KTH Electrical Engineering

Predictive Control for Wireless Networked Systems in Process Industry

ERIK HENRIKSSON

Doctoral Thesis
Stockholm, Sweden 2014

TRITA-EE 2014:004
ISSN 1653-5146
ISBN 978-91-7595-014-3

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

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i reglerteknik fredagen den 14 mars 2014, klockan 14:15 i sal F3, Kungliga Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

Tryck: Universitetsservice AB

Abstract

Wireless networks in industrial process control enable new system architectures and designs. However, wireless control systems can be severely affected by the imperfections of the communication links. This thesis proposes new methods to handle such imperfections by adding additional components in the control loop, or by adapting sampling intervals and control actions.

First, the predictive outage compensator is proposed. It is a filter which is implemented at the receiver side of networked control systems. There it generates predicted samples when data are lost, based on past data. The implementation complexity of the predictive outage compensator is analyzed. Simulation and experimental results show that it can considerably improve the closed-loop control performance under communication losses.

The thesis continues with presenting an algorithm for controlling multiple processes on a shared communication network, using adaptive sampling intervals. The methodology is based on model predictive control, where the controller jointly decides the optimal control signal to be applied as well as the optimal time to wait before taking the next sample. The approach guarantees conflict-free network transmissions for all controlled processes. Simulation results show that the presented control law reduces the required amount of communication, while maintaining control performance.

The third contribution of the thesis is an event-triggered model predictive controller for use over a wireless link. The controller uses open-loop optimal control, re-computed and communicated only when the system behavior deviates enough from a prediction. Simulations underline the methods ability to significantly reduce computation and communication effort, while guaranteeing a desired level of system performance.

The thesis is concluded by an experimental validation of wireless control for a physical lab process. A hybrid model predictive controller is used, connected to the physical system through a wireless medium. The results reflect the advantages and challenges in wireless control.

Acknowledgment

First, I want to thank my main advisor Karl Henrik Johansson. You have been excellent in encouraging, inspiring and supporting me. Second, I want to thank my co-advisor Henrik Sandberg for all the hands-on assistance in my research: I am always impressed with your skills. Thank you both for all your constructive advice and help over the years.

I want to thank all my past and present colleagues at the Department of Automatic Control for making it such a great place to work. A special thanks to Oscar Flärdh for sharing better and worse days. Thanks also to Mariette Annergren, Martin Jakobsson, Karl Henrik Johansson, Christian A. Larsson, Henrik Sandberg, Håkan Terelius and Olle Trollberg for reading and giving feedback on various parts of this manuscript.

I would also like to thank all the students that I have had the pleasure to teach over the years. I had great fun. Mikael Johansson deserves a special mentioning for supporting and encouraging me in developing my teaching.

During the work leading up to this thesis I have been fortunate to get the opportunity to visit other research groups. I would like to thank Alberto Bemporad for hosting my visit at the University of Siena in Italy and Daniel E. Quevedo for hosting my visit at The University of Newcastle in Australia. You both provided me with new insights, ideas and perspectives.

I want to express my appreciation to the Swedish Research Council (VR), the Swedish Governmental Agency for Innovation Systems (VINNOVA), the Swedish Foundation for Strategic Research (SSF), the Knut and Alice Wallenberg Foundation and the European Commission for the financial support that made this work possible.

Finally, I want to thank all my friends and especially my family for showing me what really matters in life, and that there is so much more to it than research. Your love and support has been invaluable!

Erik Henriksson
Stockholm, February 2014

Contents

Chapter 1

Introduction

Wireless communication technology is becoming an integral part of our everyday lives. It is present in most areas of society and the information exchanged over these networks is immense. The communication is made seamlessly over different technologies and platforms. For example, we use our smartphones to interact and exchange information with each other over both cellular and wireless local area networks, without really caring about which technology we use when. We just expect it to work. The low cost and high flexibility provided by wireless devices has led to that wireless appliances and systems are being used more and more.

