Model-Based Cross-Design for Wireless Networked Control Systems with Limited Resources

Modellgestütztes Cross-Design für funkbasierte Regelungssysteme mit beschränkten Ressourcen

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Kaiserslautern
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von
M.Sc. (Univ.) Dipl.-Ing. (FH) BEng (Hons) Alain Chamaken
geb. in Douala (Kamerun)

D 386

Eingereicht am:	09.01.2013
Tag der mündlichen Prüfung:	03.05.2013
Dekan des Fachbereichs:	Prof. Dr.-Ing. Norbert Wehn
Promotionskommission:	
Vorsitzender:	Prof. Dr.-Ing. Steven Liu (*Technische Universität Kaiserslautern*)
Berichterstattende:	Prof. Dr.-Ing. habil. Lothar Litz (*Technische Universität Kaiserslautern*) Prof. Dr. Reinhard Gotzhein (*Technische Universität Kaiserslautern*)

Abstract

English- This work shall provide a foundation for the *cross-design* of wireless networked control systems with limited resources. A *cross-design* methodology is devised, which includes principles for the modeling, analysis, design, and realization of low cost but high performance and intelligent wireless networked control systems. To this end, a framework is developed in which control algorithms and communication protocols are jointly designed, implemented, and optimized taking into consideration the limited communication, computing, memory, and energy resources of the low performance, low power, and low cost wireless nodes used. A special focus of the proposed methodology is on the prediction and minimization of the total energy consumption of the wireless network (i.e. maximization of the lifetime of wireless nodes) under control performance constraints (e.g. stability and robustness) in dynamic environments with uncertainty in resource availability, through the joint (offline/online) adaptation of communication protocol parameters and control algorithm parameters according to the traffic and channel conditions. Appropriate optimization approaches that exploit the structure of the optimization problems to be solved (e.g. linearity, affinity, convexity) and which are based on Linear Matrix Inequalities (LMIs), Dynamic Programming (DP), and Genetic Algorithms (GAs) are investigated. The proposed *cross-design* approach is evaluated on a testbed consisting of a real lab plant equipped with wireless nodes. Obtained results show the advantages of the proposed *cross-design* approach compared to standard approaches which are less flexible.

German- *Diese Arbeit soll die Grundlage für das Cross-Design von funkbasierten Regelungssystemen mit beschränkten Ressourcen legen. Es wird eine Cross-Design Methodik entwickelt, welche gleichzeitig die Ebenen Control (Regelalgorithmus), Communication (Protokolle) und Computing (Energie, Speicherkapazität und Rechenleistung) betrachtet. Ziel dieser Methodik ist der gemeinsame Entwurf von Regelungsalgorithmen und Kommunikationsprotokollen, die aufeinander abgestimmt werden, unter Berücksichtigung von beschränkten Ressourcen, wie Rechenleistung, Energievorrat und Speicherkapazität, der eingesetzten Funkknoten. Eines der Hauptziele ist die Vorhersagbarkeit und Optimierung der Lebensdauer der Funkknoten bei gleichzeitiger Einhaltung einer Mindestregelgüte in dynamischen Umgebungen mit Unsicherheiten bei der Verfügbarkeit von Ressourcen, durch eine gemeinsame (offline/online) Anpassung der Reglerparameter und Protokollparameter an Kanal- und Traffic-Zustände. Methodisch gesehen werden unterschiedliche Formen der Optimierung unter Einhaltung von Nebenbedingungen und Betrachtung der Eigenschaften (z.B. Linearität, Affinität, Konvexität) untersucht, sowohl auf Basis der Linearen Matrixungleichungen (LMIs), Dynamische Programmierung (DP), als auch der Genetischen Algorithmen (GAs). Die Garantie einer Mindestlebensdauer der Funkknoten bzw. deren Optimierung ist dabei das Hauptziel. Die vorgeschlagene Cross-Design Methodik wird an einem Testbed, bestehend aus einer realen Regelstrecke und ausgewählten Funkknoten, evaluiert. Ergebnisse zeigen die Vorteile des Cross-Design-Ansatzes gegenüber Standardansätzen, die weniger flexibel sind.*

Acknowledgment

This work is the summary of four and a half years of extensive research at the Institute of Automatic Control of the University of Kaiserslautern. Results obtained could not be achieved without the help, assistance, support, and encouragement of several persons.

I would like to start by expressing my gratitude to my doctoral advisor Professor Lothar LITZ for giving me the opportunity to work and do research in the new emerging field of Wireless Networked Control System (WNCS). I'm deeply grateful for his guidance, advice, encouragement and for the intellectual freedom necessary to perform independent research.

Beside Professor Lothar LITZ, I would also like to express my sincere gratitude to the other members of the examination board: The chairman Professor Steven LIU (University of Kaiserslautern) and Professor Reinhard GOTZHEIN (University of Kaiserslautern).

As a member of the Institute of Automatic Control research team, I had the privilege to be part of a wonderful family. I'm grateful to my colleagues for all the interesting discussions we had not only on my work, but also on their own works and other interesting topics. Thank you, dear colleagues, for your friendship, honesty, humor, and for being so open-minded. I always felt welcome in Kaiserslautern and you have definitely broadened my horizon through several thoughts about the German way of living and thinking. A special thank to Monika KUNZ for the unconditional help and support in coping with administrative as well as private issues. To Sven BECKER, Manfred ENGEL, and Thomas JANZ, thanks for the great support in all technical issues.

My sincere thanks to all students who contributed to this work by solving parts of the thesis relevant problems as part of their diploma and master thesis or through their work as student research assistants: Steve FANKEM for the implementation of optimal control structures for the inverted pendulum, Vanessa ROMERO for the extensive bibliography research on the state of the art in the design of Wireless Networked Control Systems, Stefan SCHNEIDER for the implementation of dedicated MAC protocols for the MicaZ wireless node, Jean NEMPE for the implementation of an IMC for the inverted pendulum, Rudi JERGES for the implementation of a serial communication interface for the inverted pendulum testbed, Jiang ZHE, Théody CHONGOUA, and Valentine TCHA-TOKEY for conducting several experiments on the inverted pendulum testbed.

This work was part of the Priority Programme 1305 of the German Research Foundation (DFG): *Control Theory of Digitally Networked Dynamical Systems*. I would like to thank all the programme participants for the interesting discussions during several seminars and workshops. A special thank to Marc KRÄMER for the constructive discussions in the field of wireless communications and for providing a code loader for the Imote2 wireless node.

To Thomas STEFFEN, Andreas HAUPT, Stefan SCHNEIDER, and Nicole SCHWALBACH: Many thanks for proofreading the manuscript and hinting at a considerable amount of flaws I have eliminated with your help.

Finally, my heartiest gratitude to my family, especially to my father who passed away two years ago, for their unconditional love, patience, support, and encouragement.

Kaiserslautern, May 2013

Alain Chamaken

In memory of my father KAMDE NGANDJUI Claude Laurent
R.I.P.

To Ulrich Adrien, Fayola Chanecia, Lise Gäelle and Line Chloe

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Motivating examples . . . 3

1.3 Important new aspects . . . 4

1.4 Target of the thesis . . . 5

1.5 Organization of the thesis . . . 7

2 Wireless Networked Control Systems **11**

2.1 History . . . 11

2.2 Structures and technologies . . . 12

2.2.1 General considerations . . . 12

2.2.2 Control loop topologies . . . 13

2.2.3 Communication technologies . . . 16

2.3 Typical problems . . . 19

2.3.1 Bandwidth limitations . . . 19

2.3.2 Time delays . . . 20

2.3.3 Losses and quasi losses . . . 21

2.3.4 Limited computing and energy resources . . . 22

2.4 Design approaches . . . 22

2.4.1 Overview . . . 22

2.4.2 Controller design approaches . . . 23

2.4.2.1 Sampling and delay . . . 23

2.4.2.2 Packet dropouts . . . 25

2.4.2.3 Approaches combining delays and packet losses . . . 26

2.4.3 Communication design approaches . . . 29

2.4.4 Cross-design approaches . . . 30

2.5 Summary and outcome . . . 32

3 Inverted Pendulum Testbed **33**

3.1 Overview . . . 33

3.2 Inverted Pendulum . . . 34

3.3 Wireless nodes . . . 34

3.4 Interface hardware . . . 36

3.5 Software . . . 37

4 Simulation environment for WNCSs **39**

4.1 Problem formulation . . . 39

4.2 HiL simulator structure . . . 40

4.3 Component description . . . 41
4.3.1 Plant component . . . 41
4.3.2 Network components . . . 42
4.3.3 Controller component . . . 42
4.4 Validation on the inverted pendulum . . . 43
4.5 Derivation of QoS bounds . . . 47
4.6 Summary . . . 48

5 Tailored WNCS communication protocols 49
5.1 Introduction . . . 49
5.2 Layered structure of the WNCS protocol stack . . . 50
5.2.1 WNCS protocol stack . . . 50
5.2.2 Network topology . . . 52
5.2.3 Superframe structure . . . 53
5.2.4 Frame format . . . 54
5.2.5 Power consumption . . . 55
5.3 MAC Protocols for WNCSs . . . 56
5.3.1 Master-Slave MAC protocol . . . 56
5.3.2 TDMA MAC protocol . . . 58
5.3.3 CSMA/CA MAC protocol . . . 59
5.4 Power control algorithm . . . 61
5.4.1 Wireless radio model . . . 62
5.4.2 PER power control . . . 63
5.5 Node lifetime models . . . 64
5.6 Experimental results . . . 67
5.7 Summary . . . 69

6 Proposed cross-design framework 71
6.1 Introduction . . . 71
6.2 Cross-design methodology . . . 71
6.3 Cross-design modeling . . . 74
6.3.1 Plant dynamics . . . 74
6.3.2 Control model . . . 75
6.3.3 State estimator . . . 76
6.3.4 Controller design . . . 78
6.4 Cross-design procedure . . . 79
6.5 Summary . . . 80

7 Results of the proposed cross-design approach 83
7.1 Experimental setup . . . 83
7.1.1 Inverted pendulum system . . . 84
7.1.2 Wireless communication protocols . . . 87
7.2 Results of the cross-design procedure . . . 89
7.3 Results of the cross-design optimization . . . 96
7.4 Summary . . . 99

8 Summaries **101**
8.1 Summary in English 101
8.2 Kurzfassung in deutscher Sprache (Extended summary in German) 103

A Appendix **109**
A.1 Inverted pendulum system 109
A.1.1 Mathematical modeling 109
A.1.2 Linearization 110
A.2 Control structures 111
A.2.1 Cascaded controller 112
A.2.2 IMC controller 112
A.2.3 State feedback controller 113
A.2.3.1 Pole placement controller design 114
A.2.3.2 LQR controller design 114
A.2.3.3 Filter design 115
A.2.3.4 Observer design 115
A.2.4 H-infinity controller 116
A.2.5 Results 117
A.2.5.1 Performance index for the QoC 117
A.2.5.2 Simulation results 118
A.2.5.3 Experimental results 120
A.3 Overview of the IEEE 802.15.4-2006 protocol standard 121
A.3.1 Architecture and components 121
A.3.2 Frame structure and data transfer 123
A.3.3 Physical layer specification 124
A.3.4 MAC sublayer specification 125

Nomenclature **127**

Bibliography **136**

1 Introduction

1.1 Motivation

Wireless technologies have been deployed in industrial environments over the past years thanks to the widespread applications of the internet, the emergence of pervasive communication and computing, the successful use of wireless technologies in consumer electronics (e.g. mobile phones, smartphones, PDAs), the availability of general purpose wireless communication technologies and standards (e.g. WLAN, Bluetooth, ZigBee), the development of new wireless communication standards tailored to the specific needs of industrial applications (e.g. WISA, WirelessHART, ISA-SP100), and most importantly the development of small size, low-cost, and low-power wireless devices. Nowadays, wireless technologies are used in areas such as mobile sensor networks, remote surgery and haptics collaboration over the internet, automated highway systems, unmanned aerial vehicles (UAVs), industrial and factory automation, automotive systems, building automation, healthcare, energy generation, IT and networks.

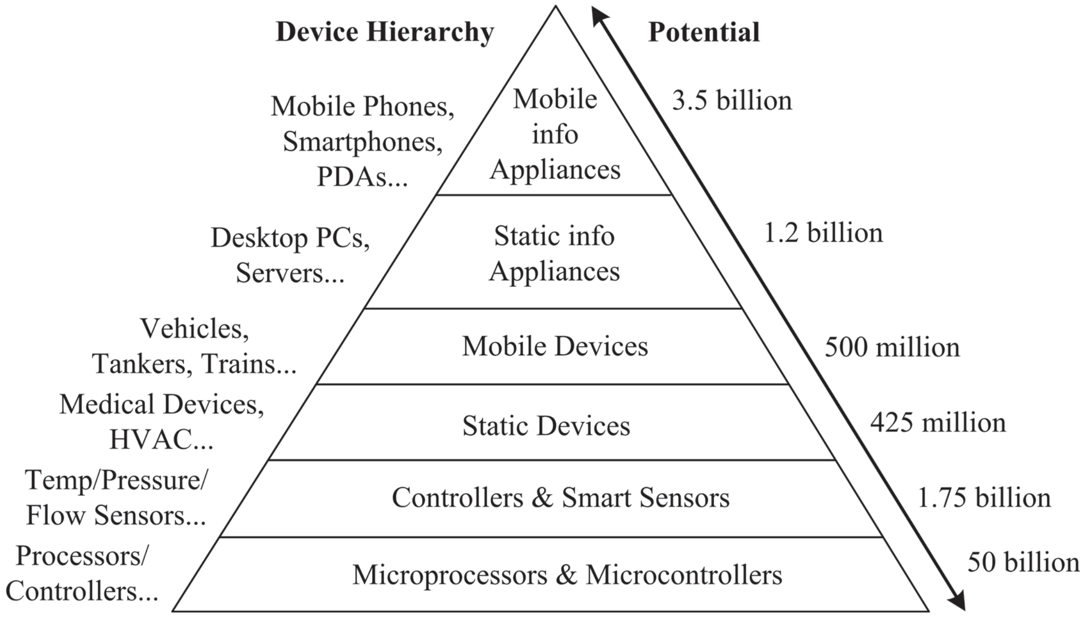


Figure 1.1: M2M device forecast [Berman & Ritorto, 2009]

Several forecasts have recently predicted exponential growths in the market of wireless devices and technologies over the next few years, which would result in a multi-billion dollar market in the near feature. Figure 1.1 shows Harbor's market analysis and forecast on machine-to-machine (M2M) and intelligent device networking for wired and wireless technologies from 2009 to 2013 [Berman & Ritorto, 2009]. Approximately 3.5 billion mobile information appliances, 500 million mobile devices, 1.75 billion controllers and smart sensors, and 50 billion microprocessors and microcontrollers will be in use by the end of 2013. According to Harbor Research, this development will drive the largest growth opportunity in the history of business.

Network-based Control Systems [Kim *et al.* , 1996] or *Networked Control Systems* (NCSs) [Walsh *et al.* , 2002] are spatially distributed control systems where the communication between sensors, controllers, and actuators takes place over a shared (wireline or wireless) and band-limited digital communication network. Wireless Networked Control Systems (WNCSs) are a class of NCSs where sensors, controllers and actuators exchange information over a wireless digital communication network. WNCSs are based on networked wireless devices which have many advantages compared to their wired counterparts such as easy deployment, flexible architecture, low installation and maintenance cost, lack of cabling, and high mobility. There is not yet a widespread use of wireless technologies in process, manufacturing and industrial applications despite their great potentials. One of the main reasons is that currently available wireless technologies do not have a framework which is easy to use and above all applicable across several application domains. Expert knowledge over the whole protocol stack from the communication layer to the application layer is often required for each specific application development.

The use of a shared digital communication network between sensors, controllers and actuators introduces new challenges such as the stability of the interconnected dynamical systems or the effects of communication on the performance of the control systems which makes *control over communication networks* one of the key feature research directions in control. For example, WNCSs applications must often satisfy tight real-time, reliability, and latency requirements regardless of the traffic conditions, even in the presence of unexpected congestion, network failures or external manipulations of the environment. In wired environments, timing and reliability are well attended by bus systems and protocols, which are a mature technology. Reliability and timing requirements are more difficult to achieve in wireless networks due to the characteristics of the radio channels. Moreover, the energy and computational efficiency of the operated wireless networks are also critical factors due to the limited lifetime and computing capabilities of battery powered wireless devices.

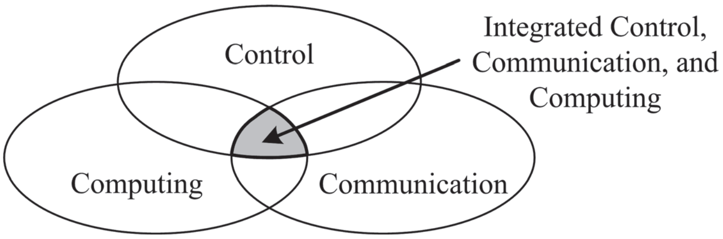


Figure 1.2: WNCSs at intersection between control communication and computing

The multidisciplinary research field of WNCSs combines control theory, communication technology, and computer science into a new dynamic environment for real-time control systems. This new environment features both resource limitations and workload variability resulting in uncertainties about the availability of the computing (e.g. CPU time, energy) and/or communication (e.g. bandwidth) resources (see Figure 1.2). Traditional control theory focuses on the interconnection of dynamical systems connected through *ideal channels* assuming equidistant sampling, negligible or constant communication delays and no data losses, whereas communication theory studies the transmission of information over *imperfect channels* with variable transmission delays and possible data losses. Moreover, systems engineers mostly assume that control and communication tasks are periodic with a set of fixed periods, having hard deadlines and known Worst-Case Execution Time (WCET). These assumptions are not always true, especially in dynamic environments with resource uncertainty. Therefore, in order to design and implement control algorithms and communication protocols for real-time WNCSs, principles and methods from computer science and communication technology should be included into the design and implementation of control systems thus, leading to the convergence of control, communication and computing. Examples of research activities related to the convergence of control, communication and computing are the European Union (EU) programs ARTIST2[1] and RUNES[2].

1.2 Motivating examples

A testbed consisting of an inverted pendulum [Chamaken & Litz, 2011] and a water flow control process [Chamaken & Litz, 2010] has been developed during this thesis to demonstrate and evaluate new solutions to the joint design of control algorithms and communication protocols (cross-layer design) for WNCSs (see Figure 1.3). Wireless position, angle, velocity, and flow sensors are used in conjunction with controllers to control both the inverted pendulum and the flow process. The position of the inverted pendulum's cart is controlled using a DC motor while the water flow is controlled by a pneumatic valve. The inverted pendulum control is implemented on a single-hop wireless network using three wireless sensors, a wireless relay, a wireless controller, and a wireless actuator whereas the flow control process is implemented on the same wireless network with one wireless sensor, a wireless controller connected to a PC running MATLAB, and a wireless actuator.

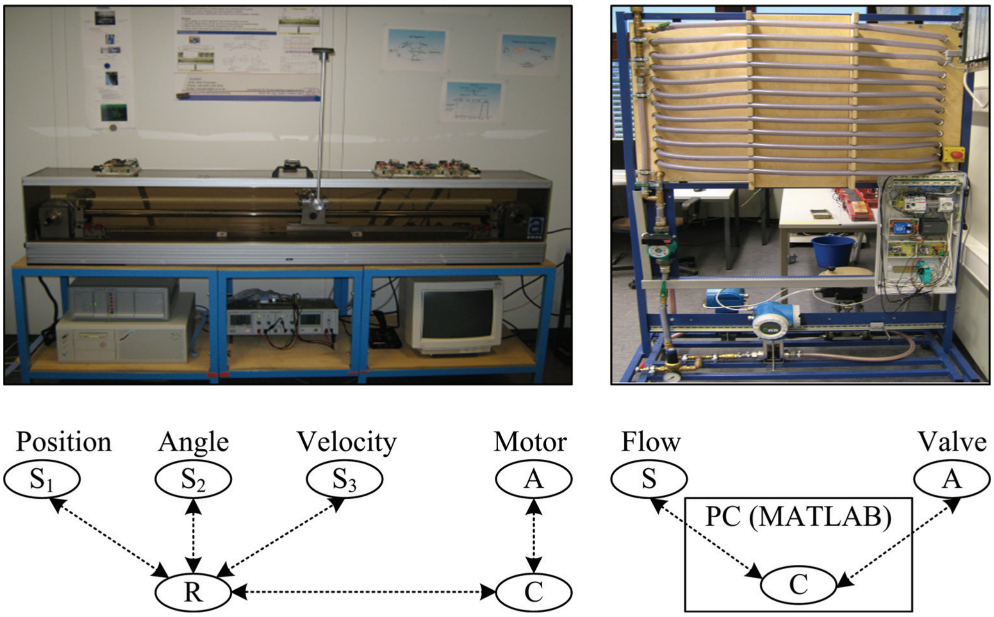


Figure 1.3: Testbed consisting of an inverted pendulum [Chamaken & Litz, 2011] and a flow process [Chamaken & Litz, 2010] with Sensors (S), Controllers (C), Actuators (A), and Relays (R)

Figure 1.4 on Page 4 shows a WNCS which was developed for a froth flotation process within the European Union project SOCRADES [Socrades, 2011]. Wireless level and flow sensors are used in conjunction with controllers to regulate the flotation process which concentrates the metal-bearing mineral in the ore. The model of the process consists of five cascade coupled tanks with tank levels $h_1, ..., h_5$. There is a variable inflow q_{in} to the first tank and the outflows $q_1, ..., q_5$ from each tank are regulated with the valve settings $u_1, ..., u_5$. The floatation process is

[1] http://www.artist-embedded.org/artist/
[2] http://www.ist-runes.org/

implemented on a wireless mesh network using five wireless sensors, five wireless actuators, two wireless relay nodes, and a wireless gateway. Time-varying LQG controllers for wireless mesh networks are used to control the floatation process.

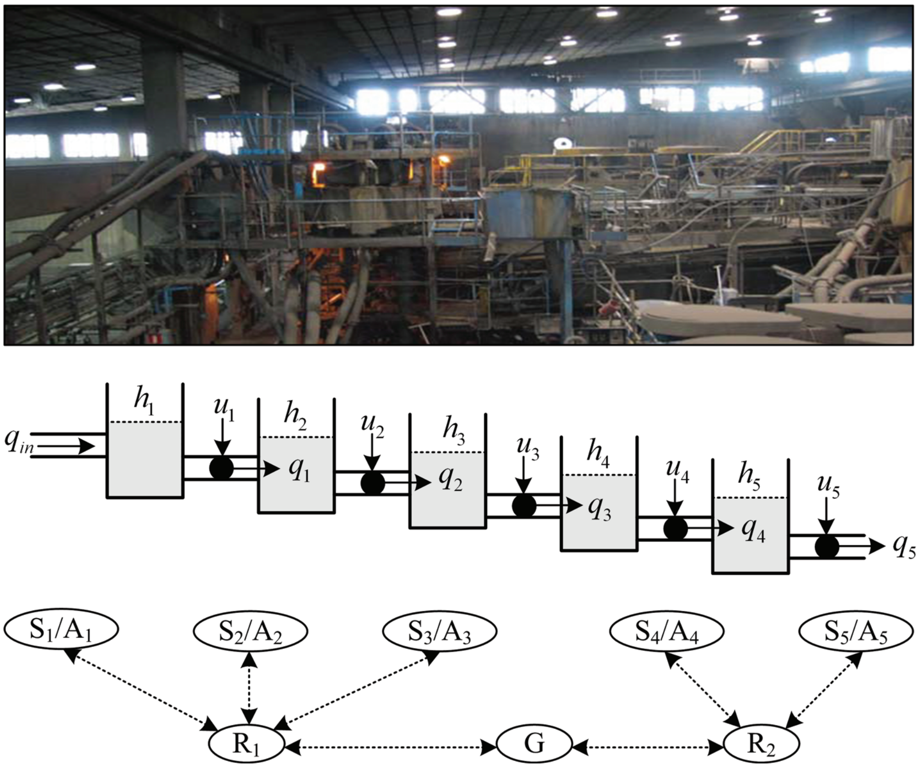


Figure 1.4: Wireless control of a froth flotation process with Actuators (A), Sensors (S), Gateways (G), and Relays (R) [Socrades, 2011]

1.3 Important new aspects

According to [Baillieul & Antsaklis, 2007], there are two important new aspects in the theoretical approaches to the analysis and design of WNCSs. The first aspect has to do with the explicit consideration of network interconnections which affect the dynamic behavior of the control system. Control systems must be redefined adding the wireless network as a new component next to plant, sensor, controller, and actuator components. Beside delays and data losses, the use of wireless communications introduces new issues such as fading and time-varying throughput in communication channels. As a result, the effect of interconnections needs to be explicitly addressed in WNCSs. The second aspect has to do with a shift of emphasis from centralize to distribute NCSs. The availability of enough processing power at low cost that can be embedded almost anywhere opens a vast array of new possibilities. Sensor data and control policies can be processed locally without needing a central decision maker. Although the remote nodes act locally, they must coordinate their actions to achieve common goals. Therefore, there is no need to exchange large amounts of data over the network.

Following aspects have to be considered when designing WNCSs:

- *Reliability:* Sensor measurements and control action losses can prevent the correct execution of control algorithms. Reliability can be maximized by using retransmissions in the event of packet losses but can result in an increase of the energy consumption of the network since nodes have to be awake for longer periods of time. Therefore, a tradeoff between reliability and network energy consumption has to be considered.

- *Latency:* Sensor measurements and control actions must reach their destinations within a given deadline since the resulting time delay influences control system performance and stability. Delay jitter with large variability is particularly difficult to compensate and a probabilistic delay requirement should be considered instead of using average delays. Also, the retransmission of packets to ameliorate the reliability may increase the delay. Therefore, a tradeoff between reliability and latency must be considered.

- *Energy:* Energy efficient operation of wireless nodes is required to prevent frequent battery replacement and increase the network lifetime. Maximizing the reliability and minimizing the latency may result in a better performance of the control system. However, the drawback is an increase of the energy consumption of the network. A tradeoff between an acceptable control performance and a minimum network lifetime should be considered.

- *Flexibility:* Selected communication protocol parameters and control algorithm parameters should be adaptable to control application requirements changes, time-varying wireless channels, and network topology variations. This requires analytical models describing the relation between protocol parameters, controller parameters, and selected performance indices such as the Quality of Control (QoC), the Quality of Services (QoS), and the energy consumption.

- *Control strategy:* The type and amount of data to be transmitted also depends on the selected control strategy. For real-time periodic control, sensor measurements and actuation commands must be received periodically within a specified deadline. Therefore, the network must be able to guarantee the delay of a signal within the specified time deadline. For event-based control and self-triggered control, sensor measurements and actuation commands do not have a time deadline. However, the network must guarantee access when events occur.

- *Scalability:* Due to the limited processing capabilities of the wireless nodes, control algorithms and communication protocols must be computationally light. Control and communication operations should be performed within the network to avoid high communication load with a central coordinator. Hence, the tradeoff between traceability and accuracy of the analytical models is of great importance.

As a result, the design of WNCSs must take into consideration a large number of competing tradeoffs. In particular, a tradeoff exists between control performance, communication performance, and network lifetime. Therefore, control and communication should be designed jointly and this thesis provides a suitable framework for this purpose.

1.4 Target of the thesis

The main contribution of this thesis is the development of a framework for the *cross-layer* modeling, analysis, design, and implementation of WNCSs that enables the combination of control with communication and computing for implementing real-time control in dynamic environments. The target of this framework is to provide principles for the design of low cost but high performance WNCSs that require less computing power, small memory space, less energy, and little upgrading. A special focus is on the optimization of the control performance and the energy consumption of the network in dynamic environments with uncertainty in resource availability through the joint adaptation of communication protocol parameters and control algorithm parameters according to the traffic and channel conditions. Parts of the thesis include:

1. Mathematical modeling of control algorithms, communication protocols, energy consumption, and performance indicators such as control performance, reliability, latency, and network lifetime.
2. Development of a methodology for the cross-layer design of WNCSs which takes into consideration the control, communication, computing, and energy aspects.
3. Resource constrained offline/online optimization of control performance and minimization of network energy consumption using tunable parameters of control algorithms and communication protocols.
4. Development of a simulation framework for WNCSs which enables the simulation of the control performance for selected control algorithms and the network effects on their control performance.
5. Joint implementation of high-performance and computationally efficient control algorithms and communication protocols on a selected hardware platform
6. Realization of a testbed for the experimental verification and validation of cross-design results.

The main objective is the minimization of the total energy consumption of the wireless network (i.e. maximization of the node lifetime) under control performance requirement constraints and limited resources. Therefore, the constrained multi-objective optimization problem with several, possibly conflicting, objectives to be satisfied is

$$\min_{\mathbf{x}} \quad \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x})\ f_2(\mathbf{x})\ f_3(\mathbf{x})] \tag{1.1a}$$

$$s.t. \quad \mathbf{x} \in \Omega_C \cup \Omega_L \cup \Omega_R \cup \Omega_P \tag{1.1b}$$

where $\mathbf{x}$ is a vector of decision variables, f_1 the objective function for the control performance, f_2 the objective function for the network performance, and f_3 the objective function for the network energy consumption. Ω_C is the feasible set for the controller parameters that meet stability constraints, Ω_R and Ω_L the feasible sets of protocol parameters that meet the reliability and latency constraints respectively, and Ω_P the set of physical layer properties which depend on the hardware platform used.

The solution of the above optimization problem determines the control strategies and the communication policies. The decision variables $\mathbf{x}$ are the controller parameters (e.g. sampling period, controller gains), the protocol parameters of the physical layer (PHY) (e.g. transmit power, data rate), the medium access control layer (MAC) (e.g. access probability, TDMA slot duration, CSMA/CA random backoff), and the routing layer (ROU) (e.g. expected transmission time of a packet). There may not exist one solution $\mathbf{x}^*$ which is the best (global minimum) with respect to all objectives. Typically, there exists a set of solutions which are better than the rest of the solutions in the decision space when all objectives are considered but are worse than other solutions in the decision space in one or more objectives. These solutions are known as *Pareto-optimal* solutions or *nondominated* solutions. The choice of one particular solution from the *Pareto-optimal* set requires problem knowledge and a number of problem-related factors. Therefore, the solution chosen by one designer may not be acceptable to another designer or in a modified environment. Genetic Algorithms (GAs) can be used to solve problem (1.1) since they work with a population of solutions.

Another way to formulate problem (1.1) is to scalarize the vector of objectives into one objective by averaging the objectives with a weight vector and results in

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + w_3 f_3(\mathbf{x}) && (1.2a) \\ s.t. \quad & \mathbf{x} \in \Omega_C \cup \Omega_L \cup \Omega_R \cup \Omega_P && (1.2b) \end{aligned}$$

where $0 \leq w_i \leq 1, i = 1, 2, 3$ and $\sum_{i=1}^{3} w_i = 1$. The preference of an objective can be changed by modifying the corresponding weight. This process allows a simpler optimization algorithm to be used, but the obtained solution largely depends on the weight vector used.

A third way to formulate problem (1.1) is to move all but one objective to the constraint set resulting in

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_1(\mathbf{x}) && (1.3a) \\ s.t. \quad & f_2(\mathbf{x}) \leq b_2 && (1.3b) \\ & f_3(\mathbf{x}) \leq b_3 && (1.3c) \\ & \mathbf{x} \in \Omega_C \cup \Omega_L \cup \Omega_R \cup \Omega_P && (1.3d) \end{aligned}$$

As with problem (1.2), a simpler optimization algorithm can be used, but the difficulty here is to find constraining values b_2 and b_3 for the former objective functions f_2 and f_3.

Analytical expressions for the control performance, the network performance, the energy consumption of the network, the reliability and the latency are essential in order to solve problems (1.1), (1.2), and (1.3) but are difficult to obtain due to the complexity of WNCSs. Therefore, approximations will be used to get tractable analytical models.

1.5 Organization of the thesis

Chapter 2 gives a summary of relevant works and interesting results in the research field of NCSs in general and WNCSs in particular. A brief review of the history of control systems from classical feedback control towards digital control to wireless networked control is given. State of the art solutions to standard WNCSs problems with advantages and limitations are presented and discussed.

In chapter 3, an inverted pendulum equipped with wireless nodes and controlled over a dedicated wireless communication network is presented, which is used as a testbed throughout the thesis. The inverted pendulum and all the hardware and software components used in the testbed are briefly presented and described. It should be pointed out however, that all the theories and methods developed in the course of this thesis are generic in the sense that they can be applied to other wireless networks based control systems.

The MATLAB®/Simulink® based WNCSs simulation framework used throughout this thesis is presented in chapter 4. This simulation framework enables the simulation of different control structures and algorithms. The integrated HiL-Simulator (Hardware in the loop simulator) which includes not only the real plant but also the real wireless network in the feedback control loop and allows fast real-time simulations with accurate simulation results is also presented. Typical network effects such as delay, jitter, data dropouts, and traffic burst can be simulated using the HiL-Simulator. Simulation results for different control structures and algorithms for the inverted pendulum are presented and discussed.

In chapter 5, selected adaptive communication protocols for the inverted pendulum testbed with the corresponding analytical models are presented. The communication protocols tailored to the requirements of the inverted pendulum are based on the IEEE 802.15.4-2006 standard. Implemented protocols include contention based random access MAC with stochastic behavior (e.g. CSMA/CA) and contention free access MAC with deterministic behavior (e.g. TDMA). Duty cycle based energy management policies and power control algorithms implemented across all the layers are presented and discussed. All communication algorithms are implemented on wireless nodes and experimental results showing that the implemented algorithms satisfy the reliability and latency requirements of the inverted pendulum while minimizing the total energy consumption of the network are also presented and discussed.

The selected approach to the cross-layer analysis, design, and optimization of WNCSs is presented in chapter 6. Our cross-design framework consisting of a physical layer, a data link layer, a routing layer, and a control layer on top of the routing layer is presented. Network effects are included in the control law derivation for selected control structures of the inverted pendulum. Optimal controller parameters are obtained by means of analytical methods or heuristics based methods with Genetic Algorithms. State estimators such as Kalman filters are used to estimate missing measurements and improve control performance in the event of measurement noise and losses. The general constrained optimization problem of the form (1.1), (1.2), or (1.3) is formulated and solved using selected degrees of freedom an appropriate optimization methods based on the structure (e.g. least-squares, linear programming) and properties (e.g. linearity, convexity) of the optimization problem. Appropriate analytical models for control, communication, and energy used in the solution of the optimization problems are derived. Stability issues are also addressed in this chapter.

In chapter 7, the selected cross-design methodology is applied to the inverted pendulum testbed. In particular, the effects of communication protocol design and protocol parameters choice on the control performance are studied. Also, methods for adapting controller parameters to changing network conditions in order to optimize the control performance are investigated. Finally, feasible regions for the control performance and the network energy consumption are derived.

A summary of the thesis is given in chapter 8 followed by an extended summary in German.

Following papers written by the author and co-authors and listed in reverse chronological order contain material that have influence the thesis:

- Chamaken, A.; Litz, L: *Modeling and Optimization of the Lifetime of Wireless Nodes in Wireless Networked Control Systems.* Proceeding of the Automation 2011, pp. 93-96, June 2011, Baden-Baden, Germany.

- Chamaken, A.; Litz, L.: *Joint Design of Control and Communication for Wireless Networked Control Systems: A Case Study.* Proceeding of the American Control Conference 2010 (ACC'10), pp. 1834-1840, June 2010, Baltimore, USA.

- Chamaken, A.; Litz, L.; Krämer, M.; Gotzhein, R.: *Model-based C3-Cross-Design for Wireless Networked Control Systems.* Proceedings of the Symposium on Recent Trends in Networked Systems and Cooperative Control (NESCOC), September 2009, Stuttgart, Germany.

- Chamaken, A.; Litz, L.: *Control Performance Optimization in Wireless Networked Control Systems: A Case Study.* Proceedings of the Automation 2010, pp. 89-93, June 2010, Baden-Baden, Germany.

- Chamaken, A.; Litz, L.: *Joint Design of Control and Communication for Wireless Networked Control Systems.* at Automatisierungstechnik, Nr. 58, Issue 4, pp. 192-205, April 2010.

- Gotzhein, R.; Krämer, M.; Chamaken, A. ; Litz, L.: *Energy-aware System Design with SDL*. 14th System Design Languages Forum (SDL2009), pp. 19-33, September 2009, Bochum, Germany.

- Chamaken, A.; Litz, L.; Krämer, M.; Gotzhein, R.: *Cross-Layer Design of Wireless Networked Control Systems with Energy Limitations*. European Control Conference 2009 (ECC'09), August 2009, pp. 2325-2330, Budapest, Hungary.

- Chamaken, A.; Litz, L.; Krämer, M.; Gotzhein, R.: *A New Approach to the Joint Design of Control and Communication in Wireless Networked Control Systems*. Proceedings of the Automation 2009, pp. 251-255, June 2009, Baden-Baden, Germany.

Following works supervised by the author and listed in reverse chronological order also contain material that have influence the thesis:

- S. Fankem: *Design and Implementation of Robust and Optimal Controllers for an Inverted Pendulum*. Master Thesis (D219), University of Kaiserslautern, department of Automatic Control, April 2009.

- S. Schneider: *Analysis of Different MAC Scenarios for the Stabilization of an Inverted Pendulum over a Wireless Network*, Bachelor Thesis (S197), University of Kaiserslautern, Department of Automatic Control, May 2009.

- J. Nempe: *Design of an IMC controller for the Inverted Pendulum System*, Bachelor Thesis (S196), University of Kaiserslautern, Department of Automatic Control, May 2009.

- R. Jerges: *Design and Implementation of a Communication Interface Between a Sensor Board and a PC*, Bachelor Thesis (S195), University of Kaiserslautern, Department of Automatic Control, May 2009.

- Z. Jiang: *Investigation of Different Control Structures on an Inverted Pendulum*, Bachelor Thesis (S192), University of Kaiserslautern, Department of Automatic Control, October 2008.

- V. Romero: *Comparison of Three Methods to Manage Time-Delays and Packet Dropouts in Networked Control Systems*, Master Thesis (M023), University of Kaiserslautern, Department of Automatic Control, October 2008.

2 Wireless Networked Control Systems

2.1 History

The timeline of the technological evolution of control technology from classical feedback control towards digital control to wired networked control and finally wireless networked control is depicted in Figure 2.1.

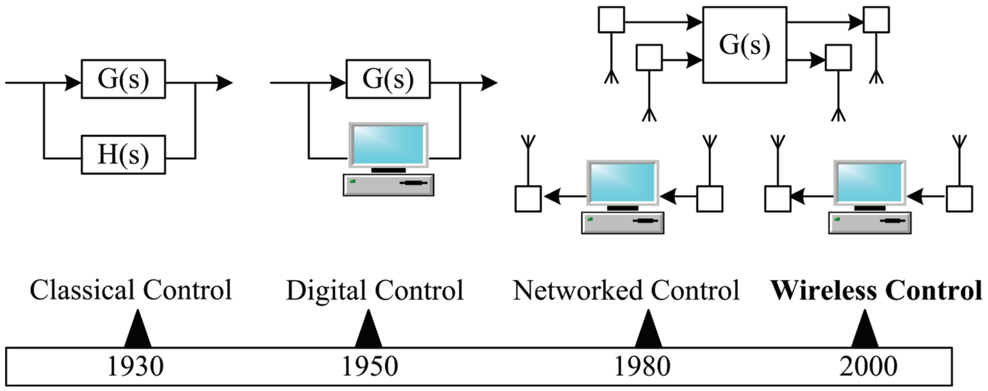


Figure 2.1: Timeline of the technological evolution from classical feedback control to wireless networked control [Baillieul & Antsaklis, 2007]

The use of digital computers as instrumentation for feedback control found large interest after 1950 and led to the transition from continuous-time/continuous-states models used in classical feedback designs to the discrete-time/quantized-states models of digital control. New issues such as sampling period selection, finite word length and the resulting quantization effects, limited computational and memory resources had to be addressed in the design of digital control with a special focus on their effects on the control performance. The foundations of digital control theory are now well established after almost half a century of research and implementation experience and can be found in several textbooks [*see* Aström & Wittenmark, 1990; Franklin *et al.* , 1997, *and references therein*].

The feasibility study of using networked devices to control different functions in passenger cars in 1983 appears to be one of the earliest steps towards modern networked control, which led to the well known Controller Area Network (CAN) communications protocol in 1986. The foundations of networked control systems theory are progressively being established and can already be found in several textbooks [*see* P. Lin & Wang, 2004; Xia & Sun, 2008; Mazumder, 2011; Huang & Nguang, 2009; Hristu-Varsakelis & Levine, 2005; Bemporad *et al.* , 2010, *and references therein*].

The long history of shared data networks started about 30 to 40 years ago with data networks such as the Slotted ALOHA [Stallings, 2007] and the ARPANET [Minoli & Schmidt, 1999]. These networks evolved to widely used modern network protocols like Ethernet and Internet. Since then, several network protocols for industrial control have been released including CAN and Profibus. Profibus is a broadcast bus protocol that operates as a multi master/slave system which was developed by five German institutes in 1987[1]. Other industrial network protocols including Foundation Fieldbus[2] and DeviceNet[3] were also developed. Most of these protocols are reliable and robust for real-time control applications with deterministic MAC protocols, meaning that the transmission delays can be calculated and are usually constant. Moreover, the number of participants in the network is known and limited (i.e., the network load can be known or predicted). The determinism can be enforced even for nondeterministic protocols such as ControlNet by means of additional mechanisms.

In order to meet the demands of control applications, the above mentioned networks are designed such that there are few data losses. Therefore, for this class of networks, the QoS parameters are well known and can be guaranteed. In wireless networks in general, and wireless ad-hoc networks in particular, nodes are mobile and can be added or removed at any time resulting in possible variation of the communication structure over time. Limited energy and computing capabilities of the nodes also result in limited communication bandwidth. The QoS metrics (e.g. average delays and packet losses) of wireless networks are usually worse than those of their wired counterparts with time-varying delays and data losses. But thanks to the development of innovative wireless communication protocols tailored to the specific demands of real-time control applications such as WirelessHART [HART, 2007] or ISA-SP100 [ISA, 2008], wireless control systems are almost everywhere and can be found nowadays at different levels of industrial systems, where the degree of integration depends on the nature and type of the wireless networks used.

2.2 Structures and technologies

2.2.1 General considerations

The continuous-time state-space model of a linear time-invariant (LTI) control system described by equation (2.1) and (2.2) is considered

$$\dot{x}(t) = Ax(t) + Bu(t) + v(t) \tag{2.1}$$
$$y(t) = Cx(t) + w(t) \tag{2.2}$$

where $x \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}^m$ the output vector, $u \in \mathbb{R}^m$ the control vector, $v(t) \in \mathbb{R}^n$ the process noise, $w(t) \in \mathbb{R}^m$ the measurement noise, A, B, and C are matrices of appropriate dimensions. Assuming fast and deterministic processors or microcontrollers with equidistant sampling and actuation leads to the discrete form in equation (2.3) and (2.4)

$$x_{k+1} = \Phi(h)x_k + \Gamma(h)u_k + v_k \tag{2.3}$$
$$y_k = Cx_k + w_k \tag{2.4}$$
$$\Phi(h) = e^{Ah} \tag{2.5}$$
$$\Gamma(h) = \int_0^h e^{As} ds B \tag{2.6}$$

where h is the constant sampling period, $kh, k \in \mathbb{Z}_+$ the sampling time instant (simplified notation k), Φ and Γ are matrices of appropriate dimensions [Aström & Wittenmark, 1990]. The control loop can be closed by state feedback using classical controller design methods such

[1] http://www.profibus.com/
[2] http://www.fieldbus.org/
[3] http://www.odva.org/

as Linear Quadratic Regulator (LQR) and the state feedback control law is given by equation (2.7)

$$u_k = K x_k \tag{2.7}$$

where K is the gain matrix.

It is well known from discrete-time control theory, that the control system specified by equation (2.3), (2.4), and (2.7) behaves correctly when the computational delay introduced by the computing platform is negligible compared to the dynamics of the system, which is the case for some control loops implemented in dedicated processors. However, control loops closed over a wireless network are subject not only to computationally induced time delays, but also to communication delays and data losses. In some cases, these effects are not negligible compared to the system dynamics and may seriously decrease the control performance and even lead to control system instability. Therefore, the type of problems to be solved when analyzing and designing WNCSs depends not only on the selected control loop structures and topologies, but also on the selected wireless communication technologies.

2.2.2 Control loop topologies

A standard single-loop control topology for WNCSs is shown in Figure 2.2. This general and realistic topology, also known as the *direct structure* [Tipsuwan & Chow, 2003], is widely used in applications such as Unmanned Aerial Vehicles (UAVs). The main components are a controlled plant, a sensor that contains an A/D (Analog-to-Digital) converter, an embedded controller, an actuator that contains a D/A (Digital-to-Analog) converter, and a wireless communication network. Sensor, controller, and actuator nodes are all equipped with wireless communication devices such as transceivers and processing units such as processors or microcontrollers.

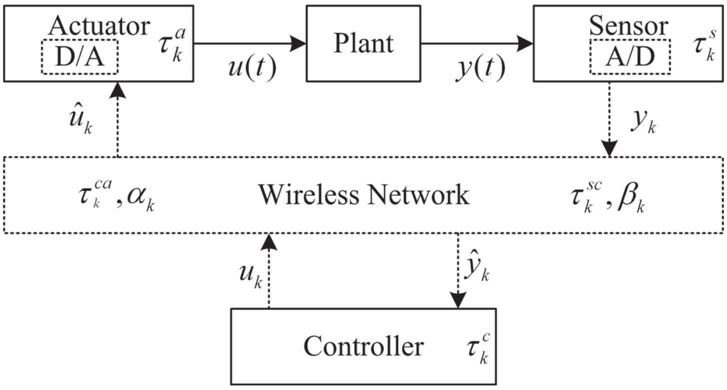


Figure 2.2: WNCS: *direct structure*

The inputs and outputs of the plant are continuous-time signals. The outputs of the plant are transformed into digital signals by the A/D converter at sampling instants k. The sampling period denoted by h is the time between two consecutive sampling instants. The sampling process includes activities such as A/D conversion, codification, and packing of data to be sent to the controller. The resulting computational delay in the sensor is τ_k^s and the communication delay between sensor and controller is τ_k^{sc}. Upon reception of the sequence of sampled data, the controller unpacks and decodes the data, uses a control algorithm to compute the control commands which are codified, packed into new packets, and sent to the actuator, thus introducing a computational delay τ_k^c and a communication delay τ_k^{ca} between controller and actuator. Control commands received by the actuator are unpacked, decoded, and applied to the plant after D/A conversion with a computational delay of τ_k^a. The overall delay τ_k^{rt} from sampling to actuation is given by equation (2.8).

$$\tau_k^{rt} = \tau_k^s + \tau_k^{sc} + \tau_k^c + \tau_k^{ca} + \tau_k^a \tag{2.8}$$

In the above setup, it is assumed that $\tau_k^{rt} \leq h\ \forall k$, i.e., a packet is lost if it is not received by the end of the selected sampling period. It is further assumed that the sensor is time-triggered and both the controller and the actuator are event-driven. That is, the controller node does not compute new control commands unless it receives the sensor measurement and the actuator continues to use the old control command until a new one is received. Under these assumptions, the probability of successful data transmission from the sensor to the controller and from the controller to the actuator is defined as $\rho^{sc} = P[\beta_k = 0]$ and $\rho^{ca} = P[\alpha_k = 0]$ respectively, with $\beta_k, \alpha_k \in \{0, 1\}$. $\beta_k = 1$ if a sensor data packet is lost and $\alpha_k = 1$ if a control data packet is lost. Therefore, the probability of the actuator receiving a packet at the k^{th} sampling instant with $\tau_k^{rt} \leq h$ is defined by equation (2.9).

$$\rho^{rt} = P[\tau_k^{rt} \leq h] = \rho^{sc} \cdot \rho^{ca} \tag{2.9}$$

Network induced delays $(\tau_k^{sc}, \tau_k^{ca})$ and data loss probabilities (ρ^{sc}, ρ^{ca}) may vary depending on the network traffic, medium access protocol, and the chosen node hardware. Computationally induced delays $(\tau_k^s, \tau_k^c, \tau_k^a)$ may also vary depending on the complexity of the implemented control algorithms and the resulting processing load, the scheduling of computational resources between control tasks and communication tasks, etc.

A state-space model formulation for the WNCS control topology in Figure 2.2 on Page 13 is given by equation (2.10), (2.11), and (2.12)

$$\begin{aligned}
x_{k+1} &= \Phi(h)x_k + \Gamma_0\left(h, \tau_k^{rt}\right)\hat{u}_k + \Gamma_1\left(h, \tau_k^{rt}\right)\hat{u}_{k-1} + v_k && (2.10)\\
\hat{y}_k &= \bar{\beta}_k\left(Cx_k + w_k\right) && (2.11)\\
\hat{u}_k &= \bar{\alpha}_k\bar{\beta}_k u_k + (1 - \bar{\alpha}_k\bar{\beta}_k)\hat{u}_{k-1} && (2.12)\\
\Phi\left(h\right) &= e^{Ah} && (2.13)\\
\Gamma_0\left(h, \tau_k^{rt}\right) &= \int_0^{h-\tau_k^{rt}} e^{As} ds B && (2.14)\\
\Gamma_1\left(h, \tau_k^{rt}\right) &= \int_{h-\tau_k^{rt}}^{h} e^{As} ds B && (2.15)
\end{aligned}$$

where $\bar{\beta}_k = 1 - \beta_k$, $\bar{\alpha}_k = 1 - \alpha_k$, $\hat{u}_k$ is the control input at the actuator, $\hat{y}_k$ the sensor input at the controller, Φ, Γ_0, and Γ_1 are matrices of appropriate dimensions [Drew *et al.* , 2005]. Using advanced controller design methods, the control loop can be closed by state feedback and the state feedback control law is given by equation (2.16), where the gain matrix K is a function of the selected sampling period h, the time varying delay τ_k^{rt}, and the data loss probability ρ^{rt}.

$$u_k = K(h, \tau_k^{rt}, \rho^{rt}) \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix} \tag{2.16}$$

Although sensor to controller delays τ_k^{sc} and data losses β_k can be known (e.g. when using time-stamping), the controller to actuator delays τ_k^{ca} or data losses α_k can not be known at the sampling instant k, unless the underlying network is deterministic with known delays and data loss probabilities. One approach to overcome these problems is to model the communication delays and the data losses as probabilistic distributions and to design controllers to account for them [Drew *et al.* , 2005]. Another approach to solve the delay problem is to predict the controller to actuator delays according to the measured sensor to controller delay, i.e., $\tau_k^{sc} = \tau_k^{ca}$ [Yépez *et al.* , 2002].

In the control loop topology in Figure 2.3, also known as the *hierarchical structure*, the remote closed-loop system is connected to a main controller over the wireless network. The main controller periodically computes and sends reference signals in a frame or packet via the wireless network to the remote system. The remote system then processes the reference signal to perform local closed-loop control. The networked control loop usually has a longer sampling period than the local control loop since the remote controller supposes to satisfy the reference signal before processing the newly arrived reference signal. This topology is widely used in applications such as mobile robots [Tipsuwan & Chow, 2002] and wireless information systems for improved fuel efficiency in vehicles [Pettersson & Johansson, 2006]. If the computing delays are negligible with respect to the dynamics of the system, the standard models from equation (2.3), (2.4), and (2.7) are applicable. This is a special case of the control loop topology in Figure 2.2 on Page 13 when $\tau_k = 0$ and $\alpha_k = \beta_k = 0$, $\forall k \in \mathbb{Z}_+$.

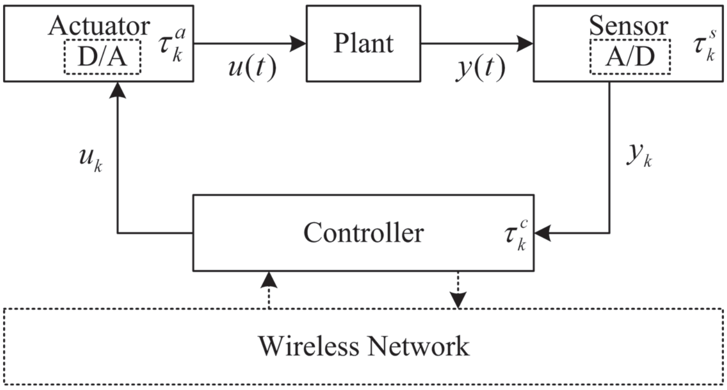


Figure 2.3: WNCS: *hierarchical structure*

The control loop topology in Figure 2.4 is a special case of the general control loop topology in Figure 2.2 on Page 13. The wireless network is located between the sensor and the controller i.e., $\tau_k^{ca} = 0$ and $\alpha_k = 0$, $\forall k \in \mathbb{Z}_+$. In some cases, controller and actuator functionalities are implemented in a single node powered using the actuator power supply. The sensor to controller delays τ_k^{sc} and data losses β_k can be known at sampling instants k using time-stamping, which greatly simplifies the controller design.

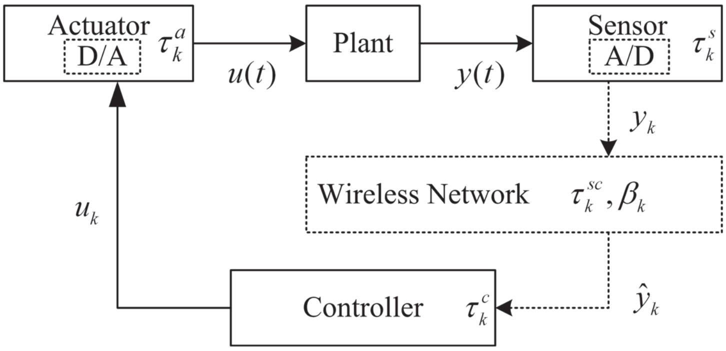


Figure 2.4: WNCS: wireless network between sensor and controller

The control loop topology in Figure 2.5 is also a special case of the general control loop topology in Figure 2.2 on Page 13. The wireless network is only between controller and actuator with $\tau_k^{sc} = 0$, $\beta_k = 0$, $\forall k \in \mathbb{Z}_+$. Usually, sensor and controller functionalities are implemented in a single node. The controller to actuator delays τ_k^{ca} or data losses α_k can not be known at the sampling instant k, unless the underlying network is deterministic with known delays and data loss probabilities, thus making the controller design difficult.

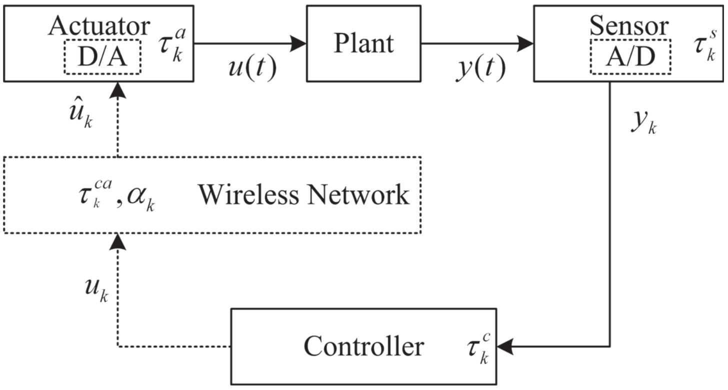


Figure 2.5: WNCS: wireless network between controller and actuator

In general, each component of the WNCS, i.e., sensor, controller, and actuator may be either time-triggered or event-triggered depending on the selected control strategy. This influences not only the sampling process, but also the computation of the control action, the actuation, and the transmission of data. In the event-triggered mode, actions are only performed when an event is generated or detected. An event is generated for example when data is received or when the sensor value crosses a predefined threshold (see [Grüne *et al.* , 2009, *and references therein*]). This mode of operation opens new possibilities such as the reduction of network traffic and the energy consumption of the nodes. The main drawback of this approach is the fact that standard digital control theory is not applicable. Unless otherwise specified, in this thesis a WNCS with time-triggered sensors and event-triggered controller and actuator will be considered.

2.2.3 Communication technologies

The use of a wireless network in the control loop introduces among others latency (e.g. delays) and reliability (e.g. losses) issues, which can be deterministic or random depending on the type of wireless communication technology used and the underlying communication protocols (e.g. MAC protocol, routing protocol). For a selected wireless communication technology, reliability and latency are usually quantified in terms of QoSs. From a control point of view, QoSs may include the average transmission delay, average delay jitter, data loss rate, average network throughput, network access probability, power consumption, data rate, etc. There is no convention as to which criteria are to be included in the QoSs and the choice is usually application specific or designer's choice.

There are a variety of wireless communication technologies designed to mitigate performance, reliability, latency, power consumption, and security issues. Each technology is designed to satisfy specific application requirements. However, there is no wireless communication that can mitigate all the problems simultaneously and there exist only solutions that perform the right tradeoffs with specific applications in mind. Some of these solutions are standard and some proprietary. Standard communication technologies are designed to interoperate, but different implementations may have slight differences whereas non-standard technologies might be adopted as de facto standards.

Many MAC and routing protocols have been proposed for a variety of applications both in academia and industry over the last years. MAC protocols can be classified into three categories based on the medium access mechanism:

- *Schedule-based* [Sohrabi *et al.* , 2000; Hoesel & Havinga, 2004]: In schedule-based MAC protocols, nodes only wake up and listen to the channel in assigned time slots and then go back to sleep in other slots. Knowledge of the network topology is required to establish a schedule that allows nodes to access the channel and communicate with other nodes without intefering with other transmissions. Once the schedule is set up, there are no collisions, no overhearing, and minimized idle listening, resulting in bounded latency, fairness, and good throughput in high loaded traffic conditions. However, slot assignment algorithms are complex due to the lack of central access point. In addition, global or local time synchronization is required not only during the initial network setup phase, but also during the runtime of the network to eliminate clock drifting. TDMA is a representative scheme of scheduled-based MAC protocols.

