, packet loss probabilities, and packet delays. The different gray color shows the different control cost. Note that the scales of color bar are different in the figures.

is very short $h \leq 0.03$ s. Since very short sampling periods increase the traffic load of the network, the packet loss probability is closer to the critical packet loss probability, above which the system is unstable. Hence, it is difficult to achieve a very low packet loss probability when the sampling period is very short. Furthermore, by comparing Figures 6.3(b) and 6.3(c), we see that the infeasibility region increases as the number of nodes increases. The infeasibility region due to the wireless network starts from the origin point where the sampling period $h = 0$, no packet loss $p = 0$, and no packet delay $\tau = 0$. No matter what communication protocol is used, the infeasibility region is started from the origin point, where the area and shape of the infeasibility region would vary according to different communication protocols.

Figure 6.4 shows the control cost and throughput of the wireless network over different sampling periods. The throughput is the average rate of successful data transmission over a communication channel, which is the common objective for a communication designer. In the figure, "J^i_∞" and "J^r_∞" refers to the cost bound

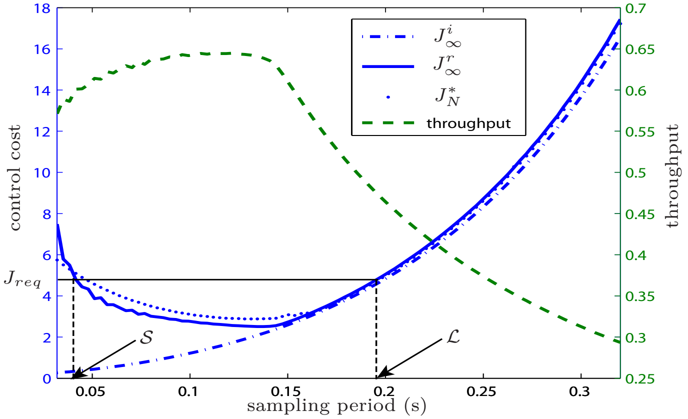


Figure 6.4: Control cost and throughput of the wireless network over different sampling periods. "J^i_∞" and "J^r_∞" refer to the cost bound $J^{\max}_\infty$ of the infinite horizon control cost given in (6.15) with ideal case and realistic model in (Park et al., 2009), respectively. "J^*_N" denotes the finite horizon control cost given in (6.12).

$J^{\max}_\infty$ given by (6.15) for the ideal and realistic model in (Park et al., 2009), respectively. Recall that J^*_N is the finite horizon control cost given by (6.12). The cost J^*_N follows the infinite horizon cost J^r_∞ based on the realistic model.

Due to the absence of delays, the control performance when using an ideal network increases monotonically as the sampling period increases. However, when using a real network, a shorter sampling period does not minimize the control cost of the control systems, because of the higher packet loss probability when the traffic load is high. In addition, the two curves of the cost J^i_∞ and J^r_∞ coincide for longer sampling periods, meaning that when the sampling period is larger, the sampling period is the dominant factor in the control cost compared to the packet loss probability and delay. Now, let us discuss the throughput of the communication network and control cost of control systems. When we flip the throughput curve on the Y-axis, we observe a similar trend of behavior with the curve of control cost. Note that the closer the throughput is to 1, the better the utility of the wireless network. As the sampling period $h \in [0, 0.13]$ s increases, the control cost decreases and the throughput increases due to mainly high packet loss. For a longer sampling period $h > 0.15$ s, the performance of both the control and the communication system degrades as the sampling period increases. The throughput decreases since the network is underutilized. We remark that the objective of both communication design and control design has a very similar trend. The dynamic interactions between these two designs, result in very interesting situations for the performance of wireless NCSs which are missing in the current literature. Let us consider a desired maximum control cost J_{req} greater than the minimum value of the control cost. Then, we have two feasible sampling periods $\mathcal{S}$ and $\mathcal{L}$ in Figure 6.4. However, the