In recent years the benefit of using wireless technology in large-scale industrial control systems, such as in process industry, utility infrastructure and transportation networks has become evident (Samad et al., 2007; Samad and Annaswamy, 2011). This has resulted in a fast growing interest in wireless automation systems tailored for these purposes, there are even a number of commercially available products from companies as Abb, Emerson and Honeywell. When fully in place, the great potential in wireless automation systems suggests that in the future, monitoring and control will represent a large portion of all traffic over wireless networks. This is indicated by the visions of cyber-physical systems and internet-of-things. However, the available technology still has some shortcomings that need to be handled to enable the full migration to these wireless automation systems. This thesis discusses some of these issues and suggests methods to adapt existing control strategies to work better in a wireless setting.

In this chapter, we first give an introduction to wireless control in process industry and some of the current issues. We then proceed to give motivating examples for the contributions of the thesis. After this, the problem formulation is given. The chapter is concluded with the thesis outline and contributions, as well as a list of notation and abbreviations.

1.1 Wireless Control in Process Industry

Advances in wireless radio and sensor technologies have enabled the engineering of networked sensing and control systems, which are now being tested and evaluated in industry for automation and process control (Neumann, 2007). Recent standardization efforts for wireless automation include the WirelessHART (Hart Communication Foundation, 2007) and the Isa100 (International Society of Automation, 2009) communication protocols, tailored for process control. Although wired communication networks have been commonly and successfully used in distributed process control systems since the 1970's (Samad et al., 2007), there are several benefits of introducing wireless networking in industrial control applications, as described next.

Reduced Implementation Cost

The introduction of wireless links in process control systems has the potential to reduce costs considerably. The most obvious effect is that wireless links lead to reduced wiring of the control system, which constitutes a substantial part of the cost in installing an automation system (Samad et al., 2007; Åkerberg et al., 2011). This is due both to the high price of cables as well as the cost for installing and maintaining the cabled infrastructure. As connectors and wires are prone to failure due to wear and tear, they lead to many faults in industrial control systems (Mhaskar et al., 2013). Wireless devices consequently have a potential to reduce the down-time for these systems. Installing the hardware for a wireless infrastructure is limited to some routers, repeaters and gateways. This leads to faster deployments, which further reduces both down-time and installation cost.

Improved Sensing and Actuation

The increased flexibility induced by using wireless instead of wired infrastructure, enables the use of more flexible automation and control structures. With wireless links there are fewer physical design limitations, in turn simplifying installation of additional sensors and actuators. This enables automation of non-stationary machinery and installation of new automation infrastructure, or reconfiguration of existing infrastructure, to a low marginal cost. Thanks to this, new and better designs can be exploited in system development and operation.

1.2 Challenges

The use of wireless technology in feedback control loops raises new challenges. The network medium introduces uncertainties such as packet loss, transmission delay, et cetera. The impact of these uncertainties on the closed-loop control system depends on many system aspects. It has become evident that new communication protocols and control strategies are needed for wireless control systems (Antsaklis and Baillieul, 2007). As the available radio spectrum is limited, care also has to be taken to use it efficiently. The problems faced may be addressed from different angles, under different paradigms, and by using different tools.

A natural approach is to focus on the wireless medium itself and develop better radio hardware and new communication standards. Together with information coding, this could significantly reduce the uncertainties introduced, so that they become transparent to the control loop. Another way is to add extra logic in the control circuit, to make it robust against imperfections in the network. This allows the use of the network at hand and whatever controller and control structure one has. An important benefit of this is modularity, making it possible to improve different parts of the automation system independently. Yet another approach is to adapt the controllers, so that they are better suited to be used over wireless networks. For example, controllers can be adapted to only transmit data over the network when new information is available and worthwhile sending. Another example is controllers which plan the time of their executions, so that it is guaranteed that both the feedback measurement as well as the actuation command can be sent over the network without conflict. This leads to a more restrictive utilization of the network. A restrictively used network in turn allows the use of the same wireless network to control several processes, while still guaranteeing reliable transmission of data and satisfactory control performance.

1.3 Motivating Examples

We now give motivating examples for the problems addressed in the thesis. The first example gives an intuitive explanation to the problem of losing data packets in the communication, and the importance of how these losses are distributed in time. In the second example, we show that there is significant performance to be gained by adding additional logic in the feedback loop, to compensate for these losses. The third example illustrates the possibilities available to reduce the communication in wireless control systems, by using sparse control updates. Finally, in the fourth example, it is shown how communication can be reduced using feedback control with adaptive sampling intervals.