- *Contention-based* [El-Hoiydi & Decotignie, 2004; Buettner *et al.* , 2006]: In contention-based MAC protocols, nodes randomly wake up to compete for the use of the wireless medium and the winner is allowed to access the medium. Neither topology knowledge nor global time synchronization is required. A node having data to transmit first senses the channel and if it is clear, it starts transmitting. Otherwise, the transmission is postponed to avoid interfering with the ongoing transmission. Contention-based MAC protocols do not rely on a central entity and are robust to node mobility, thus suitable for networks with mobility and dynamicity. Unfortunately, traditional contention-based MAC protocols are not directly applicable for most WNCS applications due to poor energy efficiency, random delays and packet losses, and high traffic load due to contention. CSMA/CA is a representative scheme of contention-based MAC protocols.

- *Hybrid-based* [W. Ye & Estrin, 2004; P. Lin & Wang, 2004]: Hybrid-based MAC protocols combine the advantages of both a random access with contention-based MAC and a deterministic access with schedule-based MAC. A reservation scheme with contention is typically used, where users contend during a reservation period, and those who succeed in this contention transmit without experiencing interference. Hybrid-based MAC protocols offer flexible QoSs to several classes of WNCS applications rather than just maximizing the throughput of the network. One of the main challenges besides understanding the coexistence of the stochastic behavior of the contention-based MAC and the deterministic behavior of the schedule-based MAC is the choice of optimal parameters (e.g. idle listening, collisions).

Routing protocols can be designed to extend the network lifetime while guaranteeing reliable data communication by employing energy management techniques. Based on the underlying network topology, relevant routing protocols can be classified as follows:

- *Topology-based* [Gnawali *et al.* , 2009; Moeller *et al.* , 2010]: In *link-state* protocols, each node first collects the neighborhood information and then floods the local topology information to the network. Thus, each node of the network has global information of the network topology. Each node uses its information to compute a routing table containing, for each destination node, an entry identifying the next hop along the shortest path according to some metric. In *distance-vector* protocols, each node exchanges and has information on what routes its neighbors have rather than a map of the network topology. Accordingly, each node knows the union of its neighbor's routes and a route to itself. While link-state protocols typically present faster convergence and less overhead, distance-vector protocols are simpler, require smaller storage and less computational capability.

- *Hierarchical-based* [Heinzelman *et al.* , 2000; Manjeshwar & Agrawal, 2002]: In hierarchical routing protocols, nodes are clustered and the one with the higher residual energy is usually chosen as the cluster head. These nodes as the leaders of their groups have some responsibilities like collecting and aggregating the data from their respective clusters and transmitting the aggregated data to the base station. Hierarchical routing lowers the energy consumption within a cluster by performing data aggregation and fusion to decrease the number of transmitted messages. Furthermore, it allows the WNCS to cope with additional load and to cover a large area of interest without degrading the service.

- *Location-based* [Xu *et al.* , 2001; Park *et al.* , 2011]: In location-based routing protocols, each node knows its own and its neighbor's positions and the source of a data is informed about the position of the destination for the energy efficient routing paths. The location information is needed to calculate the distance between two particular nodes so that energy consumption can be estimated.

In the following, some of the control application relevant protocol standards will be reviewed. Figure 2.6 shows an excerpt of available wireless communication protocol and radio standards.

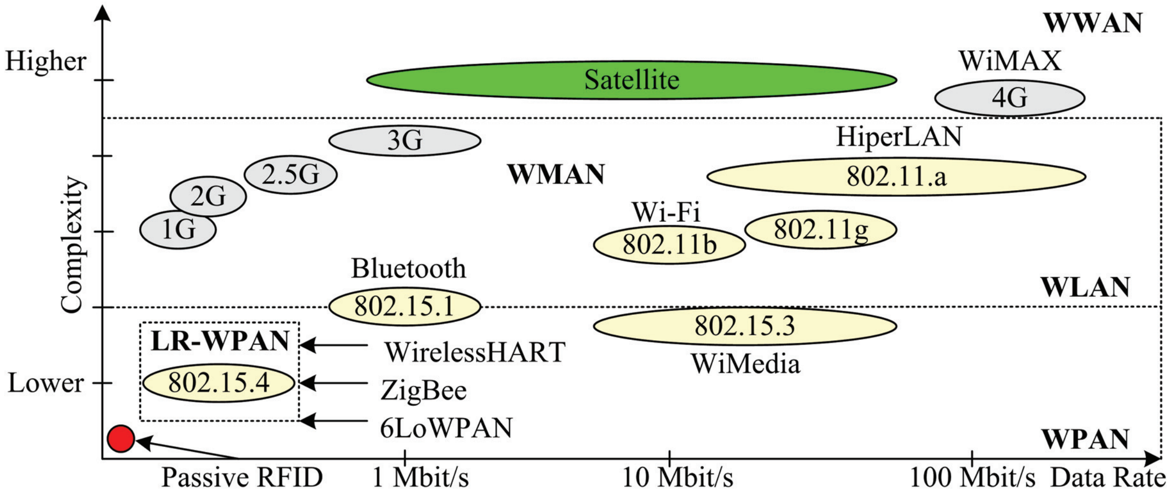


Figure 2.6: Wireless protocols and radio standards [Emerson, 2008]

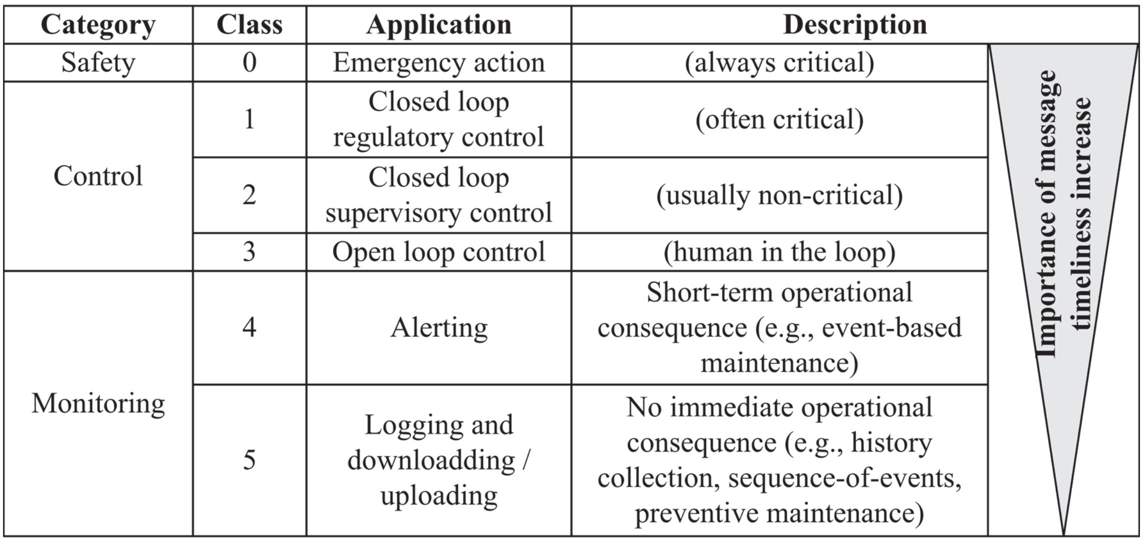
- IEEE 802.15.4 [IEEE, 2006]: This standard specifies the physical layer and the media access control for low-rate wireless personal area networks (LR-WPANs). It is the basis for ZigBee, WirelessHART, and 6LoWPAN specification, each of which further attempts to offer complete networking solution by developing the upper layers which are not covered by the standard. This standard focuses on low-cost, low-power, and low-speed ubiquitous communication between nearby devices with little to no underlying infrastructure. The basic radio range is 10 m with a maximum transfer rate of 250 kbit/s. Important features include real-time suitability by reservation of guaranteed time slots, collision avoidance through CSMA/CA and integrated support for secure communications. Devices also include power management functions such as link quality and energy detection.

- ZigBee [ZigBee, 2005]: ZigBee is defined to work on top of a modified version of the IEEE 802.15.4 standard and covers the networking layer and the application layer. ZigBee is a low-cost, low-power, wireless mesh networking technology which is widely used in wireless control and monitoring applications. ZigBee is intended to be simpler and less expensive than other WPANs such as Bluetooth. The low-power feature allows for extended life of small batteries. ZigBee allows the creation of star networks, tree networks, and particularly mesh networks, which provide high reliability and more extensive range. IEEE 802.15.4 radio and ZigBee emerge as the preferred choice for industrial and smart building applications.

- WirelessHART [HART, 2007]: WirelessHART was designed to complement the wired HART protocol in industrial environments. At the very bottom, it adopts IEEE 802.15.4-2006 as the physical layer. On top of that, WirelessHART defines its own time-synchronized

MAC layer. Some features of the WirelessHART MAC include strict $10ms$ time slot with a TDMA-based link layer, network wide time synchronization, channel hopping, channel blacklisting, and industry standard AES-128 ciphers and keys. The network layer supports self-organizing and self-healing mesh networking techniques. In this way, messages can be routed around interferences and obstacles. WirelessHART uses a central network manager responsible for maintaining up-to-date routes and communication schedules for the network, thus guaranteeing the network performance.

- ISA-SP100 [ISA, 2008]: ISA-SP100 is currently working on a series of standards addressing the adoption of wireless technologies in different industries (see Figure 2.7). ISA-SP100.11a addresses noncritical process applications that can tolerate delays up to 100 ms. Since it leverages the IEEE 802.15.4 standard, it inherits some of its properties such as low-rates (up to 250 kbit/s) and low implementation complexity for simple end devices. In addition, a data-link layer and an adaptation layer between MAC and data link layer are introduced. The data link layer controls the frequency hopping over the 16 channels offered by IEEE 802.15.4 in the 2.4 GHz ISM band and adds a TDMA scheme. Other functionalities include channel blacklisting, adaptive frequency hopping, and flow control. Finally, ISA-SP100 supports multi-hop network with mesh topology.



Category	Class	Application	Description
Safety	0	Emergency action	(always critical)
Control	1	Closed loop regulatory control	(often critical)
	2	Closed loop supervisory control	(usually non-critical)
	3	Open loop control	(human in the loop)
Monitoring	4	Alerting	Short-term operational consequence (e.g., event-based maintenance)
	5	Logging and downloadding / uploading	No immediate operational consequence (e.g., history collection, sequence-of-events, preventive maintenance)

Figure 2.7: ISA-SP100 definition of applications needs of industrial process [ISA, 2008]

2.3 Typical problems

2.3.1 Bandwidth limitations

The capacity of a wireless digital communication channel to carry information is the bandwidth and the rate at which digital data flows through a communication system (i.e. data rate) can be shown to be a measure of the communication system's bandwidth. The bandwidth requirement of WNCSs must not exceed the maximum available communication bandwidth, which is possibly shared among different systems and limited by physical bounds. Inspired by Shannon's theorem on the *maximum bit rate* that a communication channel can carry reliably, the problem of determining the minimum bit rate (i.e. the minimum communication bandwidth) that is needed to stabilize a linear system through feedback over a finite capacity channel has been studied in [Elia & Mitter, 2001] and [Wong & Brockett, 1999]. For a given wireless communication network, the data rate should be considered together with the packet or frame size and overhead if data are encapsulated into packets or frames. However, the packet size depends for example on the resolution of sensor values (i.e. quantization) whereas the header size depends on the protocol design of the wireless communication network. Both the resolution of sensor values and the size of the headers are of great importance, especially for WNCSs with low data rates.

2.3.2 Time delays

Figure 2.8 illustrates the computational and communication delays of a typical WNCS. The four layer communication model consists of a physical layer (PHY), a medium access control layer (MAC), a routing layer (ROU), and an application layer (CON) [Lian *et al.* , 2001].

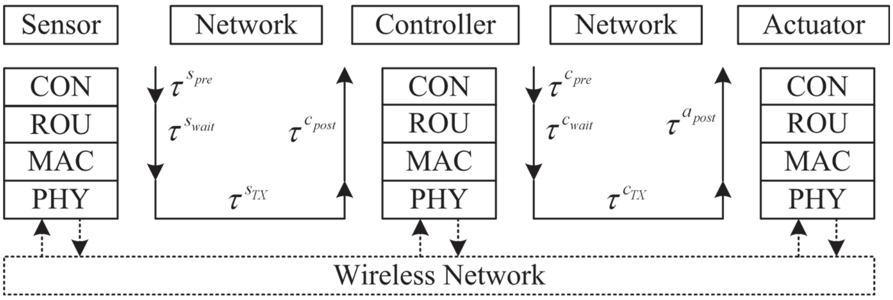


Figure 2.8: WNCS computational and communication delays

The total time delay between two nodes can be broken down into at least three parts:

1. *Time delay at the source node*: The time delay at the source node includes the preprocessing time τ^{pre} (sum of the computation time τ^{comp} and the encoding time τ^{code}), and the waiting time τ^{wait} (sum of the queue time τ^{queue} and the blocking time τ^{block}). The waiting time τ^{wait} may be significant depending on the amount of data the source must send and the traffic on the network. The blocking time τ^{block}, which is the time data must wait once a node is ready to send it, depends on the network MAC protocol and is a major factor in the determinism and performance of a WNCS. It includes waiting time while other nodes are sending data and the time needed to resend the data if a collision occurs, i.e., if acknowledgements are used. The queue time τ^{queue}, which depends on the routing protocol, is the time data must wait in the buffers of the routers or switches before being forwarded.

2. *Time delay on the wireless network channel*: The time delay on the wireless network channel includes the total transmission time of data τ^{data} and the propagation delay of the network τ^{prop}. The total transmission time of data depends on the size of the data, the overhead, and the data rate. The propagation time τ^{prop} is a function of the transmission speed and the distance between source and destination nodes.

3. *Time delay at the destination node*: The time delay at the destination node is the sum of the decoding time τ^{code} and the computation time τ^{comp}.

The total time delay between the sensor node and the controller node in equation (2.8) can be explicitly expressed by

$$\tau^{sc} = \underbrace{\tau^{s_{pre}} + \tau^{s_{wait}}}_{\text{Sensor}} + \underbrace{\tau^{s_{TX}}}_{\text{Network}} + \underbrace{\tau^{c_{post}}}_{\text{Controller}} \tag{2.17}$$

$$\tau^{sc} = \underbrace{\tau^{s_{comp}} + \tau^{s_{code}}}_{\tau^{s_{pre}}} + \underbrace{\tau^{s_{queue}} + \tau^{s_{block}}}_{\tau^{s_{wait}}} + \underbrace{\tau^{s_{data}} + \tau^{s_{prop}}}_{\tau^{s_{TX}}} + \underbrace{\tau^{c_{code}} + \tau^{c_{comp}}}_{\tau^{c_{post}}}. \tag{2.18}$$

Similarly, the total time delay between the controller node and the actuator node can be expressed by

$$\tau^{ca} = \underbrace{\tau^{c_{pre}} + \tau^{c_{wait}}}_{\text{Controller}} + \underbrace{\tau^{c_{TX}}}_{\text{Network}} + \underbrace{\tau^{a_{post}}}_{\text{Actuator}} \tag{2.19}$$

$$\tau^{ca} = \underbrace{\tau^{c_{comp}} + \tau^{c_{code}}}_{\tau^{c_{pre}}} + \underbrace{\tau^{c_{queue}} + \tau^{c_{block}}}_{\tau^{c_{wait}}} + \underbrace{\tau^{c_{data}} + \tau^{c_{prop}}}_{\tau^{c_{TX}}} + \underbrace{\tau^{a_{code}} + \tau^{a_{comp}}}_{\tau^{a_{post}}}. \tag{2.20}$$

In wireless networks with *scheduled services*, control and sensor data are transmitted in a periodic order with deterministic behaviors. Hence, the delays are periodic and can simply be modeled as a periodic function such that $\tau_k^{sc} = \tau_{k+1}^{sc}$ and $\tau_k^{ca} = \tau_{k+1}^{ca}$, $\forall k \in \mathbb{Z}_+$. The models work perfectly for the ideal case. In practice however, WNCSs may experience small variations on periodic delays due for example to discrepancies in clock generators. Wireless networks with *random services* are more involved with uncertain delays. The significant parts of the random delays are the waiting time delays due to data collisions and queuing on the network. There has been a lot of research effort to design controllers and estimators that are robust and adaptive to network induced delays (see section 2.4 on Page 22). Some approaches model network delays using various formulations based on probability and the characteristics of sources and destinations. The techniques range from simple approaches such as time stuffing to transform random delays into constant delays to the Poisson process or more sophisticated approaches such as Markov chains.

2.3.3 Losses and quasi losses

Data losses result from node failures, communication errors at the physical layer of the wireless link, or buffer overflows in switches or routers due to the congestion of the network. Longer forward delays result in data reordering, which essentially amounts to a quasi data loss if the receiver discards outdated arrivals. Some reliable transmission protocols guarantee the eventual delivery of data by using an acknowledgement mechanism. However, these protocols may not be appropriate for WNCSs since repeated retransmissions of old data is generally not useful for control applications. Quasi data losses also occur when one or more bits are corrupted in received data.

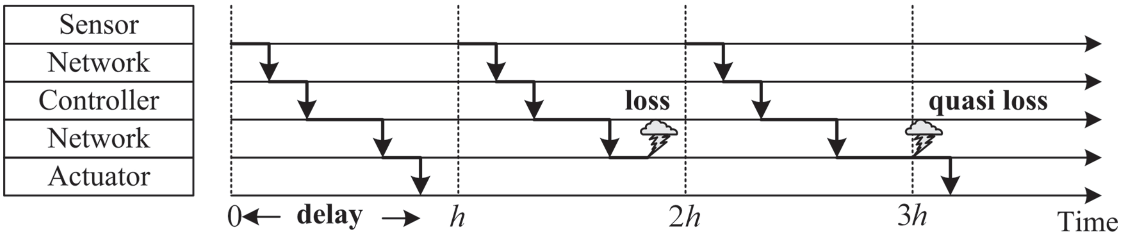


Figure 2.9: Illustration of delay, data loss and quasi loss in a WNCS

Data losses and quasi losses are illustrated in Figure 2.9. Within the first sampling period $(0, h)$, both sensor and controller data are successfully transmitted to their respective destinations. Within the second sampling period $(h, 2h)$, the sensor data is successfully transmitted to the controller but the controller data is lost. Within the third sampling period $(2h, 3h)$, the sensor data is successfully transmitted to the controller but the controller data arrives at the actuator only during the fourth sampling period $(3h, 4h)$ and is discarded.

There has been much research effort to design controllers and estimators that are adaptive and robust to data losses (see section 2.4 on Page 22). Some approaches model the data loss as a Bernoulli random process or as a random process with deterministic rates. The effects of data losses on the control performance are comparable to that of delays which degrade the control performance and may even lead to control system instability.

2.3.4 Limited computing and energy resources

Most of the currently available wireless sensor nodes (e.g. Mica2, MicaZ, Telos, Imote2) are battery powered with limited memory resources and computing (e.g. CPU time) capabilities. The limited memory, computing, and energy resources should be taken into consideration when designing WNCSs, especially when control algorithms and communication protocols are to be implemented on the same hardware platform. These resource limitations affect not only the control and communication performance but also the network lifetime. Hence, advanced model-based control strategies such as Internal Model Control (IMC) or Model Predictive Control (MPC) should not be considered since they require much computational and memory resources. One possibility of overcoming these limitations is to implement these controllers on a powerful computer such as a PC or to use wireless network devices with enough memory and computing resources connected to a power supply (e.g. actuator power supply).

On the other hand, communication protocols should be kept simple and computationally light in order to consume as less memory and computing resources as possible. Some existing wireless communication protocols are designed for byte-stream or even bit-stream radios with battery powered wireless nodes. However, there is a shift to packet-based radios used in wireless sensor nodes. With a packet-based radio, the transmission of a packet is done by radio chip because the microcontroller or processor cannot afford the overhead of these operations and has no control on each single transmitted bit. The microcontroller or processor sends data to the radio that transforms it into packets which are then sent over the wireless network. Therefore, if the communication protocols are designed for byte-stream or even bit-stream radios, they require a refinement for packet-based radios.

2.4 Design approaches

2.4.1 Overview

There has been a lot of extensive research in the field of networked control in general and wireless networked control in particular over the recent years, resulting in a tremendous amount of publications covering almost all of the problems mentioned so far. It is almost impossible to list all publications but a summary of recent results can be found in the surveys [Baillieul & Antsaklis, 2007; Hespanha *et al.* , 2007; Mazumder, 2011; Bemporad *et al.* , 2010; P. Lin & Wang, 2004; Yang, 2006; Hristu-Varsakelis & Levine, 2005; Walsh *et al.* , 2002; Zhang *et al.* , 2001; Kim *et al.* , 1996; Yuan *et al.* , 2007; Tipsuwan & Chow, 2003; Huang & Nguang, 2009; Hokayem & Abdallah, 2004].

Several methods on how to achieve optimal control performance and stability over a wireless communication network have been proposed and can be grouped in three main categories:

1. *Control system design*: This approach is aimed at designing control algorithms that compensate for communication imperfections. There has been a lot of research effort to design controllers and estimators that are adaptive and robust to communication faults such as data rate limitations, data losses, communication delays, and jitters. The control application has no influence on the communication system and can only adapt to its QoS variations using degrees of freedom at the control layer. The main idea is as follows: first collect the distributions of delay and data loss through theoretical and/or experimental analysis of the temporal behavior of the wireless networks with specified system structure and communication protocols, or just simply assume that the distributions of delay and data loss are known a priori; and then use appropriate control techniques such as stochastic optimal

control and hybrid control to design control algorithms robust to these delays, data losses, and jitters.

2. *Communication system design*: In this approach, the controller is designed based on guaranteed QoSs from the communication layers. New communication protocols are then designed or existing communication protocols modified to provide guaranteed QoSs to the control application and therefore, achieve optimal control performance for a selected control application with known or desired QoC (e.g. stability, robustness). In general, the resulting QoSs are known and fixed so that the WNCS designer can only use degrees of freedom of the control application (e.g. sampling period) to further improve the control performance. The general idea is to properly design network structures, communication protocols, bandwidth allocation algorithms, and MAC schemes to reduce network-induced delay, data loss, and jitter as much as possible.

3. *Control and communication co-design*: This approach is about the integrated design of control algorithms and communication protocols. The idea is to design control algorithms that are adaptive and robust to communication faults and flexible communication protocols with adjustable QoSs. The WNCS designer can use degrees of freedom of the control algorithms and the communication protocols to improve the control performance. Unfortunately, there has been less research effort in this area over the past years partly due to the complex interaction when integrating principles and methods from control theory, communication technology, and computer science.

2.4.2 Controller design approaches

2.4.2.1 Sampling and delay

The general state-space model for the WNCS in Figure 2.2 on Page 13 given by equations (2.10)-(2.16) can be reformulated considering time-varying sampling periods h_k and delays τ_k^{rt}. For simplicity, the notation τ_k will be used for τ_k^{rt} and ρ for ρ^{rt} in the sequel. It is assumed that there are no packet losses, i.e., $\alpha_k = \beta_k = 0$ and $\rho = \rho^{sc} = \rho^{ca} = 1$, $\forall k \in \mathbb{Z}_+$. Using the augmented state vector $z_k = [x_k^T, \hat{u}_{k-1}^T]^T$ leads to

$$z_{k+1} = \Psi\left(h_k, \tau_k\right) z_k, \ \forall k \in \mathbb{Z}_+ \tag{2.21}$$

$$\Psi\left(h_k, \tau_k\right) = \begin{bmatrix} \Phi\left(h_k\right) + \Gamma_0\left(h_k, \tau_k\right) K\left(h_k, \tau_k\right) & \Gamma_1\left(h_k, \tau_k\right) \\ K\left(h_k, \tau_k\right) & 0 \end{bmatrix} \tag{2.22}$$

$$h_k = t_{k+1} - t_k, \ \forall k \in \mathbb{Z}_+ \tag{2.23}$$

where t_k is the time at sampling instant k and equation (2.16) becomes $\hat{u}_k = K\left(h_k, \tau_k\right) x_k$ with appropriate dimension for K. When $\tau_k = 0, \forall k \in \mathbb{Z}_+$ (i.e. no delays), equation (2.5) simplifies to

$$\Psi\left(h_k\right) = \Phi\left(h_k\right) + \Gamma_0\left(h_k\right) K\left(h_k\right). \tag{2.24}$$

There has been a lot of research effort to design stabilizing controllers $K\left(h_k, \tau_k\right)$ for the open-loop unstable system (2.10). Most of these works are related to finding the upper bounds on h_k, $\forall k \in \mathbb{Z}_+$ for which stability can be guaranteed. These upper bounds are referred to as the Maximum Allowable Transfer Interval (MATI) [Zhang *et al.* , 2001].

Periodic sampling and constant delay: The discrete-time system (2.4) is time invariant for periodic sampling and constant delay. Stabilizing controllers $K\left(h, \tau\right)$ can be obtained by means of the following theorem:

Theorem 1 (Schur stability [Branicky *et al.* , 2000])*:* Assuming that there exist constants $h > \tau \geq 0$ such that

$$h_k = h, \ \tau_k = \tau, \ \forall k \in \mathbb{Z}_+$$

the WNCS (2.10)-(2.16) is exponentially stable if and only if $\Phi(h, \tau)$ is Schur, i.e., all its eigenvalues have magnitude strictly less than one. Zhang *et al.* [Zhang *et al.* , 2001] showed that the Schurness of $\Phi(h, \tau)$ is a necessary and sufficient condition for the time-invariant case.

Queuing mechanisms are sometimes used to convert random delays into deterministic delays, which are easier to model and compensate. Variable delays are equalized by using buffers to hold packets so that all packets appear to have the same delay [Luck & Ray, 1990]. However, the drawback of this method is that all packets will appear to have a delay as large as the worst case network delay, resulting in control performance degradation.

Periodic sampling and variable delay: It is assumed that the sampling intervals are constant and equal to h and the delay takes values $\tau_k = nh/N$ where $n \in \{0, 1, ..., D_{max}\}$ and $D_{max} \leq N \in \mathbb{Z}_+$. This situation happens when computational and transmission delays are negligible and only access delays due to MAC protocols are taken into consideration [Lin & Antsaklis, 2004; Lin *et al.* , 2003]. The closed-loop system (2.21) can be written as a discrete-time switched system with $D_{max} + 1$ modes as follows:

$$z_{k+1} = A_{\sigma(n)} z_k, \ \forall k \in \mathbb{Z}_+, \ n \in \{0, 1, ..., D_{max}\}$$

where the switching signal $\sigma_{(n)}$ takes values from $\{0, 1, ..., D_{max}\}$ at each time step and, when $\sigma_{(n)} = n$,

$$A_n := \Phi\left(h, n\frac{h}{N}\right), \ n \in \{0, 1, ..., D_{max}\}.$$

Stabilizing controllers $K(h, \tau_k)$ for each of the $D_{max} + 1$ modes can be designed based on the Schurness of the matrix A_n. It is assumed in [Lin *et al.* , 2003] that for the case of no delay or small delays $n \leq N_0$, the corresponding state matrix A_n is Schur stable, while for the case of large delays $n > N_0$, A_n is not Schur stable. Conditions based on *Average Dwell Time* (ADT)[4] such that the stability of the closed-loop system (2.4) is guaranteed are provided in [Zhai *et al.* , 2002].

Remark 1*:* One packet dropout can be modeled as an extra mode where $n = N$. These results are extended for the case of consecutive packet dropouts in [Lin *et al.* , 2003].

Variable sampling and delay: The closed-loop system (2.21) is not time-invariant when the network delay or the sampling interval is variable. In this case, a Lyapunov-based argument is needed to prove control system stability and derive optimal controllers $K(h_k, \tau_k)$. The following theorem gives a sufficient condition for $V(z) := z^T P z$ to be a Lyapunov function for (2.4), from which stability can be deduced.

Theorem 2 (Lyapunov stability [Hespanha *et al.* , 2007])*:* Assume that there exist constants h_{min}, h_{max}, τ_{min}, τ_{max} such that

$$\begin{aligned} &0 \leq h_{min} \leq h_k \leq h_{max} \\ &0 \leq \tau_{min} \leq \tau_k \leq \tau_{max}, \ \forall k \in \mathbb{R}. \end{aligned}$$

The WNCS (2.10)-(2.16) is exponentially stable if there exist a symmetric matrix $P > 0$ such that

$$\Phi(h, \tau)^T P \Phi(h, \tau) - P < 0, \ \forall h \in [h_{min}, h_{max}], \ \tau \in [\tau_{min}, \tau_{max}]. \tag{2.25}$$

[4] Average switching time between two consecutive modes

Finding a matrix P that satisfies (2.25) for all values of h and τ is generally not simple. However a finite set of easy to solve LMIs can be obtained when testing the existence of the matrix P that satisfies (2.25) for values of h and τ on a *finite grid*. A randomized algorithm to find the largest value of h_{max} (MATI) for which stability can be guaranteed when $h_{min} = \tau_{min} = \tau_{max} = 0$ is proposed in [Zang & M. S. Branicky, 2001; Montestruque & Antsaklis, 2003b,a].

2.4.2.2 Packet dropouts

Once again the WNCS in Figure 2.2 (see Page 13) described by equations (2.10)-(2.16) is considered and equation (2.12) can be rewritten as follows:

$$\begin{aligned} \hat{u}_k &= (1-\theta_k)u_k + \theta_k \hat{u}_{k-1} && (2.26)\\ &= \begin{cases} u_k & \theta_k = 0 \text{ (no packet dropout)} \\ \hat{u}_{k-1} & \theta_k = 1 \text{ (packet dropout)} \end{cases}, \ \forall k \in \mathbb{Z}_+ && (2.27) \end{aligned}$$

where $\theta_k = \alpha_k \beta_k$. When packets are dropped (either sensor or controller packet), the actuator uses the previous control command until a new one arrives at the actuator (zero-order hold). In some approaches, the control value is reset to zero in case of a packet dropout, i.e., $\hat{u}_k = 0$ when $\theta_k = 1$ [Hadjicostis & Touri, 2002]. Assuming periodic sampling as well as constant delay and defining the augmented state vector $z_k = [x_k^T, \hat{u}_{k-1}^T]^T$, it can be concluded that

$$\begin{aligned} z_{k+1} &= \Psi_{\theta_k} z_k, \forall k \in \mathbb{Z}_+ && (2.28)\\ \Psi_\theta &= \begin{bmatrix} \Phi(h) + (1-\theta)\Gamma_0(h,\tau)K & \Gamma_1(h,\tau) + \theta\Gamma_0(h,\tau) \\ (1-\theta)K & \theta I \end{bmatrix}, \forall \theta \in \{0,1\}. && (2.29) \end{aligned}$$

Deterministic dropouts: A deterministic dropout model with packet dropouts occurring at an asymptotic rate defined by the following time average

$$r := \lim_{T\to\infty} \frac{1}{T} \sum_{k=k_0}^{k_0+T-1} \theta_k, \ \forall k_0 \in \mathbb{Z}_+, \qquad (2.30)$$

which implicitly assumes that the limit exists is considered in [Zhang *et al.* , 2001]. The switching signal θ_k in (2.30) takes values in some index set $\{0, 1, ..., N\}$ and the rate at which the event $\theta_k = j$ occurs is defined by the following time average:

$$r_j := \lim_{T\to\infty} \frac{1}{T} \sum_{k=k_0}^{k_0+T-1} \delta_j \theta_k, \ \forall k_0 \in \mathbb{Z}_+, \ j \in \{0, 1, ..., N\} \qquad (2.31)$$

where $\delta_j\theta_k = 1$ when $\theta_k = j$ and zero otherwise. When all limits exist, then $\sum_{j=0}^{N} r_j = 1$. A quadratic Lyapunov function of the form $V(z) := z^T P z$ can be used to establish the asymptotic stability of system (2.28) with rates (2.31) and derive optimal controllers K_θ.

Theorem 3 ([Hassibi *et al.* , 1999])*:* Assume that there exist a symmetric matrix $P > 0$ and scalars $\alpha, \alpha_0, \alpha_1, ..., \alpha_n$ such that

$$\alpha_0^{r_0}\alpha_1^{r_1}...\alpha_N^{r_N} > \alpha > 1, \ \Phi_j^T P \Phi_j \le \alpha_j^{-2} P, \ \forall j \in \{0, 1, ..., N\}.$$

The system (2.28) is exponentially stable in the sense that $\lim_{k\to\infty} \alpha_k z_k = 0$ for every sequence $\delta_j\theta_k$ for which (2.31) holds.

Corollary 4 ([Hespanha *et al.* , 2007])*:* Assuming that there exist a symmetric matrix $P > 0$ and scalars $\alpha, \alpha_0, \alpha_1$ such that

$$\alpha_0^{ro}\alpha_1^{1-r} > \alpha > 1,\ \Phi_0^T(h,\tau)\,P\Phi_0(h,\tau) \le \alpha_0^{-2}P,\ \Phi_1^T(h,\tau)\,P\Phi_1(h,\tau) \le \alpha_1^{-2}P \tag{2.32}$$

The system (2.28) is exponentially stable in the sense that $\lim_{k\to\infty}\alpha_k z_k = 0$ for every sequence θ_k for which (2.30) holds. The set of LMIs that appear in (2.32) are bilinear in the unknown P, α_j and therefore generally nonconvex. However, *line search* procedures over the scalars α_0, α_1 can be used to determine the feasibility of (2.32).

Stochastic dropouts: If θ_k is a Bernoulli process with probability of dropout (i.e. $\theta_k = 1$) equal to $\rho \in [0, 1)$, then the system (2.28) is a special case of a discrete-time *Markovian jump linear system* (MJLS). In general MJLSs, θ_k in (2.28) would be the state of a discrete-time Markov chain with a finite number of states and a given transition probability matrix. For Bernoulli drops, the Markov chain has only two states, and the transition probability from any state to the dropout state $\theta_k = 1$ is equal to ρ and the transition probability from any state to the state $\theta_k = 0$ is equal to $1 - \rho$, as shown in Figure 2.10.

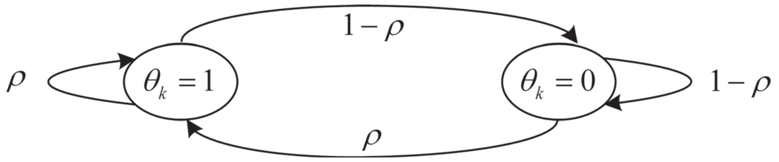


Figure 2.10: Bernoulli drops modeled as two-state discrete-time Markov chain

Following theorem can be used to check the stability of the discrete-time MJLS and obtain optimal controllers K_θ:

Theorem 5 ([Seiler & Sengupta, 2001])*:* The system (2.28) with Bernoulli dropout probability ρ is exponentially mean-square stable (ESS)[5] if there exist a symmetric matrix[6] $Z > 0$ such that

$$\begin{bmatrix} Z & \sqrt{1-\rho}\,(\Phi_0 Z)^T & \sqrt{\rho}\,(\Phi_1 Z)^T \\ * & Z & 0 \\ * & * & Z \end{bmatrix} > 0. \tag{2.33}$$

If the controller and the actuator are collocated as in Figure 2.4 on Page 15, the control law can adapt to the occurrence of packet losses by allowing the controller to use different gains at different sampling instants k, based on the value of $\theta_k \in \{0, 1\}$. In this case, the matrices Φ, Γ_0, Γ_1 in (2.29) depend on θ, but one still gets a system of the form (2.28) and similar tools can be used to analyze its stability and design optimal controller.

2.4.2.3 Approaches combining delays and packet losses

Nilsson *et al.* [Nilsson & Bernhardsson, 1998; Nilsson, 1998; Nilsson & Bernhardsson, 1996, 1997, 1998] consider the two-channel feedback WNCS in Figure 2.2 on Page 13. The corresponding discrete-time state-space model is given by (2.10)-(2.15) where v and w are zero-mean uncorrelated white noise processes. The delays τ_k^{sc} and τ_k^{ca} are assumed to be random and independent variables with known probability distributions and include the processing times τ_k^s and τ_k^c respectively. All sensor packets are assumed to be time stamped, meaning that the controller knows

[5] ESS: if for every initial state (z_0, θ_0), $\lim_{k\to\infty} E\left[\|z(k)\|^2\right] = 0$

[6] Matrix entries denoted by "*" are implicitly defined by the fact that the matrix is symmetric

the delay τ_k^{sc} when the k^{th} measurement arrives. Nilsson *et al.* show that the optimal control u_k that minimizes

$$J_k = E\left\{ x_N^T Q_N x_N + \sum_{j=k}^{N-1} \begin{bmatrix} x_j \\ u_j \end{bmatrix}^T Q \begin{bmatrix} x_j \\ u_j \end{bmatrix} \right\} Q_N \geq 0,\ Q := \begin{bmatrix} Q_{11} & Q_{12} \\ * & Q_{22} \end{bmatrix} \geq 0,\ Q_{22} > 0$$

is of the form

$$u_k = K_k(\tau_k^{sc}) \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix},\ \forall k \in \mathbb{Z}_+. \tag{2.34}$$

The computation of the matrix gain $K_k\left(\tau_k^{sc}\right)$ requires the solution of a backwards-in-time Riccati equation that involves the computation of expectations with respect to the random variables τ_k^{sc} and τ_k^{ca}. These random variables are often correlated in practice since they both depend on the network load, which typically varies at time scales lower than the sampling period h. Nilsson and Bernhardsson [Nilsson & Bernhardsson, 1997] account for this by considering three alternative delay distributions and modeling the transitions between the distributions using a three-state Markov-chain. Each state of the Markov chain would correspond to a particular network load (low, medium, or high). In this case, the optimal control strategy is of the form

$$u_k = K_k(\tau_k^{sc}, r_k) \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix} \tag{2.35}$$

where the matrix gain $K_k\left(\tau_k^{sc}, r_k\right)$ depend on both the delay τ_k^{sc} and the current state r_k of the Markov chain.

The main difficulty in using the optimal controllers (2.34) and (2.35) is the computation of the matrix gains $K_k\left(.\right)$. Usually, when stationary values for these gains exist, they can be computed offline and stored in a lookup table, which is indexed in real-time by the current value of the delay τ_k^{sc} and the network state r_k. Nilsson *et al.* extended the results for the output feedback case and showed that the separation principle holds and that the optimal control can be obtained by replacing x_k in (2.34) an (2.35) by an estimate $\hat{x}$ computed using a time-varying Kalman filter. Using the so called *Vacant Sampling* technique, the effect of a single packet dropout is also considered.

The approach of Nilsson *et al.* is further extended in [Drew *et al.* , 2005] where random delays and packet losses are considered simultaneously. Packet dropouts are modeled as a Bernoulli process, i.e., the sequence $\{\beta_k\}_0^\infty$ is assumed to be a Bernoulli random sequence. An information flow between the actuator and the controller is added so that the delays τ_{k-1}^{ca} and packet dropouts α_{k-1} between the controller and the actuator are known to the controller at the k^{th} time step. The Kalman state estimator used to estimate plant states in the event of packet loss between sensor and controller is of the following form:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k\left[\hat{y}_k - C\hat{x}_{k|k-1}\right] \tag{2.36}$$

$$\hat{x}_{k|k-1} = \Phi\hat{x}_{k-1|k-1} + \Gamma_0\left(\tau_{k-1}\right)\hat{u}_{k-1} + \Gamma_1\left(\tau_{k-1}\right)\hat{u}_{k-2} \tag{2.37}$$

$$K_k = P_{k|k-1}C^T\left(CP_{k|k-1}C^T + R\right)^{-1} \tag{2.38}$$

where $\hat{y}_k$ is the actual measurement, $C\hat{x}_{k|k-1}$ the measurement prediction, L_k the time-varying Kalman filter gain matrix, $P_{k|k-1}$ the error covariance matrix, and R the measurement noise covariance matrix. $P_{k|k-1}$ is updated according to

$$P_{k+1|k} = \Phi P_{k|k-1}\Phi^T + Q - \beta_{k+1}\Phi L_k C P_{k|k-1}\Phi^T \tag{2.39}$$

where Q is the process noise covariance matrix. Nilsson's optimal controller (2.34) can be extended and leads to the following theorem:

Theorem 6 ([Drew *et al.* , 2005])*:* When the probability of receiving control commands at the actuator with $\tau_k \le h$ is ρ (thus, loss or delayed with a probability $1-\rho$), the optimal state feedback controller is given by:

$$u_k^*(\tau_k^{sc}) = K_\rho(\tau_k^{sc}) \begin{bmatrix} x_k \\ \hat{u}_{k-1} \end{bmatrix}, \; 0 < \tau_k \le h, \; \forall k \tag{2.40}$$

where h is the sampling period, $\rho = \rho^{sc} \cdot \rho^{ca}$, and $\tau_k = \tau_k^{sc} + \tau_k^{ca}$, $\forall k$. Computational delays can be neglected or included in communication delays.

The computation of the matrix gains $K_\rho(\tau_k^{sc})$ requires solving a stochastic Riccati equation evolving backwards in time. The iteration involves expectation calculations with respect to τ^{sc} and τ^{ca} which are performed until a stationary value is found. However, if the probability of packet loss ρ is too high, then the iteration will not converge. Sinopoli *et al.* showed that bounds on ρ can be found using MJLS methods [Sinopoli *et al.* , 2004].

Gabel *et al.* [Gabel, 2008] consider the WNCS structure in Figure 2.3 on Page 15 and present an approach for the derivation of QoS-adaptive controllers for Ambient Intelligent (AmI) Systems. In this approach, the parameters of a selected robust controller are adapted online to measured network QoSs by picking suitable controller gains from a lookup table. Network QoSs include delays τ^{sc} and packet dropouts ρ^{sc} which can be deterministic or random. Optimal controller parameters $K(\rho^{sc}, \tau^{sc})$ are derived offline based on heuristics using Genetic Algorithms and saved in a lookup table. The structure of the controller parameter optimization algorithm is shown in Figure 2.11. The simulation controls the interplay between the Genetic Algorithm and the WNCS simulator, sets start signals, and provides QoS parameters to the WNCS simulator. Therefore, knowledge of the minimum and maximum values of the QoS-parameters is required. The interface between the Genetic Algorithm and the WNCS simulator are the controller gains $K(\rho^{sc}, \tau^{sc})$, and the objective function, i.e., the desired QoC. The QoC is given by equation (2.41) where λ is a weighting factor, t_E the end of the observation interval, and $e(t)$ the control error.

$$QoC = \frac{1}{\lambda \int_0^{t_E} |e(t)|\, dt + (1-\lambda) \int_0^{t_E} |e(t)|\, tdt} \tag{2.41}$$

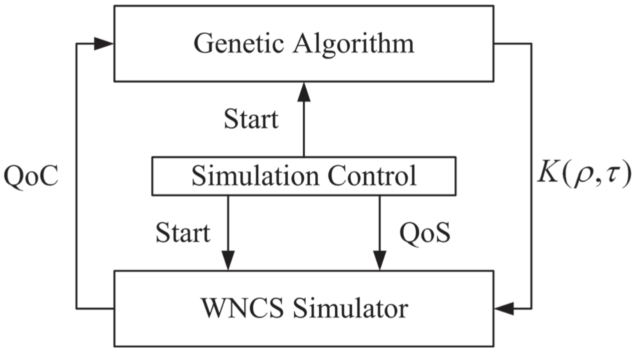


Figure 2.11: Structure of the parameter optimization [Gabel, 2008]

It is assumed that all packets are numbered, sensor and controller clocks are synchronized, and the sampling period h is known to both the sensor and the controller. Therefore, the packet number can be used to measure the QoS-parameters given below:

$$\tau_m^{sc} = t_m^c - pn_m h, \; m = kh \tag{2.42}$$

$$P_L = pn_m - pn_{m-1} - 1, \; \forall pn_m > pn_{m-1} \tag{2.43}$$

$$\rho^{sc} = \frac{\sum P_L}{P_{Total}} * 100 \tag{2.44}$$

where m is the mth successful transmission of a sensor packet to the controller, t_m^c the arrival time of the sensor packet at the controller, pn the sensor packet number, m the mth successful transmission of a sensor packet to the controller, P_L the number of packet losses, ρ^{sc} the packet loss rate, and P_{Total} the total number of transmitted packets during the observation interval t_E. Following heuristics are used for the derivation of a QoS-adaptive controller:

1. Overtaken packets are discarded
2. Controller parameters are adapted to the measured delay $\tau^{sc} \in [\tau_{min}^{sc}, \tau_{max}^{sc}]$
3. The sampling period is adapted to the measured packet losses P_L: $h_m = (P_L + 1)\, h_0$, where h_0 is the nominal sampling period

2.4.3 Communication design approaches

An adaptive IEEE 802.15.4 MAC protocol for control and monitoring applications is studied in [Hernández, 2010]. The adaptive IEEE 802.15.4 MAC protocol with beacon enabled mode is used to control an inverted pendulum. The structure of the inverted pendulum system is shown in Figure 2.12. A slotted CSMA/CA MAC protocol is used with transmissions taking place during the Contention Access Period (CAP). Energy efficient, reliable, and timely packet transmissions are achieved by tuning the MAC parameters *macMinBE*, *macMaxCSMABackoffs*, and *MacFrameRetries*. A generalized Markov chain is proposed to model the influence of these parameters on the probability of successful packet transmission, packet delay, and energy consumption.

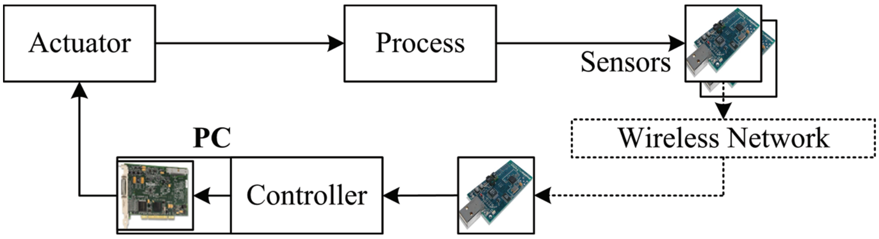


Figure 2.12: Inverted pendulum control system [Hernández, 2010]

In [Park *et al.* , 2011], an adaptive protocol (*Breath*) for industrial control applications using wireless sensor networks (WNS) is proposed. The Breath protocol is designed for the scenario depicted in Figure 2.13 on Page 30 with a plant controlled over a WSN. The outputs of the plant are periodically sampled by the sensors with a total packet generation rate of $\lambda pkt/s$. Plant states information are transmitted to the sink over a multi-hop wireless network of uniformly and randomly distributed relaying nodes. There is no direct communication between plant and sink and relay nodes forward incoming packets to their destinations. Control application requirements are allowed to vary over time, thus, putting constraints on the protocol design in terms of delay and packet loss probability.

The Breath protocol uses optimized random routing at the routing layer, a hybrid MAC protocol based on a CSMA/CA mechanism similar to IEEE 802.15.4 at the data link layer, radio power control at the physical layer, sleeping policies based on a randomized duty-cycling algorithm allowing nodes to enter sleep mode for a random amount of time, thus, saving energy. Each node in the Breath protocol is assumed to have a rough knowledge of its location obtained using the Received Signal Strength Indicator (RSSI), which is usually provided by off-the-shelf wireless devices. Location information are used to tune the transmit radio power and change the number

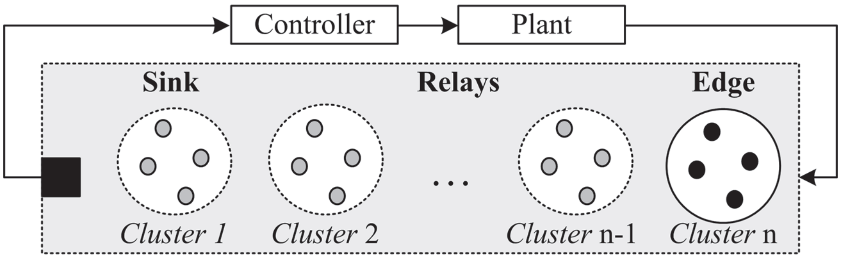


Figure 2.13: Wireless control loop using the Breath protocol [Park *et al.* , 2011]

of hops. Breath ensures a desired packet delivery and delay probabilities while minimizing the energy consumption of the network. The design approach relies on a constrained optimization problem, whereby the objective function is the total energy consumption of the network and the constraints are the packet reliability and delay probabilities.

An energy-efficient protocol for control applications over clustered WSNs that simultaneously embraces timeliness, reliability, energy efficiency, and dynamic adaptation *TREnD* is proposed in [Fischione *et al.* , 2011]. The general scenario for industrial wireless control applications considered in TREnD is similar to the one depicted in Figure 2.13 and assumes that the state of a plant must be monitored at locations where electrical cables are not available or cannot be extended, so that wireless sensor nodes can be used. Uniformly distributed and clustered sensor nodes transmit plant information to the sink node though a multi-hop wireless communication network. TREnD implements hierarchical routing algorithm with a static route at interclusters level and a dynamic routing algorithm at node level. A hybrid TDMA/CSMA protocol is implemented at the MAC layer consisting of a weighted TDMA scheme that regulates channel access among clusters and a p-persistent CSMA/CA scheme which offers flexibility for the introduction of new wireless nodes, robustness to node failures, and support for the random selection of next-hop nodes. TREnD also offers an option for data aggregation to fairly distribute traffic load and energy consumption among clusters.

A prioritized CSMA/CA protocol for real-time wireless local area networking is proposed in [Ye *et al.* , 2001]. The protocol, which is based on the IEEE 802.11 wireless standard, mixes real-time traffic with standard multimedia data in a way that the control loop stability is guaranteed. The proposed protocol uses the idea of the coexistence of the point coordination function (PCF) and distributed coordination function (DCF) in IEEE 802.11 MAC protocol to ensure that time-critical data has higher priority, exploits the degrees of freedom of IEEE 802.11 by choosing a suitable scheme to maintain the polling table and realize priority access to the wireless medium for different data types by using the inter-frame space (IFS) time defined by IEEE 802.11. Several algorithms for dynamically scheduling the traffic of wireless control systems are proposed and validated.

2.4.4 Cross-design approaches

The joint design of control and communication for WNCSs is twofold: the controller design needs to be robust and adaptive to communication faults such as random delays and packet losses, while the network should be designed with the goal of optimizing control performance and minimizing the total energy consumption of the network at the same time. Unfortunately, the cross-design of control and communication has received little attention in the past years.

Liu *et al.* [Liu & Goldsmith, 2004] present a cross-layer design methodology for the joint design of wireless network and distributed controllers. A set of networked controllers where multiple control systems coexist with their control loops closed over a shared wireless communication network that induces random delays and packet losses are considered. With the goal of optimizing the control performance, the four layer cross-design framework in Figure 2.14 is introduced. The design objective is the optimization of the control performance, which imposes implicit tradeoffs on the wireless network design as opposed to the explicit tradeoffs typical in wireless data and voice applications. The design parameters are the sampling period and performance index at the application layer, routing and flow control algorithms at the network layer, contention free and contention based medium access protocols at the MAC layer, and modulation and coding schemes at the physical layer. The optimization of the network performance is impeded by the complex tradeoffs between network throughput, time delay, and packet loss probability. A cross-layer design of the link layer, MAC layer, and sampling period selection is illustrated on a double inverted pendulum system. An iterative procedure for the cross-layer design of wireless control systems is also presented.

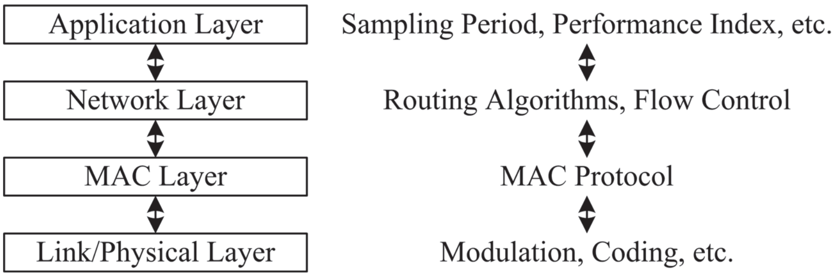


Figure 2.14: Layered structure for WNCSs [Liu & Goldsmith, 2004]

A cross-layer adaptive feedback scheduling scheme for WNCSs is proposed in [Xia & Sun, 2008]. The proposed cross-layer adaptive feedback scheduling takes advantage of the co-design of control algorithms and wireless communication protocols by the exchange of information across the physical layer and the application layer in the presence of noise interference and link transmission rate variations. By controlling the deadline miss ratio of the control loops, the overall control performance can be guaranteed. The layered structure of the cross-design framework is shown in Figure 2.15 and consists of an application layer, a MAC layer similar to IEEE 802.11b, and a physical layer based on IEEE 802.11b. Design parameters of the feedback scheduler (e.g. deadline miss ratio) are adapted to dynamic changes in the capacity of the wireless channel. The cross-design parameters are the sampling period of the control application at the application layer, the parameters of the CSMA/CA protocol at the MAC layer, and the transmission rate as well as the deadline miss ratio at the physical layer.

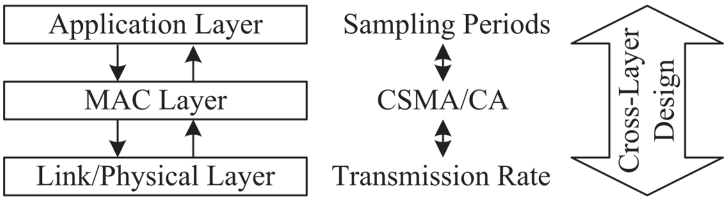


Figure 2.15: Cross-design framework for WNCSs [Xia & Sun, 2008]

A linear system in which several signals are transmitted over a wireless communication channel is considered in [Xiao *et al.* , 2003]. The problem of optimizing the stationary performance of the linear system by jointly allocating resources in the communication network and tuning parameters of the linear system is addressed. The controller synthesis and communication rate allocation is solved jointly with an iterative method. Assuming fixed source coding, channel coding, and MAC scheme for the wireless communication channel, the problem is to find the allocation of communication resources (e.g. transmit powers and bandwidths) that yields optimal performance of the linear system. For a fixed sampling frequency of the linear system, the limit on communication rate translates into a constraint on the number of bits that can be transmitted over each communication channel during one sampling period. The effectiveness of the approach is demonstrated numerically for the control of a mass-spring-system.

2.5 Summary and outcome

The cross-layer approach to the design of WNCSs is an attractive solution. However, the dynamics of the interaction among the communication protocols and the control application is quite complex due to the existence of several parameters and the nonlinear nature of the protocol state machines at different layers. Many of the cross-design solutions proposed in the literature are hard to apply because they require huge processing resources or instantaneous global network knowledge, which are usually out of reach of the capabilities of real wireless nodes. The complex interdependence of decision variables such as the sleeping policy, MAC schemes, routing and power control algorithms, sampling periods, etc., leads to difficult problems even in simple network topologies, where the analytical relation describing packet reception rate, delay, energy consumption, and sampling period may be highly nonlinear expressions.

As a consequence, following main issues will be addressed in this thesis:

- Instead of using the fastest sampling period in order to achieve best control performance, the goal will be to use the slowest possible sampling period in order to reduce the total energy consumption of the network while guaranteeing an acceptable control performance.
- The limited computing and memory resources and the fact that control algorithms and communication protocols are to be implemented on the same hardware platform will prevent the use of complex channel coding and error correction algorithms. Furthermore, complex and computationally extensive control algorithms will not be considered. Instead, simple control algorithms (e.g. PID controllers) will be implemented, whose parameters (e.g. sampling period and controller gains) can be adapted online or offline to measured network QoSs.
- Rather than implementing new communication protocols from scratch, existing communication protocols will be modified to suit the needs of the control application and provide tunable parameters (e.g. TDMA time slots, CSMA/CA backoff exponent, and transmit power) to the control application.
- Computationally extensive optimization problems will be solved offline and the results saved in lookup tables. Online optimization will then only consist of measuring control performance and communication QoSs and then selecting appropriate parameters control and communication parameters from lookup tables.

3 Inverted Pendulum Testbed

3.1 Overview

The inverted pendulum system in Figure 3.1 is used as a testbed for the cross-design of WNCSs throughout this thesis and is briefly presented in this chapter. It consists of a real inverted pendulum plant equipped with a wireless sensor and actuator network, as well as hardware interfaces between the inverted pendulum, the wireless network, and a PC running MATLAB/Simulink. New cross-design approaches of WNCSs can be tested under realistic conditions using the testbed.