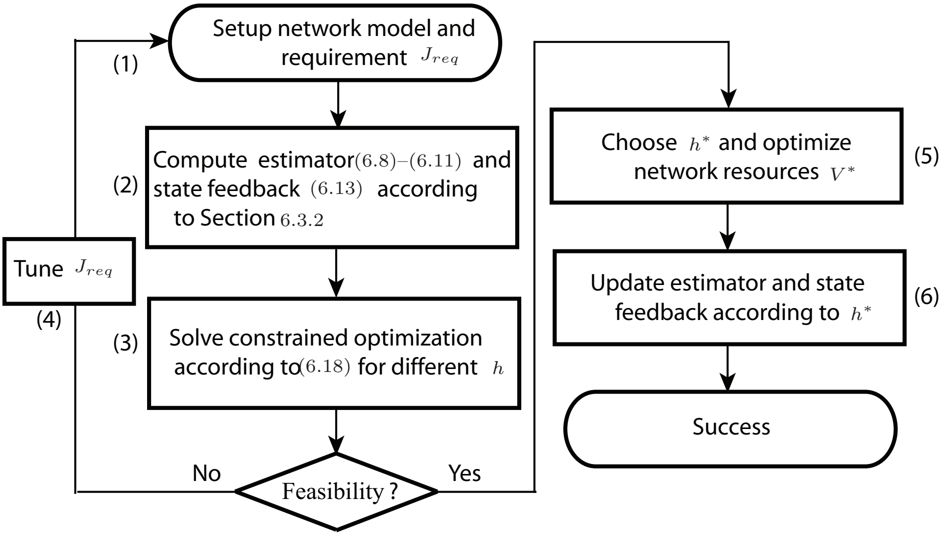

Setup network model and requirement J_{req}
(1)
Compute estimator (6.8)–(6.11) and state feedback (6.13) according to Section 6.3.2
(2)
Solve constrained optimization according to (6.18) for different h
(3)
Tune J_{req}
(4)
Feasibility ?
No
Yes
Choose h^* and optimize network resources V^*
(5)
Update estimator and state feedback according to h^*
(6)
Success

Figure 6.5: Flow diagram of co-design framework.

performance of the wireless network is still heavily affected by the choice of the sampling period of $\mathcal{S}$ and $\mathcal{L}$, as we discussed earlier. We see that by choosing $\mathcal{L}$, the NCS will be more robust from a control and communication perspective since for small perturbations in the network conditions in $\mathcal{S}$, the overall NCS might become unstable, which does not happen in the $\mathcal{L}$ case. Likewise, a longer sampling period $\mathcal{L}$ leads to lower network energy consumption than the shorter sampling period $\mathcal{S}$ (see details in (Park et al., 2009; Pollin et al., 2008)). Recall that the energy efficiency is one of the most critical issues for sensor nodes due to their limited battery power. This motivates our co-design approach of networked control systems running over WSNs.

6.4.2 Design Procedure

We remind that the problem we consider in this chapter is how to determine the optimal sampling period h^* of control systems and the protocol parameters $\mathbf{V}^*$ of the wireless network of an optimization problem given by (6.5). Figure 6.5 shows the proposed design flow that each control loop of the network follows. The application designer provides the parameters of network setup Δ and the desired maximum cost of control systems J_{req}. Δ includes the important factors for modelling the wireless network such as a network topology, length of the packets, and the number of nodes (step 1). It is also possible that each control loop has a different desired maximum cost J_{req}. The control designer then computes, off-line, an estimator (6.8)–(6.11) and a state feedback (6.13) according to Section 6.3.2 for different sampling periods, packet loss probabilities, and delays (step 2). Note that the control design process does not require any explicit considerations of the communication protocols or the network setup. The network manager formulates and solves a constrained optimization problem, whereby the objective function is the energy consumption of the network and the constraints are the packet loss probability and delay, which are derived by J_{req} for different sampling periods (step 3). More precisely, the constrained optimization problem in (6.5) is rewritten as follows

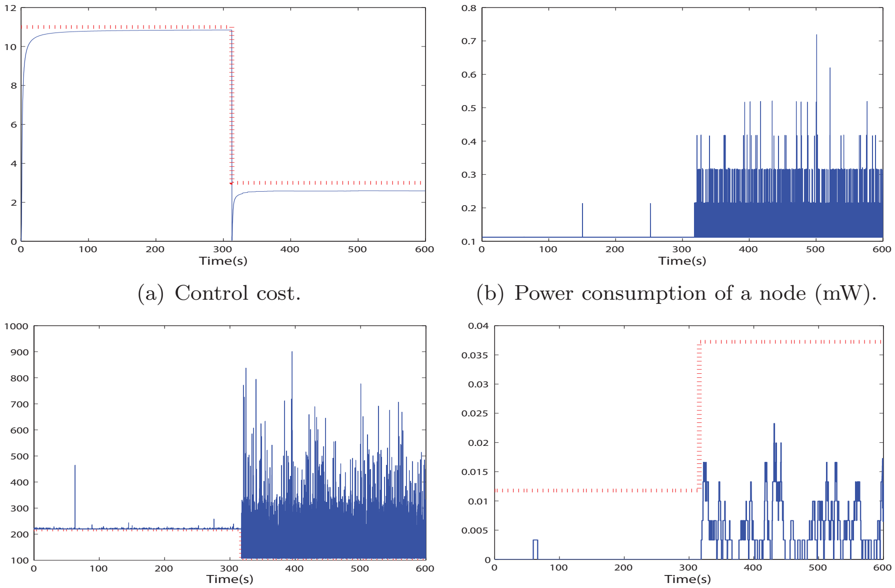
$$\begin{aligned} \min_{\mathbf{V}} \quad & E_{tot}(h, \mathbf{V}, \Delta) && (6.18a) \\ \text{s.t.} \quad & p(h, \mathbf{V}, \Delta) \leq p_{req}\,, && (6.18b) \\ & \tau(h, \mathbf{V}, \Delta) \leq \tau_{req}\,. && (6.18c) \end{aligned}$$