Compensating for Unreliable Communication

When a system is controlled over a wireless network the imperfections in the network may cause data packets sent between the sensors, controllers, and actuators to be lost. When this happens, the feedback loop is broken and additional logic needs to compensate for the lost information. Compensating for lost packets can, depending on how and when the losses occur, be a hard or relatively easy problem to handle. How the distribution of the packet loss affects the control system performance is illustrated in Example 1.1.

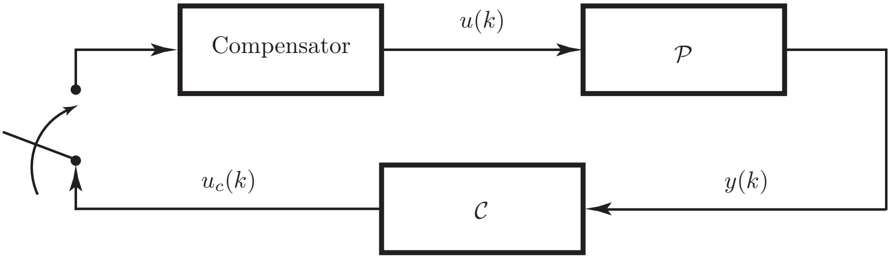


Figure 1.1: A networked control system subject to packet loss.

Example 1.1

Consider the networked control system in Figure 1.1 and let the switch represent that packets between controller and process can be lost. The process $\mathcal{P}$ is given by

$$P(s) = \frac{1}{s^2}$$

and the controller $\mathcal{C}$ is given by as sampled version of

$$C(s) = \frac{20s^2 + 12s + 1}{s^2 + 2s},$$

sampled with $T_s = 0.1\,s$. If a packet is lost, *i.e.*, the switch is open, we use the packet loss compensation policy to apply the last known control signal to the process. This additional logic is contained in the *Compensator* block. To illustrate how different loss distributions affect the performance of the system, we simulate it using two different loss profiles, both with 20 % average packet loss over time.

Studying the results in Figure 1.2 we see that for short and sparse losses the method to hold the last known value works well. Instead, looking at Figure 1.3, it becomes evident that when losses are grouped into longer connected periods a more advanced compensation scheme is needed.

Example 1.1 shows that the distribution of the communication loss is an important parameter when it comes to assessing how losses affect the control system performance. The example also highlights that depending on the packet loss profile, different kinds of actions needs to be taken to reduce the impact of the losses on the system behavior. If the system is subject to sparse and short bursts of losses it is probably valid to assume that conditions stay the same during the loss period. If the losses instead come in longer bursts, we have to use a more elaborate method or device to overcome them. One such method is the predictive outage compensator, presented in Chapter 3 and illustrated in Example 1.2.

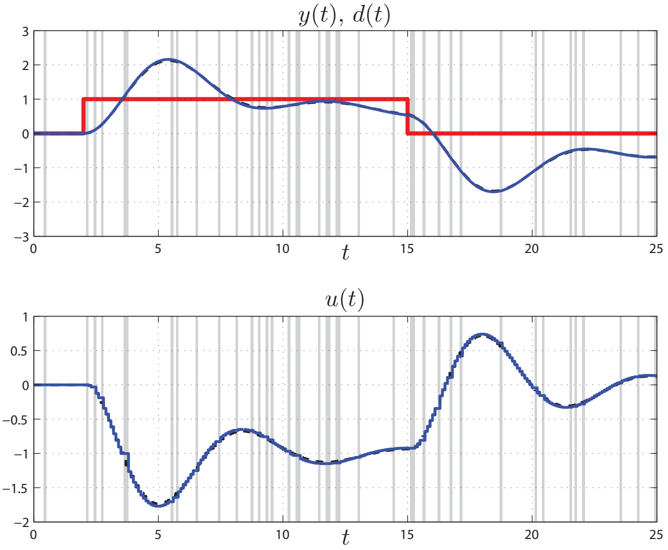


Figure 1.2: System output behavior with losses (blue) compared to the output of the system without losses (black) under disturbance d (red): 20 % packet loss, distributed in short and sparse periods (grey area). Performance using a hold compensator.