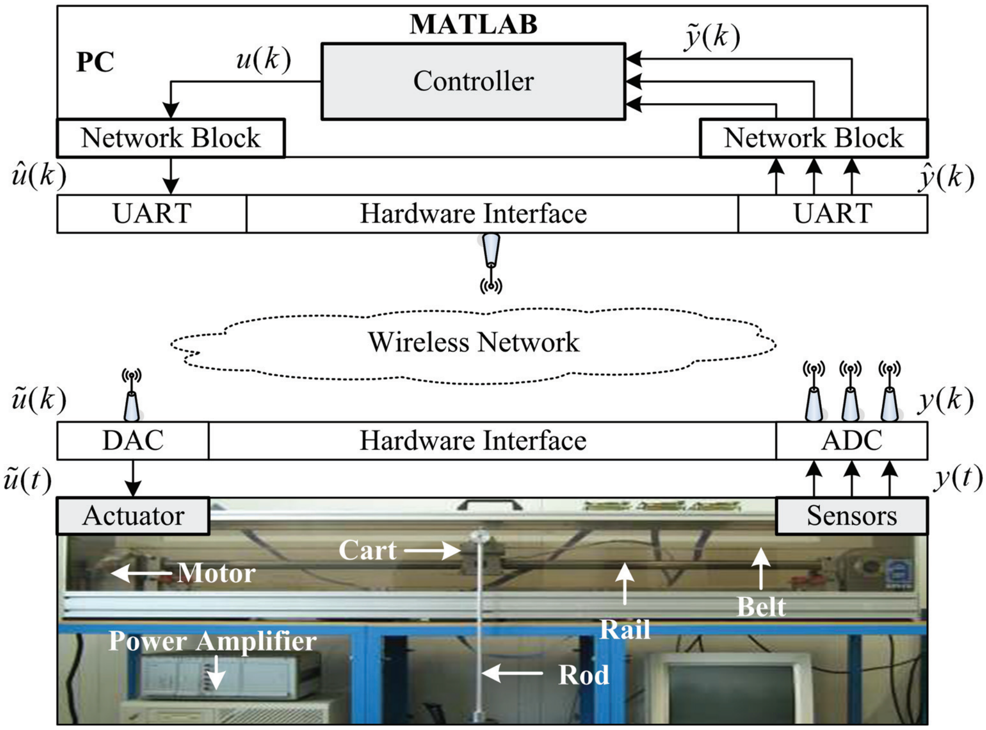


Figure 3.1: Structure of the inverted pendulum testbed

The inverted pendulum outputs $y(t)$ are periodically sampled by the sensor nodes at a sampling period h (sampling time instants $t_k = kh,\ k \in \mathbb{Z}_+$, are replaced by k for simplicity), digitized using the ADC, and sent to the controller over the wireless network. Each sensor node samples only one plant output and single sensor measurements may be delayed or lost when transmitted to the controller over the wireless network. Therefore, only estimates $\hat{y}(k)$ of sampled sensor measurements $y(k)$ are received by the input network block (e.g. $\hat{y}(k) = y(k-\tau)$ for delayed packets and $\hat{y}(k) = y(k-1)$ for lost packets). Received sensor measurement estimates $\hat{y}(k)$ can further be delayed (e.g. $\tilde{y}(k) = \hat{y}(k-\tau)$) or dropped (e.g. $\tilde{y}(k) = \hat{y}(k-1)$) by the input network block in order to obtain desired distributions of delays and packet losses. Upon reception of sensor measurement estimates $\tilde{y}(k)$, the controller computes the control action and sends the control value $u(k)$ to the actuator node over the wireless network. As for the sensor measurements, the network block at the output of the controller can be used to delay of drop control values. Upon reception of control value estimates $\tilde{u}(k)$ and conversion to analogue using the DAC, the corresponding voltage $\tilde{u}(t)$ is applied to the DC motor by the actuator node to control the inverted pendulum.

Control algorithms can be implemented directly in the wireless node or in the PC running MATLAB if more computational power and memory resources are required. If the control algorithm is implemented in MATLAB, then the wireless node sends sensor measurements to the PC using the Universal Asynchronous Receiver Transmitter (UART) connection provided by the interface hardware. Control values are sent back to the wireless node over the same UART connection. As mentioned previously, sensor measurements and control values can be manipulated in MATLAB using the two network blocks in order for example to obtain a desired distribution of delays and data losses when assessing the control performance. In the remaining of this chapter, the components of the inverted pendulum testbed will be briefly presented and described .

3.2 Inverted Pendulum

The inverted pendulum control system in Figure 3.1 on Page 33 is an open-loop unstable plant whose mathematical model is nonlinear. The inverted pendulum has been chosen because it poses high demands to the joint design of control and communication in WNCSs in terms of stability, performance, and bandwidth requirements. It consists of a vertical rod that can rotate around a fixed pivot point on a cart. The corresponding rod angle from the vertical upward position is φ which is measured using a rotational encoder. The cart is free to move along a rail and the corresponding displacement x is measured using a rotational encoder. The cart velocity $\dot{x}$ is measured using a tachometer. There is no sensor available to measure the angular velocity $\dot{\varphi}$ of the pendulum rod. The cart is driven by a DC electric motor controlled by a power amplifier with an input voltage u. The DC motor is coupled to the cart through a transmission belt. A detailed description of the inverted pendulum including mathematical modeling and model parameters is provided in the Appendix.

The inverted pendulum control system is equipped with three sensors and one actuator the characteristics of which are summarized below:

- Rotational encoder to measure the position x of the cart: -67 cm $\leq x \leq$ +67 cm normed to -10 V $\leq x \leq$ +10 V
- Rotational encoder to measure the angle φ of the pendulum rod from the vertical upward position: $-10^\circ \leq \varphi \leq +10^\circ$ normed to -10 V $\leq \varphi \leq$ +10 V
- Tachometer to measure the linear velocity $\dot{x}$ of the cart: -5 m/s $\leq \dot{x} \leq$ +5 m/s normed to -10 V $\leq \dot{x} \leq$ +10 V
- DC motor with input voltage u to drive the cart: -10 V $\leq u \leq$ +10 V

3.3 Wireless nodes

The Crossbow Imote2 [Crossbow, 2011] hardware platform in Figure 3.2 on Page 35 is used as wireless node in the inverted pendulum testbed. The hardware platform of the Imote2 is built around a low-power 32 bit fast Intel XScale$^{®}$ PXA271 processor. It integrates an 802.15.4 radio (ChipCon 2420 [Texas, 2007]) from Texas Instruments and a built in 2.4 GHz antenna. The interface between the processor and the transceiver is at the level of exchanging bytes over a Serial Programming Interface (SPI). This low-level interface has the advantage, that the MAC protocol designer has absolute control, which contrasts sharply with 802.11 WLAN equipment where the MAC is usually included as part of the chipset on a PC card. The Imote2 has a modular stackable platform that can be stacked with interface boards to customize the system to a specific application, along with a battery board to supply power to the system. The processor can operate in a low voltage (0.85 V) and a low frequency (13 MHz) mode, hence enabling low-power operation. The frequency can be scaled to 104 MHz at the lowest voltage level, and can be increased up to 416 MHz with Dynamic Voltage Scaling. The processor has many low-power modes, including sleep and deep sleep modes. The processor integrates many I/O options making it extremely flexible in supporting different sensors, ADCs, radio options, etc.

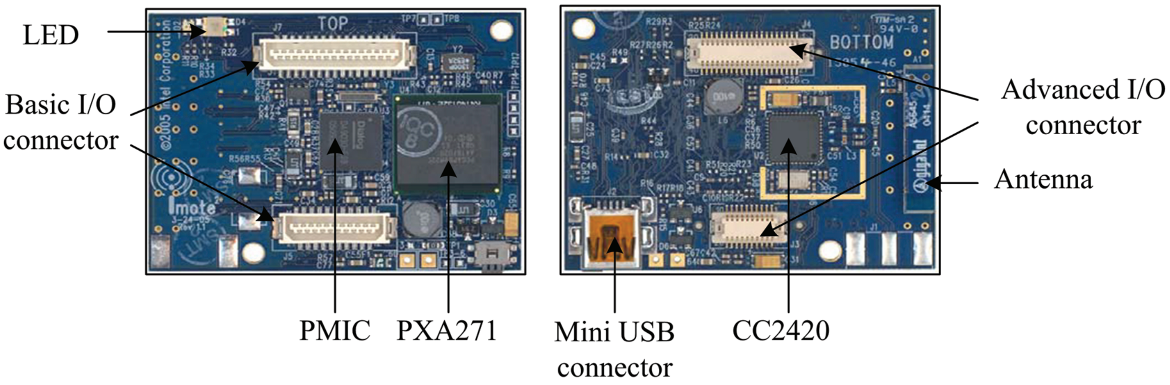


Figure 3.2: Imote2 wireless hardware platform without battery board. Left (top view), right (bottom view)

The key features of the Imote2 hardware platform are summarized below:

- PXA271 Intel XScale processor running at 13 - 416 MHz
- 256 kB SRAM, 32 MB FLASH, 32 MB SDRAM
- Integrated 802.15.4 radio and 2.4 GHz antenna
- Basic and advanced expansion connectors supporting: 3xUART, I2C, 2xSPI, SDIO, I2S, AC97, USB host, Camera I/F, GPIO
- Mini-USB port for direct PC connection
- Power Management Integrated Circuit (PMIC)
- Current: deep sleep mode 387 μA, active mode (13 MHz, radio off) 31 mA, active mode (13 MHz, radio Tx/Rx) 44 mA, and active mode (104 MHz, radio Tx/Rx) 66 mA
- Compact size 48 mm x 36 mm

The CC2420 is a true single-chip 2.4 GHz 802.15.4 compliant RF transceiver designed for low-power and low-voltage wireless applications. CC2420 includes a digital direct sequence spread spectrum (DSSS) baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbit/s. The key features of the CC2420 transceiver are summarized below:

- True single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver with baseband modem and MAC support

- DSSS baseband modem with 2 MChips/s and 250 kbit/s effective data rate
- Low current consumption (Rx: 18.8 mA, Tx: 17.4 mA @ 0 dBm transmit power)
- Low supply voltage (2.1 - 3.6 V) with integrated voltage regulator
- Programmable output power
- 128 (Rx) + 128 (Tx) byte data buffering
- Hardware MAC encryption (AES-128)
- 802.15.4 MAC hardware support:
  - Automatic preamble generator
  - Synchronization word insertion/detection
  - CRC-16 computation and checking over the MAC payload
  - Clear Channel Assessment (CCA)
  - Energy detection (ED)
  - Digital Receiver Signal Strength Indicator (RSSI)
  - Link Quality Indication (LQI)

3.4 Interface hardware

The Imote2 hardware platform has no built in ADC or DAC and cannot be directly connected to the hardware (sensors and actuator) of the inverted pendulum. The interface hardware in Figure 3.3 on Page 37 was developed to interface with the Imote2 platform using the basic I/O connectors. It also interfaces to the inverted pendulum through a 20 pin connector and to the PC via a bidirectional UART connection with a maximum data rate of 115200 baud. Sensor measurements and control values are converted to a voltage range of ±10 V by means of operational amplifiers and used for monitoring and control purposes. The main components of the interface hardware are listed below:

- 12-bit ADC operating at 200 kHz sampling rate to convert analogue sensor measurements to digital
- 12-bit DAC operating at 300 kHz update rate to convert digital control values to analogue
- UART connection to and from a PC operated at a baud rate of 115200 baud
- 20 characters and two lines LCD and LEDs for monitoring purposes
- Push-buttons for monitoring and control purposes
- 20 pins connector to the inverted pendulum

The interface hardware can also be used with a single Imote2 hardware platform to control the inverted pendulum without using the wireless network. In this case, all connections are wired and the UART connection can be used to transfer sensor measurements to the PC and control values from the PC to the actuator. All sensor values are sampled and converted to digital by the ADC driven by the Imote2 processor. After computing the control value, control voltages are applied to the DAC also driven by the Imote2 processor to control the inverted pendulum. This configuration is used to evaluate the control performance of selected control algorithm for the wired case, i.e., without network effects such as transmission delays and data losses. Further details on the interface hardware can be found in [Jiang, 2008].

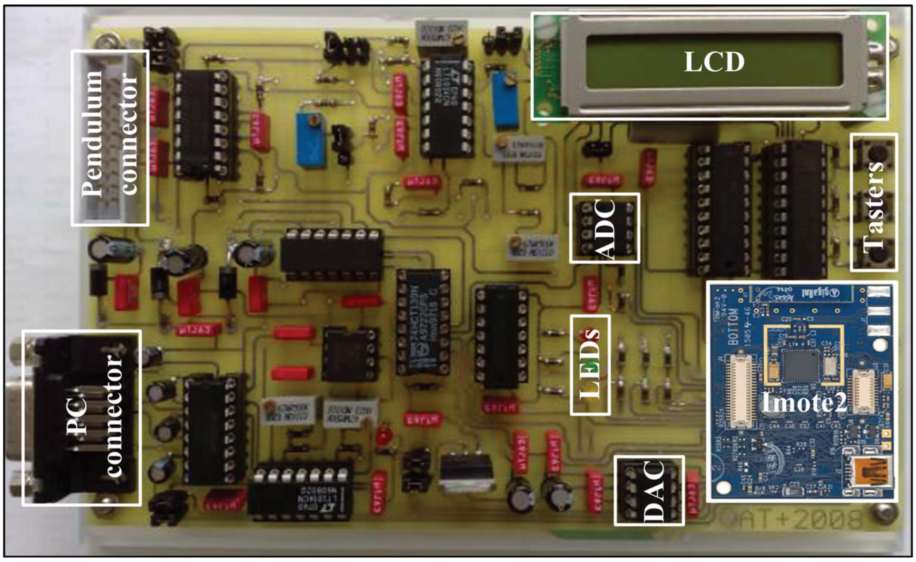


Figure 3.3: Interface hardware for the inverted pendulum testbed

3.5 Software

Several operating systems are available for WSNs among others, TinyOS [Alliance, 2011] and Contiki [Dunkels, 2011]. TinyOS[1] is an open source operating system designed for low-power wireless devices. TinyOS is suitable for event-driven applications with very small footprint. It uses the *nes*C programming language and is supported by the Imote2 platform. It provides a huge documentation with a large active community from industry and academia. However, several studies showed that TinyOS is not suitable for real-time control applications (see [Gabel, 2008]). Moreover, some functionalities of the Imote2 platform are not yet implemented and available in TinyOS.

Contiki[2] is an open source, highly portable, multi-tasking operating system for memory-efficient networked embedded systems and WSNs. It is developed by a group of people from industry and academia. Contiki is written in the C programming language and consists of an event-driven kernel, on top of which application programs can be dynamically loaded and unloaded at run time. Unfortunately, there is a lack of documentation and tutorials as well as implementation of hardware drivers for the Imote2 platform.

Due to the reasons listed above, a novel application development framework for the Imote2 hardware platform including Codeloader and Bootloader (see Figure 3.4 on Page 38) was developed which allows easy implementation of user codes in the programming language C and transfer from a standard PC to the memory of the Imote2 ([Engel & Krämer, 2009]). The main features of the framework are summarized below:

- Application interface for easy implementation of user codes including control algorithms and communication protocols in the programming language C
- Codeloader (platform independent) and Bootloader to load user compiled codes into the memory of the Imote2 using the Mini-USB connection, thereby providing full hardware control
- Control algorithms, communication protocols, and optimization algorithms are organized in tasks:

[1] http://www.tinyos.net/
[2] http://www.contiki-os.org/

- Each task has a priority represented by an integer number and tasks with small numbers have high priority
- Each task represents a function which can in turn include several functions (e.g. send, receive, and compute)
- Task periods are defined during initialization and can be changed at runtime
- Tasks are executed according to their priorities using a dynamic scheduling strategy with timers
- New tasks can be created and existing tasks removed at runtime

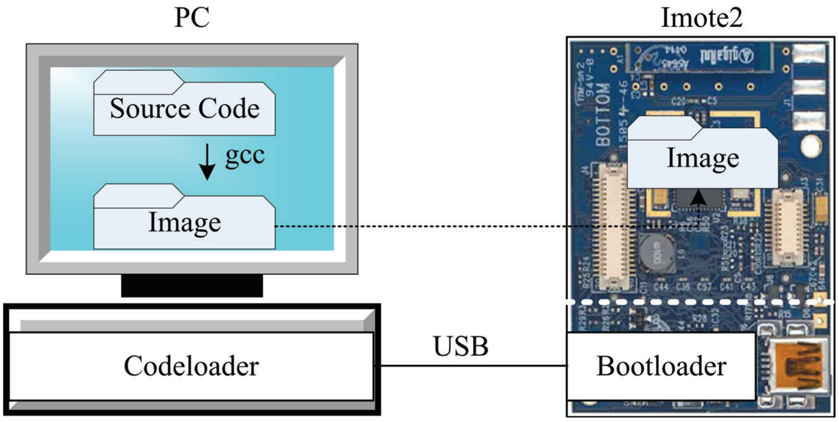


Figure 3.4: Application development framework for the Imote2 hardware platform with Codeloader and Bootloader [Engel & Krämer, 2009]

4 Simulation environment for WNCSs

4.1 Problem formulation

A simulation environment greatly simplifies not only the design process for control algorithms and communication protocols but also reduces the computational complexity and effort required during the implementation and test phase. One important issue in the design of network protocols for WNCSs is the knowledge of the QoS requirements (e.g. reliability and latency) of the control application. This knowledge allows the selection or implementation of tailored communication protocols that offer guaranteed QoSs to the control application. Therefore, a suitable simulation environment is necessary which enables not only the assessment of the control performance QoC for selected control structures in the presence of network induced effects such as delays and packet losses, but also the derivation of QoS bounds. QoS bounds include metrics such as the maximum allowable transfer interval (MATI), maximum allowable number of consecutive packet losses (MAPL), maximum allowable delay (MAD) with corresponding distributions, and the maximum allowable packet loss rate with corresponding distributions. Another important issue is the direct applicability of the simulated controller parameters to control the real plant without heuristic adaptation rules, the derivation of which is usually time consuming and not straightforward. This is one of the major drawbacks of the WNCS simulator solution proposed in [Gabel, 2008] beside the huge simulation time period compared to standard simulators and the implementation complexity which makes the realization of a WNCS structure almost impossible without expert knowledge.

In order to design and implement control algorithms and communication protocols for WNCSs, a simulation environment is needed which satisfies following requirements:

1. The simulation environment should be built around MATLAB/Simulink
2. The real plant should be included in the feedback control loop
3. Simulations should run in real time
4. The simulation of different WNCS structures with multiple sensors which can operate independently should be possible without expert knowledge
5. The simulation of stochastic as well as deterministic network induced delays and packet losses with different distributions should be possible
6. Options should be available for the selection of a desired control structure with corresponding controller parameters and different QoS parameters to simulate different network configurations

Taking into consideration all the requirements listed above led to the design, implementation, and realization of a MATLAB/Simulink based real-time Hardware in the Loop (HiL) simulation environment for WNCSs. The structure and main components of the HiL simulator will be presented and discussed subsequently. Simulation results for selected control structures and communication QoSs will also be presented and discussed.

4.2 HiL simulator structure

The structure of the HiL simulator depicted in Figure 4.1 is similar to that of the testbed described in the previous chapter. They both use the same hardware components (e.g. ADC, DAC, UART, etc.). All connections (sensor to controller and controller to actuator) of the HiL simulator are wired, i.e., there are no wireless connections between the components. The general structure of the HiL simulator is that of a feedback control loop with following main components:

- Real inverted pendulum plant with three sensors (position, angle, and velocity) and one actuator (DC motor)
- Interface hardware with data acquisition and conversion components such as ADCs and DAC and a bidirectional UART serial communication between the inverted pendulum and a PC running MATLAB
- Two network blocks implemented in MATLAB that allow the simulation of network induced delays and packet losses between sensors and controller and between controller and actuator respectively
- A controller block implemented in MATLAB that allows the simulation of different control structures and algorithms

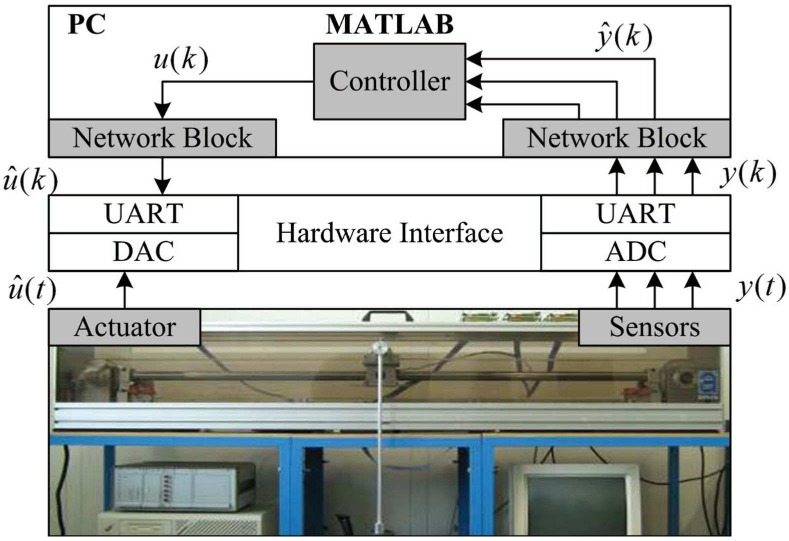


Figure 4.1: Structure of the HiL simulator

Inverted pendulum plant outputs are sampled periodically by the three sensors and transmitted to the PC over the serial communication interface. Upon reception, sensor values are passed to the input network block where they can be delayed or dropped according to a selected distribution. Delayed sensor measurements are used by the controller to compute the control action for a selected control algorithm. If all or parts of the sensor measurements are dropped, an observer can be used to reconstruct the missing measurements and compute the control action or the controller takes no action depending on the selected control strategy. Control values can also be delayed or dropped by the output network block. Delayed control values are transmitted to the actuator over the serial communication interface. Upon reception, delayed control values are applied by the actuator to the inverted pendulum. If control values are dropped, the actuator may keep the previous control value or reset the control value to zero depending on the selected actuation strategy. Although not considered in this thesis, event-based control strategies can be simulated using the HiL simulator.

4.3 Component description

4.3.1 Plant component

The plant of the HiL simulator control loop is the real inverted pendulum depicted in Figure 4.2. The inverted pendulum is a Single-Input Multiple-Output (SIMO) plant consisting of three sensors to measure the position $x(t)$ of the cart, the velocity $\dot{x}(t)$ of the cart, and the angle $\varphi(t)$ of the pendulum rod. A control voltage $u(t)$ is applied to a DC motor which serves as an actuator to control the inverted pendulum. A detailed description of the inverted pendulum and the corresponding nonlinear and linearized mathematical models is provided in the Appendix.

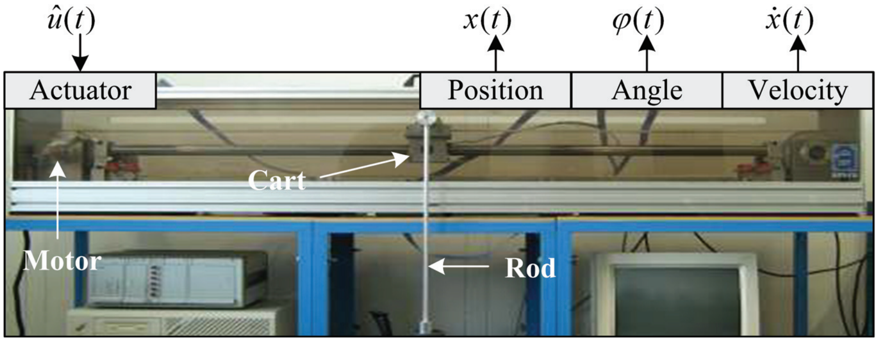


Figure 4.2: HiL plant component with one control input and three sensor outputs

The block diagram of the interface hardware of the inverted pendulum is shown in Figure 4.3. Continuous-time sensor measurements ($x(t)$, $\varphi(t)$, and $\dot{x}(t)$) are periodically sampled and converted to digital ($x(k)$, $\varphi(k)$, and $\dot{x}(k)$) by the ADC of the interface hardware. Discrete position, angle, and velocity measurements of length 2 bytes each are sent in one single packet of total length 10 bytes (including three delimiter symbols and one terminator symbol) to the PC over the UART serial connection operating at a baud rate of 115200 baud. Discrete control values of length 2 bytes are sent in a packet of total length 4 bytes (including one delimiter symbol and one terminator symbol) at the same baud rate from the PC to the actuator where they are converted to analogue by the DAC of the interface hardware. Each UART frame consists of 1 start bit, 8 data bits, and 1 stop bit. The total time required to transmit a packet is $\tau^{sc} \approx 70\ \mu s$ for sensor packets and $\tau^{ca} \approx 28\ \mu s$ for controller packets which are negligible.

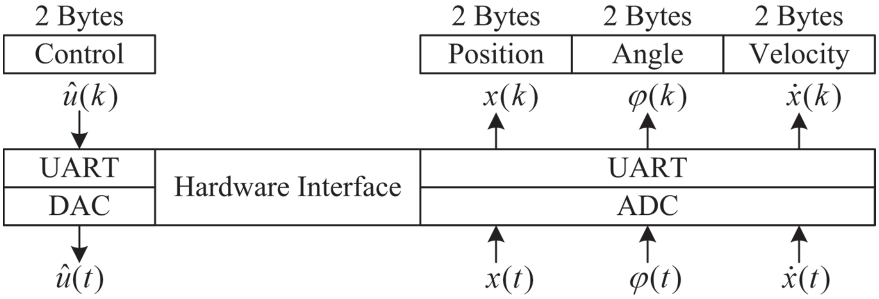


Figure 4.3: HiL interface hardware

The processor of the Imote2 hardware platform is used to drive the ADC and DAC of the interface hardware. The UART communication between the inverted pendulum and the PC is also coordinated by the processor of the Imote2 hardware platform. The transceiver of the Imote2 hardware platform is shut down and the processor is programmed to run at a frequency of 104 MHz. Details of the Imote2 hardware platform and the interface hardware were provided in the previous chapter (Inverted Pendulum Testbed).

4.3.2 Network components

The structure of the two network components used by the HiL simulator is depicted in Figure 4.4. It consists of an input network block between the sensors and the controller and an output network block between the controller and the actuator. These two network components are modeled using Simulink blocks. Network parameters can be modified by editing the properties of the Simulink blocks. Some of the available Simulink network block properties are the delay and packet loss distributions with corresponding parameters such as the mean delay and the packet loss rate. Standard distributions (e.g. Gaussian, Uniform, and Exponential) of delays and packet losses are implemented in MATLAB using *S-functions*. Special or user defined distributions of delay and packet losses can be added to existing *S-functions* or implemented separately.

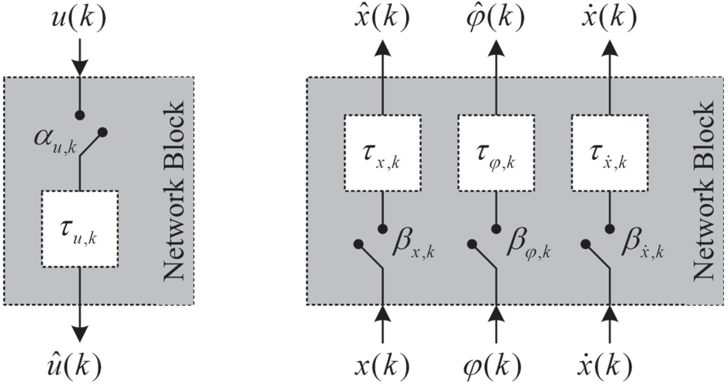


Figure 4.4: HiL network component between sensors and controller (right) and between controller and actuator (left)

The network block between the sensors and the controller has three inputs ($x(k)$, $\varphi(k)$, and $\dot{x}(k)$) and three outputs ($\hat{x}(k)$, $\hat{\varphi}(k)$, and $\hat{\dot{x}}(k)$) whereas the network block between the controller and the actuator has one input $u(k)$ and one output $\hat{u}(k)$. Packet losses are modeled by switches $\beta_{x,k}$, $\beta_{\varphi,k}$, $\beta_{\dot{x},k}$ and $\alpha_{u,k}$ which are opened if a packet is dropped (i.e. $\beta_{x,k} = 1$) and closed otherwise (i.e. $\beta_{x,k} = 0$). Delayed sensor and actuator packets are held by the network blocks for time durations $\tau_{x,k}$, $\tau_{\varphi,k}$, $\tau_{\dot{x},k}$, and $\tau_{u,k}$ respectively. At each sampling instant k, switch states and delay values from selected distributions are appended to received sensor and actuator values by the corresponding network blocks resulting in the following sets of information that are used by the controller to compute the control action and the actuator for actuation respectively:

$$I_{x,k} = \{x(k),\ \tau_{x,k},\ \beta_{x,k},\ k\} \tag{4.1}$$

$$I_{\varphi,k} = \{\varphi(k),\ \tau_{\varphi,k},\ \beta_{\varphi,k},\ k\} \tag{4.2}$$

$$I_{\dot{x},k} = \{\dot{x}(k),\ \tau_{\dot{x},k},\ \beta_{\dot{x},k},\ k\} \tag{4.3}$$

$$I_{u,k} = \{u(k),\ \tau_{u,k},\ \alpha_{u,k},\ k\} \tag{4.4}$$

where the sampling instant k is also the packet number since periodic sampling and actuation is considered.

4.3.3 Controller component

The controller block of the HiL simulator is shown in Figure 4.5 on Page 43. The four inputs to the controller block are the reference position of the cart $x_{ref}(k)$ and the three outputs $\hat{x}(k)$, $\hat{\varphi}(k)$, and $\hat{\dot{x}}(k)$ of the network block between the sensors and the controller. The output $u(k)$ of the controller block is the input to the network block between the controller and the actuator.

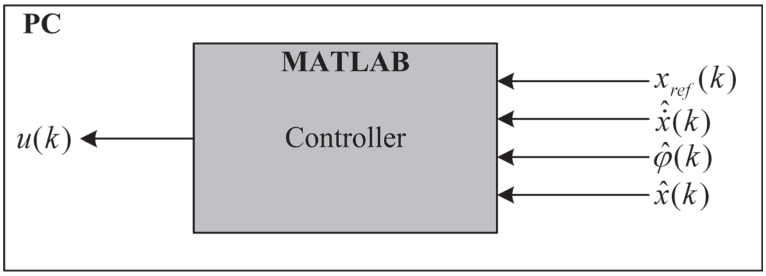


Figure 4.5: HiL controller component

Different control structures and algorithms (e.g. cascaded controller, pole placement controller, LQR, IMC, and H_∞ controller) are implemented in the controller block some of which are illustrated in Figure 4.6. All control algorithms are implemented in *S-functions* and controller parameters can be changed by editing the properties of the corresponding Simulink block. User defined control structures and algorithms for the inverted pendulum can easily be added as a Simulink block to the ones in Figure 4.6. Existing controller blocks can be used and parameterized to suit user needs. The only condition to be fulfilled is that the controller block should have four inputs and one output. A summary of all control structures and algorithms implemented for the inverted pendulum with corresponding simulation and experimental results is provided in the Appendix.

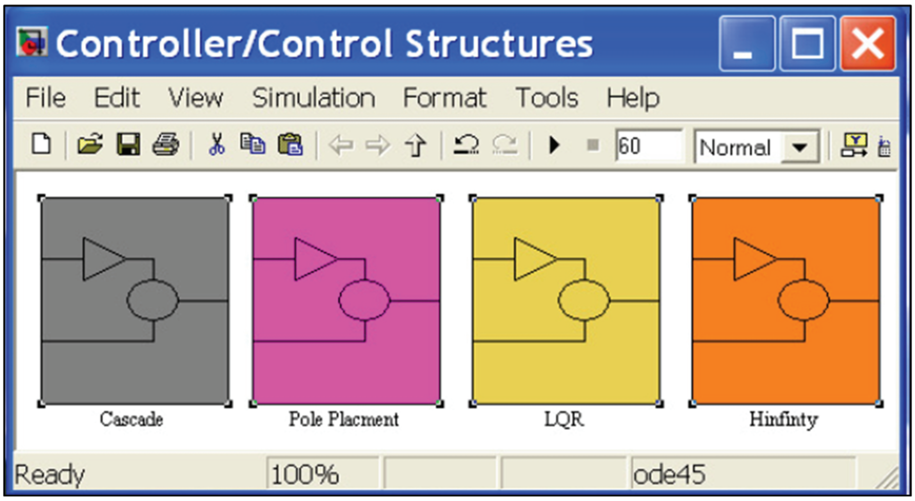

Controller/Control Structures
File Edit View Simulation Format Tools Help
60
Normal
Cascade
Pole Placment
LQR
Hinfinty
Ready
100%
ode45

Figure 4.6: HiL controller block in Simulink

4.4 Validation on the inverted pendulum

Several experiments are carried out to assess the performance of the HiL simulator. In the first set of experiments, HiL simulation results for the wired case (i.e. no delays and no packet losses) are compared simulation (MATLAB/Simulink) and experimental (Imote2) results in order to check the accuracy of the models used and the validity of the assumptions made (e.g. negligible serial communication delays). Simulation results are obtained in MATLAB/Simulink using a model of the inverted pendulum instead of the real inverted pendulum. Both model and control algorithms are implemented in MATLAB/Simulink. Experimental results are obtained with the real inverted pendulum but with the difference that the control algorithm is implemented in the Imote2 hardware and not in MATLAB/Simulink. The second set of experiments are carried out to analyze the effects of network induced delays and packet losses on the control performance from which QoS bounds for the design of suitable communication protocols can be derived.

In all simulations and experiments, a step position input from 0 m to 0.3 m is applied to the inverted pendulum at time $t = 9$ s starting from an initial state $x_0 = [0\ m,\ 0°,\ 0\ m/s,\ 0\ rad/s]^T$. The simulation duration is $t = 60$ s. The quality of control QoC is evaluated using following equation:

$$QoC = (1 - \frac{J}{J_{max}}) \cdot 100 \tag{4.5}$$

where J is the measured control error and J_{max} the maximum control error. The higher the QoC value the better the control performance. The minimum QoC value is 0 and the maximum QoC value is 100. The derivation of (4.5) is provided in the Appendix. HiL simulation results are shown in Figure 4.7 for the cascaded controller without disturbance compensator (C1), cascaded controller with disturbance compensator (C2), pole placement controller (Place), and LQR controller (LQR). These results are similar to the ones obtained experimentally (see Appendix) and the best control performance is achieved with the state feedback controllers. The relative degradation of the control performance at $h = 10$ ms compared to $h = 20$ ms and $h = 30$ ms is due to measurement noise and the effects of nonlinearities (e.g. slew rate limitation of the DC motor). The maximum achievable sampling period is $h = 100$ ms for the cascaded controllers, $h = 90$ ms for the pole placement controller, and $h = 80$ ms for the LQR controller.

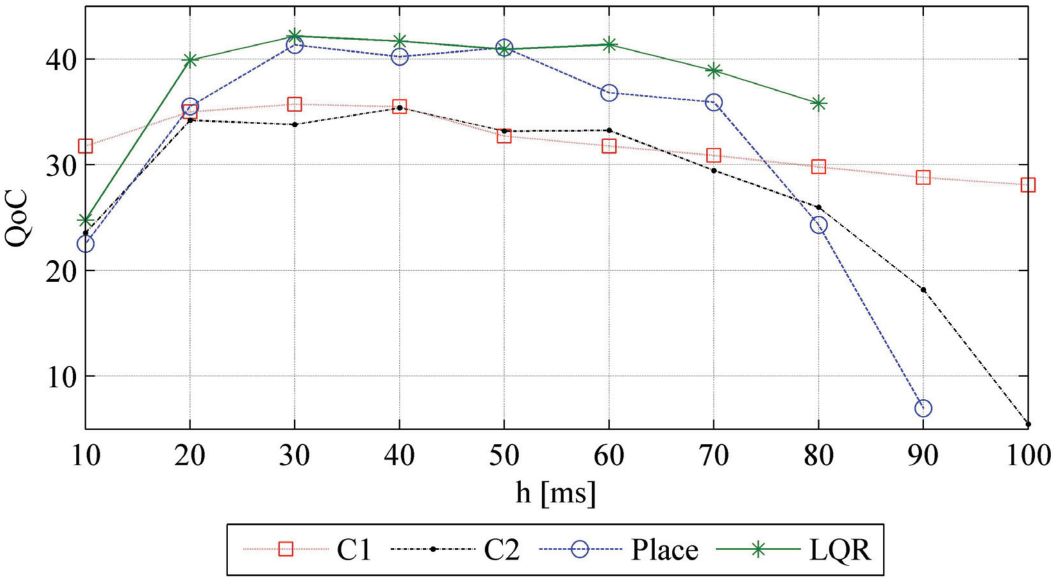


Figure 4.7: HiL simulation results for the wired case at selected sampling periods

$h[ms]$	K_x	K_φ	$K_{\dot{x}}$	$K_{\dot{\varphi}}$
10	-0.6480	2.0152	2.0957	0.3230
20	-0.5939	1.8829	1.9529	0.3058
30	-0.5444	1.7616	1.8221	0.2897
40	-0.4992	1.6504	1.7023	0.2749
50	-0.4579	1.5485	1.5927	0.2611
60	-0.4202	1.4551	1.4924	0.2483
70	-0.3857	1.3695	1.4006	0.2365
80	-0.3543	1.2911	1.3166	0.2254

Table 4.1: Optimal LQR controller gains for selected sampling periods

Optimal controller gains $K = [K_x, K_\varphi, K_{\dot{x}}, K_{\dot{\varphi}}]^T$ for selected sampling periods are summarized in Table 4.1 and Table 4.2 for the LQR controller and the pole placement controller respectively.

For the sake of completeness, the corresponding parameters of the reduced-order observer gain matrix L_{red} given by equation (4.6) are summarized in Table 4.3. Optimal controller parameters for the cascaded controllers for selected sampling period are provided in the Appendix and will not be listed here.

$$L_{red} = \begin{pmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \end{pmatrix} \tag{4.6}$$

$h[ms]$	K_x	K_φ	$K_{\dot{x}}$	$K_{\dot{\varphi}}$
10	-0.7893	1.8035	2.0734	0.3430
20	-0.7155	1.6734	1.9163	0.3197
30	-0.6493	1.5568	1.7754	0.2988
40	-0.5900	1.4521	1.6489	0.2799
50	-0.5368	1.3579	1.5352	0.2629
60	-0.4890	1.2731	1.4327	0.2476
70	-0.4460	1.1966	1.3403	0.2337
80	-0.4072	1.1276	1.2569	0.2212
90	-0.3723	1.0652	1.1815	0.2098
100	-0.3408	1.0087	1.1133	0.1994

Table 4.2: Optimal pole placement controller gains for selected sampling periods

$h[ms]$	L_{11}	L_{12}	L_{13}	L_{21}	L_{22}	L_{23}
10	0,0986	63,2141	-10,0923	0,1368	0,0015	-13,9856
20	0,1633	43,3186	-8,3388	0,1573	-0,0053	-8,0147
30	0,2258	31,8327	-7,6678	0,1619	-0,0078	-5,4799
40	0,2912	24,7726	-7,3939	0,1642	-0,0091	-4,1563
50	0,3587	20,1671	-7,2640	0,1663	-0,0098	-3,3556
60	0,4271	16,9986	-7,1848	0,1683	-0,0103	-2,8208
70	0,4954	14,7170	-7,1218	0,1702	-0,0106	-2,4383
80	0,5632	13,0110	-7,0625	0,1722	-0,0109	-2,1512
90	0,6302	11,6954	-7,0022	0,1741	-0,0111	-1,9276
100	0,6961	10,6551	-6,9394	0,1761	-0,0113	-1,7484

Table 4.3: Reduced-order observer parameters for selected sampling periods

MATLAB simulation, HiL simulation, and experimental results for the LQR controller are shown in Figure 4.8 and corresponding performance matrices are summarized in Table 4.4 both on Page 46. It can be seen that HiL simulation results are identical to those obtained with MATLAB/Simulink and experimentally for the wired case. The slight difference between HiL simulation results and those obtained with MATLAB is due to the effect of the Coulomb friction and other nonlinearities which are not considered in the MATLAB/Simulink simulation. Thus, the HiL simulator behaves correctly for the wired case and serial transmission delays can be neglected as assumed previously. Moreover, the HiL simulation is carried out in real time and has the same duration as for the real experiment. By choosing small simulation durations, several simulations can be carried out within a short period of time. This is important for the derivation of optimal controller parameters for selected network QoSs which requires a huge number of simulations.

In the second set of experiments, different distributions of delays and packet losses are considered. All three sensor measurements are sent in one single packet to the controller. A new control

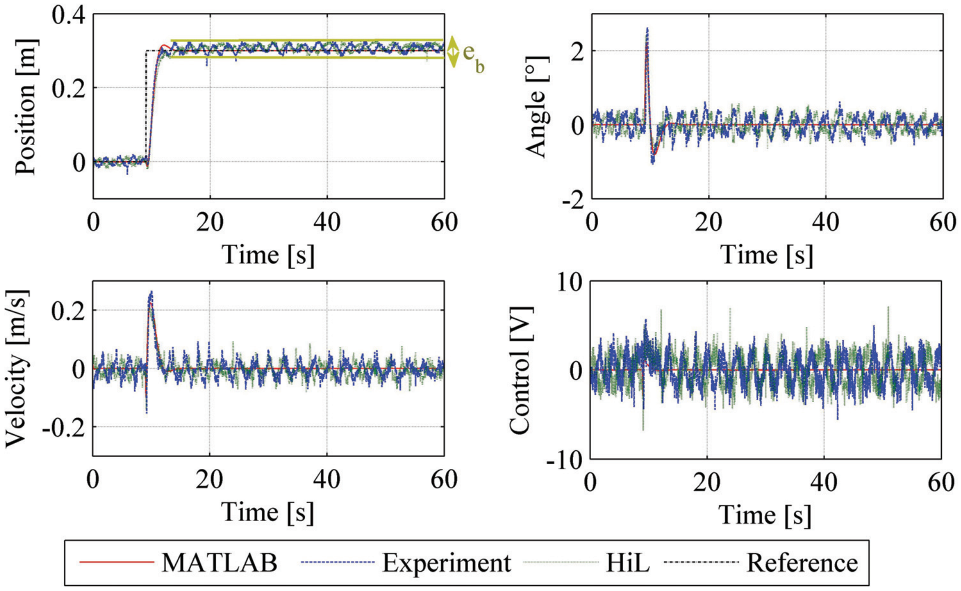


Figure 4.8: Comparison of HiL simulation results with experimental and MATLAB/Simulink simulation results for the LQR control structure and the wired case @ $h = 30\ ms$

Description	Overshot (PO)	Error Band (e_b)	Rise Time (t_r)	Settling Time (t_s)
Experiment	10.1 %	19.2 %	1.71 s	-
HiL	10.5 %	19.0 %	1.80 s	-
MATLAB	3.3 %	0 %	1.83 s	5.1 s

Table 4.4: Comparison of LQR controller performance metrics for the HiL simulation, MATLAB simulation, and experiments @ $h = 30\ ms$

value is only computed upon reception of a new sensor packet. In the event of a packet loss between sensor and controller or between controller and actuator, the old control value is used by the actuator until a new controller value arrives at the actuator. Therefore, packet loss rates represent both sensor and controller packet losses. The total transmission delay is the sum of the transmission delay between the sensor and the controller and between the controller and the actuator.

Figure 4.9 on Page 47 shows the HiL simulation results for the LQR control structure. An excerpt of the underlying delay distributions is depicted in Figure 4.10 on Page 47. The same distribution types are used for the packet losses. The effects of packet losses on the control performance are simulated at a sampling period $h = 30\ ms$ which is the optimal sampling period of the LQR controller for the wired case. The effects of delays on the control performance are simulated using a sampling period of $h = 80\ ms$ corresponding to the maximum applicable sampling period for the LQR controller (see Appendix). As can be seen in Figure 4.9 on Page 47, network induced delays and packet losses negatively affect the control performance and can even lead to control system instability. The impact of delays and packet losses on the control performance depends on the underlying distributions and control structures, some of which are less prone to delays and packet losses, as will be seen in the next section. In order to optimize the control performance, delays and packet losses should be minimized or avoided if possible by selecting or designing suitable communication protocols or compensated for by including delays and packet losses in the derivation of optimal controller parameters. Delays and packet losses at the stability bounds are very critical and the smaller the delays and packet losses, the better the control performance. Especially, packet losses are very critical as they result in an increase of the effective sampling period by the amount of consecutive packet losses. Several packet losses and delays less or equal

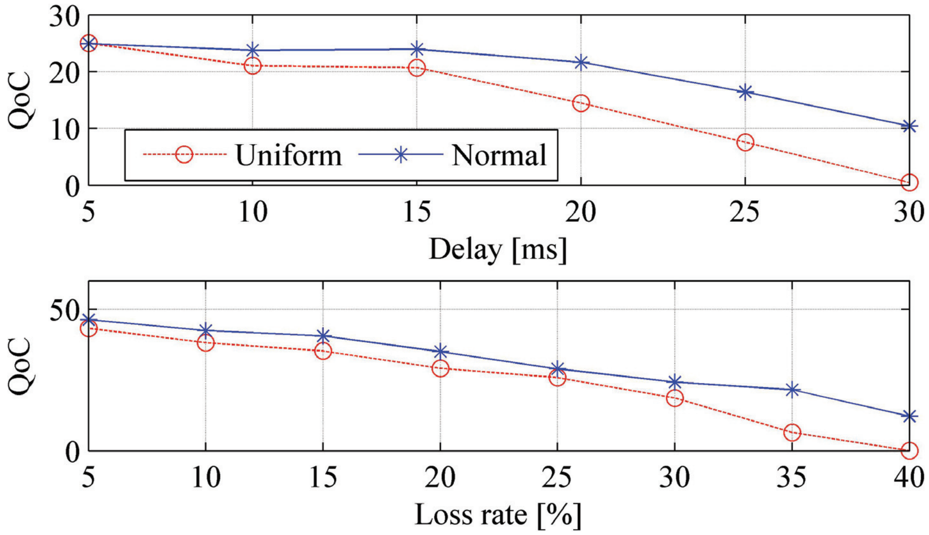


Figure 4.9: HiL simulation results for the LQR control structure with uniformly and normally distributed delays and packet losses

to the sampling period can be tolerated at fast sampling periods.

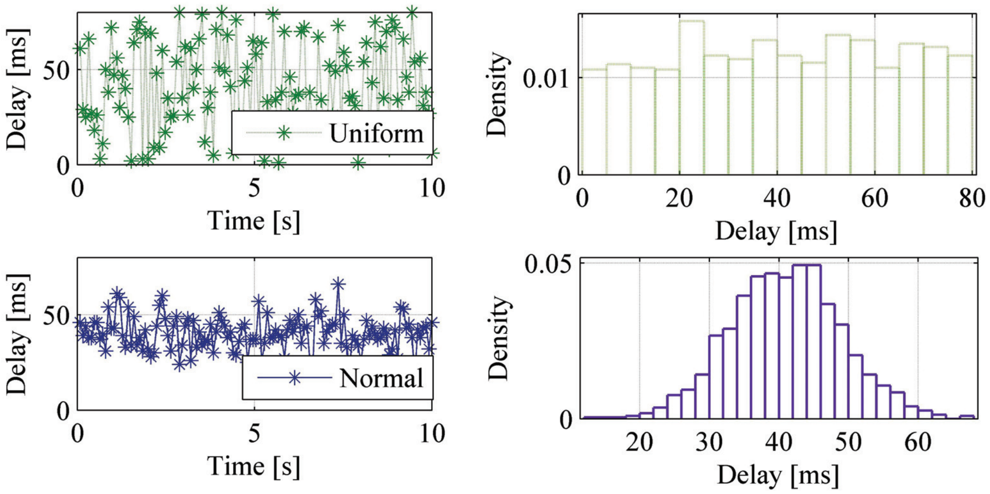


Figure 4.10: Uniform and normal distributions of delays

4.5 Derivation of QoS bounds

In this section, worst-case QoS bounds are derived for different control structures using the HiL simulator. Derived stability bounds will serve as the starting point for the implementation of suitable communication protocols for the inverted pendulum in the next chapter. These stability bounds are only valid with the optimal controller parameters obtained for the wired case. These stability bounds may change if network induced delays and packet losses are taken into consideration when deriving optimal controller parameters, i.e., larger delays and more packet losses may be tolerated. The maximum delays and packet loss rates under which the stability of the inverted pendulum is guaranteed are summarized in Table 4.5 on Page 48 and are the same for both the uniform and the normal distribution.

Controller	LQR	$Place$	$C2$	$C1$
$\tau_{max}[ms]$ @ $h = h_{max}$	35	15	15	20
$\rho_{max}[\%]$ @ $h = h_{min}$	40	30	30	35

Table 4.5: Maximum delay τ_{max} and packet loss rates ρ_{max} obtained with the uniform distribution

Other important QoS metrics beside the distributions dependent mean delay and packet loss are the maximum allowable delay (MAD) and the maximum allowable number of consecutive packet losses (MAPL) for which the stability of the controlled system can be guaranteed. MAD and MAPL values are summarized in Table 4.6 and Table 4.7 respectively. The MAD is constant during simulation and includes both the computation time of the selected control algorithm and the communication induced delay. The MAD is upper-bounded by the selected sampling period.

Controller	C1	C2	Place	LQR
h_{max} [ms]	100	100	90	80
MAD [ms]	19	32	34	14

Table 4.6: Maximum allowable delay (MAD) from sensing to actuation @ $h = h_{max}$

Controller	C1	C2	Place	LQR
h [ms]	30	30	30	30
MAPL	12	8	10	14

Table 4.7: Maximum allowable number of consecutive packet losses (MAPL) @ $h = 30\ ms$

Experimental results show that higher MAPL values can be achieved than in theory where for example one would expect MAPL = 3 for h = 30 ms instead of MAPL = 14 for the LQR. It should be noted however, that different distributions of delays and packet losses may lead to different QoS bounds and if these distributions are known, then HiL simulations can be carried out to find these bounds. QoS bounds also depend on the selected sampling period and for slow sampling periods (e.g. $h << MATI$), delays up to the value of the selected sampling period are possible. However the resulting effects on the control performance are not pronounced. Moreover, no packet loss can be tolerated when using h = MATI since losing a packet is equivalent to doubling the MATI which leads to instability. Finally, only worst-case average results are given in Table 4.5, Table 4.6, and Table 4.7.

4.6 Summary

The HiL simulation environment offers the possibility to investigate the effects of networked induced delays and packet losses on the control performance in WNCSs. Typical effects such as measurement uncertainties, nonlinearities, stochastic disturbances, and quantization are implicitly taken into consideration by introducing the real plant in the feedback control loop. This allows simulations to be run in real time and the derivation of QoS-based optimal controller parameters which are directly applicable to the real plant without further modifications. Moreover, robust and adaptive control strategies for WNCSs can be validated using the HiL simulator. Complex control algorithms such as MPC with high computational and memory requirements can be implemented in the PC which has a powerful processor and a huge amount of memory. Finally, no expert knowledge is required to modify the HiL structure to accommodate plants others than the inverted pendulum.

5 Tailored WNCS communication protocols

5.1 Introduction

Despite the ongoing standardization of low-power and low data rate wireless network protocols, there is, to the best of the author's knowledge, not yet any widely accepted complete protocol stack for industrial applications in general, and control applications in particular (see [Willig, 2008, *and references therein*]). Many of the available communication protocol solutions for industrial automation applications are based on the IEEE 802.15.4-2006 standard, which only specifies the physical layer and the MAC layer. These protocols rely on a fully centralized TDMA scheduling approach for mesh networks (e.g. WirelessHART and ISA-SP100). Dynamical schedule modifications necessary to adapt communication protocol parameters at runtime to changing network conditions while minimizing the overall energy consumption of the network (i.e. maximizing node lifetimes) are not taken into consideration by these protocols.

As far as communication protocols for WNCSs are concerned, the main contributions of this thesis are suggested modifications to the IEEE 802.15.4-2006 standard. These modifications enable the offline and most importantly the online adaptation of communication protocol parameters at the physical layer, MAC layer, and application layer to changing network conditions (e.g. interferences, congestion) and control performance (e.g. disturbances). Such a tuning is difficult because simple and accurate models of the influence of network protocol parameters on the probability of successful packet transmission, packet delay, and energy consumption are difficult to derive and are mostly not available. Moreover, adapting protocol parameters to changing network traffic conditions and control performance at runtime using algorithms that should be implemented on resource constrained wireless nodes is not straightforward.

In order to overcome the difficulties mentioned above, WNCS communication protocols constrained by the QoS bounds derived in the previous chapter are devised. These communication protocols are tailored to suit the needs of control applications in general and the inverted pendulum system in particular. Different MAC protocols are implemented with the added possibilities of switching between different MAC schemes and changing MAC protocol parameters at runtime. A simple power control algorithm is implemented at the physical layer which allows the adaptation of the transmit power of the transceiver based on the transmitter-receiver distance and the channel conditions. The influence of communication protocol parameters on network induced effects such as reliability, latency, and energy consumption are modeled using simple and accurate mathematical expressions. These mathematical expressions will be used in the next chapter to derive adaptive algorithms for the joint optimization of control and communication performance resulting in the following constrained optimization problem for a generic transmitting node:

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & E_{tot}(\mathbf{x}) + J_{control}(\mathbf{x}) && \text{(5.1a)}\\
s.t. \quad & R(\mathbf{x}) \geq R_{min} && \text{(5.1b)}\\
& E[D(\mathbf{x})] \leq D_{max} && \text{(5.1c)}\\
& \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} && \text{(5.1d)}
\end{aligned}
$$

where $\mathbf{x}$ is the vector of decision variables of the node, $J_{control}$ the control performance cost function, E_{tot} the total energy consumption of the node, $R(\mathbf{x})$ the reliability, R_{min} the minimum desired probability of successful packet delivery, $E[D(\mathbf{x})]$ the average delay for a successful received packet, and D_{max} the desired maximum average delay. The limits of the protocol and control parameters are captured by the constraint $\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}$. Equation (5.1) is a concrete formulation of the general optimization problem presented in the introduction. The total power consumption of the wireless node is the objective function and the constraints are the reliability and latency requirements of the control application in order to achieved desired or acceptable control performance. Thus, the idea is to optimize the network lifetime by taking into consideration the constraints imposed by the control application instead of just improving reliability, latency, and power consumption separately.

In the remaining of the this chapter, the selected WNCS structure is presented including details about the protocol stack, the network topology, the superframe structure, the frame format, and the power consumption of the selected wireless node hardware. This presentation is followed by a thorough description of the selected and implemented MAC protocols for WNCSs with a special focus on their available degrees of freedom as well as tunable parameters. After describing the MAC protocols, the selected power control algorithm is introduced along with the derivation of mathematical expressions for node lifetimes. Finally, simulation and experimental results are presented and discussed.

5.2 Layered structure of the WNCS protocol stack

5.2.1 WNCS protocol stack

The four layers of the proposed WNCS communication protocol stack are illustrated on a control block diagram in Figure 5.1 on Page 51. In the selected WNCS control structure, a wire connection between controller and actuator is assumed whereas sensors and controller exchange information over a wireless network. Since the connection between controller and actuator is wired, the controller may be powered using the actuator power supply. Each plant output is periodically sampled by a sensor and samples are transmitted separately over the wireless network to the controller. Upon reception of sensor measurements, the controller computes the control value which is transmitted to the actuator over the wired connection. Upon reception, control values are applied to the plant by the actuator. As will be seen later on, this configuration is advantageous since all sensor measurements and network information (e.g. delay and packet dropout statistics) are available to the controller and the optimizer at each sampling instant. This information can be used by the controller to estimate missing sensor measurements, select appropriate control structures and parameters (e.g. controller gains), and compute optimal control values. The same set of information can be used by the optimizer to select appropriate network scheduling strategies (e.g. MAC protocols) and protocol parameters (e.g. slot duration).

The main components and functionalities of the different layers of the selected WNCS communication protocol stack are described below:

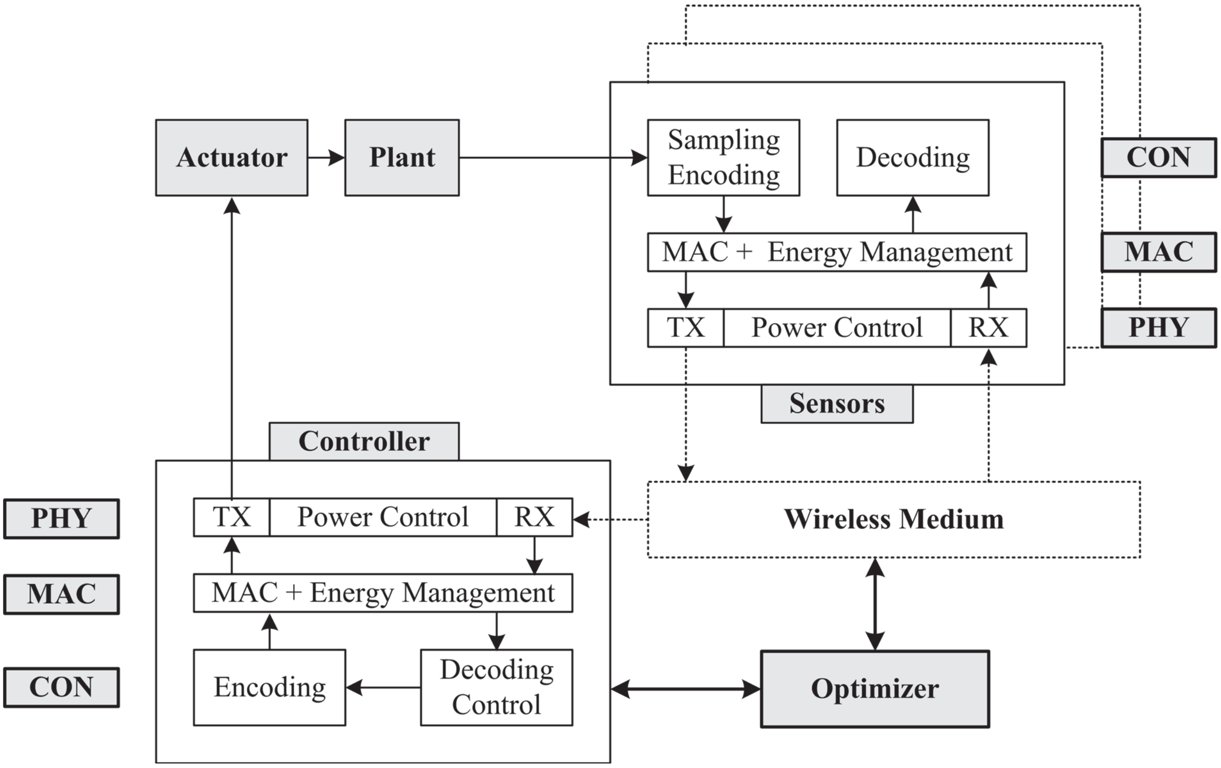


Figure 5.1: Illustration of the layers of the WNCS protocol stack including a Physical layer (PHY), a Data Link layer (MAC), a Network layer (ROU), and an Application layer (CON)) on a WNCS control block diagram

1. *Physical layer (PHY)*: Digital data are modulated into radio waves at the physical layer. Energy can be saved by means of bit rate control algorithms such as DMS (Dynamic Modulation Scaling) or by using power control algorithms to control the transmit power. DMS allows the switching of the modulation scheme under constant transmit power to improve the signal reception. From a control point of view, the adaptation of the modulation scheme can be viewed as receiving measurements with variable delays or with variable quantization levels. Power control algorithms are used to regulate the transmission quality under packet loss and signal interference. Hence, the transmit power affects the reliability and can be adapted based on the distance between the transmitting node and the receiving node or the channel conditions. Both bit rate and power control algorithms require extra communication between the transmitter node and the receiver node. Unfortunately, bit rate control is not possible on the Imote2 hardware platform since the modulation scheme and the data rate of the CC2420 transceiver are fixed. However, the transmit power can be adjusted using eight different power levels from -25 dBm to 0 dBm and channel conditions can be monitored using the RSSI, ED, and LQI signals.