The decision variables are the communication protocol parameters $\mathbf{V}$ depending on the network designer. Recall that the protocol parameters $\mathbf{V}$ are the MAC parameters (*macMinBE*, *macMaxCSMABackoffs*, *macMaxFrameRetries*) of the IEEE 802.15.4 protocol. It is a challenging task to find the global optimal solution of the problem, since the derivation of exact analytical expressions is not possible due to the uncertainty of the wireless channel. One can find a sub-optimal solution using the steps described in (Park, 2011). The network manager finds the local optimal MAC parameters $\mathbf{V}^*(h, p_{req}, \tau_{req})$ of a sub-optimization problem for a given h, p_{req}, τ_{req}. Then, the optimal solution $h^*, \mathbf{V}^*$ is given by the pair h, $\mu_c^*(h)$ that minimizes the cost function if there are feasible solutions (step 5). Otherwise, the control designer needs to tune J_{req} since the desired control performance is not realistic (step 4). The network manager adapts the optimal sampling period h^* and the optimal protocol parameters V^* of the network (step 5). The control designer updates the estimator and the state feedback according to the optimized $h^*, p(h^*, \mathbf{V}^*, \Delta), \tau(h^*, \mathbf{V}^*, \Delta)$ (step 6).

6.4.3 Simplified Computation for the Co-Design Approach

The computational complexity of the framework presented in Figure 6.5, cannot be neglected. The optimization of network energy in (6.3.2) has to perform a exhaustive search over all possible sampling period values h, to find the lowest energy consumption value for a given network configuration, which fulfils the control cost requirements J_{req}, through p_{req} and d_{req}. Therefore, relaxation of the co-design computations is required if one is interested in applying this framework in an online fashion.

In the same way as the algorithm proposed in Section 6.4.2, the network manager follows steps 1 and 2. We assume that the control cost J, which is a function of the sampling periods, packet loss probabilities, and delays for the specific system, is computed offline and stored in the network manager. Recall that for a given J_{req}, a feasibility interval with respect to the sampling period is achieved, $[h_{min}, h_{max}]$, as seen in Figure 6.4. Now, instead of solving step 3, for all possible sampling periods, we allow the network manager to select a single value of h, which we



(a) Control cost.

(b) Power consumption of a node (mW).

(c) Interval of packet generated time (ms).

(d) Packet loss probability.

Figure 6.6: Optimized control cost, power consumption of the network, interval of packet generated time, and packet loss probability of the proposed co-design approach with $M = 20$ nodes and $L = 10$ when the control requirement changes from $J_{req} = 11$ to $J_{req} = 3$ at 315 s. The particular realization is shown out of $M = 20$ nodes. The dotted line shows the requirement change of each figures.

denote by $\tilde{h} \in [h_{min}, h_{max}]$, within the feasible range of the sampling period, for which the control constraint $J < J_{req}$ is achieved. This value can be chosen as the average value between $[h_{min}, h_{max}]$, or a value close to h_{max}. As noted in Section 6.4.1 and Figure 6.4, higher sampling periods mean higher robustness from a communication perspective. However, by sampling less often, the achieved control performance is also lower. Therefore, a tradeoff must be made when selecting the sampling period $\tilde{h}$, that take the current load of the network into account. Even with a single value of the sampling period, a given range of delay and packet loss probabilities fulfills the constraints $J < J_{req}$. The achievable region of delay and packet loss probability, for which the optimization algorithm performs the network parameter optimization, may also be discretized in larger steps, such that the total amount of parameter combination is reduced. In this way, the network manager will perform much fewer computations during step 3. The optimization (6.5) is then solved for each combination of packet loss probability and delay, below the required p_{req} and d_{req}, for packet loss and delay, respectively. This optimization will then find the best network parameters that minimize the energy consumption of the network. This sampling period should then be used, together with the optimal network configuration for performing control. Before initiating the control actions, the state estimator and state feedback controller should be updated according to the sampling period $\tilde{h}$ and the delay and packet loss probability achieved through the network model introduced in Section 6.2.

In the practical application of the algorithm, real measurements could indicate the current packet loss probability and delay experienced by the wireless sensor nodes. We believe that our co-design approach could be used as a starting point for selecting the sampling period, designing the state estimator and controller, as well as selecting the network parameters, according to the required control performance. In this case, real network data could be integrated with the co-design framework such that the state estimator and the state feedback control would be updated with the real delay and packet loss probability. Likewise, the sampling period could be adjusted if too high packet loss probability is observed which was not accounted for in the proposed model. This could be due to the fact that interference would be present in the surrounding environment. Moreover, the network parameters could be tuned online according to the observed network performance. If a very high network congestion is observed, the control cost requirement may have to be increased in order to allow the communication network to reach an equilibrium. As proposed in (Park et al., 2010), the MAC parameters can also adapt to changes in the traffic load, while using the same modeling framework proposed in this chapter.

6.5 Illustrative Example

In this section, we illustrate the proposed co-design procedure described in Section 6.4.2 through numerical examples. Figure 6.6 shows the adaptation of the requirements in terms of the sampling period, and packet loss probability of the network when the control requirement changes from $J_{req} = 11$ to $J_{req} = 3$ at 315 s. The optimal h^*, p_{req}, and τ_{req} are 214.4 ms, 0.012, 74.9 ms before control requirement changes, respectively. We remark that the adaptive protocol tunes the MAC parameters to meet the requirements for packet loss probability and packet delay for a given sampling period. Figures 6.6(c), and 6.6(d) show that the adaptive communication protocol satisfies the requirements of h and p_{req}, respectively. Although not presented, the packet delay values met the requirement. The high jitter of Figure 6.6(c) is mainly due to the packet loss of Figure 6.6(d). After the control requirement changes at time 315 s, the parameters h, p_{req}, τ_{req} adapt to 102.4 ms, 0.037, 97.4 ms, respectively. Note that although the requirements of packet loss probability and packet delay are less strict after the requirement changes, the sampling period decreases to meet the requirement $J_{req} = 3$. Recall that as the sampling period decreases, the packet loss probability and packet delay increase. We observe that the control cost is satisfied and the convergence of the algorithm is very fast. By comparing Figures 6.6(a) and 6.6(b), the tradeoff between the control requirement and power consumption of the network is clearly observed.