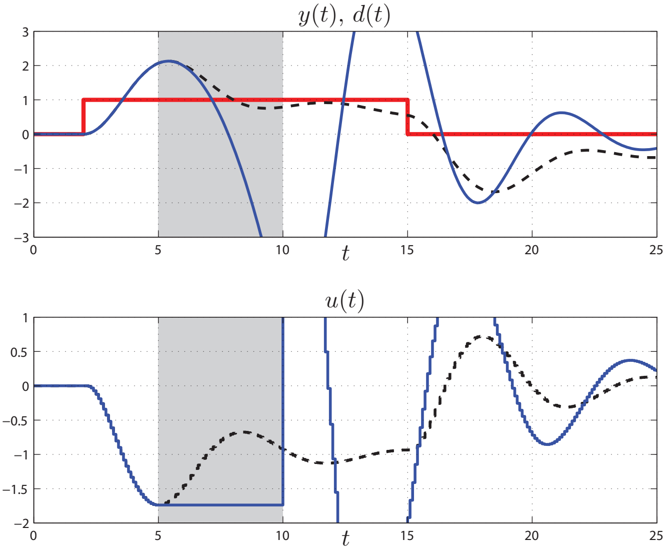


Figure 1.3: System output behavior with losses (blue) compared to the output of the system without losses (black) under disturbance d (red): 20 % packet loss in a connected period (grey area). Performance using a hold compensator.

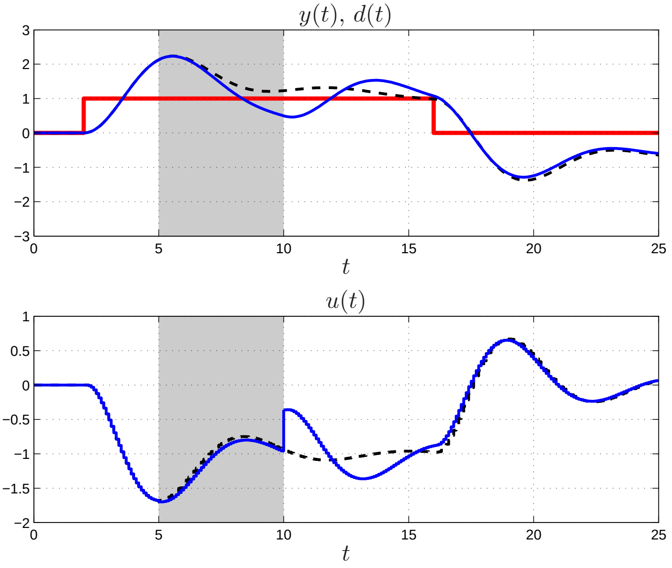


Figure 1.4: System output behavior with losses (blue) compared to the system output without losses (black) under disturbance d (red): 20 % packet loss in a connected period (grey area). Performance using a predictive outage compensator.

Example 1.2

Let us again consider the networked control system presented in Example 1.1. Consider the same process and controller but replace the device holding the last known signal in the *Compensator* block with the predictive outage compensator, presented in Chapter 3. Simulating the system under communication outage, as in Example 1.1, we then get the behavior in Figure 1.4. A comparison with the results in Figure 1.3, shows that it is possible to significantly improve system performance under outage by adding a well-designed compensation device in the control loop.

Communication-Aware Control

Wired control systems used today within process industry typically uses classical time-triggered sampled-data control where measurements are taken, control actions are computed, and then actuated at equidistantly spaced time intervals. Algorithms and methods for sampled-data control emerged when digital computers were introduced in control systems, forcing the development of new theory able to utilize their potential. In the development of this theory care then had to be taken to handle the computers' computational capacity, finite numerical accuracy and implementation constraints, see, *e.g.*, (Åström and Wittenmark, 1997).

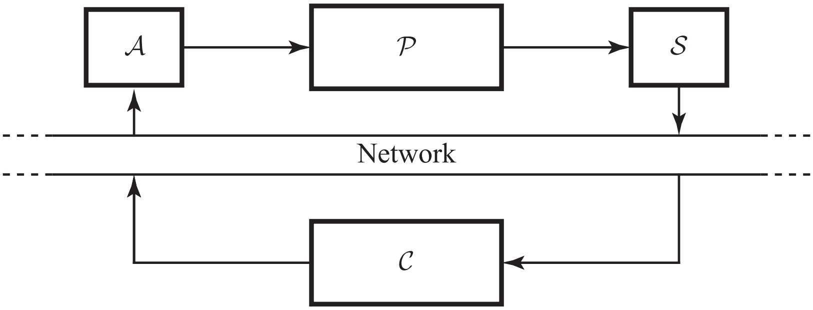


Figure 1.5: The actuator $\mathcal{A}$ and sensor $\mathcal{S}$ communicate with the controller $\mathcal{C}$ over a wireless network to control the process $\mathcal{P}$.