2. *Data Link layer (MAC)*: The access to the wireless medium shared among all the nodes is determined by the MAC layer. The selection of a MAC scheme, which can be scheduled (e.g. TDMA) or random (e.g. CSMA/CA), affects both the latency and the reliability of the network. Energy is wasted by the nodes in the idle listening mode (listening without transmission taking place), collision (simultaneous transmission leading to failed reception), overhearing (listening to message uselessly), and overhead (acknowledgement and synchronization packets). Radio and processing units can usually save energy by switching between different activity modes such as ON, OFF, Sampling, Transmit, and Receive. Energy saving can be achieved by managing the activities of the radio and processing units (e.g. duty cycling) and by tuning the parameters of the MAC protocols (e.g. slot duration, sleep duration, listening duration, access probability, and wake up probability). However,

activity mode switching consumes both time and energy and a tradeoff should be made between node awareness and energy consumption. Energy management includes scheduling policies to determine when, how long, and how deep radio and processing unit should sleep in order to save energy.

3. *Network layer (ROU)*: Data may be routed from the sensors to the controller along a series of relay nodes called path in an energy efficient manner by the routing layer. Reliability and latency can be traded by tuning the parameters of the routing protocol one performance measure of which is usually the network lifetime. However, the routing layer is not implemented in this thesis. We assume a small scale wireless network with at most single-hop connections between sensors and controller.

4. *Application layer (CON)*: The executive part of sensing, control, and actuation takes place at the application layer. Activities include among others source encoding, source decoding, and control law computation. Sensing, signal processing, and signal encoding are performed by the sensor nodes whereas source decoding and control value computation are performed by the controller node. In the sensor nodes, analogue measurements are converted to digital by the process of quantization, which introduces a loss of information depending on the quantization specifications. Encoding can reduce the amount of data to be sent over the network thus improving latency and energy consumption. Sensors and controller are distributed each operating with its own clock. Therefore, sensor nodes, controller node, and actuator node should be synchronized. Both periodic and asynchronous control (also known as event-based control) strategies can be implemented but only periodic control is addressed in this thesis.

The main difference between the proposed communication protocols and the specification in the IEEE 802.15.4-2006 standard is the use of an optimizer (see Figure 5.1 on Page 51). The optimizer, whose functionalities can be implemented in the controller node (PAN coordinator) or a separate node connected to the PAN coordinator, initiates and coordinates the online selection and switching between different network scheduling and control algorithms based on measured network QoS and control performance. The role of the optimizer will be explained in the sequel.

5.2.2 Network topology

The structure of the star network topology used throughout this thesis is depicted in Figure 5.2 on Page 53. It consists of a PAN coordinator in which the control algorithms are implemented, m sensor nodes transmitting their sensor measurements over the wireless network to the PAN coordinator, and an actuator node receiving control values from the PAN coordinator over a wired connection. The PAN coordinator is a FFD[1] whereas the sensor nodes are RFDs[2].

The PAN coordinator starts the network and controls network parameters (e.g. duty cycle). All sensor nodes contend to send data to the PAN coordinator using the beacon-enabled slotted mode for different MAC protocols. MAC protocols include scheduled access protocols and random access protocols which will be presented in the next section. A sensor wishing to transfer data to the PAN coordinator first listens for the network beacon and if found, synchronizes to the superframe structure and transmits its data frame at the appropriate time depending on the selected MAC protocol. The PAN coordinator may acknowledge successful data reception by sending an optional acknowledgement frame to the sensor. In order to transmit data to a sensor, the PAN coordinator indicates in the network beacon that it has data pending for the device. The

[1] Full Function Device

[2] Reduced Function Device

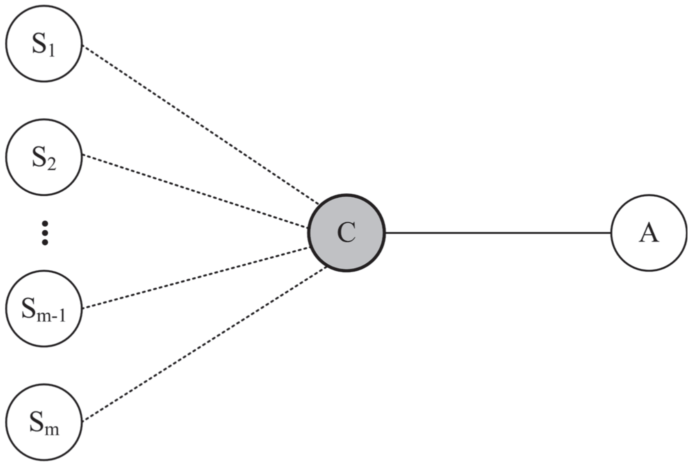


Figure 5.2: WNCS network topology with wireless connections between sensors (S_1,...,S_n) and controller (C) and a wired connection between controller (C) and actuator (A)

device then periodically listens to the network beacon and receives the data from the coordinator in dedicated time slots. Sensors may also acknowledge the successful reception of data from the PAN coordinator by transmitting an optional acknowledgment frame.

5.2.3 Superframe structure

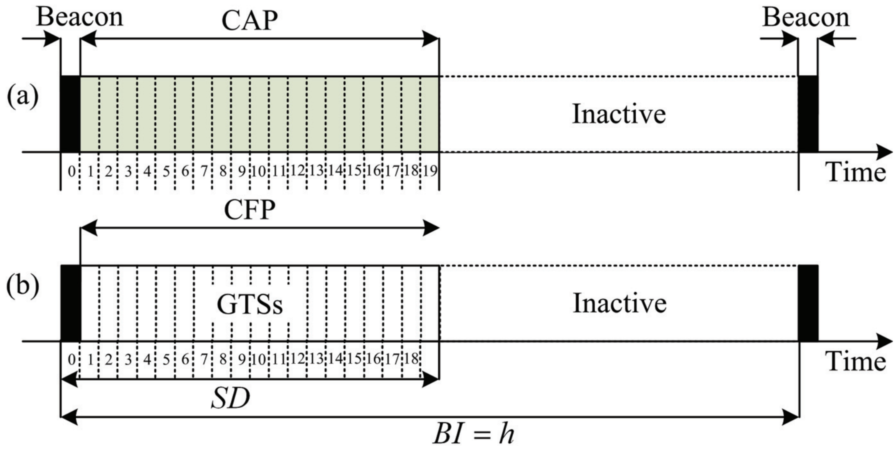


Figure 5.3: WNCS superframe structure with $S = 20$ equally sized slots: (a) superframe structure with inactive period and no GTSs, (b) superframe structure with inactive period and GTSs

The communication between the sensors and the controller is organized in temporal windows (superframe) by the PAN coordinator. The superframe structure is shown in Figure 5.3. It consists of S equally sized slots of duration s_d each and can include an active as well as an inactive portion depending on the selected values of the superframe duration SD and the beacon interval BI (details are provided in the Appendix). PAN coordinator and sensors may enter a low-power mode during the inactive portion of the superframe. The value of S depends on the selected communication protocols (e.g. MAC scheme) and can be adjusted offline and at runtime (e.g. when new nodes joint the network), thus, enabling dynamic slot allocation. All superframe slots can be accessed by contention using a CSMA/CA MAC scheme with nodes sensing the channel before transmitting a message in a *best effort* manner (possible data losses due to collision and interferences) or may be dedicated to specific nodes and accessed in a

TDMA manner (real-time *guaranteed* message transmission). This is a modification of the IEEE 802.15.4-2006 standard which has a superframe of 16 equally sized slots of length 60 symbols each, a Contention Access Period (CAP), and a Contention Free Period (CFP) with up to 7 GTSs.

The optimizer block in Figure 5.1 on Page 51 is responsible for, according to the selected MAC scheme, informing the PAN coordinator of which GTS slot is assigned to which node in the next superframe. All nodes must have their radios ON at the beginning of each superframe in order to receive the beacon message from the PAN coordinator. This beacon message contains the information necessary to configure all nodes, especially those having allocated GTSs. Beacon information include among others the selected number of superframe slots S, the slot duration s_d, the beacon interval BI, and the superframe duration (SD). The sampling period of the control application is given by the selected value of BI which also sets the duration of the inactive period of the superframe depending on the selected value of SD. In order to guarantee a minimum level of control performance (e.g. stability), the BI value must be selected such that $BI = h \leq MATI$. The network parameters S, s_d, BI, SD, SO, and BO are related as follows:

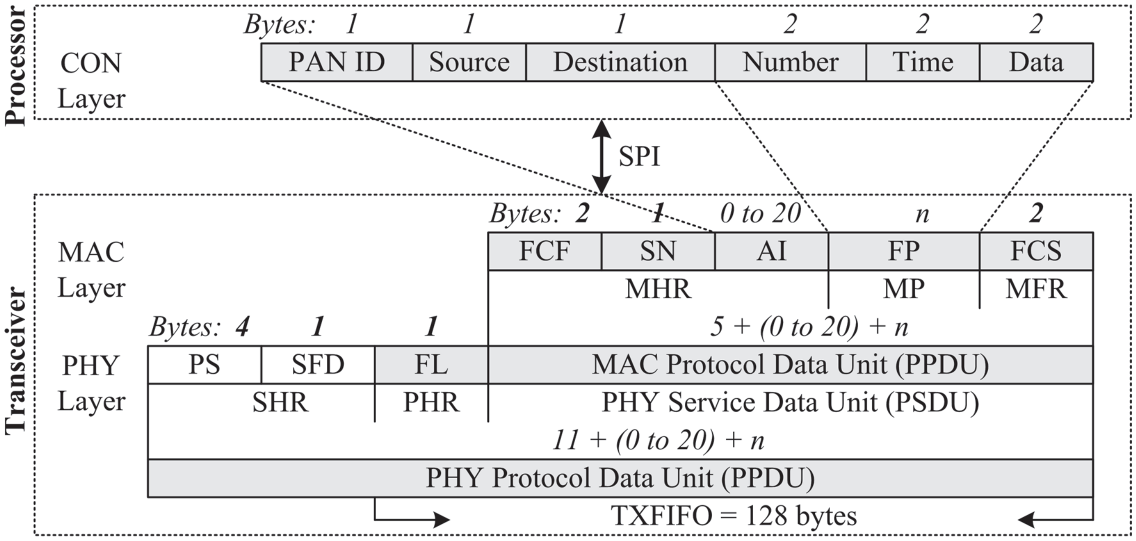
$$SD = s_d \cdot S \cdot 2^{SO} \tag{5.2}$$

$$BI = s_d \cdot S \cdot 2^{BO} \tag{5.3}$$

$$0 \leq SO \leq BO \leq 14 \tag{5.4}$$

where s_d is the slot duration, SO the superframe order, and BO the beacon order. The minimum achievable sampling period for $S = 20$ and $s_d = 1\ ms$ is $BI = h_{min} = 20\ ms$ ($BO = 0$). The corresponding maximum achievable sampling period is $BI = h_{max} = 327.68\ s$ ($BO = 14$).

5.2.4 Frame format



FCF: Frame Control Field
SN: Sequence Number
AI: Address Information
FP: Frame Payload
FCS: Frame Check Sequence

PS: Preamble Sequence
SFD: Start of Frame Delimiter
FL: Frame Length

MHR: MAC Header
MP: MAC Payload
MFR: MAC Footer
SHR: Sync Header
PHR: PHY Header

Figure 5.4: Schematic view of the IEEE 802.15.4 based WNCS data frame format

The schematic view of the WNCS data frame format is shown in Figure 5.4. This frame format is used for data frames, beacon frames, and MAC command frames. Coding and decoding schemes are used to reduce the amount of data to be sent over the wireless network resulting in a packet of total length 20 bytes including packet number and timestamp. For example, instead of sending real sensor and control values over the wireless network, only two bytes containing their quantized values are sent. Two bytes are more than enough since both the ADC and DAC only have a resolution of 12 bits. Upon reception of coded values, decoding is used to convert quantized values to real values. The small packet size allows the use of a single frame for sensor measurements and control values while up to 6 frames can be stored in the transmit buffer of the CC2420 transceiver.

Transceiver timing values are summarized in Table 5.1. The total theoretical transmission delay to transmit a single data frame of length 20 bytes from the transmitter to the receiver is $\tau_{TxFrame} = 1.2\ ms$. This delay includes the time required to transmit bytes of data between the processor and the transceiver over the SPI, the delay at the receiver, the delay until the CCA signal is valid, and the switching time between transmit and receive mode. Therefore, small sampling periods can be achieved depending on the selected control strategy, MAC scheme, and the total number of nodes in the network.

Operation	Delay
Transmit(1 byte)	$\tau_B = 32\ \mu s$
Switch(Tx/Rx)	$\tau_U = 192\ \mu s$
CCA valid	$\tau_{CCA} = 128\ \mu s$
Rx delay	$\tau_{Rx} = 32\ \mu s$
Processor/Transceiver(1 byte)	$\tau_{SPI} = 1.4\ \mu s$

Table 5.1: Transceiver timing parameters

5.2.5 Power consumption

The power consumption of the Imote2 wireless platform is given in Table 5.2. The table shows the amount of power needed by the Imote2 in different operating modes. As can be seen, computing, sending, and receiving messages are highly power demanding operations. The power consumption of the processor in the sampling mode M_4 is very high and the sampling mode duration should be kept relatively short in order to save energy. As far as the transceiver is concerned, receiving and listening (i.e. CCA detection) requires more power than sending with the maximum transmit power of 0 dBm. In order to save energy, the nodes should be kept in sleep mode (M_1) most of the time by means of duty cycling. This can be achieved by selecting large BI values with corresponding small SD values and reducing the listening time, reception and transmission times, and the computation time. However, large BI values are critical since they lead to an increase of the sampling period and a decrease of the control performance.

Mode	Description	Processor/Transceiver	Power Consumption
M_1	Power down	OFF/OFF	1.25 mW
M_2	Receive	OFF/Rx	63.3 mW
M_3	Transmit	OFF/Tx	29.3 mW @ -20 dBm, 58.7 mW @ 0 dBm
M_4	Sampling	ON/OFF	158.5 mW @ 104 MHz

Table 5.2: Power consumption of the Imote2 wireless hardware platform @ 3.3 V supply voltage and 104 MHz processor frequency

The processor of the Imote2 hardware platform is only used to control the ADC of the sensor nodes. No other computations are performed in the sensor nodes apart from coding and decoding messages to and from the controller node respectively. As a consequence, the resulting duration of the sampling mode M_4 is very small compared to the duration of the other modes and will be neglected in the sequel. Therefore, only transceiver power consumptions will be considered when deriving sensor node lifetime models. Due to the use of a wired connection between the controller and the actuator, the controller node is powered using the power supply of the actuator (i.e. both controller and actuator are not battery powered) and no lifetime model will be derived for the controller node although the approach is the same as for the sensor nodes.

5.3 MAC Protocols for WNCSs

In this section, three different MAC schemes to control the inverted pendulum over a wireless network are proposed. These MAC schemes are based on the protocol stack presented in Section 5.2 and include random as well as scheduled access schemes. The two scheduled access schemes considered are Master-Slave and TDMA. CSMA/CA is the only random access scheme that will be considered. Since the three outputs of the inverted pendulum are sampled by the sensor nodes and sent individually to the controller node, at least three data frames and one beacon frame are transmitted during each sampling period. However, the total number of transmitted frames is a function of the selected access scheme. Moreover, reliability and latency also depend on the underlying access scheme as will be seen subsequently. Following assumptions are made:

- Controller and actuator are collocated and the controller node is powered using the power supply of the actuator
- The transmission delay τ^{ca} between the controller and the actuator is negligible due to the wired connection
- The actuation delay τ^{a} is negligible
- For illustrative purposes and without loss of generality we set $S = 20$, $s_d = 1\ ms$, $SO = 0$, and $BO = 0$ (i.e. no inactive period)
- No acknowledgments are used

5.3.1 Master-Slave MAC protocol

The Master-Slave MAC protocol is a centralized scheduled access scheme where the access to the wireless network is controlled by the master (i.e. PAN coordinator) and the sensor nodes are the slaves. All communication resources are managed by the PAN coordinator and slaves only access the medium upon reception of an explicit request from the master. The superframe structure for the Master-Slave MAC scheme and the corresponding transceiver activities of the sensor nodes are depicted in Figure 5.5 on Page 57. The minimum achievable sampling period is $h_{min} = 20\ ms$. The beacon frame is sent by the master during the first time slot and contains all configuration information for the sensor nodes (e.g. sampling period). Upon reception of the beacon frame, sampling is performed during the second time slot and all sensor nodes remain in receive mode waiting for the master request. Each sensor node then sends its data to the master after receiving a corresponding request from the master and goes to sleep for the remaining time. Upon reception of all sensor values, the control value is computed by the master and transmitted to the actuator. A total of seven frames (one beacon frame, three sensor frames,

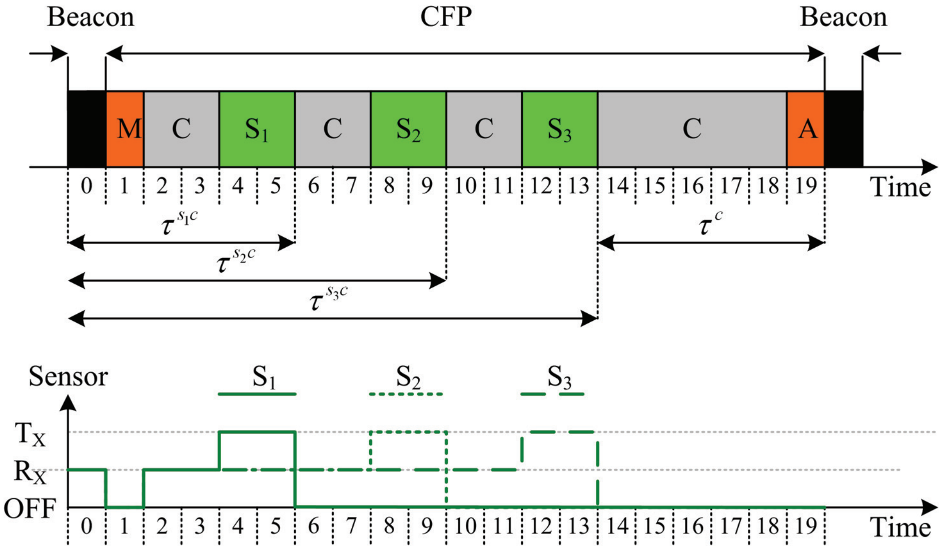


Figure 5.5: Superframe structure (top) and corresponding sensor node transceiver activities (bottom) for the Master-Slave MAC scheme

and three controller frames) are transmitted during each sampling period. Due to the scheduled nature of the Master-Slave MAC protocol, the resulting delay round-trip time delay τ^{rt} from sensing to actuation is constant and given by:

$$\tau^{rt} = \tau^{sc} + \tau^{c} \tag{5.5}$$

$$\tau^{sc} = \max\left\{\tau^{s_ic}\right\},\ i \in \{1, ..., m\} \tag{5.6}$$

where τ^c is the computational delay in the controller node and τ^{s_ic}, $i \in \{1, ..., m\}$ the sum of the processing and transmission delays of sensor node i. The vector of decision variable $\mathbf{x}$ of the constrained optimization problem (5.1) for the Master-Slave MAC protocol can be selected as $\mathbf{x} = [SO,\ BO,\ S,\ s_d]$. If we denote by s_i the number of time slots used by node i during a sampling period, then the Master-Slave MAC is constrained by the equation below:

$$\frac{1}{h}\sum_{i=1}^{N} s_i s_d \leq 1 \tag{5.7}$$

$$\sum_{i=1}^{N} s_i s_d = \tau^{rt} \tag{5.8}$$

where N is the total number of nodes in the network and s_i the number of time slots used by node i. $\frac{s_i s_d}{h}$ can be interpreted as the relative occupation of the wireless medium by node i during a sampling period. Equation (5.7) corresponds to the latency constraint in the optimization problem (5.1) for the Master-Slave MAC protocol where $E[D(\mathbf{x})] = \sum_{i=1}^{N} s_i s_d$ and $D_{max} = h$.

One of the main advantage of the Master-Slave MAC scheme beside the ease of implementation and easy inclusion of new wireless nodes in the network is the fact that no slot allocation mechanism is necessary and more importantly, no time synchronization is required. Since only one node is allowed to access the network at each time instant, there are no collisions. One of the drawbacks of this scheme is the increased latency due to the huge number of packets transmitted in each sampling period. Another problem is the increased energy consumption of the sensor nodes which should be awake until a request from the master is received.

5.3.2 TDMA MAC protocol

The TDMA MAC protocol is a scheduled access scheme where the access to the wireless network is controlled by the PAN coordinator. The superframe structure for the TDMA MAC scheme and the corresponding transceiver activities of the sensor nodes are depicted in Figure 5.6. Time slots are allocated during setup or at runtime to individual nodes by the PAN coordinator and nodes can only transmit data during the allocated time slots. Therefore, at any given time instant, only one single node is allowed to transmit, thus, avoiding collisions. However time synchronization is required in order to ensure that all nodes have the same reference time and transmission slots do not overlap. Synchronization information is sent to the nodes in each beacon by the PAN coordinator. This information is used by the nodes to update their clocks and correct eventual clock offsets and drifts.

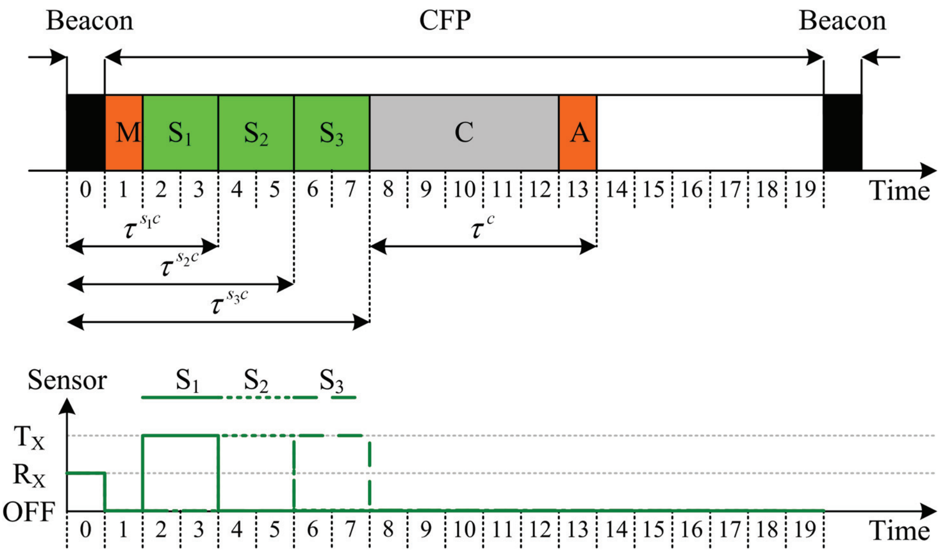


Figure 5.6: Superframe structure (top) and corresponding sensor node transceiver activities (bottom) for the TDMA MAC scheme

The major drawback of the TDMA MAC scheme is its scalability since a time slot must be assigned to each node and the latency increases with the number of nodes in the network. A slot allocation mechanism is required in the TDMA MAC scheme to allocate time slots to specific nodes. The slot allocation mechanism must be updated when new nodes are introduced into the network or when nodes leave the network. A total of four frames (one beacon frame and three sensor frames) are transmitted during each sampling period resulting in a smaller delay from sensing to actuation than with the Master-Slave MAC scheme. Only thirteen time slots are required and a value of $S = 13$ can be used resulting in a minimum sampling period $h_{min} = 13\ ms$. If we denote by s_i the number of time slots allocated to node i during a sampling period, then the slot allocation algorithm is constrained by the following equation:

$$\frac{1}{h}\sum_{i=1}^{N} s_i s_d \leq 1 \tag{5.9}$$

$$\sum_{i=1}^{N} s_i s_d = \tau^{rt} \tag{5.10}$$

where N is the total number of nodes in the network and s_i the number of time slots allocated to node i. $\frac{s_i s_d}{h}$ can be interpreted as the fraction of the sampling period during which node i is scheduled to access the wireless medium. As for the Master-Slave MAC protocol, the resulting delay τ^{rt} from sensing to actuation is constant and given by equation (5.5). The vector of decision variable $\mathbf{x}$ of the constrained optimization problem (5.1) for the TDMA MAC protocol can be selected as $\mathbf{x} = [SO,\ BO,\ S,\ s_d,\ s_i]$. Equation (5.9) corresponds to the latency constraint in the optimization problem (5.1) for the TDMA MAC protocol where $E[D(\mathbf{x})] = \sum_{i=1}^{N} s_i s_d$ and $D_{max} = h$.

5.3.3 CSMA/CA MAC protocol

The CSMA/CA MAC protocol is a random access scheme where the access to the wireless network is not directly controlled by the PAN coordinator. The superframe structure of the CSMA/CA MAC protocol and the corresponding transceiver activities of the sensor nodes are depicted in Figure 5.7. Beacon frames are transmitted periodically by the PAN coordinator and upon reception of the beacon frame, sampling is performed at the sensor nodes. All time slots are accessed by contention during the CAP period of the superframe. When a sensor node wants to transmit a frame during the CAP period, it waits for a random period of time r_b and then senses the channel for a time duration c_s until the channel is free to send its frame. Frame collisions are possible due to the random nature of the CSMA/CA access scheme and may result in random packet losses and delays. If no collisions occur during the CAP period, then a total of four frames are transmitted during one sampling period (one beacon frame and three sensor frames). The worst case scenario happens if no sensor node accesses the network during the CAP period. In this case only the beacon frame is transmitted. The algorithm of the CSMA/CA access scheme is depicted in Figure 5.8 on Page 60 and the corresponding parameters are summarized in Table 5.3 on Page 61.

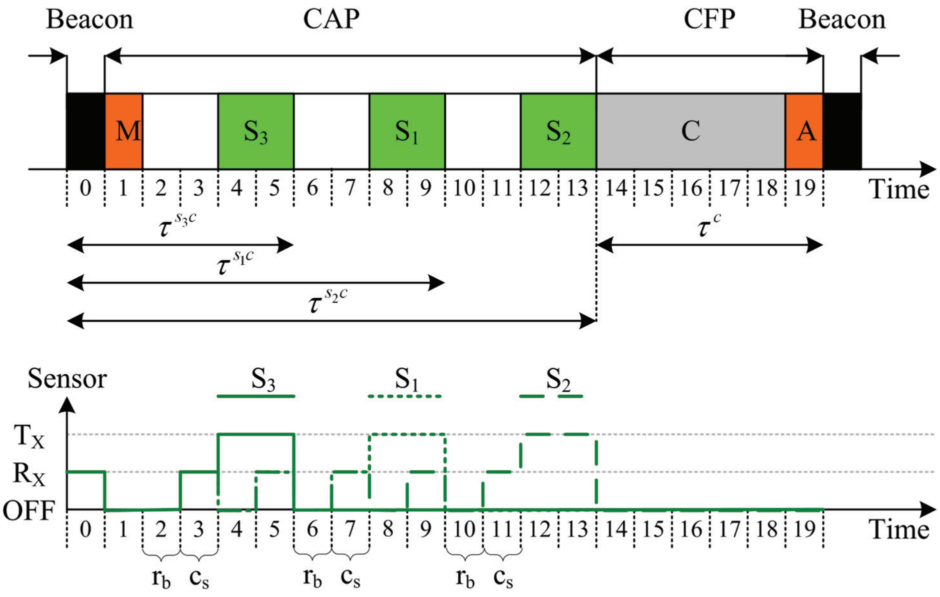


Figure 5.7: Superframe structure (top) and corresponding sensor node transceiver activities (bottom) for the CSMA/CA MAC scheme

The implementation of the algorithm is not straightforward and duty cycling is not easy to realize during the CAP period of the superframe due to contention. The parameters of the CSMA/CA algorithm that may influence energy efficiency, reliability, and timeliness are the minimum value of the backoff exponent $macMinBE$, the maximum number of backoff periods before declaring a channel access failure $macMaxCSMABackoffs$, and the maximum number of retries allowed after a transmission failure $macMaxFrameRetries$ that each node can select. Tuning is difficult because simple and accurate models of the influence of these parameters on the probability of successful packet transmission, packet delay, and energy consumption are difficult to derive.

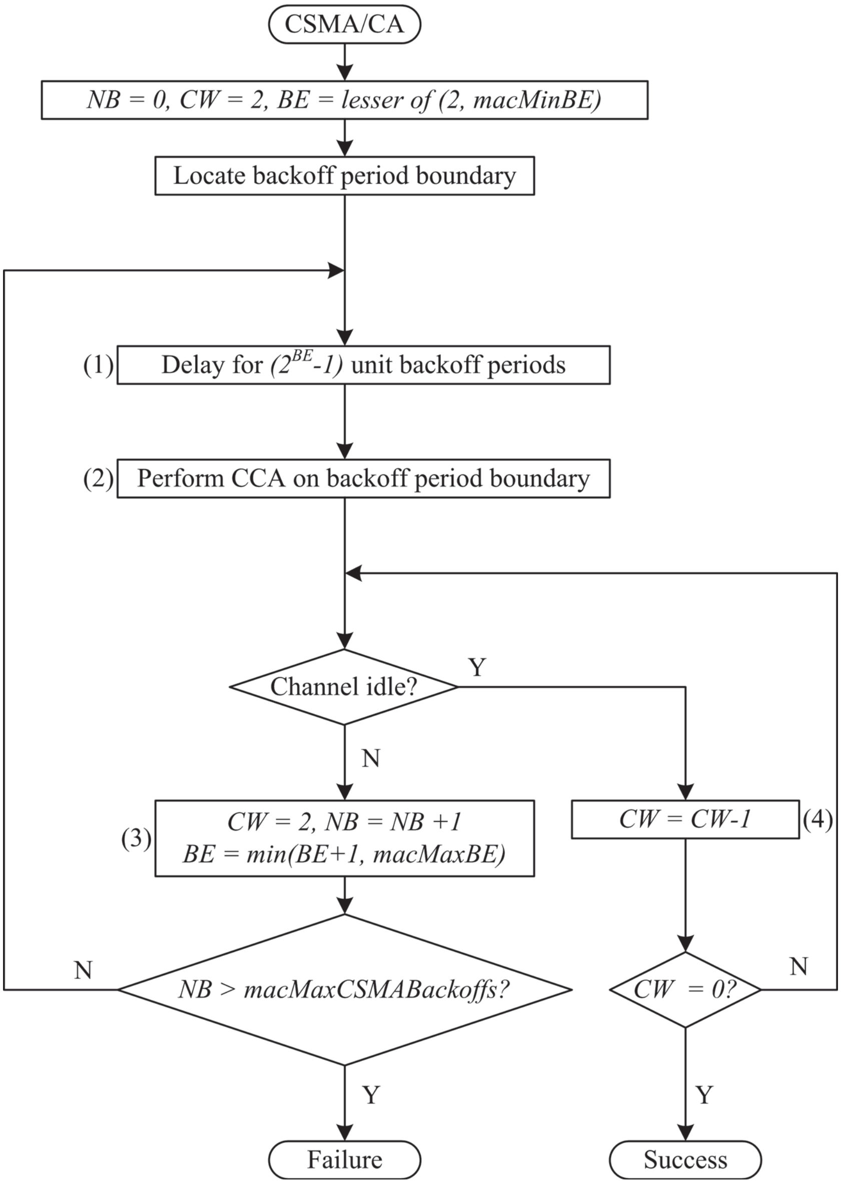

CSMA/CA
NB = 0, CW = 2, BE = lesser of (2, macMinBE)
Locate backoff period boundary
(1) Delay for (2^BE-1) unit backoff periods
(2) Perform CCA on backoff period boundary
Channel idle?
Y
N
(3) CW = 2, NB = NB +1
BE = min(BE+1, macMaxBE)
CW = CW-1 (4)
NB > macMaxCSMABackoffs?
N
Y
CW = 0?
N
Y
Failure
Success

Figure 5.8: CSMA/CA algorithm for the slotted beacon-enabled mode

Therefore, a simple model is used to model these influences resulting in the following equation:

$$\frac{1}{h}\sum_{i=1}^{N}\alpha_i s_i s_d \leq 1 \tag{5.11}$$

$$\sum_{i=1}^{N}\alpha_i s_i s_d = \tau^{rt} \tag{5.12}$$

where $\alpha_i = f\,(macMinBE,\ macMaxCSMABackoffs,\ macMaxFrameRetries)$ is the access probability of node i and N the number of contending nodes in the network. The vector of decision variable $\mathbf{x}$ of the constrained optimization problem (5.1) for the CSMA/CA MAC protocol can be selected as $\mathbf{x} = [SO,\ BO,\ S,\ s_d, c_1,\ c_2,\ c_3]$ with $c_1 = macMinBE$, $c_2 = macMaxCSMABackoffs$, and $c_3 = macMaxFrameRetries$. Equation (5.11) can be interpreted as the latency constraint in the optimization problem (5.1) for the CSMA/CA MAC protocol where $E[D(\mathbf{x})] = \sum_{i=1}^{N}\alpha_i s_i s_d$ and $R_{max} = h$.

So far simple mathematical expressions for the latency constraints of the proposed MAC schemes were derived. In the next section, we will derive mathematical formulations for the corresponding reliability constraints.

Parameter	Value
IFS duration	$\tau_{IFS} = 320\ \mu s$
Contention Window	$CW \in [0, ..., 5],\ \tau_{CW} = \tau_{IFS}$
Random Backoff	$r_b = [0, 2^{BE} - 1]$
Backoff Exponent	$BE = [3, 5]$
Carrier Sense	$c_s = 128\ \mu s$
$macMinBE$	$\in [0, ..., 3]$
$macMaxBE$	$\in [0, ..., 5]$
$macMaxCSMABackoffs$	$\in [0, ..., 4]$
$macMaxFrameRetries$	$\in [0, ..., 3]$

Table 5.3: CSMA/CA MAC protocol parameters

5.4 Power control algorithm

The power control algorithm developed for the control structure in Figure 5.1 on Page 51 and presented in this section is based on an estimation of the Packet Error Rate (PER). The transmission power P_i of sensor node i is adapted as a function of the sensor-to-controller distance or the channel conditions between sensor and controller. Figure 5.9 shows a model of the considered power control loop between the controller node and a sensor node where Ω_i is the path loss coefficient, Θ_i the additive interference power, η_i the noise power spectral density, $\tilde{\gamma}_i$ the estimate of the Signal to Interference plus Noise Ratio (SINR), and γ_r the reference SINR value chosen such as to obtain a low PER from the sensor to the controller. The difference between the estimated SINR $\tilde{\gamma}_i$ and the desired SINR γ_r is used by the controller to compute the desired transmit power level whose quantized value is sent to the sensor node over the wireless network in the next beacon frame. This power level is used by the sensor node for subsequent transmissions until a new value is received. No extra communication is needed between controller and transmitter since computed power control levels are sent periodically to the sensor nodes by the PAN coordinator in each beacon frame. Also, SINR estimates $\tilde{\gamma}_i$ are computed periodically by the controller for each received sensor packet using RSSI measurements as will be seen in the sequel.

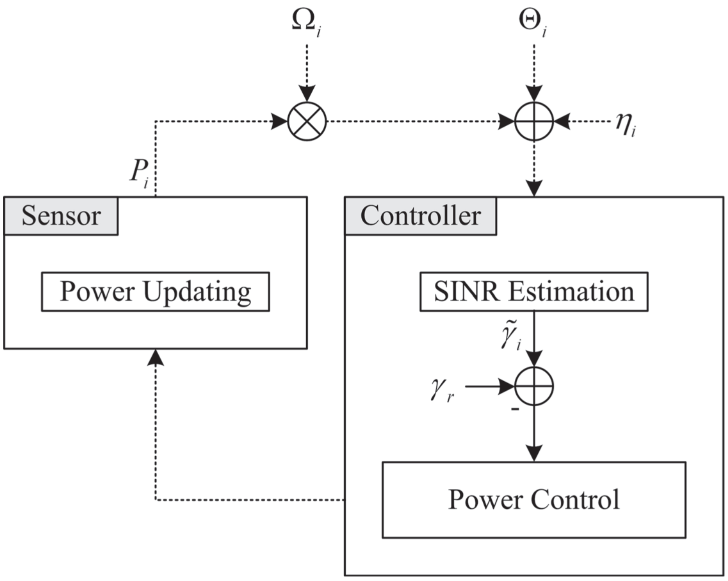


Figure 5.9: Single power control loop with transmitting node (sensor) and receiving node (controller)

5.4.1 Wireless radio model

A model of the SINR of the wireless communication link between the sensor and the controller is required in order to implement a power control algorithm for the power control loop in Figure 5.9. Following modeling assumptions based on the characteristics of the Imote2 hardware platform are made:

- The transmit power level used by a generic transmitting node i is denoted as $\{P_{il}\}_{l=\{1,\dots,n\}}$ where $n = 8$ is the number of power levels of the Imote2 transceiver with corresponding transmit powers $P_{il}|_{dBm} = \{0, -1, -3, -5, -7, -10, -15, -25\}$
- A fixed bandwidth $W = 2.4\ GHz$ is allocated for each communication link
- A fixed data rate $R_b = 250\ kbit/s$ is used for all transmissions resulting in a bit time $T_b = 4\mu s$
- Only few nodes interfering
- Bit-to-bit error independence
- Additive White Gaussian Noise (AWGN) wireless channel

Under above assumptions, the signal Z_i received by node i can be expressed as a function of the desired signal D_i, the interference signal I_i, and a Gaussian random variable N_g with zero mean and variance η_i representing thermal noise as follows:

$$Z_i(t) = D_i(t) + I_i(t) + N_g(t) \tag{5.13}$$

$$D_i(t) = \sqrt{P_i(t)\Omega_i(t)} \tag{5.14}$$

$$E\left[I_i^2(t)\right] = \frac{1}{3G} \sum_{j=1, j\neq i}^{n} \nu_i(t) P_j \Omega_j(t) + I_n(t) \tag{5.15}$$

where $G = \frac{W}{R_b}$ is the channel gain and $I_n(t)$ the interferences of the other channels. $\nu_i(t)$ is a binary random variable that models the transmission attempts of the nodes (i.e. the MAC scheme). $\nu_i(t) = 1$ if a node is allowed to transmit and $\nu_i(t) = 0$ otherwise. Moreover, $P[\nu_i = 1] = \alpha_i$ and $P[\nu_i = 0] = 1 - \alpha_i$. The path loss coefficient Ω_i is defined as:

$$\Omega_i(t) = PL_i(t) e^{\xi_i(t)} \tag{5.16}$$

$$PL_i(t)|_{dB} = -PL_i(d_r)|_{dB} - 10n log_{10}(d_i(t)/d_r) \tag{5.17}$$

where $PL_i(t)$ is the path loss, $e^{\xi_i(t)}$ the shadow fading component of the path, $\xi_i(t)$ a Gaussian random variable, $PL_i(d_r)$ the path loss at the reference distance d_r, d_i the distance between transmitter and receiver, and n the path loss decay constant. The quality of the received signal Z_i is its SINR which can be derived as follows:

$$\gamma_i(t) = \frac{P_i(t)\Omega_i(t)}{\frac{N_0}{2T_b} + \frac{1}{3G}\sum_{j=1, j\neq i}^{n} \nu_i(t) P_j \Omega_j(t) + I_n(t)} \tag{5.18}$$

where N_0 is the noise spectral density and T_b the bit time. Instead of the required SINR measurements, the Imote2 wireless hardware provides RSSI measurements which can be used to compute estimates of the SINR. Using an approach similar to the one in [Ares *et al.* , 2007] the SINR can be derived from the available RSSI measurements using following equations:

$$RSSI_i(t) = \underbrace{P_i|_{dB}}_{Transmitted\ power} + \underbrace{PL_i(t)|_{dB}}_{Path\ loss} + \underbrace{\varphi\xi_i(t)}_{Shadowing} + \underbrace{\Theta_i(t)}_{Interferences} + C + \delta(t) + 30 \tag{5.19}$$

$$\tilde{\gamma}_i(t) \approx 10^{(RSSI_i(t) - \eta_i - C - 30)/10} \tag{5.20}$$

where $\Theta_i(t)$ represents the interfering powers (which are neglected), $C = 45\ dB$ the measurement offset, $\varphi = 10log_{10}e$, and $\delta(t) = \pm 6\ dB$ the measurement uncertainty.

5.4.2 PER power control

The nodes optimal transmit power can be derived by setting a constraint on the desired node PER of the form:

$$PER \leq PER_{max} \tag{5.21}$$

which corresponds to the reliability constraint of the general optimization problem (5.1) where $R(\mathbf{x}) = PER$ and $R_{max} = PER_{max}$. Assuming bit-to-bit error independence, the packet error probability $PER(i) = F(P_i)$ of node i using a transmit power P_i can be formulated as follows [Proakis, 1995]:

$$F(P_i) = 1 - [1 - F_b(P_i)]^L \tag{5.22}$$

where L is the total number of bits in a packet and $F_b(P_i)$ the average bit error probability which depends on the modulation scheme (O-QPSK for the Imote2) and the wireless propagation statistics. Assuming an AWGN wireless channel, the average bit error probability for the Imote2 is given by the following equation:

$$F_b(P_i) = \frac{1}{2}Q(\sqrt{\bar{\gamma}(P_i)}) \tag{5.23}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt$ is the complementary standard Gaussian distribution and $\bar{\gamma}(P_i)$ the average SINR.

The SINR has a log normal distribution (see 5.20) and its average can be computed as follows [Papoulis, 1984]:

$$\bar{\gamma}(P_i) = e^{\mu_\gamma + \sigma_\gamma^2/2} \tag{5.24}$$

$$\mu_\gamma = 2lnM_i^{(1)}(P_i) - \frac{1}{2}lnM_i^{(2)}(P_i) \tag{5.25}$$

$$\sigma_\gamma^2 = lnM_i^{(2)}(P_i) - 2lnM_i^{(1)}(P_i) \tag{5.26}$$

where $M_i^{(1)}$ and $M_i^{(2)}$ are statistical expectations of the first two moments of the SINR. The average SINR $\bar{\gamma}_p$ can be derived from (5.22) and (5.23) by setting a constraint of the PER (e.g. $F(P_i) \leq p$). The transmit power is then computed in the controller such that the average SINR computed using equations (5.24)-(5.26) is met leading to the following equations:

$$\tilde{M}_i^{(1)}(nh) = \frac{1}{M}\sum_{j=1}^{M}\tilde{\gamma}_i(nh+j) \tag{5.27}$$

$$\tilde{M}_i^{(2)}(nh) = \frac{1}{M}\sum_{j=1}^{M}\tilde{\gamma}_i^2(nh+j) \tag{5.28}$$

where h is the time period of the power control (same as the sampling period of the control application) and M the number of RSSI samples collected during one power control period. Assuming $\tilde{\gamma}(nh) = \bar{\gamma}(nh)$, the transmit power can be updated using following equation:

$$P_i(nh+h) = \bar{\gamma}_p\frac{P_i(nh)}{\bar{\gamma}(nh)} \tag{5.29}$$

Following simplifications are adopted to reduce the computational complexity of the power control algorithm:

- The variance of the power control algorithm in the AWGN environment is neglected and (5.25) reduces to $\bar{\gamma}(P_i) = M_i^{(1)}(P_i)$
- An array is used to simplify the computation of the complementary standard Gaussian distribution Q in (5.23)
- Power updating commands are sent at low rate since significant variations of the average SINR are not frequent

In order to characterize the node energy consumption, the power control gain g_i for node i is defined as follows:

$$g_i = \frac{P_{max,i}}{P_{avg,i}} \tag{5.30}$$

where $P_{max,i} = 0\ dBm$ and $P_{avg,i}$ the average power consumption of node i. A high g_i and a low PER indicates a good performance of the power control algorithm. Power control results are summarized in Table 5.4 for indoor measurements. Link 1 is the wireless link between the position sensor and the controller. Link 2 is the wireless link between the angle sensor and the controller. Link 3 is the wireless link between the velocity sensor and the controller. An estimate of the PER of the sensor nodes is computed by the controller which expects periodic sensor values. The experiment was run for a duration of 10 minutes which is long enough to average out wireless channel variations. Sensor nodes and the controller node are fixed with line of sight, thus yielding the selected AWGN wireless channel model.

Link	g_i	PER [%]
1	7.42	1.06
2	10.33	2.17
3	10.33	0.45

Table 5.4: Power control gain and corresponding PER for a beacon period of 2s and required $PER_{Max} = 3\%$

A good performance is obtained both in terms of energy efficiency and PER which shows the effectiveness of the proposed power control algorithm. Different g_i values may be obtained for selected values of PER_{Max}.

5.5 Node lifetime models

The lifetime of wireless nodes in WNCSs is a complex function of the control algorithm, controller parameters (e.g. sampling period), communication protocols (e.g. MAC protocols), and protocol parameters (e.g. delays and packet losses). In this section we propose and derive a lifetime model for sensor nodes in WNCSs. Details on the derivation of the node lifetime can be found in [Chamaken & Litz, 2011]. It is assumed that all sensor nodes are powered by batteries and have different energy saving modes. A combination of the energy modes considered for the CC2420 transceiver are shown in Figure 5.10 on Page 65 including transition probabilities between the different modes. The lifetime model includes not only the energy consumption in each mode but also the energy consumption during the transition between the modes. The corresponding energy consumptions are summarized in Table 5.5 on Page 65.

Following assumptions are further made:

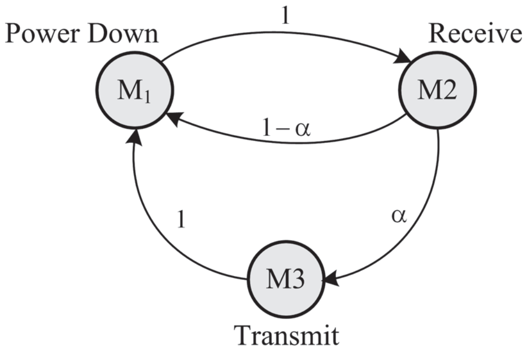


Figure 5.10: Transceiver activities for a generic sensor node during a sampling period

1. The first-order statistical properties (e.g. mean value) of all random quantities (e.g. transmission time) are known by observation and experiment
2. Radio-transmission times are *i.i.d.*[3] with arbitrary distributions
3. When a beacon frame is received the sensor node processes it and sends its sampled value to the controller node with probability α
4. During the sampling period, the radio visits a limited number of listen (Rx) states
5. All power consumptions are constant during an operation (e.g. transmission) and a fixed amount of energy is required to turn the radio ON and OFF

Mode	Radio	Power [mW]	Activity
M_1	PD	0.07	Power Down
M_2	R_X	63.3	Receive
M_3	T_X	29.3 - 58.7	Transmit
Transition	Radio	Energy $[\mu Wh]$	Time $[ms]$
M_{12}	PD$\rightarrow R_X$	0.031	2.82
M_{23}	$R_X \rightarrow T_X$	0.0018	0.98

Table 5.5: Imote2 energy consumption in different modes

As a result of the above assumptions, power state transitions can be modeled by a Semi-Markov chain. The resulting Semi-Markov chain model can be used to compute the node lifetime of wireless nodes using following Semi-Markov chain properties:

$$\pi_j = \sum_i \pi_i P_{ij}, \ \sum_j \pi_j = 1 \tag{5.31}$$

$$P_i = \frac{\pi_i \mu_i}{\sum_j \pi_j \mu_j} \tag{5.32}$$

$$E_{M_i} = p_i T P_{M_i}, \ E_{M_{ij}} = C_{M_{ij}} \bar{\eta}_{ij}(T) \tag{5.33}$$

$$\bar{\eta}_{ij} = \frac{\pi_j T}{D}, \ \bar{\eta}_{ji} = \frac{\pi_i T}{D} \tag{5.34}$$

$$\sum_{0 \leq m \leq k} \left[\underbrace{(T_W + \epsilon_m)\,\bar{P}_{W,m}}_{Awake} + \underbrace{(T_S - \epsilon_m)\,\bar{P}_{S,m}}_{Asleep} + \underbrace{C_{SW,m} + C_{WS,m}}_{Transition} \right] \leq E_{Tot} \tag{5.35}$$

[3] i.i.d.: Identically and Independently Distributed

Following results are obtained for a generic sensor node whose Semi-Markov chain model is depicted in Figure 5.10 on Page 65:

$$\pi = \begin{pmatrix} 0 & 1 & 0 \\ 1-\alpha & 0 & \alpha \\ 1 & 0 & 0 \end{pmatrix} \tag{5.36}$$

$$\pi_1 = \pi_2 = \frac{1}{2+\alpha},\ \pi_3 = \frac{\alpha}{2+\alpha} \tag{5.37}$$

$$\mu_1 = \bar{T}_{PD},\ \mu_2 = \bar{T}_{R_X},\ \mu_3 = \bar{T}_{T_X} \tag{5.38}$$

$$D = \mu_1 + \mu_2 + \alpha\mu_3 \tag{5.39}$$

$$\bar{P}_W = \frac{P_{M_2}\bar{T}_{R_X} + P_{M_3}(l)\alpha\bar{T}_{T_X} + \frac{\alpha}{3+\alpha}E_{M_{23}}}{\bar{T}_{R_X} + \alpha\bar{T}_{T_X}} \tag{5.40}$$

$$\overline{T} \leq \frac{E_{Tot}}{\left[\bar{P}_W(\alpha,l)T_W + P_{M_1}(h-T_W) + (\bar{P}_W(\alpha,l) - P_{M_1})\bar{\epsilon} + E_{M_{12}}\right]}h \tag{5.41}$$

Corresponding parameters are summarized in Table 5.6.

Parameter	Meaning
π_i	stationary probability of mode i
p_i	steady state probability of mode i
p_{ij}	transition probability from mode i to mode j
μ_i	average time spent in mode i
D	cycle duration
T_W	duration of the awake period
T_S	duration of the asleep period
P_W	power consumption of the awake period
P_S	power consumption of the asleep period
T	Node lifetime
P_{M_i}	Power consumption of mode i
E_{M_i}	Energy consumption of mode i
E_{Tot}	energy budget of a node
$C_{M_{ij}}$	transition energy of mode i to mode j
η_{ij}	number of transitions from mode i to mode j
l	transmit power level $l = \{1, ...8\}$

Table 5.6: Parameters of the Semi-Markov chain in Figure 5.10 on Page 65

Equation (5.41) provides an estimate (upper bound) of the sensor node lifetime which is a function of the selected transmit power level l set by the PER power control algorithm, the access probability α resulting from the selected MAC protocol, and the sampling period h of the control application. Instead of minimizing the total node energy consumption E_{tot} in the optimization problem (5.1), the average node lifetime $\overline{T}$given by (5.41) can be maximize resulting in the new formulation of the optimization problem as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \bar{T}(\mathbf{x}) + J_{control}(\mathbf{x}) && \text{(5.42a)} \\ s.t. \quad & R(\mathbf{x}) \geq R_{min} && \text{(5.42b)} \\ & E[D(\mathbf{x})] \leq D_{max} && \text{(5.42c)} \\ & \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}. && \text{(5.42d)} \end{aligned}$$

Figure 5.11 on Page 67 shows sensor node lifetime simulation results for different configurations of the above parameters. Sensor nodes are assumed to be powered by two AAA batteries with

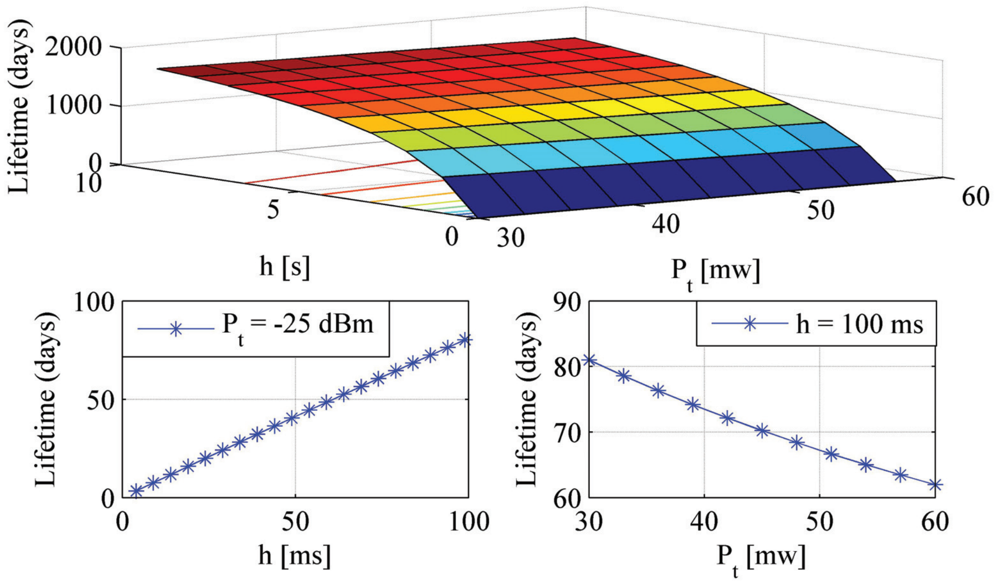


Figure 5.11: Lifetime simulation results for a generic sensor node

a total capacity of 1150 mAh with a supply voltage of 3.3 V. At the lowest transmit power of -25 dBm, a maximum node lifetime of 4 days can be achieved if the Imote2 is always in receive mode and no duty cycling is used (fast sampling). The node lifetime increases when slow sampling periods are used and can reach approximately 3 years at a sampling period $h = 10$ s. However the slower the sampling period, the worse the control performance. Hence, a tradeoff should be made between, node lifetime and control performance. This issue will be addressed in the next chapter.

For the sake of completeness it should be noted however that, when wireless nodes are not too far from each other, the PER power control scheme consumes more energy than a scheme with fixed power amplitude plus a selected margin to ensure an acceptable bit error rate given poor channel conditions. The energy consumption of the radio and processing circuitry dominates the energy consumption of the radio circuitry at short distances between the nodes. The energy saved by decreasing the transmission power under good channel conditions is wasted by the extra communication used to select the appropriate transmit power level. Power control should therefore be considered when the transmission power dominates the circuit energy consumption or in the case of distant nodes.

5.6 Experimental results

Experimental results presented in this section were obtained with the WNCS structure in Figure 5.12 on Page 68. Controller and actuator are collocated and sensor nodes are at equal distance from the controller node. A movable obstacle is placed between the sensor nodes and the controller such that there is no line of sight (NLOS) between the sensor nodes and the controller node. A constant transmit power of 0 dBm is used in all experiments (i.e. no PER power control) and the main focus will be on delay and packet loss measurements. In the line of sight case (LOS), delays are almost constant and packet losses are negligible. Therefore, only experimental results for the NLOS case will be presented and discussed in the sequel.

Sensor delay and packet loss information are stored by the controller at each sampling instant and can be used to estimate the performance of the wireless network at runtime. To this end, we define a performance metric J_{net} to measure and assess the network QoS over a defined period

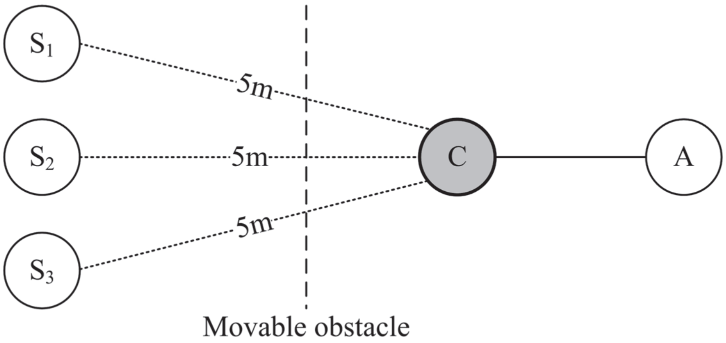


Figure 5.12: WNCS structure used in the experiments: S1 (Position), S2 (Angle), S3 (Velocity), C (Controller), A (Actuator)

of time. This performance metric is approximated by the following equation

$$J_{net} = \left(\underbrace{\frac{\lambda}{k_f} \sum_{k=1}^{k_f} \frac{\tau_k^{sc}}{h}}_{Delay} + \underbrace{\frac{1-\lambda}{k_f} \sum_{k=1}^{k_f} \beta_k}_{Loss} \right) \cdot 100, \;\; \lambda \in [0,1]\,, \;\; \beta_k \in \{0,1\}\,, \;\; \tau_k^{sc} \leq h. \tag{5.43}$$

where k_f is the measurement duration selected as an integer multiple of the sampling period h, τ_k^{sc} the sensor to controller delay which is known at each sampling instant k if the controller expects periodic sensor measurements, β_k the sensor packet loss at sampling instant k, and λ a weighting factor. Small values of the wireless network performance metric $0 < J_{net} < 100$ correspond to better network QoSs.