6.6 Discussion

The dynamic interactions between control and communications result in very interesting situations for the performance of wireless NCSs which were not yet well understood. The present work addresses the problem of joint design of control and communication for multiple control systems over the IEEE 802.15.4 wireless network. We first presented how the wireless network affects the performance of NCSs by showing the achievable region of the control performance. Moreover, a similar trend of the network throughput and the control cost was devised. Based on these results we conclude that the choice of the sampling period greatly influences the control performance, network throughput, overall energy consumption but also the robustness of the NCS. A co-design method was proposed for wireless NCSs by considering the critical aspects of both control and communication systems. In particular, a constrained optimization problem is formulated, where the objective function is the energy consumption of the network and the constraints are the packet loss probability and delay, which are derived from the desired control performance. Numerical results illustrated the efficiency of the proposed co-design approach. We also proposed computationally efficient methods to perform the co-design approach, which can be used an online fashion, and practically applied. A comparison between these two approaches is in the scope of our future work.

Chapter 7

Conclusions

In this thesis, we focused on the problem of controlling wireless NCSs with ensured closed-loop performance guarantees and efficient resource usage. The resources we considered was the energy consumption of the wireless nodes and the wireless network bandwidth. Our contributions addressed this problem, where we provided solutions that have a potential to be applied in real wireless NCSs.

In Chapters 3 and 4, we designed and experimentally evaluated an architecture that addressed the aforementioned problem. This architecture was based on three different sensor communication mechanisms which we denoted by event-based, predictive and hybrid. For each mechanism, we devised suitable scheduling and MAC parameter design such that real-time guarantees on packet delivery were provided to control-loop nodes. These mechanisms were experimentally evaluated with respect to their achievable control performance, energy efficiency as well as how efficient the network bandwidth is utilized. The results showed that all the proposed architectures achieved close-loop performances similar to the ones provided by a typical periodic implementation of the controller, 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 implementation had a clear advantage if the wireless network is shared among other nodes. However, this mechanism had the drawback of being less robust to disturbances affecting the closed-loop system. In order to solve this issue, we proposed a hybrid sensor communication mechanism which joined the benefits of the other two devised mechanisms. By utilizing a hybrid MAC and dynamic scheduling, this mechanism exhibited high control performances, robustness to disturbances and an efficient network bandwidth utilization among network nodes. The less positive side of the hybrid sensor communication mechanism is that a relatively lower battery lifetime of sensor nodes was achieved, when compared to the other two proposals.

The implementation of event-triggered PI controllers for industrial process control was proposed in Chapter 5. Motivated by the requirement of improving energy efficiency of wireless NCSs in the process industry, we presented suitable event-triggered PI controller implementations which achieved closed-loop control performances similar to a periodically sampled PI controller. A novel aperiodic sampling technique, where event generation is based on the control input value, was proposed and evaluated. We demonstrated by simulations that this technique is very promising, providing suitable closed-loop control performances with no undesirable effects, while generating a very low number of sensor transmissions.

Chapter 6 presented a framework for the joint design of control and communications in CSMA/CA wireless NCSs. We investigated the wireless network effects on the control performance and showed that the choice of the sampling period greatly influences the control performance, network throughput, overall energy consumption but also the robustness of the NCS. A co-design approach was proposed to achieve required control performances while minimizing the energy consumption of the wireless network nodes. Numerical results illustrated the efficiency of the proposed co-design approach and showed that a well-defined design procedure is required in order to achieve high performances in wireless NCSs.

7.1 Future Work

The current work can be extended in several directions. Some suggestions are given below.

The work presented in this thesis indicates that great benefits can be achieved by performing sampling and control aperiodically. However, there are still many open issues related to these mechanisms. When dealing with real applications, distributed solutions have a clear advantage since each agent can perform decisions based on local information, and no centralized unit is required. As shown in Chapters 3 and 4, synchronization messages greatly decrease energy efficiency of the wireless network. Therefore, we aim at investigating distributed solutions to the mechanisms proposed in Chapter 3. In the same line of work, we intend to study how such mechanisms would perform when a contention-based MAC is used and when loss of critical messages occur between control loop nodes.

The event-triggered PI controller implementations proposed in Chapter 5 seems suitable for industrial process control. The controllers were devised to guarantee closed-loop performances close to a periodic implementation of a PI controller, both during transient and steady-state, while reducing the number of control updates. We are currently aiming at experimentally validating such event-triggered controllers in a real industrial plant. From a real experiment, we expect to gain insight on other practical issues that were not evaluated in this thesis. Moreover, further influences of the network are also in the scope of our future work.