In the migration to wireless control systems, this now classical time-triggered sampled-data control theory needs to be developed further to fully make use of the wireless technology. In wireless communication the network utilization becomes a limiting factor. If several nodes are connected to the same network they must share the medium in a fair manner. Therefore, the available bandwidth is limited and it becomes important to use it effectively and only transmit when needed. We illustrate the potential communication savings in Example 1.3.

Example 1.3
We study the wireless control system in Figure 1.5 where the servo process

$$\mathcal{P} \begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\ y(t) = x_2(t), \end{cases} \tag{1.1}$$

corresponding to

$$P(s) = \frac{1}{s(s+1)},$$

is to be controlled from one set-point to another. The process is controlled by the controller $\mathcal{C}$, which receives sensor measurements and transmits control commands over the network. Realizing the controller $\mathcal{C}$ using a classical PI-controller

$$C(s) = K\left(1 + \frac{1}{T_I s}\right), \; K = 0.62, \; T_I = 105,$$

sampled every $T_s = 1\,\text{s}$, we obtain the system response shown in Figure 1.6. This strategy requires 62 transmissions over the network, 31 from the sensor to the controller and 31 from the controller to the actuator. If the process instead is controlled using off-line optimal control, where a trade-off between control performance and

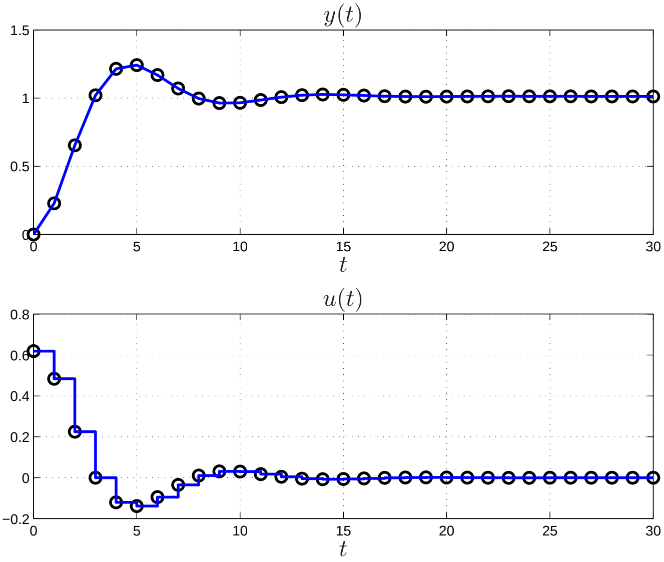


Figure 1.6: The process controlled using time-equidistant sampling and control. System output is shown in blue with sampling instances encircled in black.

the need for sampling is made, we get the results in Figure 1.7. As seen we get almost the same system performance. However, this strategy only requires a total of 8 transmissions, a reduction in communication by 87%.

Example 1.3 shows that there is great potential in developing new control algorithms better suited for wireless control, by finding feedback control laws which adapts sampling intervals and control actions to the available network resources. Such methods are developed in Chapter 4 and Chapter 5. We illustrate the performance they may exhibit in Example 1.4.

Example 1.4

Consider the setup presented in Example 1.3, where the process is controlled using off-line optimal sampling intervals and controls. Here we consider the same system, but with $y(t) = [x_1(t),\ x_2(t)]^T$ in (1.1), and use on-line optimal sampling and control based on state-feedback measurements. We do this by using the controller proposed in Chapter 4, which decides what to actuate and when to take the next sample, based on measurements of the states. Simulating the system using this controller we get the response shown in Figure 1.8. Note that only a few samples are needed to get a good performance. A comparison with the result in Figure 1.7 shows that it is possible to approximate the off-line optimal sampling and control sequence, by on-line optimal sampling and control.