MAC	RTT [ms]	$\sigma_{RTT}[ms]$	ρ [%]	σ_ρ [%]	$\overline{J}_{net}$
Master-Slave	19.12	0.55	1.33	0.13	33
TDMA	13.20	0.97	1.57	0.46	42
CSMA/CA	13.00	2.12	1.61	0.41	55

Table 5.7: Experimental delay and packet loss measurements for the NLOS case at a sampling period $h = 20ms$ for all MAC schemes. The average network performance metric $\overline{J}_{net}$ is measured over a time period of 10 minutes

Experimental delay and packet loss measurements are summarized in Table 5.7 for different MAC protocols. Relatively small delays and packet loss rates are obtained for all MAC schemes. This is due to the fact that only few nodes are used in the wireless network and there are almost no packet losses due to congestion or contention. Most of the packet losses are caused by delays larger than the selected sampling period. The minimum achievable sampling period using the Master-Slave scheme is $h \geq 20\ ms$. As far as packet losses are concerned, the Master-Slave scheme performs better. Smaller sampling periods can be achieved with the TDMA and the CSMA/CA schemes (i.e. $h = 14\ ms$) but at the expense of increased packet losses. As can be seen from the average value of the wireless performance metric $\overline{J}_{net}$, better QoSs are obtained with the Master-Slave scheme followed by the TDMA scheme.

Measured transmission delays for the different MAC schemes are shown in Figure 5.13 on Page 69 with corresponding distributions. As can be seen, delay distributions can be approximated by a Gaussian distribution. Similar observations are made for packet loss distributions which can also be approximated using a Gaussian distribution. This will play a key role in the selection and implementation of optimal control and estimation strategies for WNCSs in the next chapter.

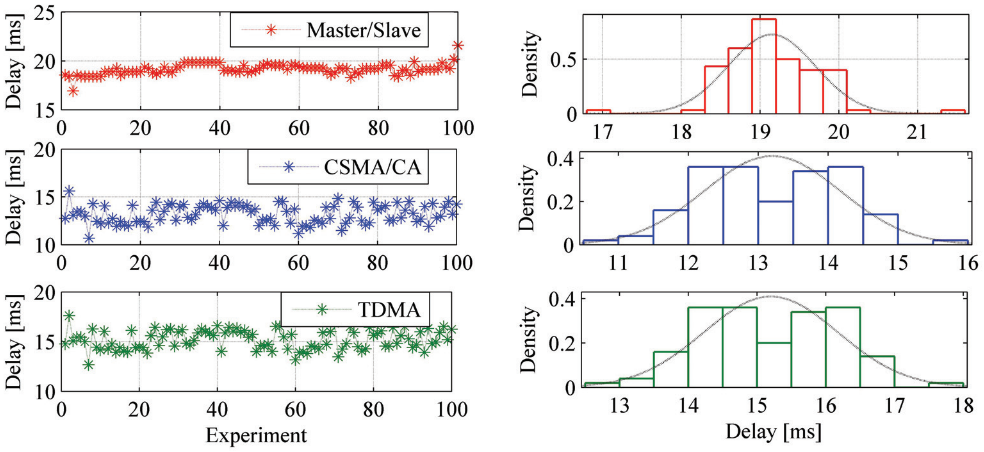


Figure 5.13: Experimental delay measurements for different MAC protocols

5.7 Summary

In this chapter we proposed, implemented, and tested a layered and flexible communication structure for the protocol stack of WNCSs based on the IEEE 802.15.4 standard. The proposed communication protocol stack for WNCSs include several adaptive MAC protocols with tunable parameters. These MAC protocol parameters can be tuned offline and online to minimize the energy consumption of the network (i.e. maximize the node lifetime) while guaranteeing reliability and latency constraints to meet a desired control performance. The proposed communication protocols are easy to implement and required less computing and energy resources. A power control algorithm based on a desired PER and available RSSI measurements is used to control the transmit power of the wireless nodes and reduce their energy consumption, thus increasing node lifetime. Furthermore, a Semi-Markov chain based node lifetime model was derived and used to assess the node lifetime of wireless nodes under different network conditions.

The proposed communication algorithms including their functionalities were implemented and tested on the inverted pendulum testbed. Simulation and experimental results presented in this chapter showed not only the effectiveness of the proposed communication algorithms but also their flexibility due to the availability of tunable parameters at different protocol layers including the control algorithm. For each of the proposed MAC protocols, simple mathematical expressions for the reliability and latency constraints in the optimization problem (5.1) where derived. These mathematical expressions are of great importance when solving cross-design optimization problems as will be seen in the next chapter.

6 Proposed cross-design framework

6.1 Introduction

The problem we aim to solve in this chapter is that of performing control over a dedicated wireless communication network. The main goal is to find solutions for control loops closed over wireless communication networks that guarantee not only control performance and stability, but also minimize the total energy consumption of the wireless network, thus, increasing the network lifetime. Abstractions of the mathematical models of the communication protocols developed in Chapter 5 are used in conjunction with optimal control design techniques to investigate the essential issues of the WNCS cross-design approach. Above mentioned optimal control design techniques are applied in order to derive a closed-loop control performance metric for WNCSs as a function of delay and packet loss requirements. To this end, the achievable control performance is obtained from a combination of HiL simulations and experiments. The proposed cross-design optimization approach is based on the constrained multi-objective optimization problem formulated in Chapter 5 for which the objective functions are the control performance J_{con} and the average wireless node lifetime $\overline{T}$ both constrained by delay and packet loss probability requirements, which are derived from the desired control cost. Moving the objective function of the control application J_{con} to the constraints leads to the following single-objective optimization function which will be solved in this chapter:

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \overline{T}(\mathbf{x}) && \text{(6.1a)}\\
s.t. \quad & J_{con}(\mathbf{x}) \leq J_{max} && \text{(6.1b)}\\
& R(\mathbf{x}) \geq R_{min} && \text{(6.1c)}\\
& E[D(\mathbf{x})] \leq D_{max} && \text{(6.1d)}\\
& \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} && \text{(6.1e)}
\end{aligned}
$$

where J_{max} is the desired maximum control cost of the WNCS. J_{max} is chosen such as to guarantee control system stability and an acceptable wireless network lifetime.

The cross-design approach developed in this chapter is illustrated through simulations and experiments with following main contributions:

- Cross-design investigation of the effects of network induced random delays and packet losses on the control performance and lifetime of WNCSs
- Proposal of a cross-design approach based on the design of suitable control algorithms and dedicated communication protocols in order to meet a desired control performance while minimizing the energy consumption of the wireless network at the same time

6.2 Cross-design methodology

The proposed cross-design methodology is based on the joint design of control algorithms and dedicated wireless communication protocols. The objective is the maximization of the lifetime of wireless nodes under control performance requirement constraints using selected control and wireless communication parameters. Control structures and corresponding controller parameters can be adapted offline or online to changing wireless network conditions in order to obtain best control performance and acceptable wireless network lifetime. At the same time, communication protocols and corresponding communication protocol parameters can be tuned both offline or online to meet a desired control performance and maximize the lifetime of the WNCS. With the goal of maximizing the lifetime of the wireless network under control performance requirement constraints, the three layer cross-design framework in Figure 6.1 is introduced.

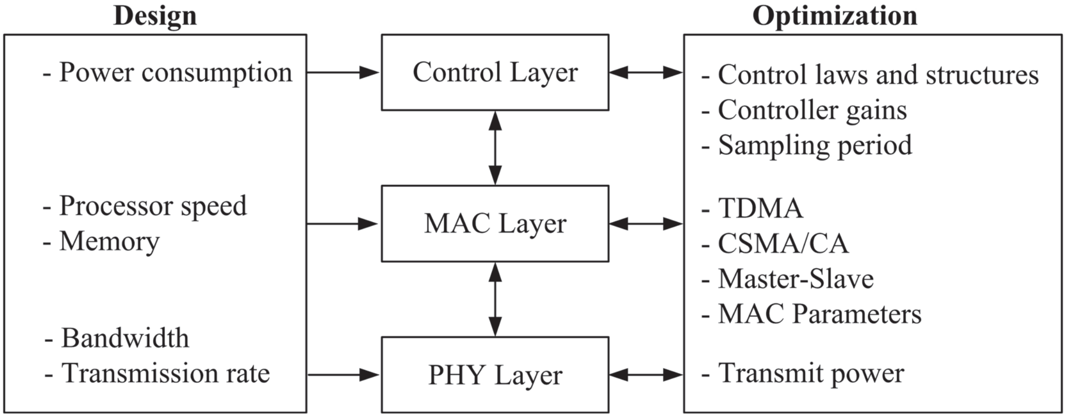


Figure 6.1: Proposed cross-design framework

The proposed cross-design approach is carried out in two phases including a design phase and an optimization phase. During the design phase [1], control system and communication system are designed for a given control application taking into consideration the limited communication, computing, and energy resources of the wireless nodes. This allows resulting control algorithms and communication protocols to be implemented on low energy wireless platforms with limited computational capabilities. At the control layer, the design phase includes the mathematical modeling and identification of the control system followed by the selection, implementation and evaluation of suitable control structures. These control structures are selected based on their performance, computational, and energy requirements. Closed-loop performance evaluation of selected control algorithms is done via simulation of real plants with suitable wireless network models and results in acceptable QoC metrics for the control application (e.g. system responses, robustness) from which QoS requirements to the underlying communication system (e.g. transmission rate, average delay, packet dropout rate) are derived. The degrees of freedom at the control layer are the control structure, the sampling period, and the controller gains. At the communication layers, the design phase includes the implementation of communication protocols that guarantee the QoS requirements from the control application in order to achieve acceptable control performance. The communication system is either capable of delivering the QoS required by the control application or, in case this turns out to be infeasible, offers guaranteed QoS so that the control application can for example change the sampling period, adapt controller parameters, or switch controllers to meet the desired QoC. The degrees of freedom at the communication layer are the MAC protocols with corresponding MAC protocol parameters and the nodes transmit powers.

The proposed structure for the design phase is depicted in Figure 6.2 on Page 73. The design workflow is aimed at finding optimal controller parameters K_{opt} for a given control structure and selected combinations of network QoSs (e.g. delay τ and packet loss rate ρ) such that the control cost J_{con} is minimized. For a given control structure, combinations of controller param-

[1] Refer to the Appendix for details on the control system design and to Chapter 5 for details on the communication system design

eters and network QoSs are selected within the stability bounds (e.g. MATI, MAD, MAPL) derived previously. Both heuristics-based optimization algorithms (CMAES[2]) and optimization algorithms using optimal control techniques (Dynamic Programming) are applied to find the optimal controller parameters K_{opt}. The HiL WNCS Simulator is used to check if the obtained optimal controller parameters really stabilize the controlled system and if the measured control cost is within the bounds computed by the design algorithm. Results are stored in a lookup table for later use during the optimization phase. An excerpt of the algorithm used in the design phase is given below:

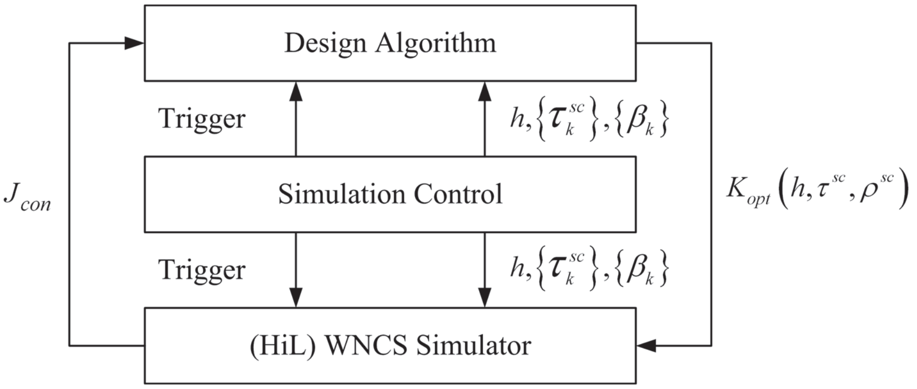

Design Algorithm
Trigger
$h, \{\tau_k^{sc}\}, \{\beta_k\}$
J_{con}
Simulation Control
$K_{opt}(h, \tau^{sc}, \rho^{sc})$
Trigger
$h, \{\tau_k^{sc}\}, \{\beta_k\}$
(HiL) WNCS Simulator

Figure 6.2: Proposed structure for the design phase with CMAES and Dynamic Programming as design algorithms

Algorithm 6.1 Design phase

```
select controller (e.g. LQR), sampling period $h$, and parameter $K$ (e.g. wired case)
select network QoS $\tau$ and $\rho$ with corresponding distributions $\tau = \{\tau_k\}_0^{k_f}$, and $\beta = \{\beta_k\}_0^{k_f}$
select $k_f, W_{k_f}, Q_x, Q_u$
$k \Leftarrow 0$, $J_{con} \Leftarrow 0$
repeat
    for $k \leq k_f$ do
        $\tau_k \Leftarrow \tau(k)$, $\beta_k \Leftarrow \beta(k)$
        $x_k \Leftarrow x(k)$, $u_k \Leftarrow u(k) = -Kx(k)$
        $J_k \Leftarrow J_{k-1} + x_{k_f}^T W_{k_f} x_{k_f} + \sum_{k=0}^{k_f-1} \{x_k^T W_k x_k + u_k^T U_k u_k\}$
        $k \Leftarrow k + 1$
        if $k = k_f$ then
            $J_{con} \Leftarrow J_{k_f}$
            $k \Leftarrow 0$
        end if
    end for
    adapt controller parameters $K$ to minimize $J_{con}$
until Optimal controller parameters $K_{opt}$ minimizing $J_{con}$ are obtained {Use of CMAES and
    Dynamic Programming algorithms. $K_{opt}(h, \tau, \rho)$ for the selected network QoS $\tau$ and $\rho$ are
    stored in a lookup table for use during the optimization phase}
```

The proposed structure for the optimization phase is depicted in Figure 6.3 on Page 74 and an excerpt of the underlying algorithm is given below. During the optimization phase, the total energy consumption of the wireless network is minimized while meeting a desired control performance using defined degrees of freedom at both the control layer and the communication

[2] Covariance Matrix Adaptation Evolution Strategy

layers. The control performance index J_{con} is computed at runtime using sensor measurements and control values. Network QoS parameters such as delay τ and packet losses β are also measured at runtime and used to compute the network performance metric J_{net} (see Chapter 5). Results are then used to estimate the node lifetime $\overline{T}$. Suitable combinations of controller and communication protocol parameters are then selected from a lookup table and applied at runtime. Both the design phase and the optimization phase will be extensively addressed in the sequel including details about the mathematical models and the optimization algorithms used.

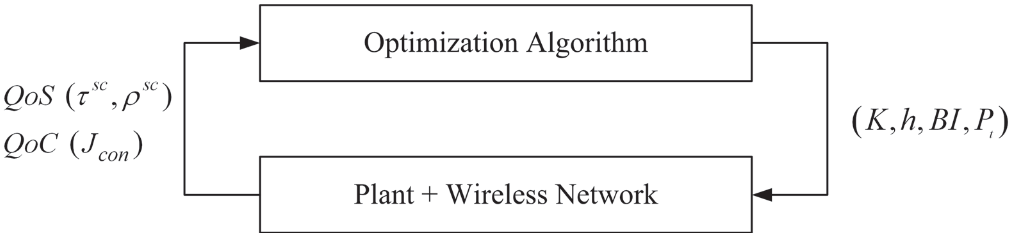


Figure 6.3: Proposed structure for the optimization phase

Algorithm 6.2 Optimization phase

```
Select MAC scheme (e.g. TDMA) and parameters (e.g. $SO, BO, S, s_d, l$)
Select controller (e.g. LQR), sampling period $h$, and parameter $K$ (from lookup table)
loop
    for $k \leq k_f$ do
        $\tau_k^{sc} \Leftarrow \tau^{sc}(k)$, $\beta_k^{sc} \Leftarrow \beta^{sc}(k)$
        $x_k \Leftarrow x(k)$, $u_k \Leftarrow u(k) = -Kx(k)$
        $J_k \Leftarrow J_{k-1} + x_{k_f}^T W_{k_f} x_{k_f} + \sum_{k=0}^{k_f-1} \left\{ x_k^T W_k x_k + u_k^T U_k u_k \right\}$
        $k \Leftarrow k + 1$
        if $k = k_f$ then
            $J_{con} \Leftarrow J_{k_f}$
            $\overline{T} = \frac{E_{Tot}}{\left[\overline{P_W}(\overline{\rho^{sc}}, l) T_W + P_{M_1}(h - T_W) + (\overline{P_W}(\overline{\rho^{sc}}, l) - P_{M_1})\overline{\epsilon} + E_{M_{12}}\right]} h$
            $k \Leftarrow 0$
        end if
        select network and controller parameters (from lookup table) in order to maximize $\overline{T}$
        while meeting $J_{con} \leq J_{max}$
    end for
end loop{Refer to Chapter 5 for the derivation of $\overline{T}$. $K_{opt}(h, \tau, \rho)$ for measured network
    QoS, $J_{con} \leq J_{max}$, and $\overline{T}$ is maximized}
```

6.3 Cross-design modeling

6.3.1 Plant dynamics

The cross-design model is depicted in Figure 6.4 on Page 75 where a plant is controlled over a wireless network using the dedicated wireless protocols developed in the previous chapter. Plant sensors transmit their measurements to the controller over the wireless network using scheduled (e.g. TDMA) or contention-based (e.g. CSMA/CA) MAC protocols. The wireless network may introduce time-varying delays and packet losses depending on the selected MAC protocol. Without loss of generality, we assume a wired connection between the controller and the actuator meaning that control commands are successfully received at the actuator. The continuous-time

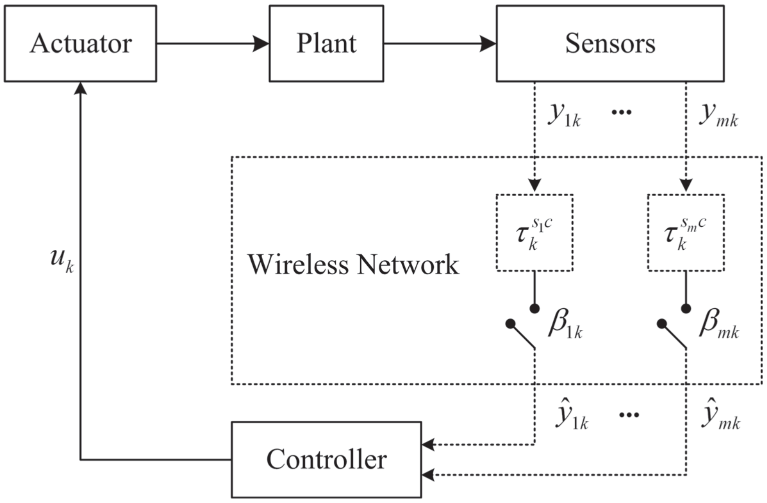


Figure 6.4: Overview of the cross-design WNCS setup

plant dynamics are modeled using the following linear stochastic differential equation

$$\dot{x}(t) = Ax(t) + Bu(t) + v(t) \tag{6.2}$$

$$y(t) = Cx(t) + w(t) \tag{6.3}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $y(t) \in \mathbb{R}^m$ the output vector, $u(t) \in \mathbb{R}$ the control input, $v(t) \in \mathbb{R}^n$ the process noise with zero mean and uncorrelated increments and $w(t) \in \mathbb{R}^m$ the measurement noise with zero mean and uncorrelated increments. A, B, and C are matrices of appropriate dimensions. Sampling the plant with time-varying sampling period $h_k = t_{k+1} - t_k$ and expressing the sampling period as $h_k = h_0 + \tau_k^{sc}$ leads to the following discrete-time formulation of the continuous-time plant dynamics (6.2)-(6.3) assuming zero-order hold

$$x_{k+1} = A_d x_k + B_d u_k + v_k \tag{6.4}$$

$$y_k = C_d x_k + w_k \tag{6.5}$$

where h_0 is constant, $0 < \tau_k^{sc} < h_k$ and $0 < h_{min} \leq h_k \leq h_{max}$. $A_d = e^{Ah_k}$, $B_d = \int_0^{h_k} e^{As} ds B$, and $C_d = C$. We assume that the random sequences $\{\tau_k^{sc}\}_0^\infty$ and $\{h_k\}_0^\infty$ are independent with known distributions.

6.3.2 Control model

Modeling the packet loss sequence $\{\beta_k\}_0^\infty$ in Figure 6.4 as a random process leads to the following expression for the sensor measurements at the controller input

$$\hat{y}_k = (1 - \beta_k)\, y_k \tag{6.6}$$

where $\beta_{ik} \in \{0, 1\}$ and $\beta_k = max\,\{\beta_{ik}\}_{i \in \{1,\ldots,m\}} \in \{0, 1\}$ is a Bernoulli random variable with packet loss probability $P\left[\beta_k = 1\right] = \rho^{sc}$ modeling the packet loss between the sensor and the controller. β_k is known to the controller which expects periodic sensor measurements. Therefore, the following set of information I_k is available to the controller and the optimizer node at each sampling time instant k:

$$I_k = \{\hat{y}_0, \cdots, \hat{y}_k; \beta_0, \cdots, \beta_k; \tau_0^{sc}, \cdots, \tau_k^{sc}\}\,. \tag{6.7}$$

A new control command u_k is only computed if all m sensor measurements are received by the controller at sampling instant k. In the event of up to $m - 1$ sensor measurement losses, missing sensor measurements are estimated using a Kalman state estimator. Available sensor measurements and estimates of missing sensor measurements are then used to compute the new control value as will be shown in the sequel.

Since delays result in the controller keeping the previous control command u_{k-1} until the new control u_k arrives at time $t_k = k + \tau_k^{sc}$, the following linear stochastic difference equations with intermittent sensor measurements are obtained using the augmented state vector $z_k = [x_k, u_{k-1}]^T$:

$$z_{k+1} = \Phi z_k + \Gamma u_k + \begin{bmatrix} v_k \\ 0 \end{bmatrix} \tag{6.8}$$

$$\hat{y}_k = (1 - \beta_k)\, C_d z_k \tag{6.9}$$

$$\Phi = \begin{bmatrix} A_d & \Gamma_1 \\ 0 & 0 \end{bmatrix} \tag{6.10}$$

$$\Gamma = \begin{bmatrix} \Gamma_0 \\ I \end{bmatrix} \tag{6.11}$$

with corresponding discrete-time system matrices $A_d = e^{Ah_k}$, $\Gamma_0 = \int_0^{h_k - \tau_k^{sc}} e^{As} ds B$, $\Gamma_1 = e^{A(h_k - \tau_k^{sc})} \int_0^{\tau_k^{sc}} e^{As} ds B$, and $\tau_k^{sc} = max\,\{\tau_{ik}^{sc}\}_{i \in \{1,\ldots,m\}}$. ($x_0$, v_k, w_k) are Gaussian, uncorrelated, white, with mean ($\bar{x}_0$, 0, 0) and covariance (P_0, Q, R) respectively. In the event of a packet loss between the sensor and the controller, the input to the controller is $\hat{y}_k = 0$. However, state estimates must be updated correctly.

In the previous chapter, a network manager was introduced between the sensors and the controller in order to achieve optimal performance and energy efficient control over wireless networks. Delay and packet loss information between the sensors and the controller are stored by the network manager and used at runtime in conjunction with the analytical model of the node lifetime to solve the following constrained optimization problem:

$$\max_{h,\mathbf{p},\varsigma} \quad \overline{T}(h, \mathbf{p}, \varsigma) \tag{6.12a}$$

$$s.t. \quad J\left(h, \rho^{sc}\left(h, \mathbf{p}, \varsigma\right), \tau^{sc}\left(h, \mathbf{p}, \varsigma\right)\right) \leq J_{max} \tag{6.12b}$$

$$\rho^{sc}\left(h, \mathbf{p}, \varsigma\right) \leq \rho_{max}^{sc} \tag{6.12c}$$

$$\tau^{sc}\left(h, \mathbf{p}, \varsigma\right) \leq \tau_{max}^{sc} \tag{6.12d}$$

where the objective function is the average wireless network lifetime $\overline{T}(h, \mathbf{p}, \varsigma)$ constrained by the desired control performance cost $J\left(h, \rho^{sc}\left(h, \mathbf{p}, \varsigma\right), \tau^{sc}\left(h, \mathbf{p}, \varsigma\right)\right)$. The decision variables are the sampling period h and the MAC protocol parameters $\mathbf{p}$. The variable ς accounts for the network protocol parameters such as the transmit power at the physical layer. The control cost $J\left(h, \rho^{sc}\left(h, \mathbf{p}, \varsigma\right), \tau^{sc}\left(h, \mathbf{p}, \varsigma\right)\right)$, the delay $\tau^{sc}\left(h, \mathbf{p}, \varsigma\right)$ and the packet loss $\rho^{sc}\left(h, \mathbf{p}, \varsigma\right)$ are functions of the sampling period h. Moreover, the control cost is also a function of the packet loss probability ρ^{sc} and delay τ^{sc}. The decision variables of the constrained optimization problem 6.12 are feasible if they satisfy a desired control cost $J_{req} \leq J_{max}$.

6.3.3 State estimator

Since measurements can be dropped out between sensors and controller, an estimator is required to reconstruct missing plant states. The Kalman filter state estimator used to estimate plant states in the event of packet loss between sensors and controller is of the following form:

$$\hat{z}_{k|k} = [E[x_k|I_k], u_{k-1}]^T \tag{6.13}$$

$$\hat{z}_{k+1|k} = \Phi \hat{z}_{k|k} + \Gamma u_k \tag{6.14}$$

$$e_{k+1|k} = \Phi e_{k|k} + v_k \tag{6.15}$$

$$P_{k+1|k} = \Phi P_{k|k} \Phi^T + Q \tag{6.16}$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + (1-\beta_{k+1}) K_{k+1} \left[y_{k+1} - C_d \hat{z}_{k+1|k} \right] \tag{6.17}$$

$$e_{k+1|k+1} = [I - (1-\beta_{k+1}) K_{k+1} C_d] e_{k+1|k} - (1-\beta_{k+1}) K_{k+1} v_{k+1} \tag{6.18}$$

$$P_{k+1|k+1} = P_{k+1|k} - (1-\beta_{k+1}) K_{k+1} C_d P_{k+1|k} \tag{6.19}$$

$$K_{k+1} = P_{k+1|k} C_d^T \left(C_d P_{k+1|k} C_d^T + R \right)^{-1} \tag{6.20}$$

where y_k is the actual sensor measurement, $C\hat{z}_{k+1|k}$ the measurement prediction, K_k the time-varying Kalman filter gain matrix, $P_{k+1|k}$ the error covariance matrix which is a function of the packet loss between sensors and controller. The derivation of the above equations is similar to the derivation for the time varying Kalman filter equations using the time varying matrices $A_k = \Phi$ and $C_k = (1-\beta_k)C_d$. Equations (6.14) to (6.16) are the time update equations and equations (6.17) to (6.20) the measurement update equations.

If sensor measurements are dropped out on one or several consecutive sampling periods, the estimator must be run in open loop up to the time step $k-1$ after successful arrival of all sensor packets at the controller at time step k until the estimate $\hat{z}_{k-1|k-1}$ is obtained before applying equation (6.14). Assuming for example that, $\hat{z}_{\epsilon|\epsilon}$, u_ϵ, and τ_ϵ^{sc} are known at sampling time instant ϵ, the estimator is run in open loop using the iteration

$$\hat{z}_{\epsilon+1|\epsilon+1} = \hat{z}_{\epsilon+1|\epsilon} = \Phi \hat{z}_{\epsilon|\epsilon} + \Gamma u_\epsilon \tag{6.21}$$

$$u_{\epsilon+1} = u_\epsilon \tag{6.22}$$

$$P_{\epsilon+1|\epsilon} = \Phi P_{\epsilon|\epsilon} \Phi^T + Q \tag{6.23}$$

until $\hat{z}_{k-1|k-1}$ is obtained.

There is a limit to how many sensor measurements can be lost before the state estimator looses stability. According to [Sinopoli *et al.* , 2004], there is a critical sensor to controller packet loss probability value ρ_{crit}^{sc} below which the estimated mean state error covariance $E[P_{k+1|k}]$ is bounded for all initial conditions (provided that the usually assumed stabilizability and detectability hypotheses are satisfied). Above this critical probability value, the estimated mean state covariance diverges for some initial conditions. The existence of this critical probability value was shown in [Sinopoli *et al.* , 2004] with bounds $\underline{\rho^{sc}} \leq \rho_{crit}^{sc} \leq \overline{\rho^{sc}}$.

A close form solution is $\underline{\rho^{sc}} = \frac{1}{\max_i(\phi_i)^2}$ for the lower bound and $\overline{\rho^{sc}} = \frac{1}{\prod_i(\phi_i)^2}$ for the upper bound where $\phi_i = |\lambda_i^u(\Phi)|$ is the spectral radius of the matrix Φ and $\lambda_i^u(\Phi)$ the unstable eigenvalues of the matrix Φ. As shown by the authors, this upper bound is the solution of the following linear matrix inequality:

$$\rho_{crit}^{sc} = argmin_{\overline{\rho^{sc}}} \Psi(Y,Z) > 0, 0 \leq Y \leq I \tag{6.24}$$

$$\Psi_{\rho^{sc}}(Y,Z) = \begin{bmatrix} Y & Y & \sqrt{\rho^{sc}} Z Q_u^{\frac{1}{2}} & \sqrt{1-\rho^{sc}}(Y\Phi^T + Z\Gamma^T) & \sqrt{\rho^{sc}} Y \Phi^T \\ Y & Q^{-1} & 0 & 0 & 0 \\ \sqrt{1-\rho^{sc}} Z Q_u^{\frac{1}{2}} Z^T & 0 & I & 0 & 0 \\ \sqrt{1-\rho^{sc}}(\Phi Y + \Gamma Z^T) & 0 & 0 & Y & 0 \\ \sqrt{\rho^{sc}} \Phi Y & 0 & 0 & 0 & Y \end{bmatrix}. \tag{6.25}$$

6.3.4 Controller design

The control objective is to minimize the following cost function

$$J_p(\mathbf{u}^{p-1}, \overline{z}_0, P_0) = E\left[z_p^T W_p z_p + \sum_{k=0}^{p-1} \left\{ z_k^T W_k z_k + u_k^T U_k u_k \right\} |\mathbf{u}^{p-1}, \overline{z}_0, P_0 \right] \tag{6.26}$$

where $\overline{z}_0 = (\overline{x}_0, 0)$, $\mathbf{u}^{p-1} = (u_{p-1}, u_{p-2}, \cdots, u_1)$, p is the optimization horizon, W_k is positive semi-definite, U_k positive definite, and W_p positive semi-definite. u_k is the control value computed by the controller upon reception of sensor measurements and is also the control value applied by the actuator at sampling instant k. In order to minimize the cost function given by equation (6.26) a control input sequence $\mathbf{u}^{*p-1}$ is required, which is a function of the available set of information I_k at the controller at sampling time instant k, i.e. $u_k = g_k(I_k)$. This leads to the following equation:

$$J_p^*(\overline{z}_0, P_0) = \min_{u_k = g_k(I_k)} J_p(\mathbf{u}^{p-1}, \overline{z}_0, P_0) \tag{6.27}$$

The derivation of the optimal control law and the corresponding control cost are based on the dynamic programming approach using the cost-to-go iterative procedure. Defining the final cost function to be $V_p = E[z_p^T W_p z_p | I_p]$ leads to the following cost-to-go function:

$$V_k = \min_{u_k} E[z_k^T W_k z_k + u_k^T U_k u_k + V_{k+1}(z_{k+1}) | I_p] \tag{6.28}$$

where $k = p-1, \cdots, 1$. $V_k(z_k)$ in equation (6.28) can be written as

$$V_k(z_k) = E[z_k^T S_k z_k | I_k] + \nu_k, k = p, \cdots, 0 \tag{6.29}$$

$$S_k = \Phi^T S_{k+1} \Phi + W_k - \Phi^T S_{k+1} \Gamma (\Gamma^T S_{k+1} \Gamma + U_k)^{-1} \Gamma^T S_{k+1} \Phi \tag{6.30}$$

$$\nu_k = \text{trace}((\Phi^T S_{k+1} \Phi + W_k - S_k) P_{k|k}) + \text{trace}(S_{k+1} Q) + E[\nu_{k+1} | I_k] \tag{6.31}$$

with initial values $S_p = Q_p$ and $\nu_p = 0$. The optimal control input u_k is given by

$$u_k = -L_{\rho^{sc}}(\tau_k^{sc}, S_\infty) \hat{z}_{k|k} = -\left[\Gamma^T S_{k+1} \Gamma + U_k\right]^{-1} \Gamma^T S_{k+1} \Phi \hat{z}_{k|k} \tag{6.32}$$

where $L_{\rho^{sc}}(\tau_k^{sc}) = L_{\rho^{sc}}(\tau_k^{sc}, S_\infty)$ is a function of the packet loss probability ρ^{sc} as well as the delay τ^{sc} between the sensors and the controller. Since $J_p^*(\overline{z}_0, P_0) = V_0(z_0)$, the cost function of the optimal LQG is given by

$$\begin{aligned} J_p^*(\overline{z}_0, P_0) = {} & \overline{z}_0^T S_0 \overline{z}_0 + \text{trace}(S_0 P_0) + \\ & \sum_{k=0}^{p-1} \left\{ \text{trace}(\Phi^T S_{k+1} \Phi + W_k - S_k) E_{\rho^{sc}}[P_{k|k}] \right\} \end{aligned} \tag{6.33}$$

where $E_{\rho^{sc}}[P_{k|k}]$ explicitly shows that the expectation of the error covariance matrix is computed taking into consideration the packet loss probability between the sensors and the controller. Therefore, the error covariance matrices are stochastic and cannot be computed analytically. However, deterministic bounds on the error covariance matrices can be computed as shown in [Sinopoli *et al.* , 2004] resulting in the following equations:

$$\underline{P_{k|k}} \leq E_{\rho^{sc}}[P_{k|k}] \leq \overline{P_{k|k}}, \forall k \geq 0 \tag{6.34}$$

$$\begin{aligned} \overline{P_{k+1|k}} = {} & \Phi \overline{P_{k|k-1}} \Phi^T + Q - \\ & (1-\rho^{sc}) \Phi \overline{P_{k|k-1}} C_d^T (C_d \overline{P_{k|k-1}} C_d^T + R)^{-1} C_d \overline{P_{k|k-1}} C_d^T \Phi^T \end{aligned} \tag{6.35}$$

$$\overline{P_{k|k}} = \overline{P_{k|k-1}} - (1-\rho^{sc}) \Phi \overline{P_{k|k-1}} C_d^T (C_d \overline{P_{k|k-1}} C_d^T + R)^{-1} C_d \overline{P_{k|k-1}} C_d^T \Phi^T \tag{6.36}$$

$$\underline{P_{k+1|k}} = \rho^{sc} \Phi \underline{P_{k|k-1}} \Phi^T + Q \tag{6.37}$$

$$\underline{P_{k|k}} = \rho^{sc} \underline{P_{k|k-1}} \tag{6.38}$$

where the initial conditions are $\overline{P_{0|0}} = \underline{P_{0|0}} = P_0$. The minimum achievable control cost J_p^* in equation (6.33) cannot be computed analytically, but bounds can be found as follows:

$$J_p^{min} \leq J_p^* \leq J_p^{max} \tag{6.39}$$

$$\begin{aligned}
J_p^{max} &= \bar{z}_0^T S_0 \bar{z}_0 + \text{trace}(S_0 P_0) + \sum_{k=0}^{p-1} \text{trace}(S_{k+1} Q) + \\
&\sum_{k=0}^{p-1} \left\{ \text{trace}(\Phi^T S_{k+1} \Phi + W_k - S_k) \overline{P_{k|k}} \right\} \\
J_p^{min} &= \bar{z}_0^T S_0 \bar{z}_0 + \text{trace}(S_0 P_0) + \sum_{k=0}^{p-1} \text{trace}(S_{k+1} Q) + \\
&\sum_{k=0}^{p-1} \left\{ \text{trace}(\Phi^T S_{k+1} \Phi + W_k - S_k) \underline{P_{k|k}} \right\}
\end{aligned} \tag{6.40}$$

Following the same procedure as for the finite horizon case presented above, the optimal controller gain for the infinite horizon case is given by

$$\lim_{k\to\infty} K_k = L_\infty = \left[\Gamma^T S_\infty \Gamma + U\right]^{-1} \Gamma^T S_\infty \Phi \tag{6.41}$$

and the expected minimum cost is bounded by the following two deterministic sequences

$$\frac{1}{p} J_p^{min} \leq J_p^* \frac{1}{p} \leq \frac{1}{p} J_p^{max} \tag{6.42}$$

$$\begin{aligned}
J_\infty^{max} &= \lim_{p\to\infty} J_p^{max} \\
&= \text{trace}(S_\infty Q) + \text{trace}((\Phi^T S_\infty \Phi + W - S_\infty)(\overline{P_\infty} - \\
&(1-\rho^{sc})\overline{P_\infty} C_d^T (C_d \overline{P_\infty} C_d^T + R)^{-1} C_d \overline{P_\infty})) \\
J_\infty^{min} &= \lim_{p\to\infty} J_p^{min} \\
&= \text{trace}(S_\infty Q) + \rho^{sc} \text{trace}(\Phi^T S_\infty \Phi + W - S_\infty)
\end{aligned} \tag{6.43}$$

where the matrices S_∞, $\overline{P_\infty}$, and $\underline{P_\infty}$ are positive definite solutions to the following set of equations

$$S_\infty = \Phi^T S_\infty \Phi + W - \Phi^T S_\infty \Gamma (\Gamma^T S_\infty \Gamma + U)^{-1} \Gamma^T S_\infty \Phi \tag{6.44}$$

$$\overline{P_\infty} = \Phi \overline{P_\infty} \Phi^T + Q - (1-\rho^{sc}) \Phi^T \overline{P_\infty} C_d (C_d^T S_\infty C_d + U)^{-1} C_d^T \overline{P_\infty} \Phi^T \tag{6.45}$$

$$\underline{P_\infty} = \rho^{sc} \Phi \underline{P_\infty} \Phi^T + Q. \tag{6.46}$$

J_p^{min} and J_p^{max} are explicit functions of the sampling period h, the packet loss probability ρ^{sc} and the delay τ^{sc}. Both the finite and the infinite horizon costs are used as performance indicators during the optimization phase (online) and the design phase (offline) respectively as mentioned in Section 6.2 on Page 71.

6.4 Cross-design procedure

The WNCS cross-design problem considered so far is that of finding the optimal sampling period h^* of the control system and the corresponding optimal parameters $\mathbf{p}^*$ of the dedicated communication protocols presented in Chapter 5 solving the constrained optimization problem given by equation (6.12). The main steps of the proposed cross-design procedure are depicted in Figure 6.5. Details about the WNCS network setup including network topology, protocols, packet length, transmit power levels, and total number of wireless nodes are stored in the variable ς. This network setup information determines the network QoS (e.g. delay and packet loss probabilities) and should be provided in **Step 1** along with the maximum desired control cost J_{req} required to achieve acceptable control performance. Although not considered in this thesis, several control loops can be closed over the same wireless network and in this case, each control loop may have a different desired control cost J_{req}. During **Step 2**, state estimates (6.13)-(6.20), state feedback control law (6.32), and control costs (6.40), (6.43) may be computed offline for different sampling periods h, delays τ^{sc}, and packet loss probabilities ρ^{sc}. Results may then be verified using the HiL WNCS Simulator and stored in a lookup table for later use. The constrained optimization problem (6.12) is formulated and solved by the network manager in **Step 3**, whereby the objective function is the total energy consumption of the network constrained by the delay and packet loss requirements. Delay and packet loss requirements are derived from the desired control cost J_{req} for different sampling periods h. For given sampling period h, delay ρ^{sc}_{max} and packet loss probability τ^{sc}_{max}, the network manager finds the local optimal MAC parameters $\mathbf{p}^*(h, \rho^{sc}_{max}, \tau^{sc}_{max})$ of a sub-optimization problem. The optimal solution h^*, $\mathbf{p}^*$ is given in **Step 5** by h and $\mathbf{p}$ that minimize the cost function if there are feasible solutions. In case the constrained optimization problem turns out to be infeasible, the desired control cost J_{req} must be refined in **Step 4**. Otherwise, the network manager adapts the optimal sampling period h^* and the protocol parameters $\mathbf{p}^*$ of the wireless network in **Step 5**. Accordingly, state estimates and the feedback control law are updated in **Step 6** using h^*, $\tau^{sc}(h^*, \mathbf{p}^*, \varsigma)$, $\rho^{sc}(h^*, \mathbf{p}^*, \varsigma)$.

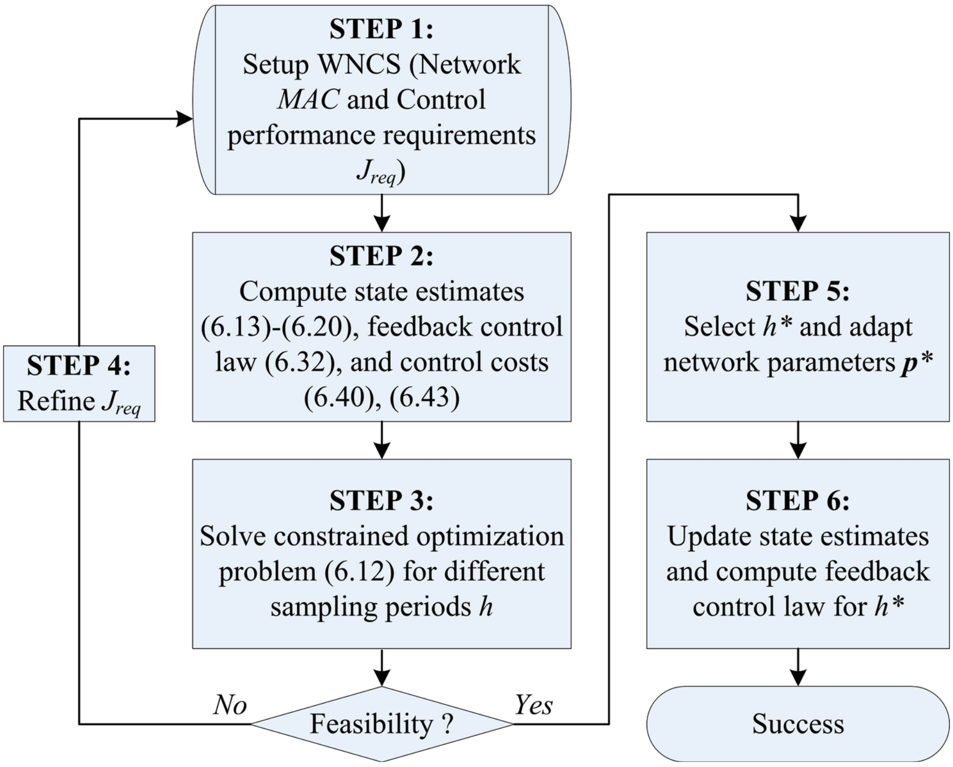


Figure 6.5: Procedure of the cross-design: STEPS 1-4 can be carried out offline or online while STEPS 5-6 are carried out online. The HiL WNCS Simulator is used in STEP 2

6.5 Summary

The interactions between the wireless communication network and the control application are critical factors that influence the performance of a WNCS. In this chapter, we proposed a cross-design framework for WNCSs and showed the applicability to real control applications. In order to achieve best control performance and optimal wireless network lifetime, control algorithms and wireless communication protocols should be jointly designed. This can be achieved during the design phase (offline) or at runtime (online) during the optimization phase using selected degrees of freedom at both the control and the communication layer. The proposed constrained optimization problem is a powerful tool that can be used when solving cross-design problems. The energy consumption of the network is the objective function and packet loss probability and delay are constraints that can be derived from a desired control cost. Next, the proposed cross-design approach and the corresponding cross-design procedure will be illustrated on the inverted pendulum testbed.

7 Results of the proposed cross-design approach

In this chapter, we illustrate the proposed cross-design approach for WNCS on an inverted pendulum testbed under realistic conditions. To this end, we apply the different steps of the cross-design procedure presented in Chapter 6 and discuss the results obtained. The main focus here is to show the applicability of the proposed cross-design approach to real wireless networked control processes. In Section 7.1, the experimental setup considered is presented including a brief description of the inverted pendulum system and the dedicated wireless communication protocols. In Section 7.2, the general cross-design optimization procedure described in Chapter 6 is applied to the inverted pendulum system and experimental results discussed. Cross-design optimization results for the inverted pendulum system are presented and discussed in Section 7.3. Finally, Section 7.4 concludes the chapter.

7.1 Experimental setup

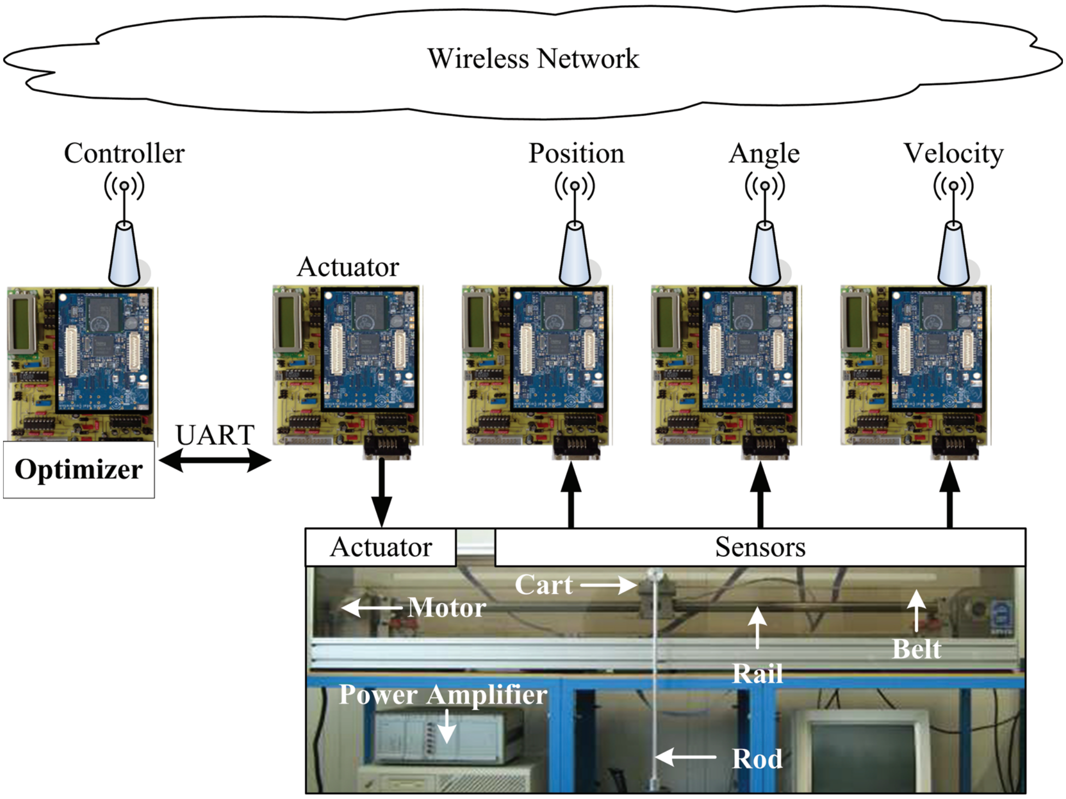


Figure 7.1: Cross-design experimental setup for the inverted pendulum testbed

Figure 7.1 shows the considered cross-design experimental setup consisting of the real inverted pendulum and Imote2 wireless nodes. The interface between the hardware of the Imote2 wireless nodes and the inverted pendulum hardware, i.e., sensors, controller, and actuator is assured by a special hardware platform which was developed in the course of this thesis. Details on the hardware interface platform and the selected Imote2 wireless nodes can be found in Chapter 4. The inverted pendulum system (refer to Figure A.1 in Section A.1 of the Appendix) is an open-loop unstable plant with nonlinear system dynamics. It consists of a vertical rod that can rotate around a fixed pivot point on a cart. The corresponding rod angle from the vertical upward position φ is measured using a rotational encoder connected to a wireless node and angle measurements are sent to the controller node over the wireless network. The cart is free to move along a rail and the corresponding displacement x is measured using a rotational encoder connected to a wireless node and position measurements are sent to the controller node over the wireless network. The cart velocity $\dot{x}$ is measured using a tachometer connected to a wireless node and the velocity measurements are sent to the controller node over the wireless network. The cart is driven by a DC electric motor controlled by a power amplifier with an input voltage u. The DC motor is coupled to the cart through a transmission belt. There is a wired connection between the controller node and the actuator node controlling the power amplifier. Thus, the controller node is powered by the power supply of the power amplifier. There is no sensor available to measure the angular velocity $\dot{\varphi}$ of the inverted pendulum. Therefore, the angular velocity $\dot{\varphi}$ of the inverted pendulum system is estimated using the Kalman estimator introduced in Chapter 6. Finally, it should be pointed out, that all the functionalities of the Optimizer node, i.e., control and wireless network models as well as performance monitoring functionalities are implemented in the controller node.

7.1.1 Inverted pendulum system

The motion of the pendulum rod and cart can be described by the following two nonlinear equations[1]:

$$(M_r + M_c)\ddot{x} + M_r l_r(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2 \sin\varphi) + f_c\dot{x} = C_v u \tag{7.1}$$

$$(J_r + M_r {l_r}^2)\ddot{\varphi} + f_r\dot{\varphi} + M_r l_r(\ddot{x}\cos\varphi - g\sin\varphi) = 0. \tag{7.2}$$

Choosing the state vector $x = [x; \varphi; \dot{x}; \dot{\varphi}]$ and linearizing equations (7.1) and (7.2) around the equilibrium point $x = [a; 0; 0; 0], a \in \mathbb{R}$ leads to the following continuous-time state space equations:

$$\dot{x} = Ax + Bu \tag{7.3}$$

$$y = Cx \tag{7.4}$$

with the corresponding continuous-time state space system matrices:

$$A = \begin{pmatrix} 0 & 0 & -1.950 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.156 & -1.948 & 0.009 \\ 0 & 23.509 & 28.805 & -0.134 \end{pmatrix} \tag{7.5}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ -6.243 \\ 92.288 \end{pmatrix} \tag{7.6}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{7.7}$$

[1] Refer to Section A.1 of the Appendix for a detailed derivation of the equations of motion of the inverted pendulum and corresponding parameter values

The following two control problems are considered for the inverted pendulum system:

1. *Stabilization of the inverted pendulum*: Keep the pendulum rod in the vertical upward position, i.e. $\varphi = 0$

2. *Trajectory tracking*: Follow a given position trajectory while keeping the pendulum rod in the vertical upward position

Given the selected initial states $x_0 = 0m$, $\varphi_0 = 4°$, $\dot{x}_0 = 0m/s$, and $\dot{\varphi}_0 = 0rad/s$, different control structures for the inverted pendulum system were designed to meet the following control performance requirements and constraints:

- Fast transient response with little overshot and zero steady state error
- Control value saturation: $|u| \leq 10V$
- Angle value limitation due to linearization assumptions: $|\varphi| \leq 10°$
- Mechanical limitations of the cart's position: $|x| \leq 0.67m$

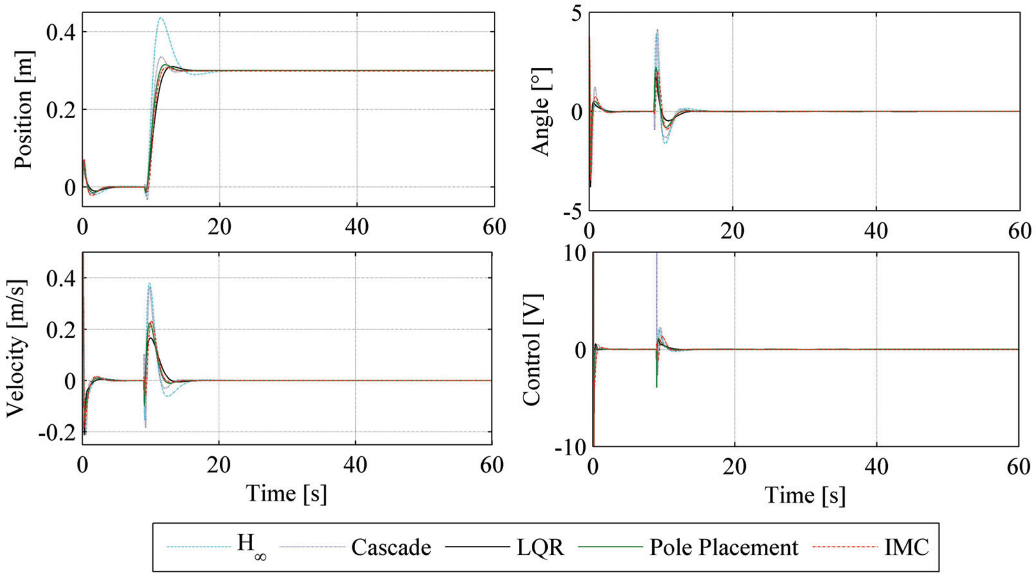


Figure 7.2: Simulation results of the continuous-time case for selected control structures

Controller	Overshot	Steady State Error	Rise Time	Settling Time
Cascade	11.5%	0%	0.86s	3.6s
Pole Placement	5%	0%	1.32s	4.1s
LQR	3.3%	0%	1.83s	5.1s
IMC	2.6%	0%	1.3s	3.7s
H_∞	45%	0%	1.13s	9.2s

Table 7.1: Control performance results from simulation for the continuous-time case

Simulation results for the continuous-time case, i.e., wireless network effects such as transmission delays and packet losses are not taken into consideration, are shown in Figure 7.2 and corresponding performance results are summarized in Table 7.1. All implemented control structure meet the desired control performance requirements and better control performance can be achieved by tuning the corresponding controller parameters[2]. However, depending on the selected control structure and algorithm, the number of tunable parameters, i.e., the number of degrees of freedom for the cross-design optimization at the control layer is different. The H_∞ control structure has 16 tunable parameters, the internal model controller (IMC) has 14 tunable parameters, the cascaded control structure has 7 tunable parameters, and the state feedback control structures pole placement and LQR both have 4 tunable parameters. However, since the angular velocity of the inverted pendulum is not measurable and must be estimated using and observer, the parameters of the observer must be added to the list of tunable parameters for the state feedback control structures. H_∞, cascade, and IMC do not require an observer for the estimation of the angular velocity but the number of tunable parameters is still high compared to that of the state feedback control structures.

All control structures and algorithms developed in the course of this thesis can be used to illustrate the proposed cross-design approach on the inverted pendulum system and applying the cross-design procedure is straight forward. Using different control algorithms at the same time even offers the added flexibility of switching between different control algorithms and parameters at runtime. In the remaining of this chapter, the state feedback control structure in Figure 7.3 is selected to illustrate the proposed cross-design approach. The Kalman estimator is used to estimate not only missing sensor measurements in the event of packet losses, but also the Coulomb's friction d_u which is modeled as a disturbance input to the inverted pendulum, and the angular velocity $\dot{\varphi}$ of the inverted pendulum. The filter allows the inverted pendulum to follow step reference inputs without steady state errors.