Chapter A

Appendix

A.1 PI tuning

We now introduce the tuning methods we used for the PI controller.

In order to evaluate the performance of a PID controller, several performance metrics have been proposed, see (Åström and Hägglund, 2006) for a detailed overview. In our study we will use the Integrated Absolute Error (IAE) as the performance metric to characterize how well the PID controller is performing, under all conditions. The calculation of the IAE is as follows,

$$IAE = \int_0^\infty |e(t)|. \tag{A.1}$$

From the PI control literature, there exist many tuning methods ranging from pure heuristics, pole placement, optimization based methods, etc. See see (Åström and Hägglund, 2006) for a detailed overview. A simple and efficient tuning method called AMIGO (Approximate M-Constrained Integral Gain Optimization) was proposed in (Åström and Hägglund, 2006). This method is derived by performing step response analysis over a large range of heterogeneous processes, which are denoted as "the test batch" , in the same spirit of the classic Ziegler Nichols methods. The PI controller parameters are achieved according to the parameters of a first-order plus time-delay (FOTD) approximation of the real plant. A FOTD model is also typically known as a KLT model in the control literature and is defined as,

$$G(s) = \frac{K}{1+sT}e^{-sL}, \tag{A.2}$$

with static gain K, time-constant T and a time-delay L. The controller parameters given by the AMIGO tuning rules are (Åström and Hägglund, 2006),

$$\begin{aligned} K_p &= \frac{0.15}{K} + \Big(0.35 - \frac{LT}{(L+T)^2}\Big)\frac{T}{KL} \\ T_i &= 0.35L + \frac{13LT^2}{T^2+12LT+7L^2} \end{aligned} \tag{A.3}$$

Bibliography

J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *9th IEEE International Conference on Industrial Informatics (INDIN)*, July 2011. URL `http://www.mrtc.mdh.se/index.php?choice=publications&id=2455`.

I.F. Akyildiz and I.H. Kasimoglu. Wireless sensor and actuator networks: research challenges. *Ad hoc networks*, 2(4):351–367, 2004.

J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Transactions on Wireless Communications*, 11(6):6–28, 2004.

F. Altaf, J. Araujo, A. Hernandez, H. Sandberg, and K. H. Johansson. Wireless event-triggered controller for a 3d tower crane lab process. In *19th Mediterranean Conference on Control and Automation*, june 2011.

A. Anta and P. Tabuada. Self-triggered stabilization of homogeneous control systems. *Proceedings of the American Control Conference*, 2008.

A. Anta and P. Tabuada. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. *30th IEEE Real-Time Systems Symposium*, pages 3–12, 2009a.

A. Anta and P. Tabuada. Isochronous manifolds in self-triggered control. *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009b.

A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030 –2042, 2010a.

A. Anta and P. Tabuada. On the minimum attention and anytime attention problems for nonlinear systems. *49th IEEE Conference on Decision and Control*, 2010b.

A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010c.

P. Antsaklis and J. Baillieul. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1), 2007.

P.J. 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, jul 2000. ISSN 0018-9219. doi: 10.1109/JPROC.2000.871299.

D. Antunes, J. P. Hespanha, and C. Silvestre. Stochastic hybrid systems with renewal transitions. In *Proc. of the 2010 Amer. Contr. Conf.*, June 2010.

D. Antunes, J. P. Hespanha, and C. Silvestre. Stochastic networked control systems with dynamic protocols. In *Proc. of the 2011 Amer. Contr. Conf.*, June 2011.

D. Antunes, J. P. Hespanha, and C. Silvestre. Volterra integral approach to impulsive renewal systems: Application to networked control. *IEEE Trans. on Automat. Contr.*, 2012. To appear.

T. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *IEEE International Symposium on Intelligent Control, MCCA*, pages 719–724, 2005.

K.E. Årzén. A simple event-based PID controller. *Preprints 14th World Congress of IFAC. Beijing, China*, 1999.

K.J. Åström and B.M. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *14th IFAC World Congress*, 1999.

K.J. Åström and B.M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. *Proceedings of the 41st IEEE Conference on Decision and Control*, 2, 2002.

K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA, NC, 2006.

K.J. Åström and T. Hägglund. The future of pid control. *Control Engineering Practice*, 9, No. 11:1163–1175, 2011.

K.J. Åström and M. Lundh. Lund control program combines theory with hands-on experience. *Control Systems, IEEE*, 12(3):22 –30, jun 1992. ISSN 1066-033X.

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.

J. Baillieul and P.J. Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9 –28, jan. 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.887290.

R. Blind and F. Allgöwer. Analysis of networked event-based control with a shared communication medium: Part I - Pure ALOHA. *Submitted to IFAC World Congress*, 2011a. URL `http://www.ist.uni-stuttgart.de/~blind/blind_IFAC11_I.pdf`.

R. Blind and F. Allgöwer. Analysis of networked event-based control with a shared communication medium: Part II - Slotted ALOHA. *Submitted to IFAC World Congress*, 2011b. URL `http://www.ist.uni-stuttgart.de/~blind/blind_IFAC11_II.pdf`.