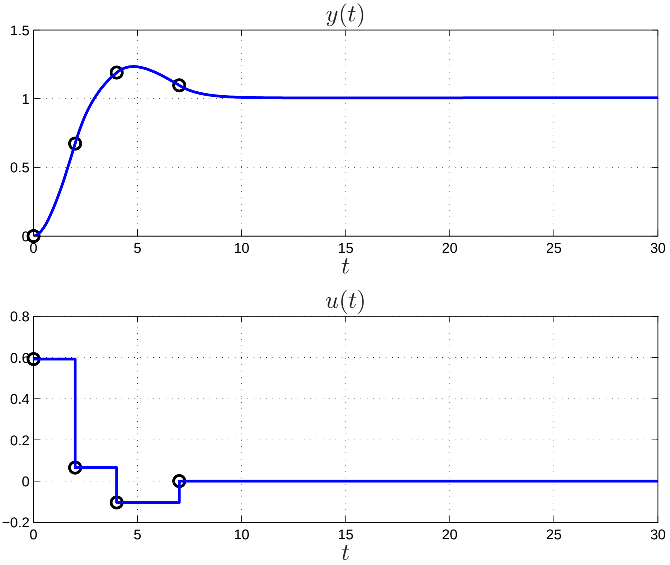


Figure 1.7: The process controlled using an adaptively sampled off-line optimal controller. System output is shown in blue with sampling instances encircled in black.

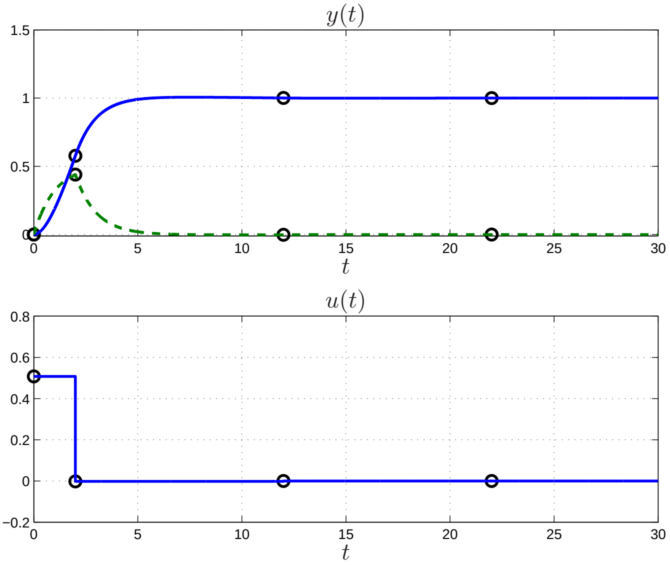


Figure 1.8: The process controlled using an adaptively sampled state-feedback controller. System output is shown in blue with additional state in green and sampling instances encircled in black.

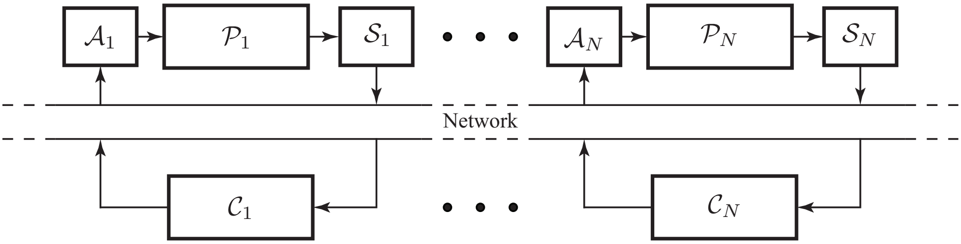


Figure 1.9: Sensors $\mathcal{S}$ and actuators $\mathcal{A}$ communicate with the controllers $\mathcal{C}$ over a wireless network to control the processes $\mathcal{P}$.

1.4 Problem Formulation

The thesis discusses wireless networked control in process industry, and how to adapt existing control strategies to work better when feedback measurements and actuation commands are sent over a wireless communication link. Consider the networked control system shown in Figure 1.9, where processes and controllers communicate over a shared wireless network. The thesis addresses two problems faced when controlling such a system. First, it is considered how to introduce compensators in the control loop to overcome losses in the communication, without having to change the involved controllers or the control structure. Second, communication-aware control algorithms taking the communication into account are derived.