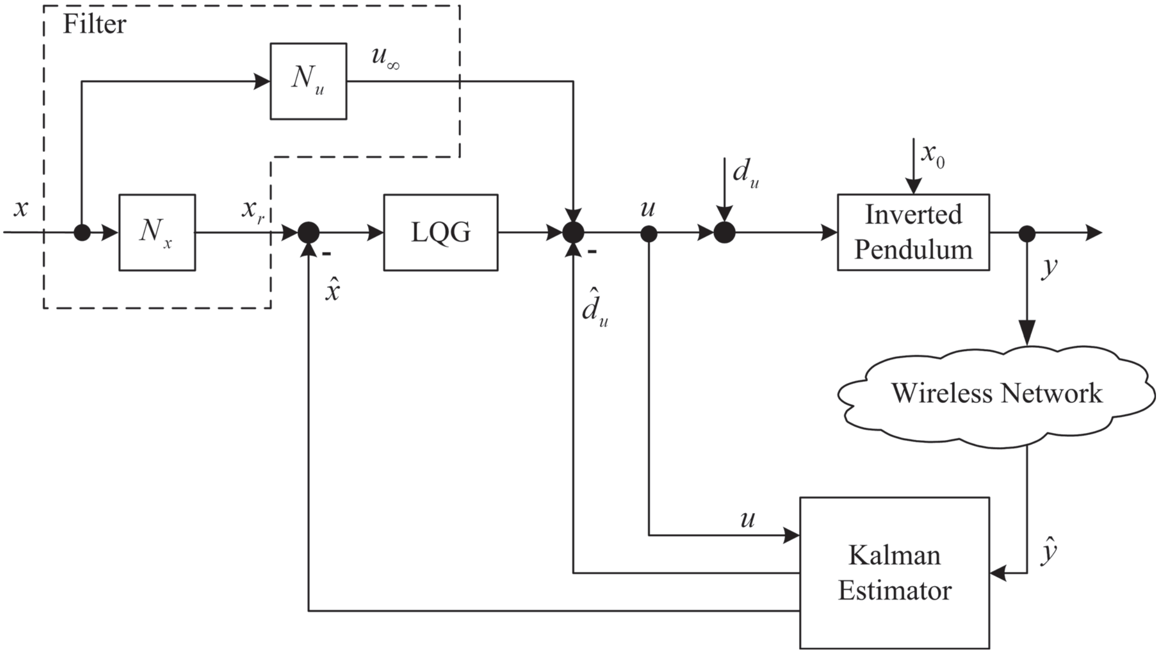


Figure 7.3: State feedback control structure for the cross-design with LQG controller and Kalman estimator

Following assumptions are made for the cross-design based on simulation and experimental results from Chapter 4 and the Appendix respectively:

- Filter parameters: $N_x = [1; 0; 0; 0]$ and $N_u = 0$
- LQG controller gain: $K = [K_x; K_\varphi; K_{\dot{x}}; K_{\dot{\varphi}}]$

[2] Refer to Section A.2 of the Appendix for details about the implemented control structures and parameters

- Kalman estimator gain: $K_k = [K_{k_x}; K_{k\varphi}; K_{k\dot{x}}; K_{k\dot{\varphi}}; K_{kd_u}]$
- Sampling period: $h_0 = 10ms$ and $h = n \cdot h_0, n \in \{1, ..., h_{max}/h_0\}$ where h_0 is the basic sampling period and $h_{max} = MATI$ the maximum applicable sampling period
- Delay: $\tau_0 = 5ms$ and $\tau = m \cdot \tau_0 < h, m \in \{1, ..., h/\tau_0\}$ where τ_0 is the basic delay
- Packet losses: $\rho_0 = 2\%$ and $\rho = q \cdot \rho_0 \leq \rho^{sc}_{crit}, q \in \{1, ..., \rho^{sc}_{crit}/\rho_0\}$ where ρ_0 is the basic packet loss probability and ρ^{sc}_{crit} the critical packet loss probability above which the Kalman estimator loses stability

The cross-design parameters at the control layer are the LQG controller gain K, the Kalman estimator gain K_k, and the sampling period h. During the cross-design optimization, K, K_k, and h are adapted to varying wireless network conditions such as delays τ^{sc} and packet loss probabilities ρ^{sc}.

7.1.2 Wireless communication protocols

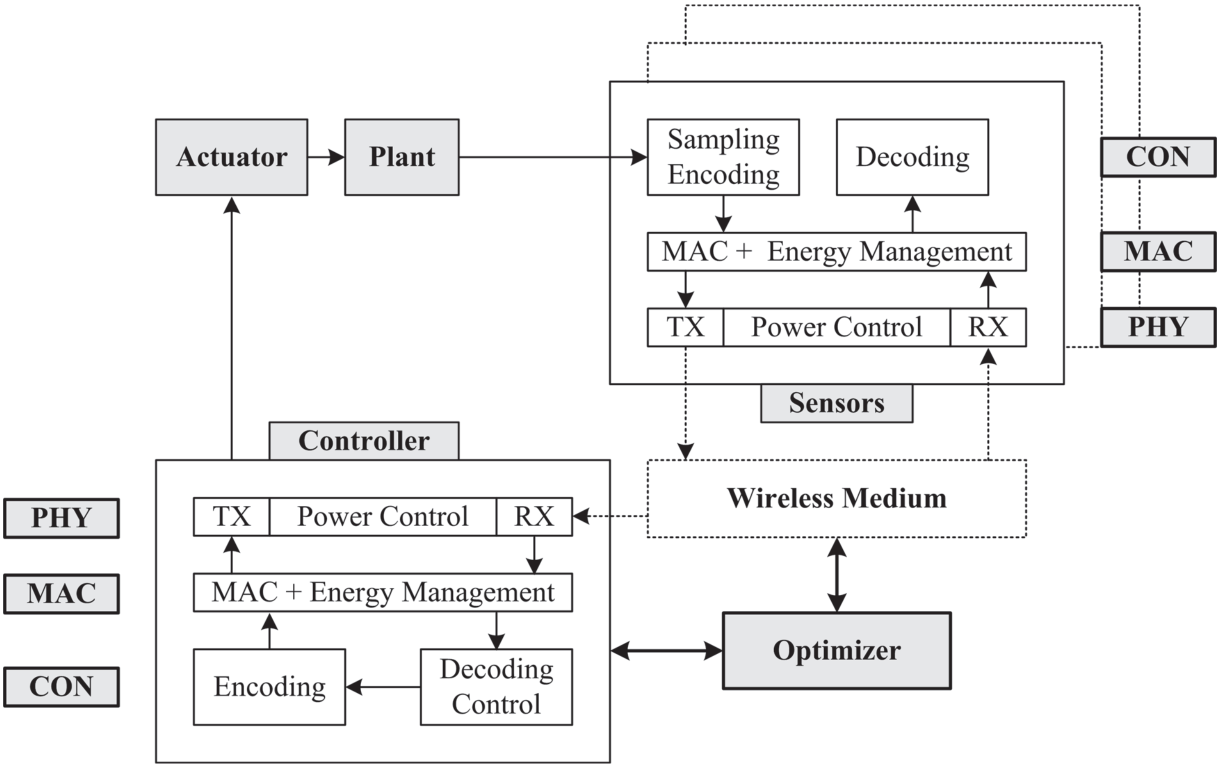


Figure 7.4: Layered wireless communication structure for the cross-design

The dedicated wireless communication protocols developed in Chapter 5 are suitable for the inverted pendulum system. These protocols are based on a slight modification of the IEEE 802.15.4-2006 standard [3]. The wireless communication structure considered for the cross-design is illustrated on the control block diagram of the inverted pendulum in Figure 7.4. In order to implement the dedicated wireless communication protocols on the Imote2 hardware platform, an application development environment was developed including a Codeloader and a Bootloader (see Chapter 3). The application development environment includes a library of functions implemented in the C programming language. The embedded application interface enables the easy implementation of user code (e.g. control algorithms and communication protocols) which can be loaded into the memory of the Imote2 using the platform independent Bootloader.

[3] Refer to Section A.3 of the Appendix for details about the IEEE 802.15.4-2006 standard

The main features of the dedicated wireless communication protocols [4] are:

- Different MAC protocols at the communication layer including schedule-based (e.g. TDMA and Master-Slave) and contention-based (e.g. CSMA/CA)
- Time synchronization algorithm
- Power control algorithm at the physical layer for the online adaptation of the transmit power based on RSSI signal measurements
- Analytical model for the computation of the average wireless node lifetime at the control layer used by the optimizer node during the cross-design optimization phase

MAC Protocol	RTT $[ms]$	σ_{RTT} $[ms]$	ρ^{sc} $[\%]$	$\sigma_{\rho^{sc}}$ $[\%]$
Master-Slave	19.12	0.55	1.33	0.13
TDMA	13.20	0.97	1.57	0.46
CSMA/CA	13.00	2.12	1.61	0.41

Table 7.2: Experimental round trip time (RTT) and packet loss probability measurements for selected MAC schemes

Experimental round trip time and packet loss probability measurements for selected MAC schemes are summarized in Table 7.2. The mean round trip time RTT and the corresponding standard deviation σ_{RTT} are the results of several experiments carried out for different sampling periods h. As can be seen, the inverted pendulum can be stabilized using each of the implemented MAC schemes at the sampling period $h = 20ms$ and shorter sampling periods $15ms \leq h \leq 20ms$ are only achievable with the TDMA and CSMA/CA MAC schemes. The measured mean packet loss probability ρ^{sc} and the corresponding standard deviation $\sigma_{\rho^{sc}}$ are very small and do not significantly affect the control performance. However, the number of tunable MAC protocol parameters, i.e., the degrees of freedom of the cross-design varies depending on the selected MAC scheme. The CSMA/CA MAC scheme has more tunable parameters than the TDMA MAC scheme and the Master-Slave MAC scheme. The complexity of the CSMA/CA MAC scheme increases with the number of nodes in the wireless network which in turn complicates the cross-design optimization. The TDMA MAC scheme has a better performance in terms of the minimization of the network energy consumption through its duty cycling capability. The small round trip time of the TDMA MAC scheme at the selected sampling period $h = 20ms$ compared to the round trip time of the Master-Slave MAC scheme allows the sensor nodes to sleep longer, thus, saving energy and maximizing the average wireless network lifetime. As the number of nodes in the wireless network increases, the round trip time of the Master-Slave MAC scheme increases faster than with the TDMA MAC scheme, thus, reducing the minimum applicable sampling period for the control application. All MAC schemes developed in the course of this thesis can be used to illustrate the proposed cross-design approach on the inverted pendulum system and applying the cross-design procedure using these MAC schemes is straight forward. The proposed MAC schemes are implemented such that switching between the different MAC schemes at runtime is possible. However, the increased number of cross-design parameters, i.e., the MAC protocol parameters, complicates the cross-design optimization.

In the remaining of this chapter, the TDMA MAC scheme is selected to illustrate the proposed cross-design approach on the inverted pendulum testbed and the cross-design procedure is the same for the other MAC schemes. The superframe structure of the selected TDMA MAC scheme is depicted in Figure 7.5 on Page 89.

[4] Refer to Chapter 5 for details about the wireless communication protocols

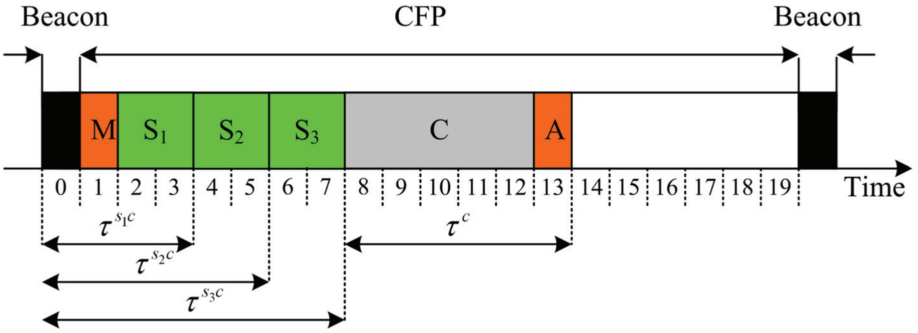


Figure 7.5: TDMA MAC protocol structure of the cross-design for $S = 20$. S1: Position, S2: Angle, S3: Velocity, C: Controller, A: Actuator, M: Sampling process

Following communication protocol assumptions are made for the cross-design based on simulation and experimental results from Chapter 5 and the assumptions about the inverted pendulum system made in Section 7.1 on Page 83:

- Number of time slots in a superframe: $S = 20$ for $h = 20ms$
- Slot assignment: $s_i = 2$ for each sensor node (i.e., position, angle, velocity), $s_i = 5$ for the controller node, $s_i = 1$ for the actuator node, and $s_i = 1$ for the sampling process and beacon duration
- Slot duration: $s_d = 1ms$
- Superframe duration: $SD = BI = h$
- Beacon Interval, superframe order, and beacon order: $BI = h$, $SO = BO = 0$

The cross-design parameters at the communication layer for the TDMA MAC scheme are the Beacon interval BI which determines the sampling period at the control layer and the transmit power P_t. Since SD is constant, BI also determines the length of the inactive portion of the superframe during which nodes can be switched off to save energy using duty cycling. Due to the scheduled nature of the TDMA MAC scheme, there are no packet collisions and packet losses only occur if the transmit power is low or the nodes are not in line of sight. Packet collisions may also occur if the nodes lose synchronization which is the case when one or more sensor nodes do not receive the synchronization beacon from the controller node. Moreover, transmission delays as well as computational delays are assumed to be constant and integer multiples of the slot duration $s_d = 1ms$.

7.2 Results of the cross-design procedure

Prior to the cross-design optimization procedure as well as results presented and discussed in Section 7.3, performance measurements are carried out on the inverted pendulum testbed during the design phase in order to verify both the control as well as wireless communication assumptions made in Section 7.1.1 and Section 7.1.2 respectively. The goal is to determine the feasible control performance requirement J_{req} as a function of the selected sampling period h, delay τ^{sc}, packet loss probability ρ^{sc}, and desired average wireless node lifetime $\overline{T}$. Combinations of feasible control performance requirements, sampling periods, delays, packet loss probabilities, average wireless node lifetime, and corresponding optimal controller and Kalman estimator gains are stored in lookup tables for use during the optimization phase of proposed the cross-design procedure.

The choice of the Kalman estimator was motivated among others by the assumption that measurement noise on sensor values are Gaussian, uncorrelated, and with mean zero. In order to verify this assumption, several experiments were carried out on the inverted pendulum testbed to estimate the measurement noise. To this end, controller and actuator were switched off and the inverted pendulum positioned at the origin, i.e., $x = 0$, $\varphi = 0$, and $\dot{x} = 0$. Noise measurement results are shown in Figure 7.6. As can be seen, the measurement noise on the inverted pendulum position, angle, and velocity is approximately Gaussian with zero mean which justifies the choice of the Kalman estimator.

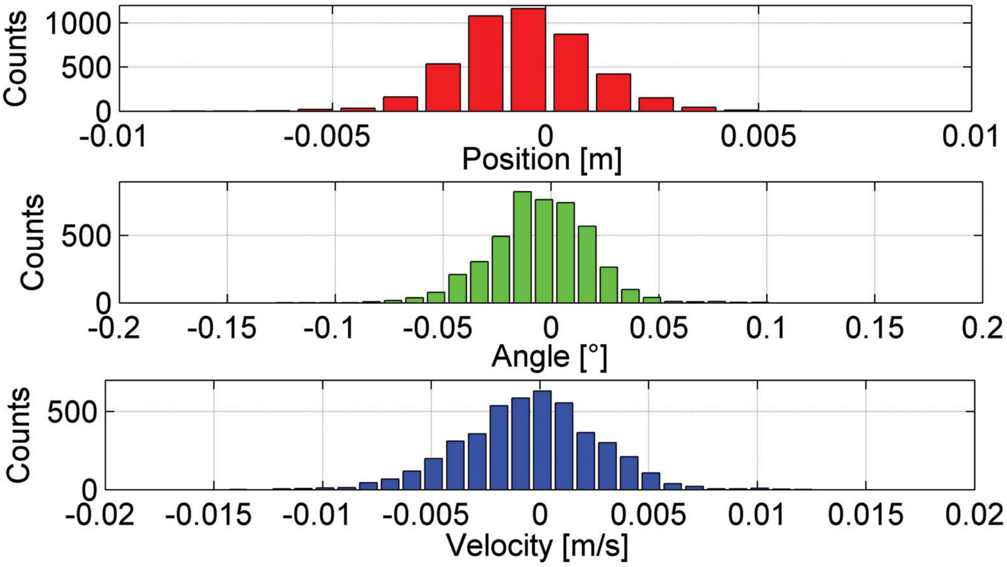


Figure 7.6: Sensor noise measurement results for the inverted pendulum testbed

Kalman estimator equations for the inverted pendulum system used in **Step 2** and **Step 6** of the cross-design procedure (see Figure 7.11 on Page 95) are derived according to equations (6.13)-(6.20). Critical packet loss probabilities above which the Kalman estimator theoretically loses stability are summarized in Table 7.3 for different sampling periods with constant delay $\tau^{sc} = 15ms$ corresponding to the minimum delay (integer multiple of the basic delay τ_0) achievable with the TDMA MAC scheme. These critical packet loss probabilities are the solutions of the LMI formulated in equation (6.24) and are functions of the sampling period h and delay τ^{sc}. Up to 25% packets can be lost at the sampling period $h = 20ms$ before the Kalman estimator loses stability. As the sampling period increases, the critical packet loss probability decreases and for $h = 100ms$ and $\tau^{sc} = 95ms$ the Kalman estimator still tolerates up to 20% packet losses before losing stability. HiL Simulation results for the inverted pendulum in Chapter 4 showed that the worst case MAPL is smaller than the critical number of packet losses for which the Kalman estimator loses stability. Therefore, the Kalman estimator will not limit the performance of the inverted pendulum system. In the sequel, the maximum packet loss probability will be limited to $\rho^{sc}_{max} = 20\%$ without loss of generality.

$h[ms]$	20	30	40	50	60	70	80	90	100
$\rho^{sc}_{crit}[\%]$	25.27	25.21	25.02	24.71	24.27	23.73	23.09	22.37	21.57

Table 7.3: Computed critical packet loss probabilities for the Kalman estimator and $\tau^{sc} = 15ms$

The continuous-time state space system equations for the inverted pendulum system are given by equations (7.3)-(7.4) with system matrices (7.5), (7.6), and (7.7). The corresponding linear stochastic difference equations with intermittent sensor measurements are obtained for the inverted pendulum system according to equations (6.8)-(6.11) derived in Chapter 6 resulting in the following system matrices:

$$A_d \approx I + Ah_k + \frac{1}{2}A^2h_k^2 + \frac{1}{3}A^3h_k^3 + ... \tag{7.8}$$

$$\Gamma_0 \approx \left(I(h_k - \tau_k^{sc}) + \frac{1}{2}A(h_k - \tau_k^{sc})^2 + \frac{1}{6}A^2(h_k - \tau_k^{sc})^3 + ... \right) B \tag{7.9}$$

$$\begin{aligned} \Gamma_1 \approx & \left(I + A(h_k - \tau_k^{sc}) + \frac{1}{2}A^2(h_k - \tau_k^{sc})^2 + ... \right) \cdot \\ & \left(I\tau_k^{sc} + \frac{1}{2}A(\tau_k^{sc})^2 + \frac{1}{6}A^2(\tau_k^{sc})^3 + ... \right) B. \end{aligned} \tag{7.10}$$

Following the procedure described in Section 6.3.4 of Chapter 6, the optimal LQG control input u^* and the corresponding control cost J_p^* can be derived using equations (6.32) and (6.33) respectively. The optimal control input u^* and the corresponding control cost J_p^* are both functions of the measured delay τ^{sc} and packet loss probability ρ^{sc} between sensors and controller as well as the selected sampling period h. The bounds of the LQG control cost for the infinite horizon case (i.e. $p = \infty$) given by equation (6.43) are computed offline during the design phase and stored in lookup tables for use during the cross-design optimization phase. For the finite horizon case (i.e. p finite), the bounds of the LQG control cost given by equation (6.40) are computed online and used to select appropriate LQG controller parameters from the lookup tables during the cross-design optimization. The value of p is chosen within the stability bounds of the Kalman estimator. Optimal LQG controller parameters for selected sampling periods within the stability bounds of the inverted pendulum and constant delay $\tau^{sc} = 15ms$ for the TDMA MAC scheme are summarized in Table 7.4. If the delay is not constant, then a similar lookup table is obtained for each possible delay τ equal to an integer multiple of the basic delay τ_0 resulting in a two dimensional lookup table for the optimal LQG controller gains. As will be seen later on, a three dimensional lookup table for the optimal LQG controller gains is obtained by considering the dependency on the packet loss probability ρ^{sc}.

$h[ms]$	K_x	$K_{\dot{x}}$	K_φ	$K_{\dot{\varphi}}$
20	-0.5939	1.8829	1.9529	0.3058
30	-0.5444	1.7616	1.8221	0.2897
40	-0.4992	1.6504	1.7023	0.2749
50	-0.4579	1.5485	1.5927	0.2611
60	-0.4202	1.4551	1.4924	0.2483
70	-0.3857	1.3695	1.4006	0.2365
80	-0.3543	1.2911	1.3166	0.2254
90	-0.3256	1.2192	1.2397	0.2152
100	-0.2995	1.1533	1.1693	0.2058

Table 7.4: Optimal LQG controller gains for selected sampling periods

$h[ms]$	20	30	40	50	60	70	80	90	100
$\tau^{sc}[ms]$	12.89	12.64	13.34	13.70	13.59	12.95	12.84	12.98	13.40
$\sigma_{\tau^{sc}}[ms]$	0.98	0.80	0.92	0.65	0.89	1.09	1.04	0.81	0.86

Table 7.5: Experimental delay measurement results for the inverted pendulum testbed

Experimental delay measurement results for the inverted pendulum testbed are summarized in Table 7.5. For each of the selected sampling period, several measurements were carried out and the mean delay τ^{sc} as well as the corresponding standard deviation $\sigma_{\tau^{sc}}$ were computed.

Figure 7.7 shows an excerpt of delay measurement results for ten different experiments. As can be seen, the total delay from sensing to actuation for the LQG controller and the TDMA MAC scheme is $\tau^{sc} < 15ms$.

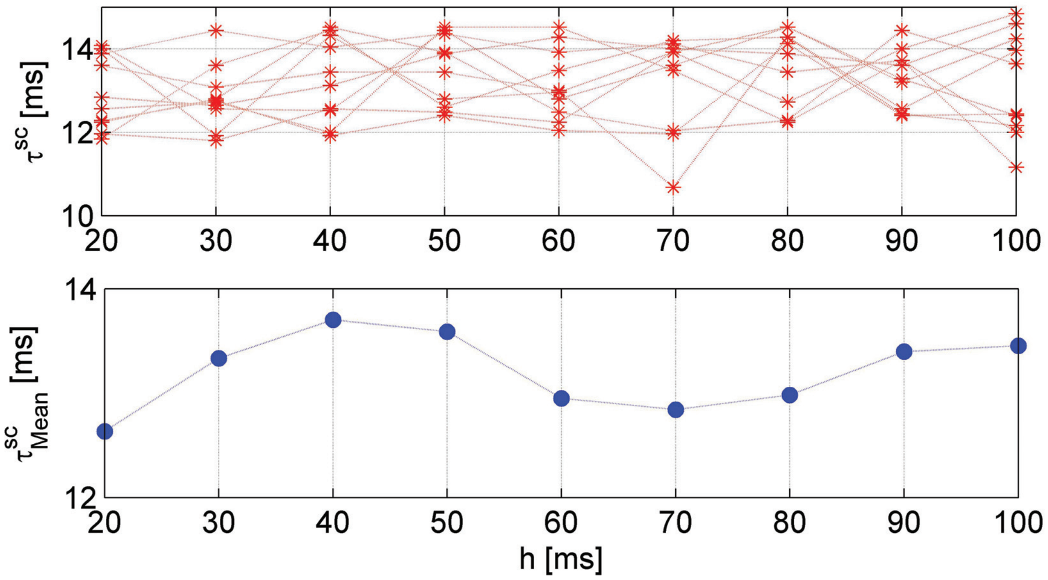


Figure 7.7: Excerpt of experimental delay measurement results for the inverted pendulum tesbed

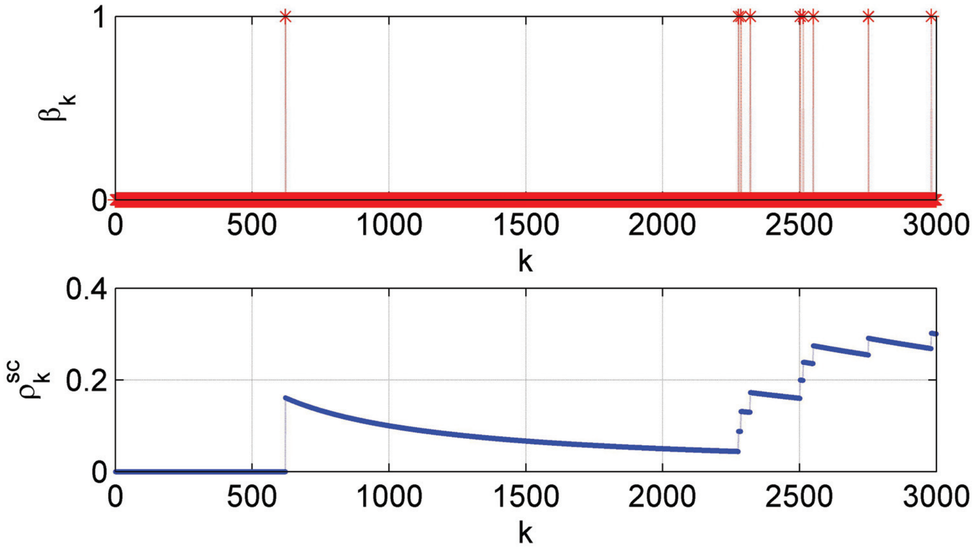


Figure 7.8: Experimental results of packet loss measurement at $h = 20ms$

Figure 7.8 shows the result of experimental packet loss measurements for the inverted pendulum tesbed and overall results are summarized in Table 7.6 on Page 93. Several experiments similar to the one in Figure 7.8 were carried out in order to compute the mean packet loss rate ρ^{sc} and the corresponding standard deviation $\sigma_{\rho^{sc}}$. The obtained packet loss probabilities are estimated from the packet loss statistics over the wireless network variable β_k. $\beta_k = 1$ if at least one of the three sensor packets is lost and $\beta_k = 0$ if all three sensor packets are received by the controller during the same sampling period. The relative packet loss probability ρ^{sc}_k is obtained by summing up the values of β_k and dividing the result by the total number of transmitted packets at sampling time instant k. As can be seen, there are few sensor packet losses due to the scheduled nature of the TDMA MAC scheme and the corresponding packet loss probability is very small. The rare packet losses occur when sensor nodes lose synchronization.

$\rho^{sc}[\%]$	$\sigma_{\rho^{sc}}[\%]$
1.33	0.13

Table 7.6: Experimental packet loss measurement results for $h = 20ms$

Figure 7.9 shows the computed control cost J_{req} for selected sampling periods h and packet loss probabilities ρ^{sc}. The total delay from sensing to actuation is assumed to be constant $\tau^{sc} = 15ms$ for the TDMA MAC Scheme. The feasible control cost J_{req} is computed using equation (6.43) derived in Chapter 6. A point $M = (J_{req}, h, \rho^{sc})$ in Figure 7.9 is feasible if it satisfies the required control cost J_{req} and $\rho^{sc} \leq \rho^{sc}_{max}$ for the selected sampling period h. The feasible region Ω_M is the set of all feasible points M. As the control performance requirement J_{req} becomes strict, the infeasible region increases and the feasible region Ω_M shrinks. As can be seen, longer sampling periods increase the feasible control cost and hence, the average wireless node lifetime. Also, packet losses at higher sampling periods are very critical for the stability of the inverted pendulum. Moreover, the sampling period is the dominant factor in the control cost compared to the packet loss probability. For the inverted pendulum testbed, all points for which $\rho^{sc} > \rho^{sc}_{crit}$ are not feasible since the Kalman estimator may lose stability. Results in Figure 7.9 are stored in a lookup table and used during the optimization phase of the proposed cross-design approach. A third dimension can be added to the lookup table for the required control cost by considering variable wireless communication delays τ^{sc}. Alternatively, separate lookup tables for the required control cost can be generated for selected delays τ^{sc} equal to integer multiples of the basic delay τ^{sc}_0.

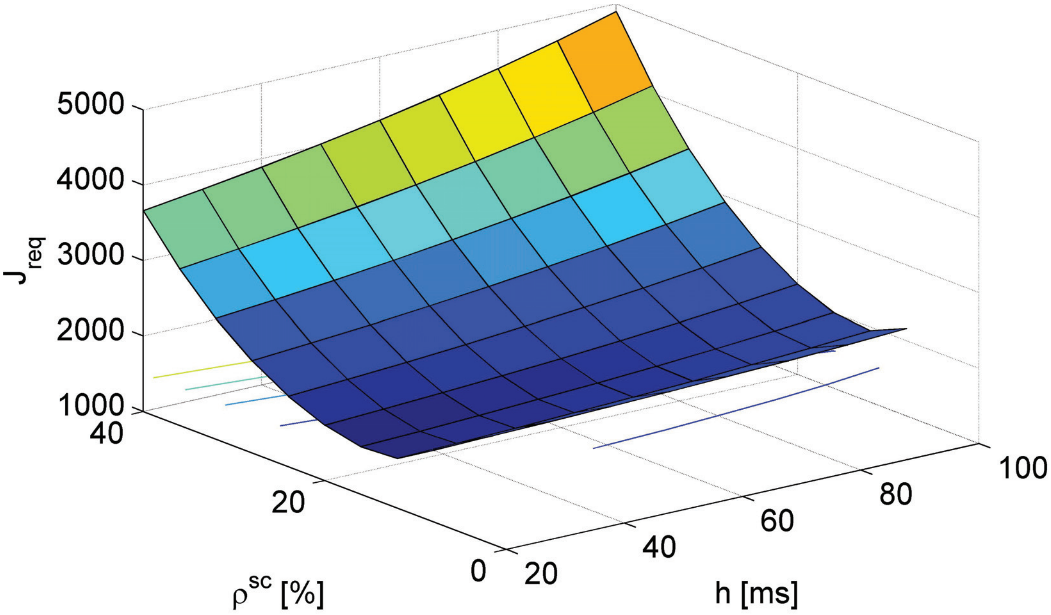


Figure 7.9: Feasible control cost of the inverted pendulum system with the LQG controller and TDMA MAC scheme for selected sampling periods and packet loss probabilities

The constrained cross-design optimization problem formulated in equation (6.12) derived in Chapter 6 and solved during **Step 3** of the cross-design procedure (refer to Figure 7.11 on Page 95) can be reformulated as follows for the inverted pendulum system:

$$\max_{h,\mathbf{p},\varsigma} \quad \overline{T}(h, \mathbf{p}, \varsigma) \tag{7.11}$$

$$s.t. \quad J\left(h, \rho^{sc}\left(h, \mathbf{p}, \varsigma\right), \tau^{sc}\left(h, \mathbf{p}, \varsigma\right)\right) \leq J_{req} \tag{7.12}$$

$$\rho^{sc}\left(h, \mathbf{p}, \varsigma\right) \leq \rho^{sc}_{crit} \tag{7.13}$$

$$\tau^{sc}\left(h, \mathbf{p}, \varsigma\right) < \tau^{sc}_{max} \tag{7.14}$$

where $\mathbf{p} = \{BI, P_t\}$, $\varsigma = TDMA$, and $\tau^{sc}_{max} = 15ms$ for the TDMA MAC scheme. The average wireless node lifetime is given by

$$\overline{T} \approx \frac{E_{Tot}}{\left[\bar{P}_W(\alpha, l)T_W + P_{M_1}(h - T_W) + E_{M_{12}}\right]} h \tag{7.15}$$

$$\bar{P}_W = \frac{P_{M_2}\bar{T}_{R_X} + P_{M_3}(l)\alpha\bar{T}_{T_X} + \frac{\alpha}{3+\alpha}E_{M_{23}}}{\bar{T}_{R_X} + \alpha\bar{T}_{T_X}} \tag{7.16}$$

where $\bar{P}_W$ is the average wireless node power consumption during the awake period, P_{M_2} the power consumption during the receive mode, P_{M_1} the power consumption during the sleep mode, P_{M_3} the power consumption during the transmit mode, $E_{M_{23}}$ the power consumption during the transition from receive to sleep mode, $\bar{T}_{T_X}$ the average duration of the transmit mode, $\bar{T}_{R_X}$ the average duration of the receive mode, and l the transmit power level for the transmit power P_t.

Following assumptions are true for the computation of the average wireless node lifetime[5] using the TDMA MAC scheme:

- The duration of the awake period T_W is constant and identical for each sensor node
- The node access probability is $\alpha = 1$
- The power consumption during the awake period depends only on the selected sampling period, i.e., transmission and reception times are constant

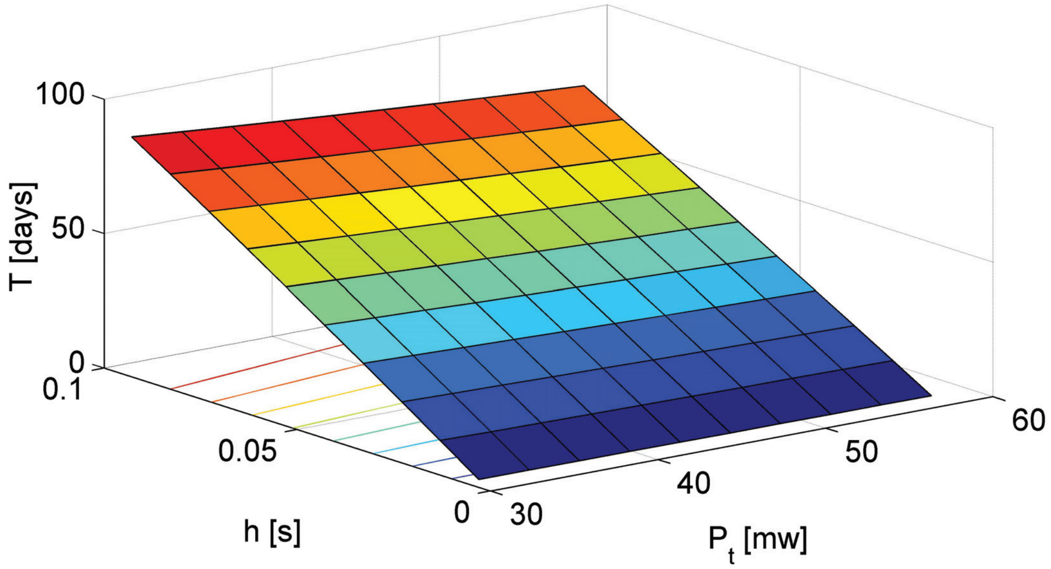


Figure 7.10: Average node lifetime for selected sampling periods and transmit powers at constant delay $\tau^{sc} = 15ms$ for the TDMA MAC scheme

Figure 7.10 shows the computed average wireless node lifetime for selected sampling periods h, transmit powers P_t, and constant delay $\tau^{sc} = 15ms$ for the TDMA MAC scheme. Complete results are summarized in Table 7.7 on Page 95 which are stored in a lookup table for use during the cross-design optimization phase. As can be seen, the average node lifetime increases for higher sampling periods but decreases with increasing transmit powers. Decreasing the transmit power may increase the packet loss probability if the nodes are not closed enough to each other. The maximum average node lifetime $\overline{T} \approx 90days$ is achieved at the maximum sampling period $h = 100ms$ and the minimum transmit power $P_t = 30mW$. However, the corresponding

[5] Refer to Chapter 5 for more details

feasible control cost increases with the sampling period resulting in a poor control performance (see Figure 7.9 on Page 93). Therefore, a tradeoff should be made between acceptable control performance and desired average wireless node lifetime. The sampling period is the dominant factor in the average wireless node lifetime compared to the transmit power.

$h[ms]/P_t[mW]$	30	33	36	39	42	45	48	51	54	57
10	2.99	2.93	2.88	2.83	2.78	2.73	2.68	2.64	2.59	2.55
20	12.91	12.67	12.44	12.22	12.00	11.80	11.60	11.40	11.22	11.04
30	22.74	22.32	21.92	21.53	21.15	20.79	20.44	20.10	19.77	19.45
40	32.49	31.89	31.32	30.76	30.23	29.71	29.21	28.73	28.26	27.81
50	42.15	41.38	40.64	39.92	39.23	38.56	37.92	37.29	36.69	36.11
60	51.73	50.79	49.88	49.01	48.16	47.34	46.56	45.79	45.05	44.34
70	61.23	60.12	59.05	58.02	57.02	56.06	55.13	54.23	53.35	52.51
80	70.64	69.37	68.14	66.95	65.81	64.70	63.63	62.59	61.59	60.62
90	79.97	78.54	77.15	75.82	74.52	73.28	72.07	70.90	69.77	68.67
100	89.23	87.63	86.10	84.61	83.17	81.79	80.44	79.14	77.89	76.67

Table 7.7: Summary of the average sensor node lifetime $\overline{T}$ days for selected sampling periods and transmit powers

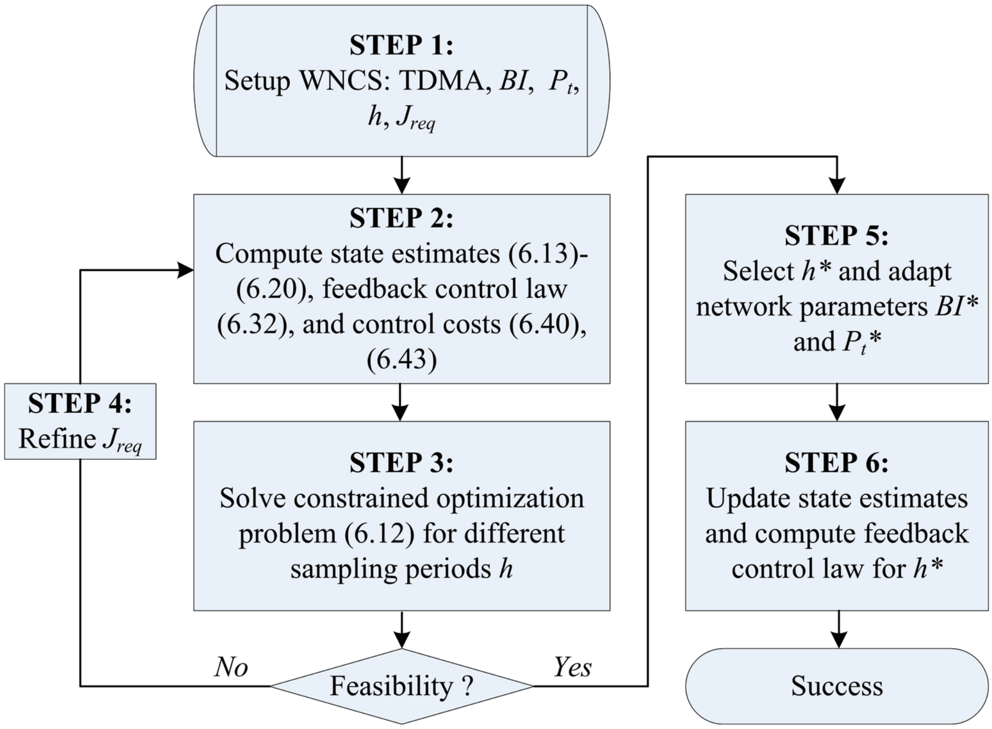

STEP 1:
Setup WNCS: TDMA, BI, P_t, h, J_{req}
STEP 2:
Compute state estimates (6.13)-(6.20), feedback control law (6.32), and control costs (6.40), (6.43)
STEP 3:
Solve constrained optimization problem (6.12) for different sampling periods h
STEP 4:
Refine J_{req}
Feasibility ?
No
Yes
STEP 5:
Select h^* and adapt network parameters BI^* and P_t^*
STEP 6:
Update state estimates and compute feedback control law for h^*
Success

Figure 7.11: Cross-design procedure for the inverted pendulum system with the LQG controller and the TDMA MAC protocol

Now, the general cross-design procedure presented in Chapter 6 can be adapted as shown in Figure 7.11 for the inverted pendulum system with the selected TDMA MAC scheme using the results derived in this section. Since the MAC scheme is fixed, **Step 1** of the cross-design procedure is carried out once and controller and wireless communication parameters are initialized with default values, e.g., $BI = h$, $h = h_0$, and $P_t = 0dBm$. The control cost J_{req} is selected such as to obtain satisfactory control performance and acceptable average node lifetime. **Step 2**, **Step 3**, and **Step 4** are carried out until a feasible solution to the constrained cross-design optimization problem is found. Optimal LQG controller gains and TDMA MAC protocol parameters are adapted in **Step 5**. In **Step 6** obtained optimal LQG controller gains are used to update the state estimates and compute the feedback control law for the optimal sampling period h^*. Optimal LQG controller gains and corresponding TDMA MAC protocol parameters are stored in lookup tables.

7.3 Results of the cross-design optimization

In this section, we illustrate the cross-design optimization on the inverted pendulum system using the optimization structure in Figure 7.12. The TDMA MAC protocol is selected during initialization and default parameters for the transmit power $P_t = 0dBm$, sampling period $h = h_0$, LQG controller gain K_0 (e.g. optimal LQG controller for the wired case), and control performance requirement $J_{req} = J_{max}$ (e.g. maximum LQG control cost for the wired case) are also selected and applied. The optimization procedure itself includes **Step 1** to **Step 5** which are executed at each sampling period. During **Step 1**, QoS metrics (e.g. delay τ_k^{sc} and packet loss β_k) and QoC metric (e.g. optimal LQG control cost J_p^* where p is the finite control horizon) metrics are measured and computed respectively at each sampling time instant k. In **Step 2**, QoS and QoC metrics are used to estimate the average node lifetime $\overline{T}_k$ at sampling time instant k. If the estimated average node lifetime is not satisfactory, the required control performance J_{req} is adapted in **Step 3** and corresponding optimal LQG controller parameters (h^* and K^*) and communication parameters (P_t^* and BI^*) are taken from the available lookup tables and applied at sampling time instant k. Then the optimization procedure starts from **Step 1** again.

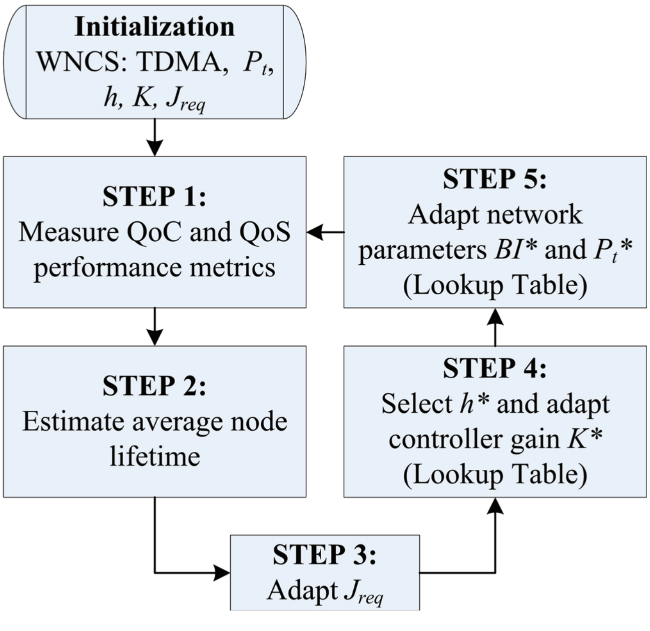

Initialization
WNCS: TDMA, P_t, h, K, J_{req}
STEP 1:
Measure QoC and QoS performance metrics
STEP 2:
Estimate average node lifetime
STEP 3:
Adapt J_{req}
STEP 4:
Select h^* and adapt controller gain K^* (Lookup Table)
STEP 5:
Adapt network parameters BI^* and P_t^* (Lookup Table)

Figure 7.12: Structure of the cross-design optimization for the inverted pendulum system

In order to demonstrate the effectiveness of the proposed cross-design approach using the optimal LQG controller, a standard LQG controller is used as the basic controller. On the contrary of the optimal LQG controller, the gains of the standard LQG controller are not adapted online based on the measured control performance J_{meas} and measured QoS metrics τ^{sc} and ρ^{sc}, but are computed offline using measured average wireless network delays and packet loss probabilities. Since packet losses especially at higher sampling periods significantly affect the control performance by increasing the control cost, additional packet losses are added to the measured packet losses of the TDMA MAC scheme using the wireless network block embedded in the controller node. Additional packet losses are carefully added in such a way that the stability bounds of both the inverted pendulum system and the Kalman estimator are not violated, i.e., $\rho_k^{sc} < \rho_{crit}^{sc}$. While the standard LQG controller gains are computed for the average packet losses of the TDMA MAC scheme $\rho_{mean}^{sc} \approx 1.5\%$, the optimal LQG controller adapts its gains to the measured packet loss rates ρ_k^{sc} at each sampling time instant k resulting in a better control performance for the optimal LQG controller. The packet loss probabilities in Figure 7.13 are obtained using the network block embedded in the controller node during one cross-design optimization cycle. The resulting average packet loss probability $\rho^{sc} \approx 6\%$ is almost four times the measured average packet loss probability of the TDMA MAC scheme.

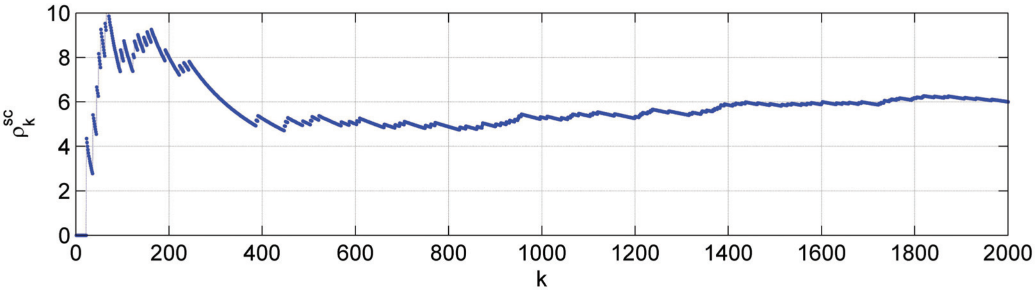


Figure 7.13: Measured packet loss rates during one cross-design optimization cycle

A comparison of the measured control cost J_{meas} of the standard LQG controller and the optimal LQG controller is shown in Figure 7.14 for a selected control cost $J_{req} = 4000$. For this selected control cost, all sampling periods within the stability bounds of the inverted pendulum system are applicable and both the optimal LQG controller and the standard LQG controller are able to stabilize the inverted pendulum. Starting with the basic sampling period $h = h_0$, **Step 1** to **Step 5** of the optimization procedure in Figure 7.12 on Page 96 are carried out. As long as the required control cost is feasible, the sampling period is increased and the network protocol parameters adapted accordingly until the maximum achievable average wireless node lifetime is obtained.

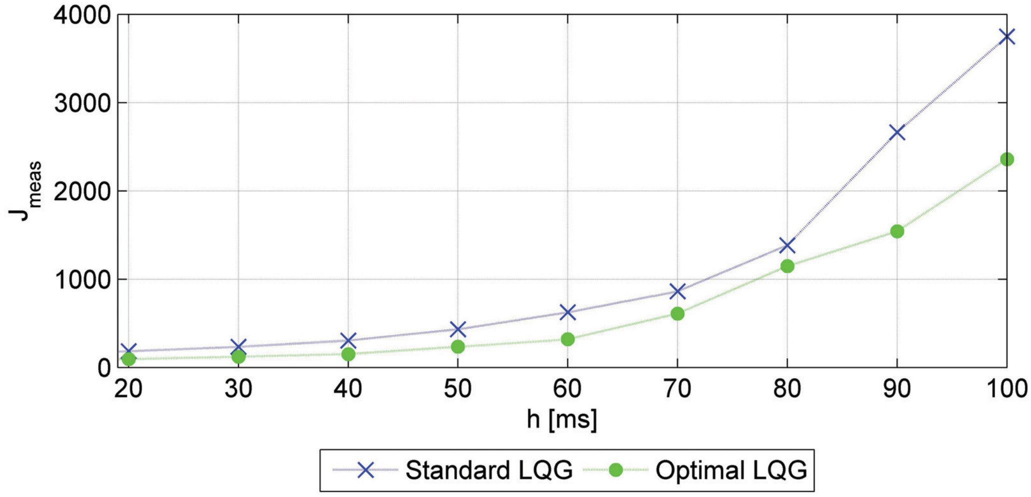


Figure 7.14: Measured control performance of the optimal and standard LQG controllers

As can be seen, the optimal LQG controller performs better than the standard LQG controller over the range of applicable sampling periods. Control costs $J_{req} \leq 1000$ are achievable for sampling periods $h \leq 70ms$ but at the expense of a reduced average wireless node lifetime. For sampling periods $h > 80ms$, the optimal LQG controller outperforms the standard LQG controller although both controllers achieve the same average wireless node lifetime. As before,

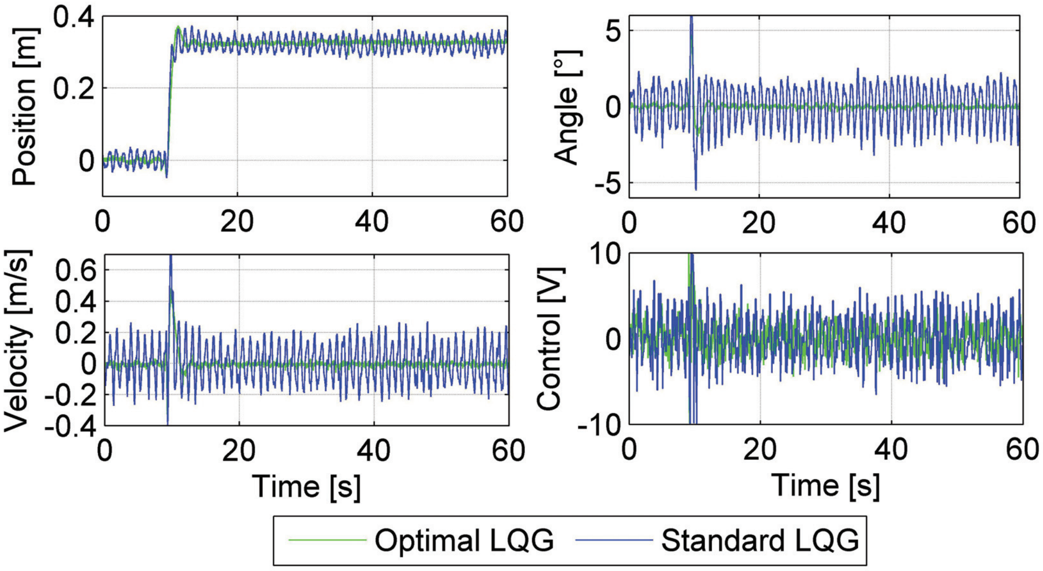


Figure 7.15: Step responses of the standard and optimal LQG controllers

a tradeoff should be made between acceptable control performance and desired average wireless node lifetime during the cross-design optimization.

The step responses of the optimal LQG controller and the standard LQG controller are shown in Figure 7.15 for a selected sampling period $h = 100ms$. A step position input of amplitude $x = 0.3m$ is applied to the inverted pendulum at time $t = 10s$. Inverted pendulum position, angle, velocity, and control values are recorded and plotted. As can be seen, the optimal LQG controller has a better control performance than the standard LQG controller which further proves the effectiveness of the proposed cross-design approach.

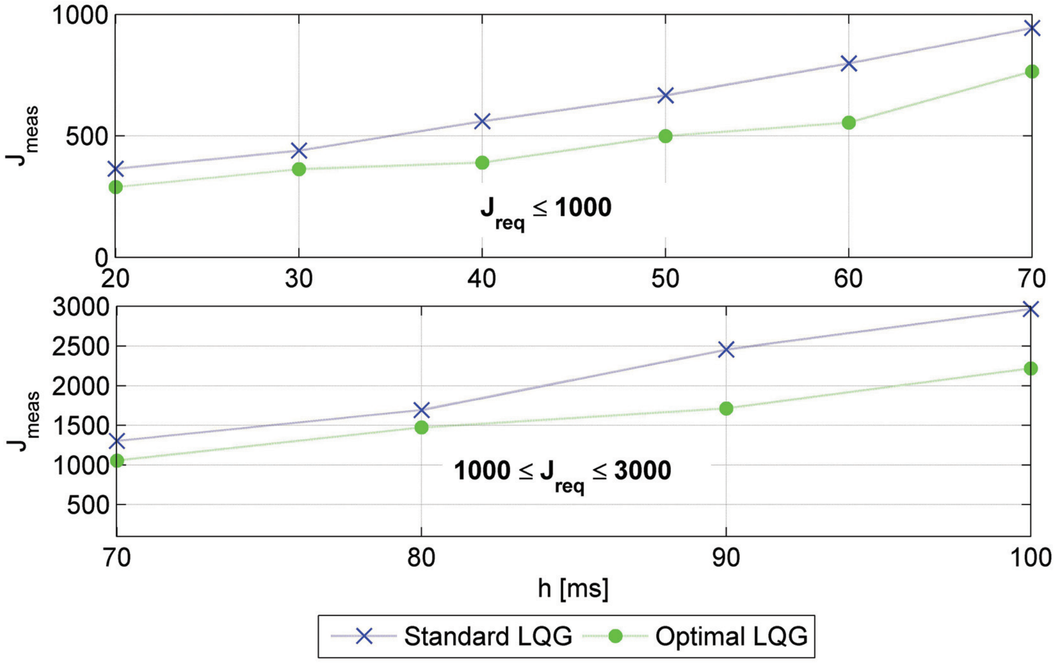


Figure 7.16: Sampling period adaptation due to performance requirements changes

Next, the performance of the proposed cross-design approach is further illustrated on a second example. To this end, the packet loss rate is modified between two cross-design optimization cycles using the network block embedded in the controller node. Figure 7.16 illustrates the adaptation of the control performance requirement in terms of the sampling period when the required control cost changes from $1000 \leq J_{req} \leq 3000$ to $J_{req} \leq 1000$ within two optimization cycles. The performance requirement change is caused by an increase in the packet loss probability as shown in Figure 7.17 which limits not only the range of applicable sampling periods, but also the range of achievable average wireless node lifetimes. Sampling periods up to $h = 70ms$ are feasible for a desire control cost $J_{req} \leq 1000$ and a corresponding average packet loss probability $\rho^{sc} = 2\%$ obtained using the controller network block. The maximum achievable average wireless node lifetimes are $\overline{T} = 62days$ and $\overline{T} = 53days$ for selected transmit powers $P_t = 0dBm$ and $P_t = -25dBm$ respectively. For the control cost before switching $1000 \leq J_{req} \leq 3000$, sampling periods $70ms \leq h \leq 100ms$ are feasible with average node lifetimes $\overline{T} = 90days$ and $\overline{T} = 77days$ for selected transmit powers $P_t = 0dBm$ and $P_t = -25dBm$ respectively. Again, the optimal LQG controller performs better than the standard LQG controller.

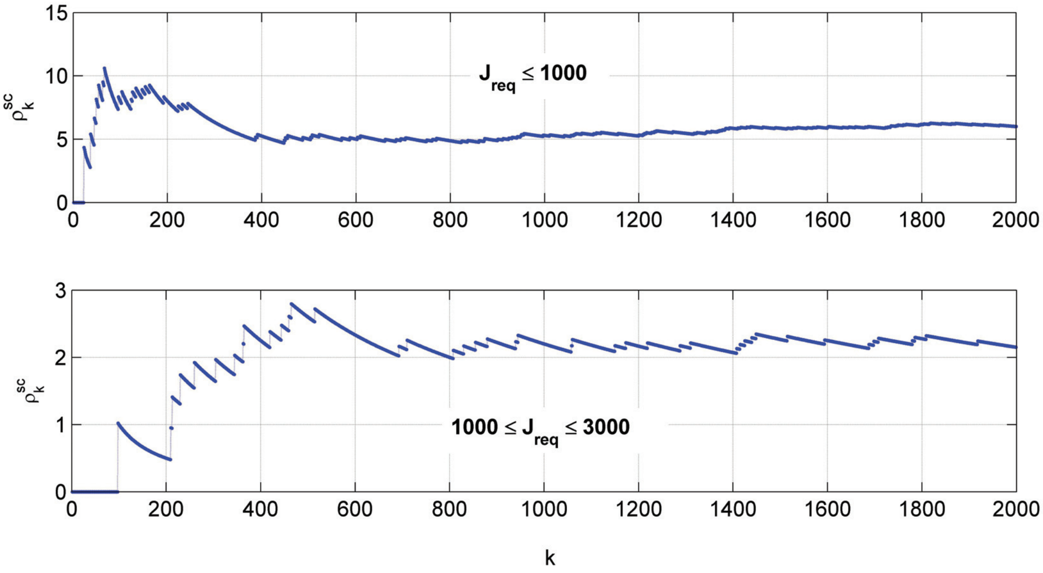


Figure 7.17: Measured packet loss probabilities corresponding to the performance requirements change in Figure 7.16 on Page 98

7.4 Summary

The dynamic interactions between the control system and the underlying wireless communication network are critical factors that determine the performance (e.g. stability and robustness) of WNCSs. As demonstrated in this chapter, the proposed cross-design framework and the corresponding cross-design approach are applicable to real wireless networked control applications. While applying the cross-design procedure to the inverted pendulum, we showed how wireless network effects such as delay and packet losses affect the performance of a control system by analyzing the feasible regions of the control performance. We further observed, that optimizing the traffic load of the wireless network is equivalent to either optimizing the wireless communication throughput or the control cost. Experimental cross-design results further showed that the sampling period of the control application significantly influences not only the performance of the control application, but also the throughput and the total energy consumption of the wireless network. Therefore, the sampling period is a key cross-design optimization factor which should be chosen carefully and wireless communication protocols should provide mechanisms to adapt the sampling period at runtime. Finally, the proposed control algorithms, dedicated wireless communication protocols, and cross-design optimization algorithms can be implemented on off the shelves wireless nodes with limited energy and computing resources.

8 Summaries

8.1 Summary in English

This thesis was aimed at developing a new approach to the model-based cross-design of WNCS[1] with limited computing, communication, and energy resources. An extensive review of the existing literature revealed the existence of three main approaches to the cross-layer-design of WNCSs: The first approach, also known as *communication aware controller design*, consists in implementing optimal control algorithms for a selected wireless communication system with known and fixed QoSs[2] (e.g. delay, packet loss, and network throughput). In this approach, the degrees of freedom are only available at the control level and are limited. Moreover, the control designer has no influence on the performance of the wireless network. The second approach, also known as *control aware communication design*, consists in designing suitable communication protocols for selected control applications with desired QoC[3] bounds such as stability and robustness. In this approach, the degrees of freedom are at the communication level and are also limited. Unfortunately, there are only few available communication protocols standards for WNCS and most control system designers do not have the required wireless communication knowledge. The third approach, also known as *cross-design*, is based on the joint design of control algorithms and communication protocols, thus, offering degrees of freedom at both the control and the communication level. This third approach was selected in this thesis due to its flexibility and the offered possibility of choosing cross-design degrees of freedom at both the control level and the communication level, thus, yielding more flexibility.