E. Bonnema. *Large Hospital 50% Energy Savings Technical Support Document.* NREL/TP-550-47867., Golden, CO: National Renewable Energy Laboratory. xiv, 170 pp., 2010.

R.W. Brockett. Minimum attention control. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 3, 1997.

G.C. Buttazzo. *Hard real-time computing systems: predictable scheduling algorithms and applications.* Springer-Verlag New York Inc, 2005.

C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Springer, second edition, 2008.

A. Cervin and K.J. Åström. On limit cycles in event-based control systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 3190 –3195, dec. 2007. doi: 10.1109/CDC.2007.4434325.

A. Cervin, M. Velasco, P. Marti, and A. Camacho. Optimal on-line sampling period assignment: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 2010.

R. Cogill, S. Lall, and J.P. Hespanha. A constant factor approximation algorithm for event-based sampling. In *American Control Conference, 2007. ACC '07*, pages 305 –311, july 2007. doi: 10.1109/ACC.2007.4282991.

IEEE CSS. *The Impact of Control Technology.* IEEE Control Systems Society, 2011. URL `www.ieeecss.org`.

B. Demirel, Z. Zou, P. Soldati, and M. Johansson. Modular co-design of controllers and transmission schedules in wirelesshart. In *50th IEEE Conference on Decision and Control/European Control Conference*, 2011.

M. Deru, E. Bonnema, I. Doebber, A. Hirsch, M. McIntyre, and J. Scheib. Thinking like a whole building: A whole foods market new construction case study. 2011.

D.V. Dimarogonas and E. Frazzoli. Distributed event-triggered control strategies for multi-agent systems. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 906 –910, 30 2009-oct. 2 2009. doi: 10.1109/ALLERTON.2009.5394897.

D.V. Dimarogonas and K.H. Johansson. Event-triggered control for multi-agent systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 7131 –7136, dec. 2009. doi: 10.1109/CDC.2009.5399776.

M.C.F. Donkers, W.P.M.H. Heemels, D. Bernardini, A. Bemporad, and V. Shneer. Stability analysis of stochastic networked control systems. In *American Control Conference (ACC), 2010*, pages 3684 –3689, 30 2010-july 2 2010.

M.C.F. Donkers, P. Tabuada, and W.P.M.H. Heemels. On the minimum attention and the anytime attention control problems for linear systems: A linear programming approach. In *50th IEEE Conference on Decision and Control/European Control Conference 2011 (invited paper)*, 2011.

S. Durand and N. Marchand. Further Results on Event-Based PID Controller. In *Proceedings of the European Control Conference 2009*, pages 1979–1984, Budapest, Hongrie, August 2009.

Dust Networks, September 2011. URL `http://www.dustnetworks.com/`.

V.L. Erickson, M.A. Carreira-Perpinan, and A.E. Cerpa. Observe: Occupancy-based system for efficient reduction of hvac energy. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 258 –269, april 2011.

L. Eriksson. *PID Controller Design and Tuning in Networked Control Systems.* PhD thesis, Helsinki University of Technology, 2008.

D. Fontanelli, L. Greco, and A. Bicchi. Anytime control algorithms for embedded real-time systems. *Hybrid Systems: Computation and Control*, 4981:158–171, 2008.

E. Garcia and P. Antsaklis. Adaptive stabilization of model-based networked control systems. In *American Control Conference 2011*, june 2011a.

E. Garcia and P. Antsaklis. Model-based event-triggered control with time-varying network delays. In *to appear IEEE CDC-ECC 2011*, 2011b.

T. Glad and L. Ljung. *Control Theory: Multivariable and Nonlinear Methods.* Taylor & Francis, 2000.

V. Gupta. On an Anytime Algorithm for Control. *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009.

HART Communication Foundation. *WirelessHART Data Sheet*, 2007. Datasheet.

J. Hauer. TKN15.4: An IEEE 802.15.4 MAC Implementation for TinyOS 2. *TKN Technical Report Series, Telecommunication Networks Group, TU-Berlin*, (TKN-08-003), Mar 2009. URL `http://www.tkn.tu-berlin.de/publications/papers/TKN154.pdf`.

W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch. Analysis of event-driven controllers for linear systems. *Int. J. of Control*, pages 81(4), 571–590, 2008.

W.P.M.H. Heemels, A. R. Teel, N. van de Wouw, and D. Nesic. Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Transactions on Automatic Control*, 2010.

W.P.M.H. Heemels, M.C.F. Donkers, and A.R. Teel. Periodic event-triggered control based on state feedback. In *To appear in IEEE Conference on Decision and Control 2011*, 2011.

T. Henningsson, E. Johannesson, and A. Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, November 2008.

D. Henriksson and A. Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *IEEE CDC*, 2005.

A. Hernandez. Wireless process control using IEEE 802.15.4 protocol. Master's thesis, Royal Institute of Technology (KTH), November 2010.

A. Hernandez. Modification of the IEEE 802.15.4 implementation extended gts implementation. Technical report, KTH Royal Institute of Technology, July 2011. URL `http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-2.x-contrib/kth/index.html`.

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.

S. Hirche, M. Buss, P. Hinterseer, and E. Steinbach. Towards deadband control in networked teleoperation systems. In *IN: PROCEEDINGS OF THE 16.TH IFAC WORLD*, 2005.

IEEE. *IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE, 2006. URL `http://www.ieee802.org/15/pub/TG4.html`.

IEEE. *IEEE 802.15 task group 4e: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE, 2011. 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.

O. Kaltiokallio, L. M. Eriksson, and M. Bocca. On the performance of the pidplus controller in wireless control systems. In *Control Automation (MED), 2010 18th Mediterranean Conference on*, pages 707 –714, june 2010. doi: 10.1109/MED.2010.5547788.

Younghun Kim, Thomas Schmid, Mani B. Srivastava, and Yan Wang. Challenges in resource monitoring for residential spaces. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 1–6, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-824-7.

KTH Wireless NCS Testbed, September 2011. URL `http://code.google.com/p/kth-wsn/`.

M. Leach. *Technical Support Document Strategies for 50% Energy Savings in Large Office Buildings.* NREL/TP-550-49213, Golden, CO: National Renewable Energy Laboratory xviii, 141 pp., 2010.

D. Lehmann. *Event-based state-feedback control.* PhD thesis, Ruhr-Universitat Bochum, 2011.

D. Lehmann and J. Lunze. Extension and experimental evaluation of an event-based state-feedback approach. *Control Engineering Practice*, 19(2):101 – 112, 2011. ISSN 0967-0661.

M. Lemmon, T. Chantem, X.S. Hu, and M. Zyskowski. On Self-Triggered Full-Information H-infinity Controllers. *Hybrid Systems: Computation and Control*, 2007.

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.

L. Li and M.D. Lemmon. Performance and average sampling period of sub-optimal triggering event in event triggered state estimation. In *IEEE Conference on Decision and Control*, December 2011.

C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM*, 20:46–61, January 1973. ISSN 0004-5411.

X. Liu and A. J. Goldsmith. Wireless network design for distributed control. In *IEEE CDC*, 2004.

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 8th ACM Conference on Embedded Networked*

Sensor Systems, SenSys '10, pages 211–224, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0344-6.

J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, January 2010.

M. Mazo Jr. and P. Tabuada. On Event-Triggered and Self-Triggered Control over Sensor/Actuator Networks. 2008.

M. Mazo Jr. and P. Tabuada. On the robustness of self-triggered control for sensor/actuator networks. 2009.

M. Mazo Jr. and P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *To appear in IEEE Transactions on Automatic Control*, April 2010.

M. Mazo Jr., A. Anta, and P. Tabuada. On self-triggered control for linear systems: Guarantees and complexity. 2009.

M. Mazo Jr., A. Anta, and P. Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46(8):1310–1314, 2010.

A. Molin and S. Hirche. On LQG joint optimal scheduling and control under communication constraints. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 5832 –5838, 2009.

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.

R. M. Murray et al. *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems.* Society for Industrial and Applied Mathematics, 2003. URL `http://www.cds.caltech.edu/~murray/cdspanel`.

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.

J. Nilsson. *Real-Time control systems with delays.* PhD thesis, Lund Institute of Technology, Jan 1998. Ph.D. thesis.

P.G. Otanez, J.R. Moyne, and D.M. Tilbury. Using deadbands to reduce communication in networked control systems. In *American Control Conference, 2002. Proceedings of the 2002*, volume 4, pages 3015 – 3020 vol.4, 2002. doi: 10.1109/ACC.2002.1025251.

P. Park. *Modeling, Analysis, and Design of Wireless Sensor Network Protocols.* PhD thesis, Royal Institute of Technology (KTH), March 2011. TRITA-EE 2011:001.

P. Park, P. Di Marco, P. Soldati, C. Fischione, and K. H. Johansson. A generalized markov chain model for effective analysis of slotted IEEE 802.15.4. In *IEEE MASS*, 2009.

P. Park, C. Fischione, and K. H. Johansson. Adaptive IEEE 802.15.4 protocol for energy efficient, reliable and timely communications. In *ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010.

K. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In *PDCS*, 2008.

J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch. Comparative Study of Energy-Efficient Sampling Approaches for Wireless Control Networks. *IEEE Transactions on Industrial Informatics*, (99):1, 2010.

J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. *Information Processing in Sensor Networks 2005. Fourth International Symposium on*, Apr. 2005.

S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L.V. Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor. Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer. *IEEE Transactions on Wireless Communication*, 7(9):3359–3371, 2008.

A. Prayati, Ch. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos. A modeling approach on the telosb wsn platform power consumption. *J. Syst. Softw.*, 83(8):1355–1363, 2010. ISSN 0164-1212.

Quanser. *Coupled Water Tanks*. Quanser, 2011, September 2011. URL `http://www.quanser.com/english/downloads/products/Specialty/CoupledTanks_PIS_031708.pdf`.

M. Rabi. *Packet based Inference and Control*. PhD thesis, Institute for Systems Research, University of Maryland, 2006. URL `http://hdl.handle.net/1903/6612`.

M. Rabi and K. H. Johansson. Event-triggered strategies for industrial control over wireless networks. In *International Conference on Wireless Internet*, pages 1–7, 2008.