Prior to the development of the proposed cross-design for WNCS itself, the particularities of WNCS design as compared to classical control design were studied including the effects of wireless networked induced random delays and packet losses as well as computing and energy resource limitations on the control and communication performance. To this end, a HiL simulation environment for the inverted pendulum system was developed which allows the simulation of networked induced effects on the control performance using a real control plant. This HiL simulation environment allows not only the simulation of time-based control strategies but also the simulation of event-based control strategies with realistic network models. Different control strategies for the inverted pendulum system where implemented including among others a cascaded controller, state feedback controllers including pole placement and LQR, an IMC controller, and an H_∞ controller.

Based on the HiL simulation results, QoS bounds such as MAD [4], MAPL [5], MATI [6] where derived for each of the implemented control structure and used to design and implement a dedicated wireless communication protocol for the inverted pendulum system. The proposed dedicated wireless communication protocol includes several MAC protocols such as scheduled-based MAC (e.g. TDMA and Master/Slave), contention-based MAC (CSMA/CA) with corresponding MAC

[1] WNCS: Wireless Networked Control Systems

[2] QoS: Quality of Services

[3] QoC: Quality of Control

[4] MAD: Maximum Allowable Delay

[5] MAPL: Maximum Allowable number of consecutive Packet Losses

[6] MATI: Maximum Allowable Transfer Interval

parameters, and the added capability of adapting the transmit power at the physical layer at runtime. The proposed communication protocols are light and can be implemented on wireless nodes with limited computing, energy, and memory resources.

The proposed cross-design approach for WNCS includes two phases: A design phase which is carried out offline and an online optimization phase. The proposed cross-design approach is based on the selection, design, and implementation of suitable control structures and algorithms as well as dedicated wireless communication protocols. Control algorithms and wireless communications protocols are designed in order to meet desired control performance requirements while minimizing the total energy consumption of the wireless network at the same time (i.e. maximizing the average wireless node lifetime) using the available degrees of freedom at the control layer (e.g. control structure, controller gains, and sampling period) and the available degrees of freedom at the communication layer (e.g. MAC protocols, MAC parameters, and transmit power). Control structures and corresponding controller parameters can be adapted offline and most importantly online to changing network conditions in order to obtain best control performance. At the same time, communication protocols and corresponding protocol parameters can be tuned both offline and online to meet a desired control performance and maximize the lifetime of the wireless network. In order to reduce the computational complexity of the proposed cross-design algorithms, optimal controller parameters (e.g. gains and sampling period) for selected network QoS are computed offline and stored in lookup tables for use online during the optimization phase. During the online optimization phase, a constrained single-objective optimization problem is solved, where the objective function is the average wireless network lifetime and the constraints are the delay and packet loss probability requirements, which are derived from a desired control cost.

The applicability of the proposed cross-design approach to a real control plant with selected wireless nodes was illustrated on the inverted pendulum testbed developed in the course of this thesis. Simulation and experimental results of the design phase and the optimization phase of the proposed cross-design approach showed not only the performance of the proposed cross-design approach compared to existing standard cross-design approaches, but also the effective implementation of the selected control algorithms and the dedicated wireless communication protocols on the same wireless hardware platform with limited computing, energy and memory resources.

Outlook

The research field of NCS[7] in general and WNCS in particular is growing very fast and the availability of wireless communication protocol standards, tailored to the specific needs of control applications, as well as the availability of pervasive, energy efficient, and computationally efficient wireless nodes is opening new research possibilities and directions. One interesting research issue is the extension of the proposed cross-design methods to distributed WNCS with several control loops sharing the same wireless communication network and the inclusion of a wireless network path between the controller node and the actuator node of the individual control loops. Another research issue would be the extension of the proposed cross-design approach to distributed and event-based WNCS using for example the inverted pendulum system and the flow control process presented in this work. In both of the above named research fields, bottleneck issues such as dynamic routing and bandwidth allocation should be carefully addressed at the communication layers. The distributed dynamic adaptation of controller parameters for the different control loops to changing wireless network conditions at the control layer should also be carefully addressed. The resulting distributed multi-objective cross-design optimization problem could be solved by extending the single-objective optimization problem presented in this thesis using selected appropriate degrees of freedom at both the control layer and the communication layers.

[7] NCS: Networked Control Systems

8.2 Kurzfassung in deutscher Sprache (Extended summary in German)

Motivation und wichtige Beiträge der Arbeit:

Die Verfügbarkeit von Funkkommunikationsstandards (z.B. WLAN, Bluetooth und ZigBee), die Entwicklung und Verabschiedung von neuen Funkkommunikationsstandards, zugeschnitten auf die Bedürfnisse der industriellen Automatisierung (z.B. WISA, WirelessHART und ISA-SP100), sowie die Entwicklung von kleinen, kostengünstigen und energiearmen Funkknoten, haben in den letzten Jahren den Einsatz von Funktechnologien in der industriellen Automatisierungstechnik begünstigt. Heute werden Funktechnologien unter anderem in mobilen Sensornetzwerken, unbemannten Fahrzeugen, automotiven Systemen, Gebäudeautomatisierung und Fabrikautomatisierung eingesetzt.

Funkbasierte Regelungssysteme sind Regelungssysteme, in denen Sensoren, Regler und Aktuatoren Informationen über ein Funknetzwerk austauschen. Die Vorteile gegenüber verdrahteten Regelungssystemen sind unter anderem die flexible Architektur, die Mobilität, niedrige Installations- und Unterhaltungskosten und wenig Kabel. Echtzeitanforderungen an ein funkbasiertes Regelungssystem sind von der spezifischen Regelungsapplikation abhängig und Netzwerkeffekte, wie zufällige Zeitverzögerungen und Paketverluste, die abhängig von der gewählten Funknetzwerkstruktur und der eingesetzten Funktechnologie entstehen, beeinträchtigen die Regelgüte und müssen beim Reglerentwurf berücksichtigt werden. Die Regelgüte eines über ein Funknetzwerk geregelten Systems ist eine komplexe Funktion von Reglerstruktur, Reglerparametern, Netzwerkstruktur und Netzwerkparametern. In funkbasierten vernetzten Regelungssystemen mit batteriebetriebenen Funkknoten stellt der Energieverbrauch der Funkknoten einen zusätzlichen limitierenden Faktor, sowohl für die Lebensdauer des gesamten Kommunikationssystems, als auch für die Güte des zu regelnden Systems, dar. Anders als bei reinen Sensornetzwerken muss nicht nur der Energieverbrauch der Funkknoten minimiert werden, sondern und vor allem auch eine akzeptable Regelgüte, wie zum Beispiel eine minimale Stabilität des Regelungssystems, garantiert werden. Neben dem Entwurf oder der Auswahl einer Hardware-Plattform mit sehr niedrigem Energieverbrauch, kann dieser zusätzlich dadurch minimiert werden, dass Hardwarekomponenten, zum Beispiel Prozessor und Transceiver, abgeschaltet werden, wenn diese nicht arbeiten. Dies erfordert allerdings den Einsatz von energieeffizienten Regelungsalgorithmen und Kommunikationsprotokollen, die gemeinsam entworfen und aufeinander abgestimmt werden müssen.

Bei der Entwicklung von funkbasierten Regelungssystemen haben sich in den letzten Jahren drei Methoden durchgesetzt. Bei der ersten Methode werden stabilisierende Regler ohne Berücksichtigung von Netzwerkeffekten entworfen und anschließend ein existierendes Kommunikationssystem ausgewählt und die QoS (z.B. Verzögerungen, Paketverluste und Durchsatz) des Kommunikationssystems auf die Bedürfnisse des Regelungssystems angepasst, um die beste Regelungsperformance zu erzielen (*Kommunikation trifft auf Regelung*). Diese Methode hat den Nachteil, dass der Regelungstechniker zwar Einfluss auf das Kommunikationssystem hat, die erzielbare Regelgüte allerdings von der Güte des ausgewählten Kommunikationssystems abhängt. Außerdem sind tiefe Kenntnisse über das Kommunikationssystem notwendig, um dessen Parameter zu ändern, die der Regelungstechniker nicht unbedingt hat. Bei der zweiten Methode wird zuerst ein vorhandenes Kommunikationssystem mit bekannten QoSs ausgewählt. Danach werden stabilisierende Regler unter Berücksichtigung der Eigenschaften des gewählten Kommunikationssystems entworfen, um die beste Regelungsperformance zu erzielen (*Regelung trifft auf Kommunikation*). Hier können zwar die Reglerparameter beim Reglerentwurf auf schwankende QoS Parameter des ausgewählten Kommunikationssystems anpasst werden, um die beste Regelgüte zu erzielen, aber der Regelungstechniker hat keinen Einfluss auf das Kommunikationssystem, welches weiterhin die Güte des geregelten Systems maßgeblich beeinflusst. Bei der letzten Methode, unsere Wahl in dieser Arbeit, werden Regelungssystem und Kommunikationssystem gemeinsam entworfen und aufeinander abgestimmt, um die beste Regelungsperformance zu erzielen. Wichtig hierbei ist die Klärung der Frage, welche Regelungsalgorithmen bzw. welche Kommunikationsprotokolle für gegebene Regelungsstrecken geeignet sind, und vor allem, welche Parameter zur Laufzeit dynamisch zwischen Regelung und Kommunikation ausgetauscht werden können, um die Regelgüte zu verbessern, die knappen Netzwerkressourcen optimal zu nutzen und Energie zu sparen.

Der methodische Kern dieser Arbeit ist die Entwicklung einer neuen Cross-Design-Methodik, die die Aspekte Control (Regelungsalgorithmus), Communication (Protokolle) und Computing (Energie, Rechenleistung) beim Entwurf funkbasiert vernetzter Regelungssysteme gemeinsam berücksichtigt. Hier werden synchron arbeitende Sensoren, Regler und Aktuatoren, sowie funkbasierte Single-Hop-Netzwerke betrachtet, die in der Entwurfsphase aufeinander abgestimmt und statisch konfiguriert werden. Abhängig von der aktuellen Regelungs- und Kanalgüte wird dazu dynamisch zwischen Regelalgorithmen, Reglerparametern und Netzwerkparametern umgeschaltet. Hierbei werden sowohl die Regelgüte des Regelungssystems, als auch der Energieverbrauch des Kommunikationssystems in dynamischen Umgebungen mit Unsicherheiten in der Ressourcenverfügbarkeit optimiert. Am Beispiel des Testbeds eines invertierten Pendels zeigt diese Arbeit die Verbesserungen gegenüber existierender Cross-Design-Ansätze. Insbesondere werden folgende wichtige Themen behandelt:

1. Aufbau und Inbetriebnahme eines invertierten Pendel-Testbeds für die experimentelle Validierung der Cross-Design-Ergebnisse (Kapitel 3)
2. Entwicklung einer HiL-Simulationsumgebung für die Untersuchung von Netwerkeffekten, wie zufällige Zeitverzögerungen und Paketverluste, auf die Performance eines ausgewählten Regelungssystems (Kapitel 4)
3. Entwurf und Implementierung von zugeschnittenen und parametrierbaren Kommunikationsprotokollen für funkbasierte Regelungssysteme (Kapitel 5)
4. Entwurf einer Cross-Design-Methodik für funkbasierte Regelungssysteme mit beschränkten Ressourcen (z. B. Rechenleistung und Energie) (Kapitel 6)
5. Umsetzung der Cross-Design-Methodik auf das invertierte Pendel-System und experimentelle Validierung der Ergebnisse (Kapitel 7)

1- Testbed für das invertierte Pendel

Um die erarbeiteten Cross-Design-Ergebnisse unter realistischen Bedingungen prüfen zu können wird ein Testbed, bestehend aus einem invertierten Pendel, ausgestattet mit ausgewählten Imote2-Funkknoten, aufgebaut und in Betrieb genommen. Der Imote2-Funkknoten verfügt über einen energiearmen 32-bit Intel PXA271 Prozessor mit skalierbarer Taktfrequenz von $13MHz$ bis $416MHz$ und den IEEE 802.15.4-fähigen Transceiver CC2420 im $2.4GHz$ ISM Band. Das modular aufgebaute Testbed ermöglicht Experimente mit unterschiedlichen Netzwerkstrukturen (z.B. Funknetzwerk zwischen Sensoren und Regler oder zwischen Regler und Aktuator oder jede beliebige Kombination je nach Bedarf) und Kommunikationsprotokollen (z.B. MAC Protokolle), sowie Regelungsstrukturen und Algorithmen (z.B. Kaskade, LQG, H_∞ und IMC). Wie im Bild 8.1 auf Seite 105 zu sehen, können Regelalgorithmen sowohl in den Imote2-Funkknoten, als auch in einem PC implementiert werden. Im PC können leistungsstarke Algorithmen wie MPC[8] umgesetzt werden, die viel Rechenzeit und Speicher benötigen. Über die vorhandenen Netzwerkblöcke können beliebige Verteilungen von Zeitverzögerungen und Paketverlusten zusätzlich simuliert werden, um deren Effekte auf die Güte der Regelung zu untersuchen.

[8] MPC: Modell prädiktive Regelung

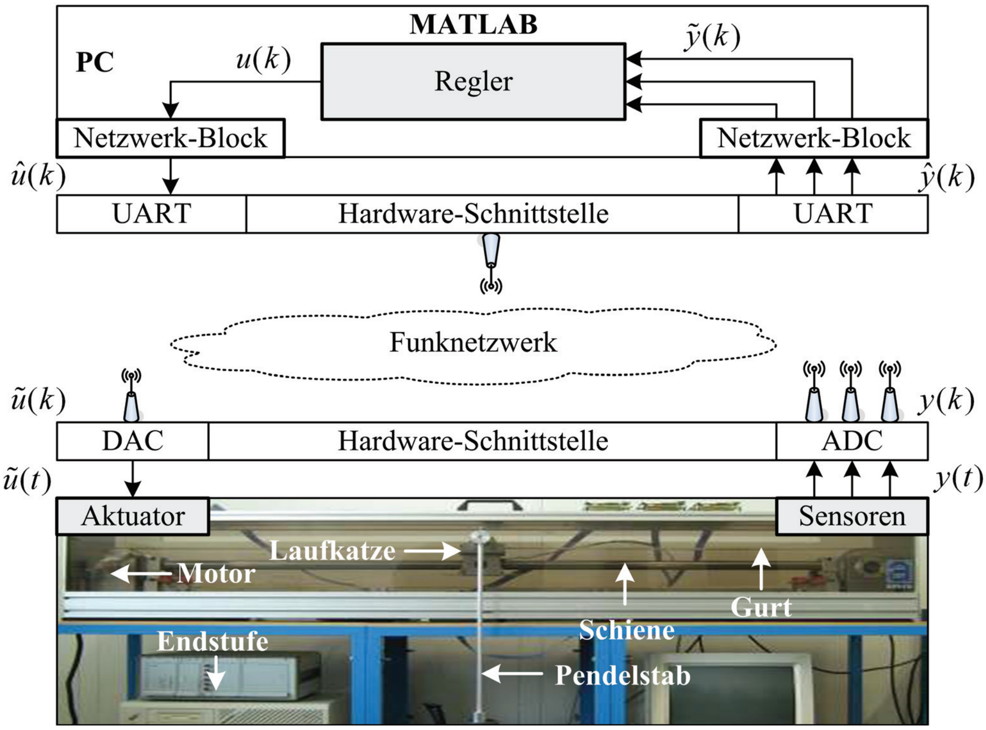


Bild 8.1: Testbed für das invertierte Pendel

2- Simulationsumgebung für funkbasierte Regelungssysteme

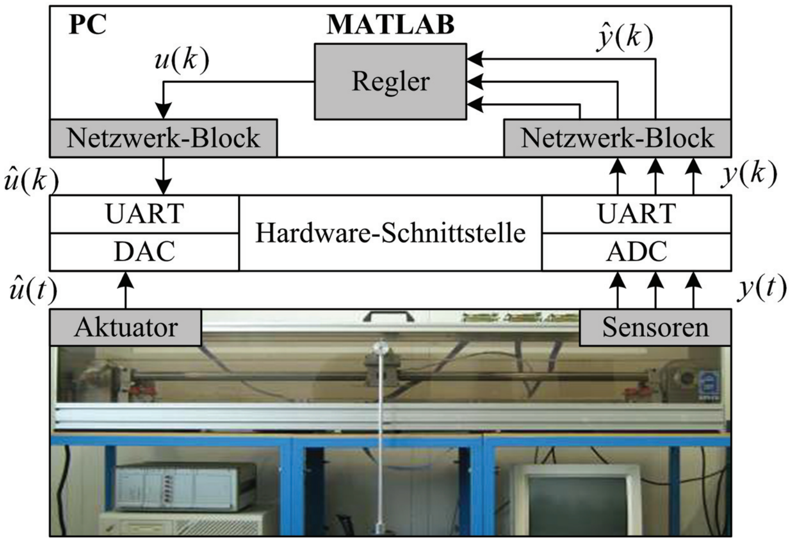


Bild 8.2: HiL-Simulationsumgebung für das invertierte Pendel

Im Hinblick auf die Zielsetzung dieser Arbeit wird eine HiL-Simulationsumgebung entwickelt, um die Auswirkungen von Netzwerkeffekten auf die Güte eines ausgewählten Regelungssystems in Echtzeit untersuchen zu können. Die entwickelte HiL-Simulationsumgebung (Siehe Bild 8.2) ermöglicht die Simulation von unterschiedlichen Regelungsstrukturen und -algorithmen mit realistischen Funknetzwerkmodellen und besteht aus dem im vorherigen Abschnitt beschriebenen invertierten Pendel, einem PC und einer Hardware-Schnittstelle. Die Hardware-Schnittstelle ermöglicht den Austausch von Informationen zwischen dem invertierten Pendel und dem in Matlab implementierten Regler. Realistische Netzwerkeffekte, wie stochastische Zeitverzögerungen, Paketverluste, Jitter und Burst können über beide Netzwerkblöcke simuliert werden. Ergebnisse der HiL-Simulationen dienen als Grundlage für den Entwurf von zugeschnittenen Kommunikationsprotokollen für das invertierte Pendel. Kenngrößen, wie die maximal zulässige Abtastperiode (MATI), die maximal zulässige Anzahl an aufeinander folgenden Paketverlusten (MAPL) und die maximale Verzögerung (MAD) können mit der HiL-Umgebung für beliebige Regelalgorithmen untersucht werden. Diese Grenzen müssen von dem zu entwerfenden Kommunikationsprotokoll eingehalten werden.

3- Zugeschnittene Kommunikationsprotokolle für funkbasierte Regelungssysteme

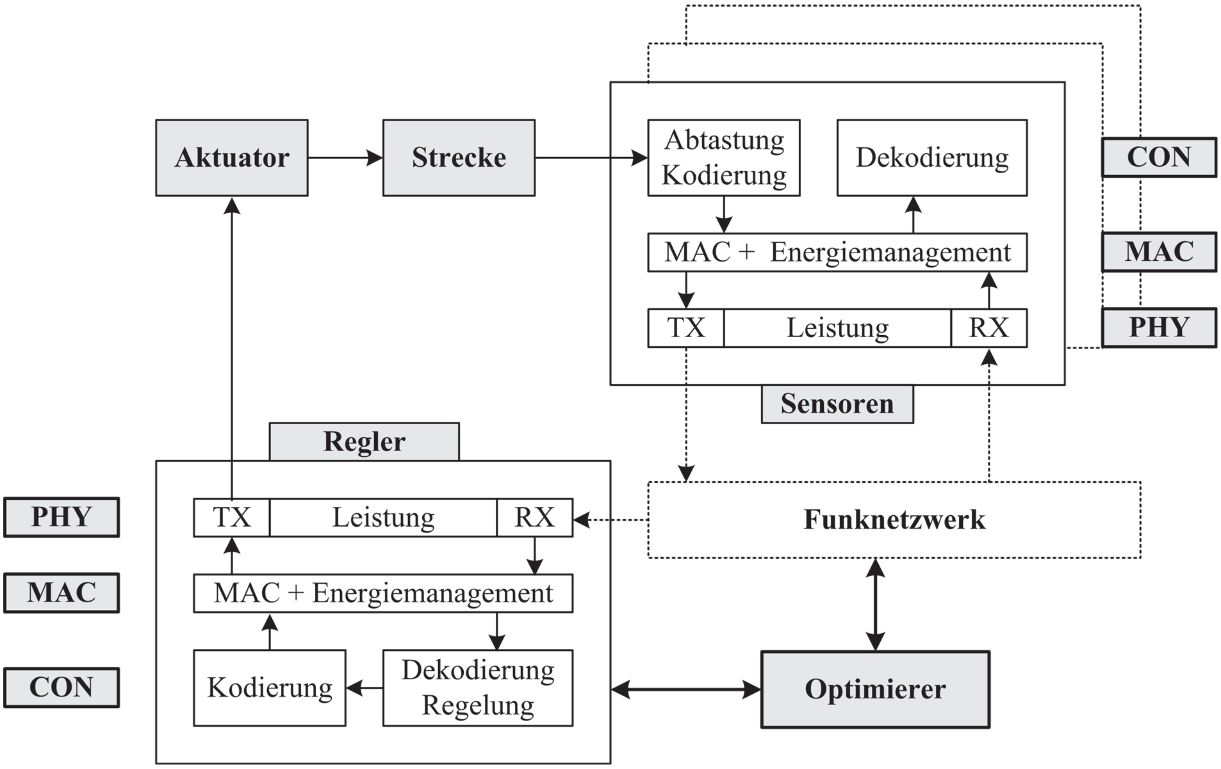


Bild 8.3: Schichten der ausgewählten Kommunikationsstruktur für funkbasierte Regelungssysteme

Adaptive Kommunikationsprotokolle, zugeschnitten auf das invertierte Pendel, werden entwickelt und mathematisch modelliert. Unter anderem wird ein mathematisches Modell für die Berechnung der Lebensdauer der Funkknoten erstellt, sowie ein mathematisches Modell für die Berechnung der optimalen Sendeleistung, basierend auf RSSI[9] Messungen. Eine Besonderheit dieser Kommunikationsstruktur ist der Einsatz eines Optimierers, der alle mathematischen Modelle beinhaltet und zur Laufzeit ein Optimierungsproblem zur Bestimmung der optimalen Reglerparameter und Netwerkparameter löst, wobei das Optimierungsziel die Lebensdauer der Funkknoten ist und als Nebenbedingung die Performance der Regelung. Ausgangspunkt für die Implementierung dieser Kommunikationsprotokolle sind HiL-Simulationsergebnisse, aus denen Kennwerte gewonnen werden (z.B. maximal zulässige Verzögerungen und Paketverluste). Dabei werden gezielte Modifikationen an dem vorhandenen IEEE 802.15.4-2006 Funkkommunikationsprotokoll vorgenommen. Unter anderem werden wettbewerbsbasierte MAC-Protokolle (z.B. TDMA) und wettbewerbsfreie MAC-Protokolle (z.B. CSMA/CA) implementiert mit definierten Freiheitsgraden (z.B. Sendeleistung, Telegrammlänge und Dauer eines Zeitfensters), die in der Cross-Design-Methodik für die Optimierung der Regelgüte eines ausgewählten Regelungssystems verwendet werden. Bild 8.3 zeigt die ausgewählte Kommunikationsstruktur. Quer über alle Schichten wird ein Energiemanagementkonzept implementiert, um die Lebensdauer der Funkknoten zu maximieren indem inaktive Funkknoten zyklisch abgeschaltet werden und die Sendeleistung angepasst wird, um Energie zu sparen. Alle entwickelten Kommunikationsprotokolle werden gemeinsam mit den Regelungsalgorithmen auf die gemeinsame Imote2-Plattform implementiert. Experimentelle Ergebnisse zeigen, dass diese Kommunikationsprotokolle die An-

[9] RSSI: Receiver Signal Strenght Indicator

forderungen des invertierten Pendels erfüllen.

4- Cross-Design-Methodik für funkbasierte Regelungssysteme

Die entwickelte Cross-Design-Methodik in Bild 8.4 beinhaltet eine Entwurfsphase und eine Optimierungsphase.

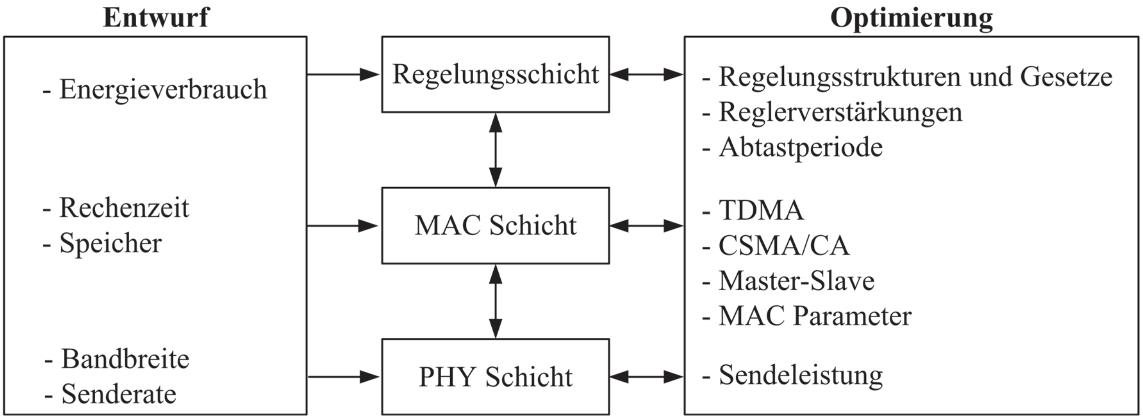


Bild 8.4: Struktur der Cross-Design-Methodik

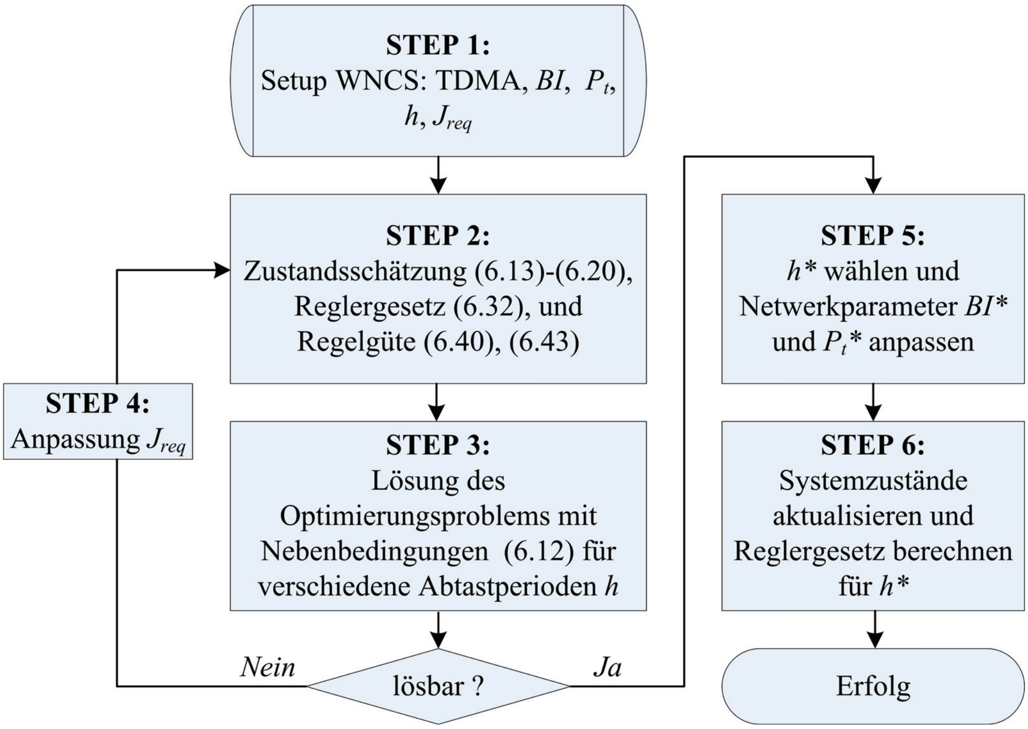

STEP 1:
Setup WNCS: TDMA, BI, P_t, h, J_{req}
STEP 2:
Zustandsschätzung (6.13)-(6.20), Reglergesetz (6.32), und Regelgüte (6.40), (6.43)
STEP 3:
Lösung des Optimierungsproblems mit Nebenbedingungen (6.12) für verschiedene Abtastperioden h
STEP 4:
Anpassung J_{req}
lösbar ?
Nein
Ja
STEP 5:
h* wählen und Netwerkparameter BI* und P_t* anpassen
STEP 6:
Systemzustände aktualisieren und Reglergesetz berechnen für h*
Erfolg

Bild 8.5: Struktur der Cross-Design-Entwurfsphase für das invertierte Pendel-System

Während der Entwurfsphase in Bild 8.5 werden Regelungsalgorithmen und Kommunikationsprotokolle für ein gegebenes Regelungssystem entworfen und aufeinander abgestimmt. Dabei wird ein besonderes Augenmerk auf die beschränkte Energie- und Rechenleistung geworfen. Dies ermöglicht die spätere gemeinsame Implementierung der entwickelten Regelungsalgorithmen und Kommunikationsprotokolle auf ressourcenbeschränkten Funkknoten. Die Entwurfsphase bei der Regelung beinhaltet die mathematische Modellbildung und Identifikation des Regelungssystems, sowie die Implementierung von geeigneten Regelungsalgorithmen. Die Auswahl der geeigneten Regelungsalgorithmen basiert auf der Regelgüte, der benötigten Rechenzeit und dem Energieverbrauch. Die Performance der ausgewählten Regelungsalgorithmen (z.B. Systemantwort und Robustheit) wird mit der Simulationsumgebung untersucht und Anforderungen an das zu entwerfende Kommunikationssystem abgeleitet (z.B. maximal zulässige Abtastperiode, Zeitverzögerungen und Paketverluste). Die Freiheitsgrade bei den Regelalgorithmen sind unter anderem die Abtastperiode und die Reglerverstärkungen. Bei der Kommunikation beinhaltet die Entwurfsphase die Implementierung von Kommunikationsprotokollen, die die Anforderungen der Regelalgorithmen erfüllen. Die Freiheitsgrade bei den Kommunikationsprotokollen sind die MAC-Protokolle mit ihren Parametern und die Sendeleistung.

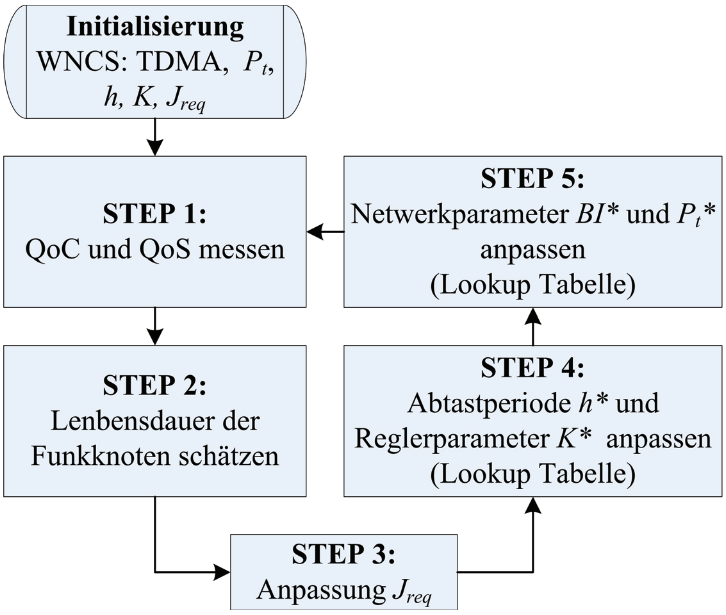

Initialisierung
WNCS: TDMA, P_t, h, K, J_{req}
STEP 1:
QoC und QoS messen
STEP 2:
Lenbensdauer der Funkknoten schätzen
STEP 3:
Anpassung J_{req}
STEP 4:
Abtastperiode h^* und Reglerparameter K^* anpassen (Lookup Tabelle)
STEP 5:
Netwerkparameter BI^* und P_t^* anpassen (Lookup Tabelle)

Bild 8.6: Struktur der Cross-Design-Optimierungsphase für das invertierte Pendel-System

Während der Optimierungsphase, dargestellt in Bild 8.6, wird die Regelgüte der Regelalgorithmen optimiert und die Lebensdauer der Funkknoten maximiert. Dabei werden die Freiheitsgrade der Regelalgorithmen und der Kommunikationsprotokolle zur Laufzeit verwendet. Experimentelle Ergebnisse am invertierten Pendel-Testbed zeigen die Performance der entwickelten Cross-Design-Methodik für funkbasierte Regelungssysteme.

A Appendix

A.1 Inverted pendulum system

A.1.1 Mathematical modeling

The inverted pendulum system in Figure A.1 is an open-loop unstable plant whose mathematical model is nonlinear. It consists of a vertical rod that can rotate around a fixed pivot point on a cart. The corresponding rod angle from the vertical upward position φ is measured using a rotational encoder. The cart is free to move along a rail and the corresponding displacement x is measured using a rotational encoder. The cart velocity $\dot{x}$ is measured using a tachometer. The cart is driven by a DC electric motor controlled by a power amplifier with an input voltage u. The DC motor is coupled to the cart through a transmission belt. There is no sensor available to measure the angular velocity $\dot{\varphi}$.

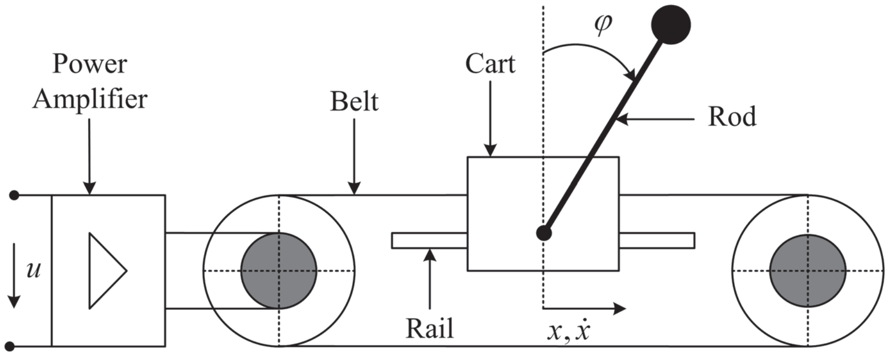


Bild A.1: Inverted pendulum system

The free-body diagram of the inverted pendulum's rod and cart as well as the motions of the center of gravity of the pendulum rod are illustrated in Figure A.2 on Page 110. Applying Newton's laws of motion [Franklin *et al.* , 1997] yields

$$M_c\ddot{x} + f_c\dot{x} - u + N = 0 \tag{A.1}$$

$$M_r\ddot{x} - M_r l_r\ddot{\varphi}\cos\varphi + M_r l_r\dot{\varphi}^2\sin\varphi - N = 0 \tag{A.2}$$

$$J_r\ddot{\varphi} + f_r\dot{\varphi} + N l_r\cos\varphi - P l_r\sin\varphi = 0 \tag{A.3}$$

$$P = M_r g \tag{A.4}$$

where N and P are unknown reaction forces. Equations (A.1)-(A.4) can be manipulated to eliminate the reaction forces N and P resulting in two nonlinear equations describing the motion of the pendulum rod and cart respectively:

$$(M_r + M_c)\ddot{x} + M_r l_r(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2\sin\varphi) + f_c\dot{x} = C_v u \tag{A.5}$$

$$(J_r + M_r {l_r}^2)\ddot{\varphi} + f_r\dot{\varphi} + M_r l_r(\ddot{x}\cos\varphi - g\sin\varphi) = 0. \tag{A.6}$$

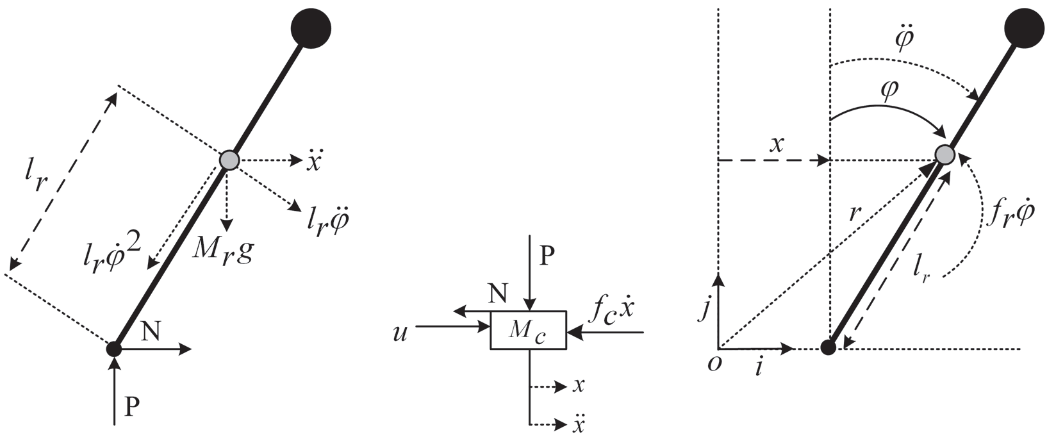


Bild A.2: Free-body diagram of pendulum rod (left), cart (middle), rotational and translational motions of the rod's center of gravity (right)

For an accurate calculation of the motion of the inverted pendulum, the nonlinear equations (A.5)-(A.6) need to be solved, which is usually difficult. Instead, linearized versions of these equations for small motions of the inverted pendulum around an equilibrium point (e.g. $\varphi = 0$) are used and will be derived in the sequel. The parameters of the inverted pendulum are summarized in Table A.1.

Parameter	Value	unit	Description
M_c	3.2	kg	cart mass
M_r	0.329	kg	rod mass
l_r	0.488	m	rod length
f_c	6.2	kg/s	cart friction constant
f_r	0.009	kgm^2/s	rod friction constant
C_v	2.6	N/V	force conversion constant
J_r	0.0743	kgm^2	rod moment of inertia
g	9.81	m/s^2	gravity acceleration
n_{11}	14.9	V/m	scaling factor: position [m]
n_{22}	−52.27	V/rad	scaling factor: angle [rad]
n_{33}	−7.64	Vs/m	scaling factor: velocity [m/s]
n_{44}	−52.27	Vs/rad	scaling factor: angular velocity [rad/s]

Table A.1: Inverted pendulum system parameters

A.1.2 Linearization

Choosing the state vector $x = [x, \varphi, \dot{x}, \dot{\varphi}]^T$ and linearizing (A.5) and (A.6) around the equilibrium point $x = [a, 0, 0, 0]^T, a \in \mathbb{R}$, leads to the continuous-time linear state space model

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{A.7}$$

$$y(t) = Cx(t) \tag{A.8}$$

where

$$A = \begin{pmatrix} 0 & 0 & -1.950 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.156 & -1.948 & 0.009 \\ 0 & 23.509 & 28.805 & -0.134 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ -6.243 \\ 92.288 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{A.9}$$

The corresponding discrete-time state space model for periodic sampling with constant sampling period h is given by

$$x_{k+1} = A_d x_k + B_d u_k \tag{A.10}$$

$$y_k = C_d x_k \tag{A.11}$$

$$A_d = e^{Ah} \approx I + Ah + \frac{1}{2!}A^2h^2 + ... \tag{A.12}$$

$$B_d = \int_0^h e^{As} ds B \approx \left(Ih + \frac{Ah^2}{2!} + \frac{A^2h^3}{3!} + ... \right) B \tag{A.13}$$

$$C_d = C \tag{A.14}$$

where I is a unit matrix of appropriate dimension [Aström & Wittenmark, 1990].

Figure A.3 shows the block diagram of the inverted pendulum. The corresponding transfer functions (A.15)-(A.17) can be obtained from (A.9) using MATLAB and are given below:

$$G_{ux}(s) = \frac{12.1745(s+4.665)(s-4.544)}{s(s-4.714)(s+5.067)(s+1.729)} \tag{A.15}$$

$$G_{u\varphi}(s) = \frac{92.2875s}{(s-4.714)(s+5.067)(s+1.729)} \tag{A.16}$$

$$G_{u\dot{x}}(s) = \frac{-6.2425(s+4.665)(s-4.544)}{(s-4.714)(s+5.067)(s+1.729)}. \tag{A.17}$$

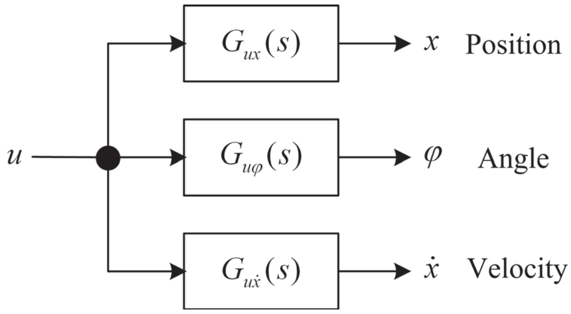


Bild A.3: Inverted pendulum block diagram

A.2 Control structures

Following two control problems are considered for the inverted pendulum system:

1. *Stabilization of the inverted pendulum*: Keep the pendulum rod in the vertical upward position (i.e. $\varphi = 0$)

2. *Trajectory tracking*: Follow a given trajectory while keeping the pendulum rod in the vertical upward position

Given the initial state $x_0 = [0\ m,\ 4°,\ 0\ m/s,\ 0\ rad/s]^T$, all control structures presented in the sequel are designed to meet following requirements:

- Fast transient response with little overshot and zero steady state error

- Control value saturation: $|u| \leq 10$ V
- Angle value limitation due to linearization assumptions: $|\varphi| \leq 10°$
- Mechanical limitations of the cart's position: $|x| \leq 0.67$ m

A.2.1 Cascaded controller

The structure of the cascaded controller is shown in Figure A.4. The optional disturbance compensator is used to compensate the Coulomb friction which can be modeled as a disturbance input voltage d_u to the inverted pendulum. The structure and parameters of the disturbance compensator are given in section A.2.3.4. Controller transfer functions and corresponding parameters are summarized in Table A.2. Design details can be found in [Jiang, 2008].

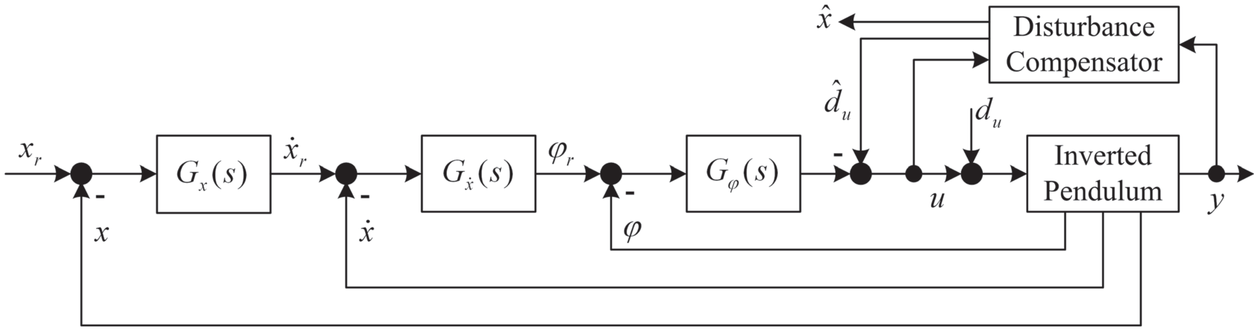


Bild A.4: Cascaded controller structure with optional disturbance compensator

Controller	Transfer function	Parameters
Position	$G_x(s) = k_x$	$k_x = -0.50$
Velocity	$G_{\dot{x}}(s) = k_{\dot{x}} \frac{s+z_{\dot{x}}}{s+p_{\dot{x}}}$	$k_{\dot{x}} = -0.90,\ z_{\dot{x}} = -6.69,\ p_{\dot{x}} = 6.12$
Angle	$G_\varphi(s) = k_p + \frac{k_i}{s} + k_d s$	$k_p = 5.80,\ k_i = 6.90,\ k_d = 0.059$
Position	$G_x(z) = k_x$	$k_x = -0.50$
Velocity	$G_{\dot{x}}(z) = k_{\dot{x}} \frac{1+(p_{\dot{x}}h-1)z^{-1}}{1+(z_{\dot{x}}h-1)z^{-1}}$	$k_{\dot{x}} = -0.90,\ z_{\dot{x}} = -6.69,\ p_{\dot{x}} = 6.12$
Angle	$G_\varphi(z) = k_p + k_i h \frac{z^{-1}}{1-z^{-1}} + \frac{k_d}{h} \frac{1-z^{-1}}{z^{-1}}$	$k_p = 5.80,\ k_i = 6.90,\ k_d = 0.059$

Table A.2: Cascaded controller transfer functions and parameters

A.2.2 IMC controller

Figure A.5 on Page 113 shows the structure of the Internal Model Controller (IMC) for the inverted pendulum. Unfortunately, standard IMC design cannot be applied to the inverted pendulum system since it is nonlinear and unstable. Moreover, the transfer function of the inverted pendulum is not invertible. Therefore, a special IMC structure is realized to control the position of the inverted pendulum's cart. Plant and model use the same transfer functions as in (A.15)-(A.17) but with added disturbances (e.g. friction terms) to the plant. Both plant and model are stabilized using the cascaded controller structure in section A.2.1. The transfer function of the IMC filter used to compensate model discrepancies (i.e. $\hat{e} = y - y_M$) and the corresponding parameters are summarized in Table A.3 on Page 113. Since the structure of the stabilizing controllers for model and plant are the same in Table A.2, only controller parameter values are provided. Further details can be found in [Nempe, 2008].

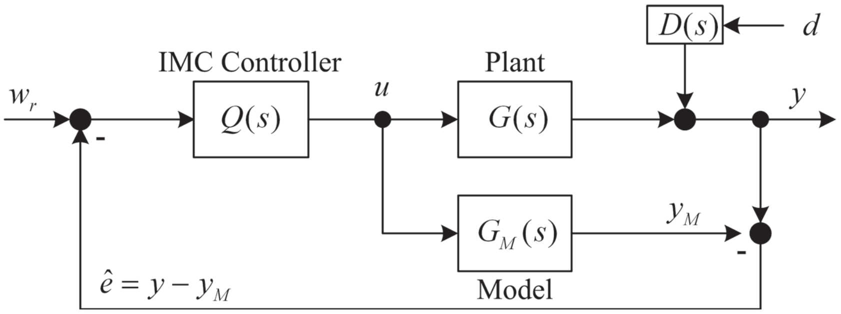


Bild A.5: IMC controller structure

Controller	Model	Plant
Position	$k_{x,m} = -0.35$	$k_{x,p} = -0.4$
Velocity	$k_{\dot{x},m} = -0.46$, $z_{\dot{x},m} = -12.3$, $k_{d,m} = 0.1$	$k_{\dot{x},p} = -0.46$, $z_{\dot{x},p} = -12.3$, $k_{d,p} = 0.1$
Angle	$k_{p,m} = 8.8$, $k_{i,m} = 18.4$, $p_{d,m} = 4.98$	$z_{\dot{x},m} = -12.3$, $z_{\dot{x}} = -12.3$, $p_{\dot{x},m} = 4.98$
Controller	Transfer function	Parameters
Filter	$Q(s) = k_f \frac{1}{s+p_f}$	$k_f = 5.0$, $p_f = 0.3475$
Filter	$Q(z) = k_f h \frac{z^{-1}}{1+(p_f h-1)z^{-1}}$	$k_f = 5.0$, $p_f = 0.3475$

Table A.3: IMC filter parameters

A.2.3 State feedback controller

The general structure of the state feedback controller for the inverted pendulum is depicted in Figure A.6 and includes following components:

1. *Controller*: Compensates initial state disturbances and stabilizes the inverted pendulum
2. *Filter*: Allows the inverted pendulum to follow step reference inputs without steady state errors
3. *State and disturbance observer*: Reconstructs the angular velocity $\dot{\varphi}$ and the Coulomb's friction d_u (modeled as a disturbance input) from measured outputs y

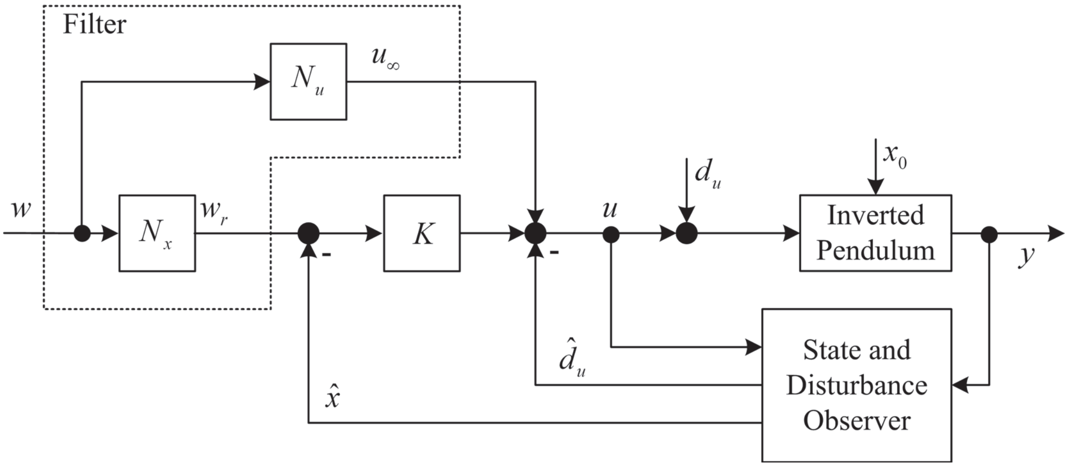


Bild A.6: State feedback controller structure

State feedback controllers are designed using the *emulation* approach including the following steps:

1. Design of continuous-time controllers
2. Digitization of the continuous-time controllers using discrete-time methods
3. Design verification using analysis, simulation, or experiments

In the sequel, some details about the design of state feedback controllers for the inverted pendulum for the continuous-time case will be provided. Discrete-time equivalent controllers are easily obtained using MATLAB and will not be addressed here. For details please refer to [Fankem, 2009].

A.2.3.1 Pole placement controller design

Choosing $u = -Kx$, equation (A.7) can be rewritten as follows:

$$\dot{x} = (A - BK)x \tag{A.18}$$

where $(A - BK)$ is the matrix of the closed-loop system whose eigenvalues are to be appropriately selected. Following steps are carried out for the design of K:

1. Choice of desired closed-loop eigenvalues
2. Calculation of the gain matrix K using MATLAB
3. Simulation in MATLAB/Simulink

If the performance requirements are not met, then steps 1 to 3 are repeated choosing different closed-loop eigenvalues. The resulting controller gain matrix K and the corresponding eigenvalue vector p for the inverted pendulum are given below:

$$K = [-0.8718,\ 1.9487,\ 2.2489,\ 0.3690] \tag{A.19}$$

$$p = [-1.0500 - 1.0712i;\ -1.0500 + 1.0712i;\ -10;\ -10]\,. \tag{A.20}$$

A.2.3.2 LQR controller design

The objective of the LQR controller design is to find a controller K ($u = -Kx$) that minimizes a cost function of the form

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt,\ Q \geq 0,\ R > 0. \tag{A.21}$$

Following design steps are carried out:

1. Choice of suitable weighting matrices Q and R
2. Calculation of the optimal controller matrix gain K using MATLAB
3. Simulation in MATLAB/Simulink

If the performance requirements are not met, then steps 1 to 3 are repeated choosing different weighting matrices Q and R. The resulting controller gain matrix K and the corresponding eigenvalue vector p for the inverted pendulum system are given below:

$$K = [-0.7071,\ 2.1592,\ 2.2514,\ 0.3416] \tag{A.22}$$

$$p = [-0.8663 - 0.7855i;\ -0.8663 + 0.7855i;\ -8.9092 - 7.3530i;\ -8.9092 + 7.3530i]\,. \tag{A.23}$$

A.2.3.3 Filter design

The selected two components filter structure for the state feedback-controllers is depicted in Figure A.6 on Page 113. The matrix N_x transforms the input vector w into the desired reference state w_r which should be attained in the stationary state and the matrix N_∞ transforms the input vector w into a control value u_∞ necessary in order attain desired stationary accuracy. Following filter parameters were chosen for the inverted pendulum:

$$N_x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \; N_u = 0. \tag{A.24}$$

A.2.3.4 Observer design

A reduced-order state observer is used to reconstruct the angular velocity $\dot{\varphi}$ of the inverted pendulum. Since only the angular velocity $\dot{\varphi}$ of the inverted pendulum is not measurable, the state vector x can be written as $x = [x_1, x_2]^T$, where $x_1 = [x,\ \varphi,\ \dot{x}] = y^T$ and $x_2 = \dot{\varphi}$. Equations (A.7)-(A.8) can be rewritten as

$$\begin{pmatrix} \dot{y} \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} y \\ x_2 \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u,\; y = (I\ 0) \begin{pmatrix} y \\ x_2 \end{pmatrix} \tag{A.25}$$

The reduced-order observer equations are given below [Aström & Wittenmark, 1990]:

$$\dot{\hat{x}}_2 = (A_{22} - L_{red}A_{12})\hat{x}_2 + A_{21}y + B_2u + L_{red}(\dot{y} - A_{11}y - B_1u) \tag{A.26}$$

$$\dot{\tilde{x}}_2 = (A_{22} - L_{red}A_{12})\, x_2 \tag{A.27}$$

where $\tilde{x}_2 = x_2 - \hat{x}_2$.

As for the angular velocity, disturbances can also be reconstructed from available state and control input values. Therefore, assuming a constant or stepwise disturbance input d_u, following state space representation is obtained:

$$\underbrace{\begin{pmatrix} \dot{x} \\ \dot{d}_u \end{pmatrix}}_{\dot{z}} = \underbrace{\begin{pmatrix} A & B \\ 0 & 0 \end{pmatrix}}_{A_z} \underbrace{\begin{pmatrix} x \\ d_u \end{pmatrix}}_{z} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{B_z} u,\; y = \underbrace{(C\ 0)}_{C_z} \underbrace{\begin{pmatrix} x \\ d_u \end{pmatrix}}_{z} \tag{A.28}$$

where the augmented state vector z includes the measurable output vector y, the angular velocity $\dot{\varphi}$, and the input disturbance d_u. The corresponding system matrices are given below:

$$A_z = \begin{pmatrix} 0 & 0 & -1.950 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -0.156 & -1.948 & 0.009 & -6.243 \\ 0 & 23.509 & 28.805 & -0.134 & 92.288 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A.29}$$

$$B_z = \begin{pmatrix} 0 \\ 0 \\ -6.243 \\ 92.288 \\ 0 \end{pmatrix}, \; C_z = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{A.30}$$

The selected vector of eigenvalues p_{red} and the resulting reduced-order observer gain matrix L_{red} are

$$p_{red} = [-100, \ -200] \tag{A.31}$$

$$\tag{A.32}$$

$$L_{red} = \begin{pmatrix} 0.0000 & 99.8789 & -14.7838 \\ 0.0000 & 0.0286 & -32.0386 \end{pmatrix}. \tag{A.33}$$

A.2.4 H-infinity controller

The general structure of the H_∞ controller for the inverted pendulum is depicted in Figure A.7. The response to set point changes is given by the following transfer function

$$G_{zw}(s) = \begin{pmatrix} W_e S_y W_r & -W_e S_y W_{d_y} & -W_e S_y G W_{d_u} \\ W_u K S_u W_r & -W_u K S_u W_{d_y} & -W_u T_u G W_{d_u} \end{pmatrix} \tag{A.34}$$

where $S_y = (I + GK)^{-1}$ and $S_u = (I + KG)^{-1}$ are the sensitivity function to input respectively output disturbances. W_r, W_{d_y}, W_{d_u}, W_e, and W_u are weighting matrices.

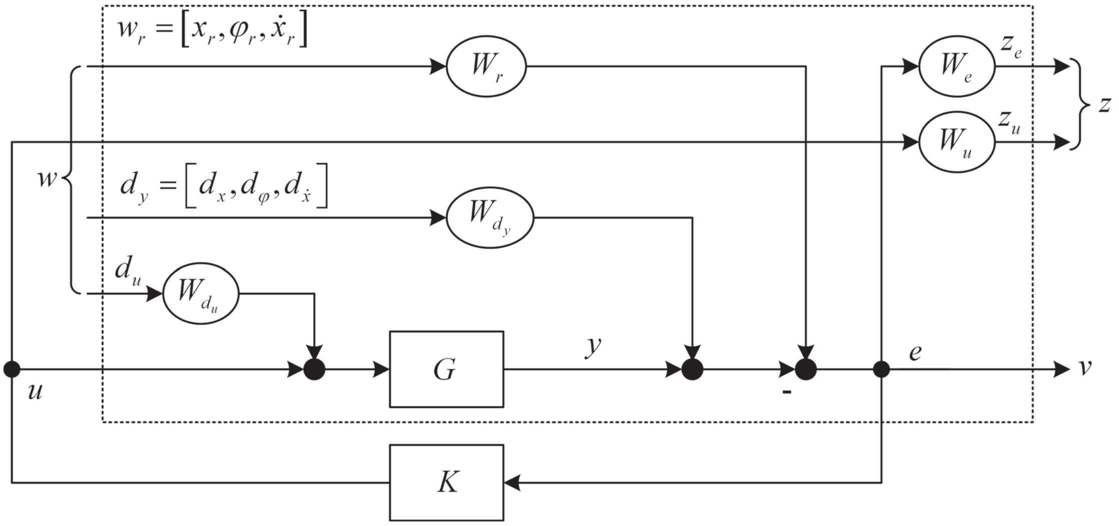


Bild A.7: H_∞ controller structure

The H_∞ controller design consists in finding a controller $K(s)$ which minimizes the H_∞-Norm of the transfer function $G_{zw}(s)$. The optimization problem can be formulated as follows:

$$\min_{K(s)} \ \|G_{zw}(s)\|_\infty \tag{A.35}$$

$$\|G_{zw}(s)\|_\infty = \sup_{w \in \mathbb{R}} \sigma_{max} G_{zw}(jw) \tag{A.36}$$

where $\|G_{zw}(s)\|_\infty$ is the H_∞-Norm of the matrix G_{zw} and σ_{max} the maximum singular value of the matrix G_{zw}.