M. Rabi and K. H. Johansson. Scheduling Packets for Event-Triggered Control. *European Control Conference*, 2009.

M. Rabi, K. H. Johansson, and M. Johansson. Optimal stopping for event-triggered sensing and actuation. In *Proceedings IEEE Conference on Decision and Control*, 2008.

C. Ramesh, H. Sandberg, and K. H. Johansson. Steady state performance analysis of multiple state-based schedulers with csma. In *to appear IEEE Conference on Decision and Control*, 2011a.

C. Ramesh, H. Sandberg, and K. H. Johansson. On the dual effect in state-based scheduling of networked control systems. In *American Control Conference 2011*, june 2011b.

R. Rom and M. Sidi. *Multiple access protocols: performance and analysis*. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97253-6.

C.J. Rozell and D.H. 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.

Tariq Samad, Paul McLaughlin, and Joseph Lu. System architecture for process automation: Review and trends. *Journal of Process Control*, 17(3):191 – 201, 2007. ISSN 0959-1524. doi: 10.1016/j.jprocont.2006.10.010. URL `http://www.sciencedirect.com/science/article/pii/S0959152406001156`. <ce:title>Special Issue ADCHEM 2006 Symposium</ce:title>.

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.

G. Seyboth, D.V. Dimarogonas, and K.H. Johansson. Control of multi-agent systems via event-based communication. In *18th IFAC World Congress*, 2011.

L. Shi, K. H. Johansson, and L. Qiu. Time and event-based sensor scheduling for networks with limited communication resources. In *World Congress of the International Federation of Automatic Control (IFAC)*, Aug 2011.

B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453—-1464, 2004.

J. Song, A. K. Mok, D. Chen, M. Nixon, T. Blevins, and W. Wojsznis. Improving PID Control with Unreliable Communications. *ISA EXPO Technical Conference*, 2006.

M. Tabbara and D. Nesic. Number 5, pages 1160 –1175, june 2008. doi: 10.1109/TAC.2008.923691.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

P. Tabuada and X. Wang. Preliminary results on state-triggered scheduling of stabilizing control tasks. *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.

U. Tiberi, C. Fischione, K.H. Johansson, and M.D. Di Benedetto. Adaptive self-triggered control over ieee 802.15.4 networks. In *IEEE Conference on Decision and Control*, December 2010.

U. Tiberi, C. Fischione, K.H. Johansson, and M.D. Di Benedetto. Self-triggered control of multiple loops over ieee 802.15.4 networks. In *IFAC World Congress, Milan, Italy*, 2011.

UK Department of Trade and Industry. *DTI, Energy Trends 2005*. Department of Trade and Industry, London, 2011. URL `http://webarchive.nationalarchives.gov.uk/+/http://www.berr.gov.uk/files/file10735.pdf`.

V. Ungan. Networked PID controllers for wireless systems. Master's thesis, Royal Institute of Technology (KTH), November 2010.

U.S. Department of Energy. *Energy Efficiency Trends in Residential and Commercial Buildings*. U.S. Department of Energy, Oct. 2008. URL `http://apps1.eere.energy.gov/buildings/publications/pdfs/corporate/bt_stateindustry.pdf`.

V. Vasyutynskyy and K. Kabitzsch. Simple pid control algorithm adapted to deadband sampling. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pages 932 –940, sept. 2007. doi: 10.1109/EFTA.2007.4416884.

M. Velasco, J. Fuertes, and P. Martí. The self triggered task model for real-time control systems. *24th IEEE Real-Time Systems Symposium (work in progress)*, pages 67–70, 2003.

X. Wang and M. Lemmon. State based Self-triggered feedback control systems with L2 stability. *17th IFAC world congress*, 2008a.

X. Wang and M. Lemmon. Self-triggered feedback control systems with finite-gain l2 stability. *IEEE Transactions on Automatic Control*, 45:452–467, 2009a.

X. Wang and M. Lemmon. Self-triggered feedback control systems with finite-gain l2 stability. *IEEE Transactions on Automatic Control*, 45:452–467, 2009b.

X. Wang and M.D. Lemmon. Event design in event-triggered feedback control systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 2105 –2110, dec. 2008b. doi: 10.1109/CDC.2008.4739105.

X. Wang and M.D. Lemmon. Event-triggering in distributed networked control systems. *Automatic Control, IEEE Transactions on*, 56(3):586 –601, march 2011a. ISSN 0018-9286. doi: 10.1109/TAC.2010.2057951.

X. Wang and M.D. Lemmon. Minimum attention controllers for event-triggered feedback systems. In *to appear in IEEE Conference on Decision and Control - European Control Conference*, 2011b.

A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2):102 –124, May 2008. ISSN 1551-3203. doi: 10.1109/TII.2008.923194.

J.K. Yook, D.M. Tilbury, and N.R. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *Control Systems Technology, IEEE Transactions on*, 10(4):503 –518, jul 2002. ISSN 1063-6536. doi: 10.1109/TCST.2002.1014671.

M. Yu, L. Wang, G. Xie, and T. Chu. Stabilization of networked control systems with data packet droupout via switched system approach. In *IEEE CACSD*, 2004.

Q. Zheng and K.G. Shin. On the ability of establishing real-time channels in point-to-point packet-switched networks. *IEEE Transactions on Communications*, 42 (234):1096–1105, 1994.