Following weighting matrices were chosen for the inverted pendulum and corresponding parameters are summarized in Table A.4 on Page 117:

$$W_r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ W_{d_y} = \begin{pmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{pmatrix}, \ w_{du} = 2 \tag{A.37}$$

$$W_e = \begin{pmatrix} w_p & 0 & 0 \\ 0 & w_a & 0 \\ 0 & 0 & w_v \end{pmatrix}, \ w_u = 1 \tag{A.38}$$

Parameter	Value	Parameter	Value	Parameter	Value
M_p	1.8	M_a	0.5	M_v	0.02
A_s	10^{-4}	A_a	0.001 rad/s	A_v	0.001 rad/s
w_b	1.5 rad/s	$w_{1,a}$	0.1 rad/s	$w_{1,v}$	0.2 rad/s
-	-	$w_{2,a}$	15 rad/s	$w_{2,v}$	12 rad/s

Table A.4: H_∞ controller parameters

where

$$w_p = \frac{1}{M_p}\frac{s + w_b}{s + A_p w_b}, \ w_a = \frac{1}{M_a}\frac{s + A_a}{(s + w_{1,a})(s + w_{2,a})}, \ w_v = \frac{1}{M_v}\frac{s + A_v}{(s + w_{1,v})(s + w_{2,v})}. \tag{A.39}$$

The resulting H_∞ controller $K(s) = [K_p(s),\ K_a(s),\ K_v(s)]$ is given below:

$$K_p(s) = \frac{0.15808(s - 8.666e004)(s^2 + 0.8965s + 0.2454)(s^2 + 21.05s + 234.4)}{(s + 9180)(s + 28.22)(s + 0.00015)(s^2 + 1.323s + 0.4797)} \tag{A.40}$$

$$K_a(s) = \frac{-0.12485(s + 2.383)(s - 6.067e005)(s^2 + 12.99s + 43.93)}{(s + 9180)(s + 28.22)(s^2 + 1.323s + 0.4797)} \tag{A.41}$$

$$K_v(s) = \frac{0.77501(s - 2.324e004)(s - 12.14)(s + 7.865)(s + 1.77)}{(s + 9180)(s + 28.22)(s^2 + 1.323s + 0.4797)}. \tag{A.42}$$

A.2.5 Results

Simulation and experimental results for the different control structures studied so far are presented and discussed in this section. First, a cost function is derived in order to assess the control performance followed by a comparison of simulation and experimental results. Following abbreviations will be used in the sequel:

- *C1*: Cascaded controller without disturbance observer
- *C2*: Cascaded controller with disturbance observer
- *Place*: Pole placement controller with state observer and disturbance observer
- *LQR*: LQR controller with state observer and disturbance observer
- *Ref*: Reference input
- *Sim*: Simulation

A.2.5.1 Performance index for the QoC

The equation below is used to compute the performance index for a given control structure:

$$QoC = (1 - \frac{J}{J_{max}}) \cdot 100 \tag{A.43}$$

where J is the measured control cost and J_{max} the maximum control cost. The higher the QoC performance index, the better the control performance. According to (A.43), the minimum achievable control performance index is $QoC_{min} = 0$ and the maximum achievable control performance index is $QoC_{max} = 100$. The control cost J is computed as follows:

$$J = \alpha_1 IAE + \alpha_2 ISE + \alpha_3 ITAE + \alpha_4 ITSE \tag{A.44}$$

where α_1, α_2, α_3, and α_4 are weighting factors. IAE (Integral of the Absolute Error) and ISE (Integral of the Squared Error) emphasize the contribution of initial errors whereas ITAE (Integral of Time-weighted Absolute Error) and ITSE (Integral of Time-weighted Square Error) tend to reduce the contribution of large initial errors and emphasize late errors. IAE, ISE, ITAE, and ITSE are computed using following equations:

$$IAE = \sum_{k=1}^{k_f} \left[|e_{x,k}| + |e_{\varphi,k}| + |e_{\dot{x},k}| + |u_k| \right] \tag{A.45}$$

$$ITAE = \sum_{k=1}^{k_f} \left[|e_{x,k}| + |e_{\varphi,k}| + |e_{\dot{x},k}| + |u_k| \right] k \tag{A.46}$$

$$ISE = \sum_{k=1}^{k_f} \left[e_{x,k}^2 + e_{\varphi,k}^2 + e_{\dot{x},k}^2 + u_k^2 \right] \tag{A.47}$$

$$ITSE = \sum_{k=1}^{k_f} \left[e_{x,k}^2 + e_{\varphi,k}^2 + e_{\dot{x},k}^2 + u_k^2 \right] k \tag{A.48}$$

where e_x is the position error, e_φ the angle error, $e_{\dot{x}}$ the velocity error, and u the control value. For simplicity and without loss of generality, $\alpha_2 = 0.5$, $\alpha_4 = 0.5$, $\alpha_1 = 0$, and $\alpha_3 = 0$ in the sequel.

A.2.5.2 Simulation results

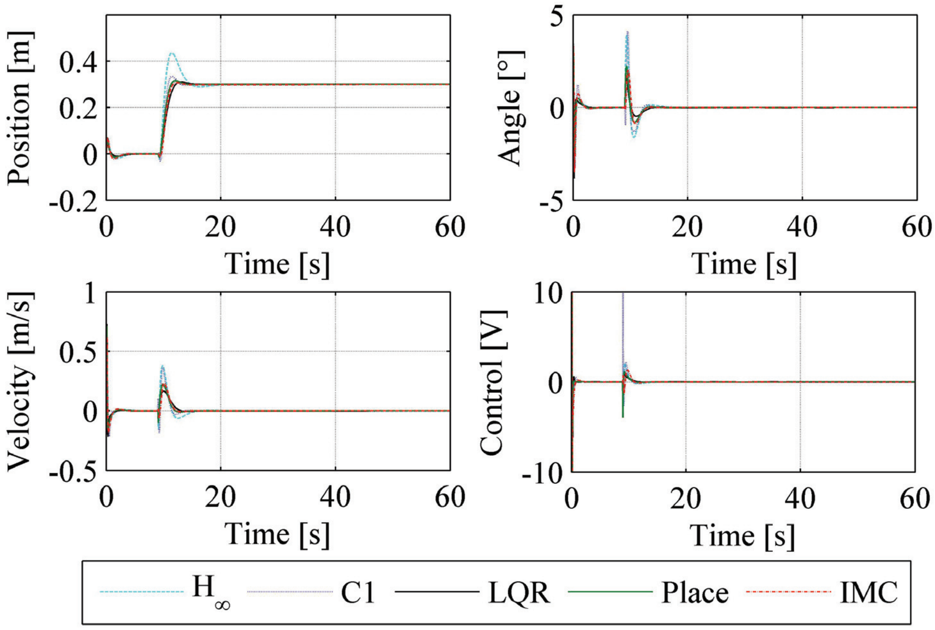


Bild A.8: Simulation results for all control structures for the continuous-time case

Simulation results for all control structures implemented so far are shown in Figure A.8 and the corresponding performance metrics (overshoot, steady state error, error band, rise time, and settling time) are summarized in Table A.5 on Page 119. The H_∞ controller and the cascaded controller have the fastest rise time but at the expense of increased overshot and high control voltages (saturation). The worst control performance is achieved by the H_∞ controller. The final position of the inverted pendulum is reached approximately 5 s after the step input position is applied with all controllers except for the H_∞ controller. The smallest angle deviations

Controller	Overshot (PO)	Steady State Error (e_{ss})	Rise Time (t_r)	Settling Time (t_s)
C1	11.5 %	0 %	0.86 s	3.6 s
Place	5 %	0 %	1.32 s	4.1 s
LQR	3.3 %	0 %	**1.83 s**	5.1 s
IMC	2.6 %	0 %	1.3 s	3.7 s
H_∞	**45 %**	0 %	1.13 s	**9.2 s**

Table A.5: Comparison of controller performance metrics from simulation

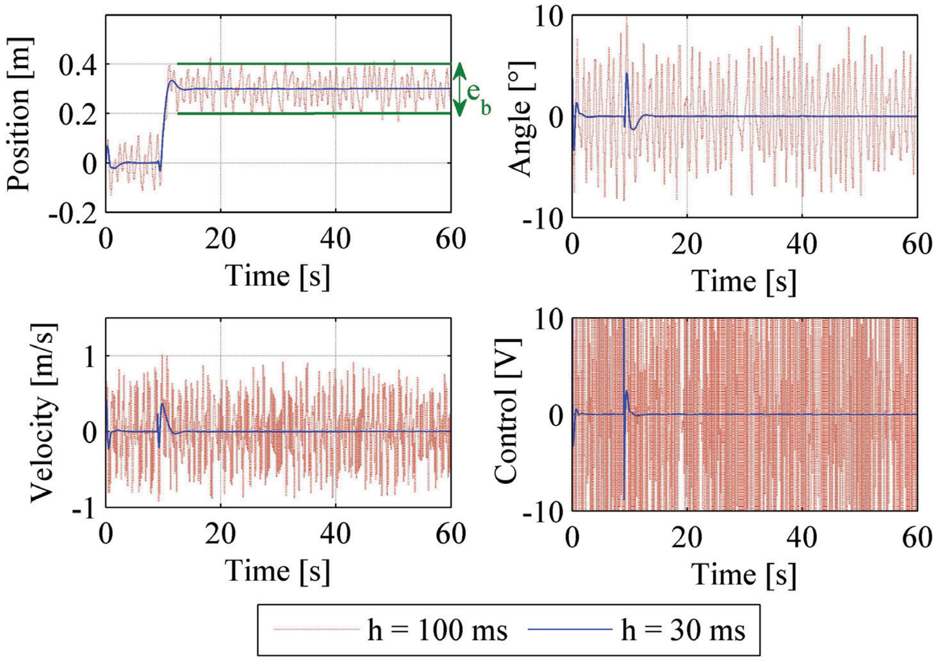


Bild A.9: Simulation results for the cascaded controller for $h = 30\ ms$ and $h = 100\ ms$

and control effort are obtained with the state feedback controllers and the best overall control performance is achieved with the IMC controller.

Similar results are obtained for the discrete-time case for selected sampling periods as shown in Figure A.9. As the sampling period increases, the control performance deteriorates with increased oscillations and overshoots. The range of applicable sampling periods varies with the selected control structure. The sampling period is lower bounded by the round-trip time (i.e. sum of the control algorithm computation time, sensor data acquisition time, and actuation time) and upper bounded by the inverted pendulum dynamics. Measured maximum sampling periods and round-trip times (minimum sampling achievable sampling periods) for each control structure are summarized in Table A.6.

Sampling period	Place	LQR	C1	C2
$h_{max}[ms]$	90	80	100	100
$RTT[ms]$	3.50	3.46	2.93	3.60
$\sigma_{RTT}[ms]$	0.06	0.04	0.03	0.05

Table A.6: Minimum and maximum applicable sampling periods and round trip times for all control structures

In the next section, experimental results for the cascaded controller (with and without disturbance compensation), pole placement controller, and LQR controller will be presented and discussed. The H_∞ controller and the IMC will not be considered for the cross-design due to their computational complexity and the huge number of controller parameters. Moreover, the limited range of applicable sampling periods for both control structures will considerably affect the choice and implementation of communication protocols.

A.2.5.3 Experimental results

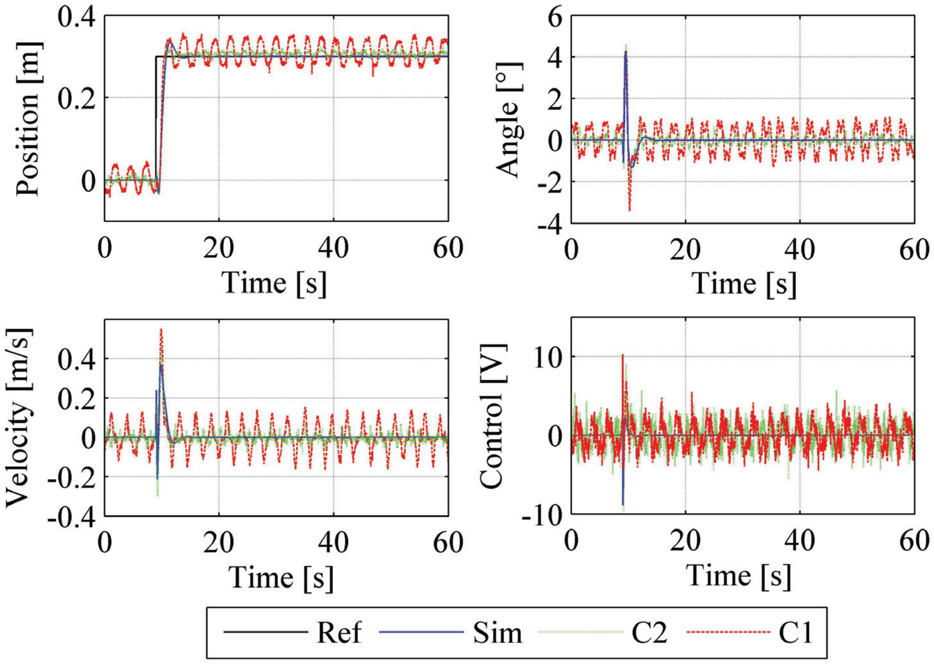


Bild A.10: Simulation and experimental results for the cascaded controller with $h = 30\ ms$

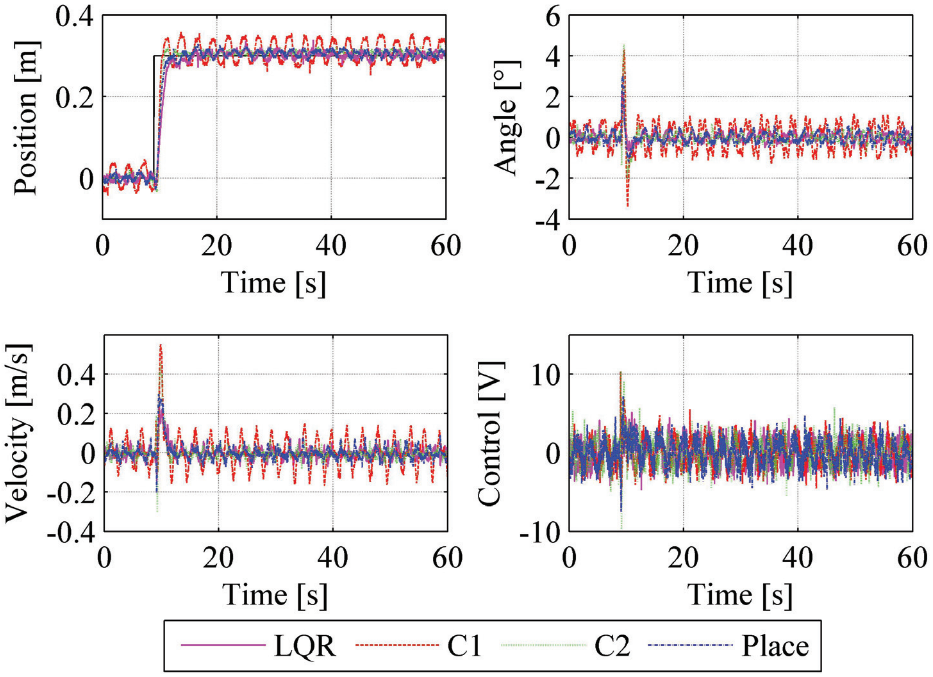


Bild A.11: Experimental results for all control structures with $h = 30ms$

Figure A.10 shows a comparison of simulation and experimental results for the cascaded controller (with and without disturbance compensation) for a selected sampling period $h = 30\ ms$. It can be seen that simulation and experimental results are almost identical which shows the accuracy of the inverted pendulum model used to design the controllers. When the disturbance observer is not used, the response of the inverted pendulum to a step input is oscillatory due to the effect of the uncompensated Coulomb friction which was neglected in the linearization process. Hence, the disturbance compensator effectively compensates the Coulomb friction. Similar results are obtained with the pole placement controller and LQR controller (see Figure A.11 on Page 120) and the corresponding performance metrics (overshoot, error band, rise time, and settling time) are summarized in Table A.7.

Controller	Overshot (PO)	Error Band (e_b)	Rise Time (t_r)
C1	**19.4 %**	**17.3 %**	0.6 s
C2	8.3 %	6.5 %	0.8 s
Place	10.2 %	12 %	1.31 s
LQR	10.1 %	10 %	**1.71 s**

Table A.7: Comparison of controller performance metrics from experiments for $h = 30ms$

Measurements at the stability bounds (e.g. $h \geq h_{max}$) are difficult to realize since the pendulum rod constantly falls down due to the nondeterministic disturbances acting on the inverted pendulum. The control performance of all control structures for selected sampling periods are summarized in Table A.8. The relative degradation of the control performance at $h = 10\ ms$ compared to $h = 20\ ms$ and $h = 30\ ms$ is due to measurement noise and the effects of nonlinearities which were neglected during the linearization process (e.g. slew rate limitation of the DC motor). The optimal sampling period is $h = 30\ ms$ for all controllers except for the cascade controller with observer whose optimal sampling period is $h = 40\ ms$. Best control performance is obtained with the state feedback controllers and the largest range of applicable sampling periods is obtained with the cascaded controllers.

$h[ms]$	QoC_{C1}	QoC_{LQR}	QoC_{C2}	QoC_{Place}
10	31.7460	24.7615	23.5531	22.5108
20	35.0087	39.8949	34.2125	35.5161
30	**35.7308**	**42.1587**	33.8159	**41.3538**
40	35.4828	41.6965	**35.3863**	40.2185
50	32.7208	40.9249	33.1843	41.1013
60	31.7587	41.3673	33.2507	36.8089
70	30.8791	38.8989	29.4703	35.9234
80	29.8048	35.7997	25.9630	24.3098
90	28.7918	-	18.1652	6.9554
100	28.0878	-	5.4718	-

Table A.8: Control performance index for all control structures and selected sampling periods $10\ ms \leq h \leq 100\ ms$

A.3 Overview of the IEEE 802.15.4-2006 protocol standard

A.3.1 Architecture and components

The layered architecture of IEEE 802.15.4 is depicted in Figure A.12 on Page 122 and is based on the open systems interconnection (OSI) seven-layer model. The physical layer (PHY) contains

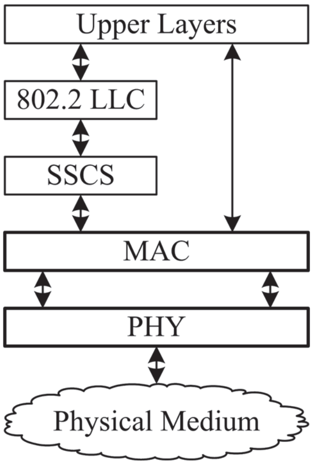


Bild A.12: LR-WPAN device architecture [IEEE, 2006]

the radio frequency (RF) transceiver along with its low-level control mechanisms. The MAC sublayer provides access to physical channels for all types of data transfer. The upper layers consist of a network layer, which provides network configuration, manipulation, and message routing, and an application layer, which provides the intended function of the device. The definition of these upper layers is not provided in the standard.

The most basic component in IEEE 802.15.4 is the device which can be a Full-Function Device (FFD) or Reduced-Function Device (RFD). At least one FFD should be included in a network and operate as a Personal Area Network (PAN) coordinator. A FFD can operate in three modes: PAN coordinator, coordinator or device. RFDs are intended for very simple applications with small amounts of data to be transmitted over the network. An FFD can talk to FFDs whereas RFDs can only talk to FFDs.

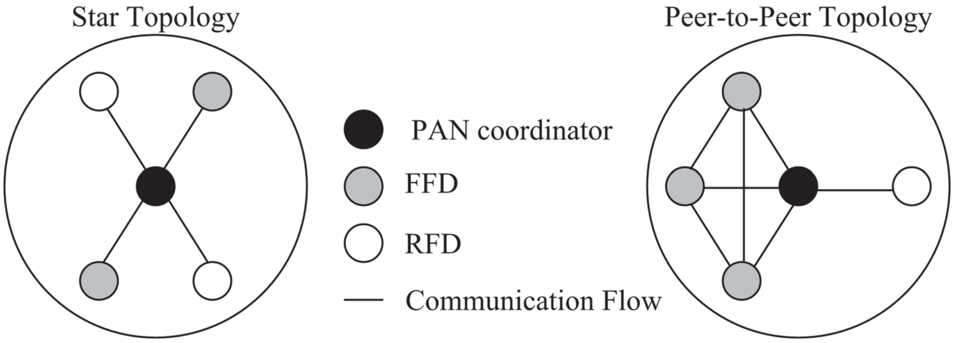


Bild A.13: IEEE 802.15.4 star and peer-to-peer network topology examples

The standard supports a combination of different components that could generate two types of network topologies shown in Figure A.13. In the basic star topology, all nodes contend to send data to the PAN coordinator. The star topology is widely used in home automation, personal computer (PC) peripherals, games and toys, and personal health care. In contrast to the star topology, any device can communicate with any other in the peer-to-peer topology as long as they are in range of one another. The peer-to-peer topology allows more complex network formations to be implemented, such as mesh networking topology and is widely used in applications such as industrial control and monitoring, and wireless sensor networks (WSN). A peer-to-peer network can be ad hoc, self-organizing, and self-healing. It may also allow multiple hops to route messages from any device to any other device on the network. Such functions can be added at the higher layer, but are not part of the standard.

Two channel access schemes are defined by the standard:

1. The beacon-enabled access scheme, which uses a slotted CSMA/CA and the optional guaranteed time slot (GTS) allocation mechanism
2. Unslotted CSMA/CA without beacons

A.3.2 Frame structure and data transfer

The communication is organized in temporal windows called superframes whose structure is depicted in Figure A.14 for the beacon-enabled mode. Beacon frames are periodically sent by the network coordinator in each Beacon Interval (BI) to identify its PAN and synchronize nodes that communicate with it. The format of the superframe is defined by the coordinator; it is divided into 16 equally sized slots and can have an active and an inactive portion. Coordinator and all devices may enter a low-power mode during the inactive portion. During the contention access period (CAP), any device wishing to communicate competes with other devices using the CSMA/CA mechanism. The PAN coordinator may dedicate portions of the active superframe to specific nodes. These portions are called guaranteed time slots (GTSs) and form the contention-free period (CFP).

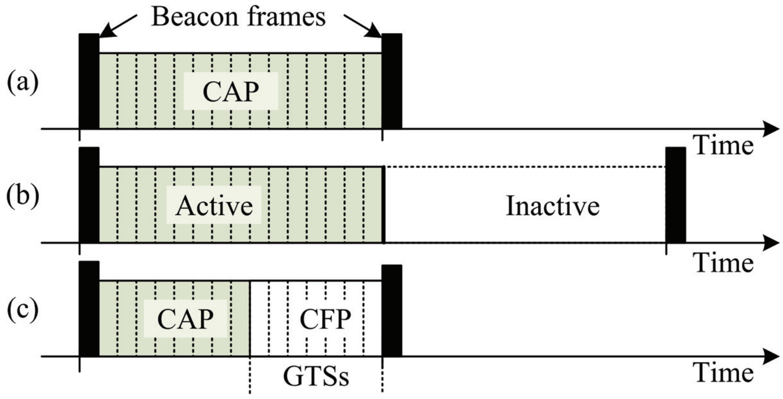


Bild A.14: IEEE 802.15.4 superframe structure for the beacon enabled mode. (a) superframe structure without inactive period, (b) superframe structure with inactive period, (c) superframe structure with CFP and GTSs without inactive period

The standard defines four frame structures:

- *Beacon frame*: Used by the coordinator to transmit beacons and set the network configuration
- *Data frame*: Used for all data transfers
- *Acknowledge packet (ACK) frame*: Used to acknowledge successful frame reception
- *MAC command frame*: Used for handling all MAC peer entity control transfers

The superframe structure example in Figure A.15 on Page 124 is described by the value of Beacon Order (BO) and Superframe Order (SO). The interval at which the coordinator shall transmit its beacon frames is given by the BI value. The relations between the SD, SO, BI, and BO are given below:

$$BI = aBaseSlotDuration \cdot aNumSuperframeSlots \cdot 2^{BO},\ 0 \leq BO \leq 14. \tag{A.49}$$

$$SD = aBaseSlotDuration \cdot aNumSuperframeSlots \cdot 2^{SO},\ 0 \leq SO \leq BO \leq 14. \tag{A.50}$$

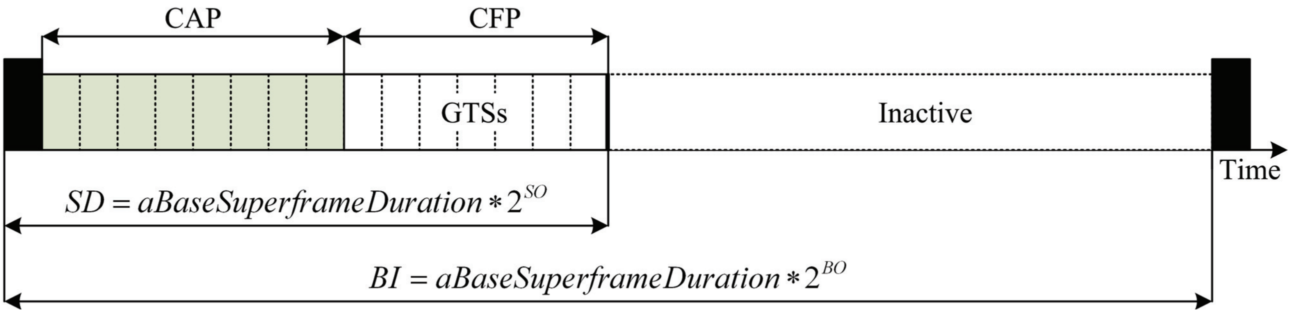

CAP
CFP
GTSs
Inactive
Time
$SD = aBaseSuperframeDuration * 2^{SO}$
$BI = aBaseSuperframeDuration * 2^{BO}$

Bild A.15: An example of the superframe structure

A superframe is composed of three parts: a beacon, a CAP and a CFP. Transmission within the CFP should not use a CSMA/CA mechanism to access the channel. However, all frames transmitted during the CAP should use a slotted CSMA/CA mechanism to access the channel except for ACK frame and any data frames that immediately follow the ACKs of a data request command.

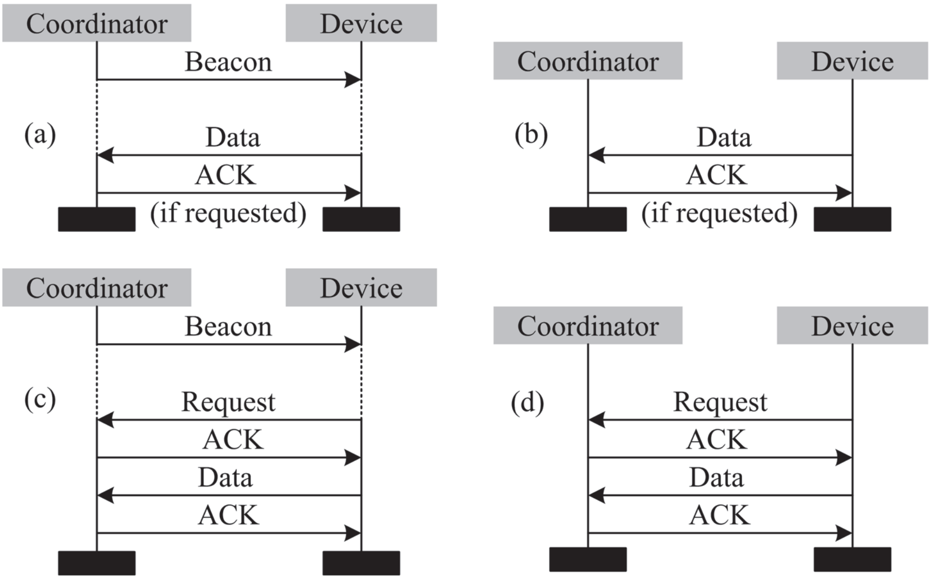


Bild A.16: Data transfer: (a) to a coordinator in beacon-enabled PAN, (b) to a coordinator in nonbeacon-enabled PAN, (c) from a coordinator in beacon-enabled PAN, (d) from a coordinator in nonbeacon-enabled PAN

Figure A.16 shows two of the three types of data transfer that are possible with IEEE 802.15.4 depending on whether beacons transmission is supported or not: data transfer from a device to a coordinator for slotted and unslotted CSMA/CA, data transfer from a coordinator to a device whose radio needs to be enabled when communication starts for slotted and unslotted CSMA/CA, and peer-to-peer data transfer in which every device may communicate with any other device in its radio range using for example unslotted CSMA/CA.

A.3.3 Physical layer specification

The standard defines four physical sublayer, but the CC2420 radio chip [Texas, 2007] uses a 2450-2485.5 MHz Direct Sequence Spread Spectrum (DSSS) RF transceiver with O-QPSK modulation with a maximum data rate of 250 kbit/s. This band is part of the Industrial and Scientific Medical (ISM) band.

The physical sublayer is responsible of the following tasks:

- Activation and deactivation of the radio transceiver
- Energy detection (ED) within the current channel
- Link Quality Indication (LQI) for received packets
- Clear Channel Assessment (CCA) for CSMA/CA
- Channel frequency selection: 868 - 868.6 MHz (e.g. Europe), 902 - 928 MHz (e.g. North America), and 2400 - 2483.5 MHz (worldwide)
- Data transmission and reception

A.3.4 MAC sublayer specification

The MAC sublayer is responsible for the following tasks:

- Generate network beacons for coordinator devices
- Synchronize to network beacons
- Support PAN association and disassociation
- Support device security
- Use the CSMA/CA mechanism for channel access
- Handle and maintain the GTS mechanism
- Provide reliable link between two peer MAC entities

In the following only a brief description of the CSMA/CA MAC algorithm as specified by the standard will be given. The steps of the CSMA/CA algorithm are illustrated in Figure A.17 on Page 126. In slotted CSMA/CA, the backoff period of every device in the PAN should be aligned with the superframe slot boundaries of the PAN coordinator while in unslotted CSMA/CA, the backoff periods of one device are not related in time to the backoff periods of any other device in the PAN. Each device maintains three variables for each transmission attempt: the number of times the CSMA/CA algorithm is required to backoff while attempting to access the channel (NB), the Contention Window length (CW) which defines the number of backoff periods that need to be clear of channel activity before transmission starts, and the Backoff Exponent (BE) which gives the number of number of backoff periods a device should wait before attempting to access a channel. The steps of the CSMA/CA algorithm are as follows:

1. Initialize NB and CW for slotted CSMA/CA or NB and BE for unslotted CSMA/CA
2. Delay for a random number of complete backoff periods from 0 to $2^{BE} - 1$
3. Perform a CCA: the CCA starts immediately for unslotted CSMA/CA or on the boundary of a backoff period for slotted CSMA/CA
4. Increment both NB and BE by one if channel is busy
5. If the channel is idle, transmission begins immediately for the unslotted CSMA/CA or on the boundary of the next backoff period for slotted CSMA/CA

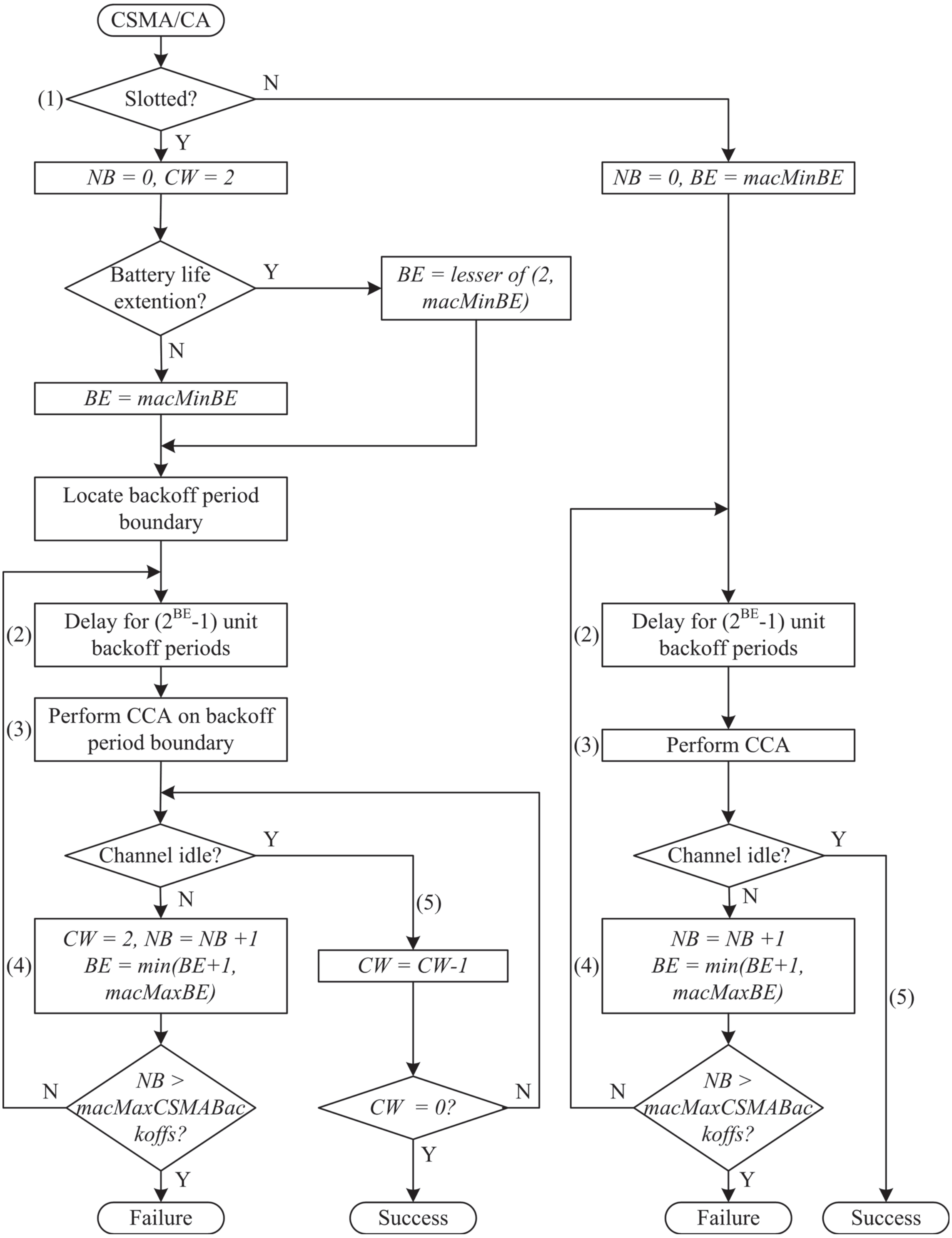


Bild A.17: CSMA/CA algorithm [IEEE, 2006]

Nomenclature

Formula symbols:

α	Actuator packet loss switch , see equation (2.15), page 14
β	Sensor packet loss switch , see equation (2.15), page 14
$\ddot{\varphi}$	Pendulum angular acceleration , see equation (A.4), page 109
$\dot{\varphi}$	Pendulum angular velocity , see equation (A.4), page 109
Φ, Γ_0, Γ_1	Discrete-time state-space matrices , see equation (2.7), page 13
ρ	Packet transmission probability , see equation (2.15), page 14
τ_k^a	Computational delay in sensor node , see equation (2.9), page 14
τ_k^{ca}	Delay between controller and actuator , see equation (2.9), page 14
τ_k^c	Computational delay in controller node , see equation (2.9), page 14
τ_k^{sc}	Delay between sensor and controller , see equation (2.9), page 14
τ_k^s	Computational delay in sensor node , see equation (2.9), page 14
φ	Pendulum angle , see equation (A.4), page 109
A, B, C	Continuous-time state-space matrices , see equation (2.2), page 12
A_d, B_d, C_d	Discrete-time state space matrices , see equation (A.14), page 111
h	Sampling period, see equation (2.7), page 13
K	State-space gain matrix , see equation (2.7), page 13
L	Kalman filter gain , see equation (2.38), page 27
P_L	Number of packet losses , see equation (2.44), page 29
P_{Total}	Total number of packet losses , see equation (2.44), page 29
pn	Packet number , see equation (2.44), page 29
t	Reference time , see equation (2.44), page 29
u	Control vector , see equation (2.2), page 12
v	Process noise , see equation (2.2), page 12
w	Measurement noise , see equation (2.2), page 12
x	State vector , see equation (2.2), page 12

y	Output vector , see equation (2.2), page 12

Mathematical operators:

$\cdot$	Scalar multiplication , page 14
$E[X]$	Expected value of the random variable X , page 50
$P[X = x]$	Probability of X being equal to x , page 14

Notions and abbreviations:

$1G$	First-Generation of Wireless Mobile Telecommunications , page 18
$2.5G$	Second and a Half Generation of Wireless Mobile Telecommunications , page 18
$2G$	Second-Generation of Wireless Mobile Telecommunications , page 18
$3G$	Third-Generation of Wireless Mobile Telecommunications , page 18
$4G$	Fourth-Generation of Cellular Wireless Standards , page 18
$6LoWPAN$	IPv6 over Low power Wireless Personal Area Networks , page 18
MacFrameRetries	number of frame retransmissions , page 29
macMaxCSMABackoffs	maximum number of backoffs , page 29
macMinBE	minimum number of backoffs , page 29
A/D	Analog-to-Digital , page 13
ADT	Average Dwell Time, page 24
AmI	Ambient Intelligence , page 29
$ARPANET$	Advanced Research Projects Agency Network , page 12
$AWGN$	Additive White Gaussian Noise , page 62
CAN	Controller Area Network , page 11
CAP	Contention Access Period, page 123
CAP	Contention access period , page 29
CFP	Contention Free Period, page 123
CPU	Central Processing Unit , page 3
$CSMA/CA$	Carrier Sense Multiple Access Collision Avoidance, page 8
$CSMA/CA$	Carrier Sense Multiple Access with Collision Avoidance , page 6
D/A	Digital-to-Analog , page 13
DCF	Distributed Coordination Function, page 30
DMS	Dynamic Modulation Scaling , page 52
EU	European Union , page 3

GAs	Genetic Algorithms, page 6
GTS	Guaranteed Time Slot, page 123
$HART$	Highway Addressable Remote Transducer , page 1
HiL	Hardware in the loop, page 7
$HiperLAN$	High Performance Radio Local Area Network , page 18
$HVAC$	Heating, Ventilating and Air Conditioning , page 1
IAE	Integral of the absolute value of error , see equation (A.48), page 118
IFS	Inter-frame space, page 30
IFS	Interframe spacing , page 59
IMC	Internal Model Control , page 22
ISA	International Federation of the National Standardizing Associations , page 1
ISE	Integral of the squared error , see equation (A.48), page 118
IT	Information Technology , page 1
$ITAE$	Integral of time-weighted absolute error , see equation (A.48), page 118
$ITSE$	Integral of time-weighted square error , see equation (A.48), page 118
LQG	Linear Quadratic Gaussian , page 4
LQR	Linear-quadratic regulator, page 13
$LR-WPAN$	Low-Rate Wireless Personal Area Network , page 18
LTI	Linear Time Invariant , see equation (2.2), page 12
$M2M$	Machine-to-Machine , page 1
MAC	Medium Access Control, page 6
$macMaxBE$	Maximum value of the backoff exponent , page 59
$macMaxCSMABackoffs$	Maximum number of backoffs before declaring a channel access failure , page 59
$macMaxFrameRetries$	Maximum number of retries after a transmission failure , page 59
$macMinBE$	Minimum value of the backoff exponent , page 59
MAD	Maximum allowable delay , page 39
$MAPL$	Maximum allowable number of consecutive packet losses , page 39
$MATI$	Maximum Allowable Transfer Interval , page 23
$MJLS$	Markovian jump linear system , page 26
MPC	Model Predictive Control , page 22
MSS	Mean-square stable , page 26

N	Normal force , see equation (A.4), page 109
NCS	Networked Control System, page 2
$O-QPSK$	Offset Quadrature Phase Shift Keying , page 63
P	Perpendicular force , see equation (A.4), page 109
PAN	Personal Area Network , page 122
PCF	Point Coordination Function, page 30
$PDAs$	Personal Digital Assistant
PER	Packet Error Rate , page 61
QoC	Quality of Control, page 5
QoS	Quality of Service, page 5
$RFID$	Radio-frequency identification , page 18
$RSSI$	Received Signal Strength Indicator , page 29
RTT	Round Trip Time , page 119
$RUNES$	Reconfigurable Ubiquitous Networked Embedded Systems , page 3
SI	International System of Units , page 110
$SINR$	Signal to Interference plus Noise Ratio , page 61
$TDMA$	Time Division Multiple Access , page 6
$TDMA$	Time Division Multiple Access, page 8
UAV	Unmanned Aerial Vehicle , page 1
$WCET$	Worst Case Execution Time , page 3
$Wi-Fi$	Wireless Fidelity , page 18
$WiMAX$	Worldwide Interoperability for Microwave Access , page 18
$WISA$	Wireless Interface for Sensors and Actuators, page 1
$WLAN$	Wireless Local Area Network , page 1
$WMAN$	Wireless Metropolitan Area Network , page 18
$WNCS$	Wireless Networked Control System, page 2
$WPAN$	Wireless Personal Area Network , page 18
WSN	Wireless Sensor Networks , page 29
$WWAN$	Wireless Wide Area Network , page 18

Some specific sets:

$\mathbb{R}$	Real numbers , page 13
$\mathbb{R}^n$	Real n-vectors ($n\times$ 1 matrices) , page 13
$\mathbb{Z}_+$	Nonnegative integers , page 13

Bibliography

Alliance, TinyOS. 2011 (August). *TinyOS*. www.tinyos.net.

Ares, B. Zurita, Park, P. G., Fischione, C., Speranzon, A., & Johansson, K. H. 2007. On Power Control for Wireless Sensor Networks: System Model, Middleware Component and Experimental Evaluation. *In: Proceeding of the European Control Conference (ECC'07)*.

Aström, Karl Johan, & Wittenmark, Björn. 1990. *Computer Controlled Systems: Theory and Design (3rd Edition)*. Printice Hall.

Baillieul, John, & Antsaklis, Panos. 2007. Control and Communication Challenges in Networked Real-Time Systems. *Proceedings of the IEEE*, **95**(1), 9 – 29.

Bemporad, Alberto, Heemels, Maurice, & Johansson, Mikael. 2010. *Networked Control Systems*. Lecture Notes in Control and Information Sciences (LNCIS), no. 406. Springer-Verlag.

Bennett, Stuart. 1979. *A History of Control Engineering, 1800-1930*. Institution of Electrical Engineers Stevenage.

Bennett, Stuart. 1993. *A History of Control Engineering 1930-1955*. Peter Peregrinus Hitchin.

Berman, Jonathan, & Ritorto, Mark. 2009. *2009-2013 Pervasive Internet & Smart Services Market Forecast Report*. Tech. rept. Harbor Research, Inc.

Branicky, M.S., Phillips, S.M., & Zhang, Wei. 2000. Stability of networked control systems: explicit analysis of delay. *Pages 2352 – 2357 of: Proceedings of the American Control Conference (ACC2000)*, vol. 4.

Buettner, Michael, Yee, Gary V., Anderson, Eric, & Han, Richard. 2006. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. *In: Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys 2006)*.

Chamaken, A., & Litz, L. 2010 (June). Joint Design of Control and Communication for Wireless Networked Control Systems: A Case Study. *Pages 1834–1840 of: Proceedings of the American Control Conference ACC'10*.

Chamaken, Alain, & Litz, Lothar. 2011 (June). Modeling and Optimization of the Lifetime of Wireless Nodes in Wireless Networked Control Systems. *Pages 93 – 96 of: Proceedings of the Automation 2011 (VDI/GMA-Congress 2011)*.

Crossbow. 2011. *Imote2 Platform for Wireless Sensor Network*. http//:www.xbow.com/.

Drew, Michael, Liu, Xiangheng, Goldsmith, Andrea, & Hedrick, Harl. 2005 (December). Networked Control System Design over a Wireless LAN. *Pages 6704 – 6709 of: Proceedings of the 44th IEEE Conference on Decision and Control*.

Dunkels, Adam. 2011 (August). *The Contiki OS*. http://www.sics.se/contiki/.

El-Hoiydi, A., & Decotignie, J.-D. 2004. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. *Pages 244 – 251 of: Proceedings of the ninth International Symposium Computers and Communications (ISCC 2004)*, vol. 1.

Elia, N., & Mitter, S. K. 2001. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, **46**(9), 1384 – 1400.

Emerson. 2008. *Wireless Technology Landscape.* http://emerson.com/.

Engel, Markus, & Krämer, Marc. 2009. *New Bootloader for the Imote2 platform.* Tech. rept. 373/09. University of Kaiserslautern, Department of Networked Systems.

Fankem, Steve. 2009 (April). *Design and Implementation of Robust and Optimal Controllers for an Inverted Pendulum.* http://www.eit.uni-kl.de/litz.

Fischione, Carlo, Park, Pangun, Marco, Piergiuseppe Di, & Johansson, K. H. 2011. *Design Principles of Wireless Sensor Networks Protocols for Control Applications.* Springer. Chap. 9, pages 203 – 238.

Franklin, Gene F., Powell, J. David, & Workman, Michael. 1997. *Digital Control of Dynamic Systems (3rd Edition).* Addison Wesley Longman.

Gabel, Oliver. 2008 (June). *Design of QoS-adaptive Control for AmI-Systems with heuristic methods.* Ph.D. thesis, Departement of Automatic Control, University of Kaiserslautern, Kaiserslautern, Germany.

Gnawali, Omprakash, Fonseca, Rodrigo, Jamieson, Kyle, Moss, David, & Levis, Philip. 2009. Collection tree protocol. *In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009.*

Grüne, Lars, Jerg, Stefan, Junge, Oliver, Lehmann, Daniel, Lunze, Jan, Müller, Florian, & Post, Marcus. 2009. Two complementary approaches to event-based control. *at-Automatisierungstechnik (Special Issue on Networked Control Systems)*, **58**(4), 173 – 182.

Hadjicostis, C.N., & Touri, R. 2002. Feedback control utilizing packet dropping network links. *Pages 1205 – 1210 of: Proceedings of the 41st IEEE Conference on Decision and Control (CDC2002)*, vol. 2.

HART. 2007. *HART Communication Protocol.* http://www.hartcomm.org/.

Hassibi, A., Boyd, S. P., & How, J. P. 1999. Control of asynchromous dynamical systems with rate constraints on events. *Pages 1345 – 1351 of: Proceedings of the 38th Conference on Decision and Control (CDC1999)*, vol. 2.

Heinzelman, Wendi Rabiner, Chandrakasan, Anantha, & Balakrishnan, Hari. 2000. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *In: Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, vol. 8.

Hernández, Aitor. 2010. *Wireless Process Control using IEEE 802.15.4 Protocol.* M.Phil. thesis, KTH Royal Institute of Technology, Departement of Electrical Engineering, Stockholm, Sweden.

Hespanha, By Joao P., Naghshtabrizi, Payam, & Xu, Yonggang. 2007. A Survey of Recent Results in Networked Control Systems. *Proceedings of the IEEE*, **95**(1), 138 – 162.

Hoesel, L Van, & Havinga, P. 2004. A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks. *Computer and Information Science*, **12**(3), 493 – 506.

Hokayem, Peter F., & Abdallah, Chaouki T. 2004. Inherent Issues in Networked Control Systems: A Survey. *Pages 4897 – 4902 of: Proceeding of the 2004 American Control Conference.*

Hristu-Varsakelis, Dimitrios, & Levine, William S. 2005. *Handbook of Networked and Embedded Control Systems.* Control Engineering. Birkhäuser.

Huang, Dan, & Nguang, Sing Kion. 2009. *Robust Control for Uncertain Networked Control Systems with Random Delays.* Lecture Notes in Control and Information Sciences (LNCIS), no. 386. Springer-Verlag.

IEEE. 2006. *Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks.* http://www.ieee802.org/15/pub/TG4.html.

ISA. 2008. *ISA100, Wireless Systems for Automation.* http://www.isa.org/.

Jiang, Zhe. 2008 (October). *Inverstigation of Different Control Structures on an Inverted Pendulum.* http://www.eit.uni-kl.de/litz.

Kim, Yong Ho, Kwon, Wook Hyun, & Park, Hong Seong. 1996. Stability and a scheduling method for network-based control systems. *Pages 934 – 939 of: Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 2.

Lian, Feng-Li, Moyne, J.R., & Tilbury, D.M. 2001. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems*, **21**(1), 66 – 83.

Lin, Hai, & Antsaklis, P.J. 2004. Persistent disturbance attenuation properties for networked control systems. *Pages 953 – 958 of: 43rd IEEE Conference on Decision and Control (CDC2004)*, vol. 1.

Lin, Hai, Zhai, Guisheng, & Antsaklis, P.J. 2003. Robust stability and disturbance attenuation analysis of a class of networked control systems. *Pages 1182 – 1187 of: 42nd IEEE Conference on Decision and Control (CDC2003)*, vol. 2.

Liu, Xiangheng, & Goldsmith, Andrea. 2004. Wireless Network Design for Distributed Control. *Pages 2823 – 2829 of: Proceedings of the 43rd IEEE Conference on Decision and Control (CDC2004)*, vol. 3.

Luck, Rogelio, & Ray, Asok. 1990. An observer-based compensator for distributed delays. *Automatica*, **26**(5), 903 – 908.

Manjeshwar, A., & Agrawal, D.P. 2002. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. *Pages 2009 – 2015 of: Proceedings of the 15th International Parallel and Distributed Processing Symposium.*

Mazumder, Sudip K. 2011. *Wireless Networking Based Control.* Springer Science.

Minoli, Daniel, & Schmidt, Andrew. 1999. *Internet Architectures.* Wiley.

Moeller, Scott, Sridharan, Avinash, Krishnamachari, Bhaskar, & Gnawali, Omprakash. 2010. Routing without routes: the backpressure collection protocol. *In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2010).*

MONTESTRUQUE, L. A., & ANTSAKLIS, P. J. 2003a. On the model-based control of networked systems. *Automatica*, **39**(10), 1837 – 1843.

MONTESTRUQUE, L. A., & ANTSAKLIS, P. J. 2003b. Stochastic stability for model-based networked control systems. *Pages 4119 – 4124 of: Proceedings of the American Control Conference (ACC2003)*, vol. 5.

NEMPE, JEAN. 2008 (November). *Design of an IMC controller for the inverted pendulum system.* http:
www.eit.uni-kl.de/litz.

NILSSON, J. 1998 (January). *Real-Time control systems with delays.* Ph.D. thesis, Departement of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NILSSON, J., & BERNHARDSSON, B. 1996. Analysis of real-time control systems with time delays. *Pages 3173 – 3178 of: Proceedings of the 35th Conference on Decision and Control (CDC 1996)*, vol. 3.

NILSSON, J., & BERNHARDSSON, B. 1997. LQG control over a Markov communication network. *Pages 4586 – 4591 of: Proceedings of the 36th Conference on Decision and Control (CDC 1997)*, vol. 5.

NILSSON, J., & BERNHARDSSON, B. 1998. Some topics in real-time control. *Pages 2386 – 2390 of: Proceedings of the American Control Conference (ACC 1998)*, vol. 4.

NILSSON, J., BENHARDSSON, B., & WITTENMARK, B. 1998. Stochastic analysis and control or real-time systems with random time delays. *Automatica*, **34**(1).

P. LIN, C. QIAO, & WANG, X. 2004. Medium access control with a dynamic duty cycle for sensor networks. *Pages 1534 – 1539 of: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2004)*, vol. 3.

PAPOULIS, ATHANASIOS. 1984. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill.

PARK, PANGUN, FISCHIONE, C., BONIVENTO, A., JOHANSSON, K.H., & SANGIOVANNI-VINCENT, A. 2011. Breath: An Adaptive Protocol for Industrial Control Applications Using Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, **10**(6), 821 – 838.

PETTERSSON, N., & JOHANSSON, K. H. 2006. Modelling and control of auxiliary loads in heavy vehicles. *International Journal of Control*, **79**(5), 479 – 495.

PROAKIS, J. G. 1995. *Digital Communications, 3rd ed.* McGraw Hill International Editions.

SEILER, P., & SENGUPTA, R. 2001. Analysis of communication losses in vehicle control problems. *Pages 1491 – 1496 of: Proceedings of the American Control Conference (ACC2001)*, vol. 2.

SINOPOLI, BRUNO, SCHENATO, LUCA, FRANCESCHETTI, MASSIMO, POOLLA, KAMESHWAR, JORDAN, MICHAEL I., & SASTRY, SHANKAR S. 2004. Kalman Filtering with Intermittent Observations. *IEEE Transactions on Automatic Control*, **49**(9), 1453 – 1464.

SOCRADES. 2011. *SOCRADES project: European research and advanced development project.* http://www.socrades.eu/.

SOHRABI, K., GAO, J., AILAWADHI, V., & POTTIE, G. 2000. Protocols for selforganization of a wireless sensor network. *IEEE Personal Communication*, **7**(5), 16 – 27.

Stallings, William. 2007. *Data & Computer Communications*. Prentice Hall.

Texas. 2007. *CC2420 Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBe Ready RF Transceiver*.

Tipsuwan, Yodyium, & Chow, Mo-Yuen. 2002. Gain adaptation of networked mobile robot to compensate QoS deterioration. *Pages 3146 – 3151 of: The 28th annual conference of the IEEE industrial electronics society (IECON 02)*, vol. 4.

Tipsuwan, Yodyium, & Chow, Mo-Yuen. 2003. Control mehtodologies in networked control systems. *Control Engineering Practice*, **11**, 1099 – 1111.

W. Ye, J. Heidemann, & Estrin, D. 2004. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, **12**(3), 493 – 506.

Walsh, G.C., Ye, Hong, & Bushnell, L.G. 2002. Stability Analysis of Networked Control Systems. *IEEE Transactions on Control Systems Technology*, **10**(3), 438 – 446.

Willig, Andreas. 2008. Recent and emerging topics in wireless industrial communications. *IEEE Transactions on Industrial Informatics*, **4**(2), 102 – 124.

Wong, W. S., & Brockett, R. W. 1999. Systems with finite communication bandwidth constraints-II: Stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, **44**(5), 1049 – 1053.

Xia, Feng, & Sun, Youxian. 2008. *Cross-Layer Adaptive Feedback Scheduling*. Springer. Chap. 8, pages 216 – 243.

Xiao, Lin, Johansson, Mikael, Hindiz, Haitham, Boyd, Stephen, & Goldsmith, Andrea. 2003. Joint optimization of communication rates and linear systems. *IEEE Transactions on Automatic Control*, **48**(1), 148 – 153.

Xu, Ya, Heidemann, John, & Estrin, Deborah. 2001. Geography-informed Energy Conservation for Ad Hoc Routing. *Pages 70 – 84 of: Proceedings of the International conference on Mobile Computing and Networking (MobiCom 2001)*.

Yang, T.C. 2006. Networked control system: a brief survey. *Control Theory and Applications*, **153**(4), 1350 – 2379.

Ye, H., Walsh, G. C., & Bushnell, L. 2001. Real-Time Mixed Traffic Wireless Networks. *IEEE Transactions on Industrial Electronics*, **48**(5), 883 – 890.

Yépez, José, Matí, Pau, & Fuertes, Josep M. 2002. Control Loop Performance Analysis over networked Control Systems. *Pages 2880 – 2885 of: IEEE 2002 28th Annual Conference of the Industrial Electronics Society (IECON 02)*, vol. 4.

Yuan, GE, TIAN, Li, & LIU, Zhenan. 2007. Survey on the stability of networked control systems. *Journal of Control Theory and Applications*, **5**(4), 374 – 379.

Zang, W., & M. S. Branicky, M. 2001. Stability of networked control systems with time-varying transmission period. *In: Proceedings of the Allerton Conference on Communication Control and Computing*.

Zhai, Guisheng, Hu, Bo, Yasuda, K., & Michel, A.N. 2002. Qualitative analysis of discrete-time switched systems. *Pages 1880 – 1885 of: Proceedings of the American Control Conference (ACC2002)*, vol. 3.

Zhang, W., Branicky, M., & Phillips, S. 2001. Stability of Networked Control Systems. *IEEE Control System Magazine*, **21**, 84 – 99.

ZigBee. 2005. *ZigBee Protocol Spezification.* http://www.zigbee.org/.

About the author

Alain Chamaken was born on April 1973 in Cameroon and currently lives in Langenargen, Germany. After the studies of Communications and Electronics both at the University of Applied Sciences in Kaiserslautern and at the University of East London followed by studies of Automatic Control at the University of Kaiserslautern, he joined the Institute of Automatic Control of the University of Kaiserslautern in October 2007. His main research area was the field of Wireless Networked Control Systems. The author works as a development engineer for the company ZF in Friedrichshafen since January 2012.