The first problem considered is how to compensate for subsequent losses of packets, so called outages, in the control loop. This problem is addressed in Chapter 3 where we consider how to compensate for such outages in networked control systems by placing devices at the receiver sides of the network, without requiring any modifications to the existing control design. We call these devices predictive outage compensators. The predictive outage compensators are designed to overcome losses in the network by suggesting replacement commands in the event of an outage.

The second problem is on how to make controllers communication-aware. This problem can be formulated as how to find a controller which jointly decides what control signal to apply, as well as when the next measurement needs to be taken. It should by this a priori adapt the time between measurements in order to only communicate as often as needed. Chapter 4 shows how such a controller can control several processes sharing a common communication network, and still enable collision-free transmissions. Another variation of the communication-aware control problem, is that of finding a control strategy which approximates the performance and behavior of classical time-triggered model predictive control, but only uses the network to transmit samples and control commands when absolutely needed. This is studied in Chapter 5, where it is considered how processes can be controlled using sequences of open-loop optimal control commands, updated only when the measured process behavior differs significantly from the predicted behavior.

1.5 Outline and Contributions

We now give a summary of the remainder of the thesis, presenting the outline and content. For the chapters containing novel contributions by the author, the publications upon which the chapter is based are also stated.

Chapter 2: Background

In this chapter we give some background to the topics discussed in the thesis and discuss how it relates to existing research.

Chapter 3: Predictive Outage Compensation

A method to compensate for unpredictable interruptions in the communication by introducing a predictive outage compensator (POC) is proposed. The POC is a filter to be implemented in the receiver at the actuator and controller sides of a networked control system, without having to modify the implemented controller. If the receiver node does not receive data, the POC suggests an open-loop input to replace the missing data based on a history of past transmissions. We show how to design, tune and implement a POC. Theoretical bounds, simulation and experimental results show that a POC can considerably improve the closed-loop control performance under communication outages. We also show that it is possible to achieve good performance with a low-order implementation based on Hankel-norm approximation. Trade-offs between achievable performance, outage length and POC order are discussed. The results are illustrated via simulations and experiments on a wirelessly controlled two-tank process.

The chapter is based on the following publications:

- E. Henriksson, H. Sandberg, and K. H. Johansson. Predictive outage compensation for networked control systems. *Journal of Process Control*, 2013b. Under Review
- E. Henriksson, H. Sandberg, and K. H. Johansson. Reduced-order predictive outage compensators for networked systems. In *Proceedings of IEEE Conference on Decision and Control*, Shanghai, P.R. China, 2009
- E. Henriksson, H. Sandberg, and K. H. Johansson. Predictive compensation for communication outages in networked control systems. In *Proceedings of IEEE Conference on Decision and Control*, Cancun, Mexico, 2008

The experiments have previously been reported in the following master theses supervised by the author:

- J. Wallander. Implementation of a wireless control system: network routing and unreliable communication link compensation. Master's thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2012

- I. Cornell. Implementation of a collection tree routing protocol and a predictive outage compensator. Master's thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2012

Chapter 4: Self-Triggered Model Predictive Control

The chapter presents an algorithm for controlling multiple linear time-invariant processes on a shared communication network, by using adaptive sampling intervals. At every sampling instant the controller not only computes the new control command, but also decides the time interval to the next sample. The approach relies on model predictive control, where the cost function depends on the control performance as well as the cost for sampling. The latter is introduced in order to generate an adaptive sampling scheme for the overall system. The chapter presents a method for synthesizing such a predictive controller and gives explicit sufficient conditions for when it is stabilizing. Further, explicit conditions are given that guarantee conflict-free transmissions on the network. It is shown that the optimization problem can be solved off-line and that the controller can be implemented as a lookup table of state-feedback gains.

The chapter is based on the following publications:

- E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. Multiple-loop self-triggered model predictive control for network scheduling and control. *IEEE Transactions on Control Systems Technology*, 2013a. Under Review

- E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. Self-triggered model predictive control for network scheduling and control. In *Proceedings of IFAC International Symposium on Advanced Control of Chemical Processes*, Singapore, 2012

Chapter 5: Event-Triggered Model Predictive Control

An approach to event-triggered model predictive control for discrete-time linear systems subject to input and state constraints as well as exogenous disturbances is proposed. Stability properties are derived by evaluating the difference between the event-triggered implementation and the conventional time-triggered scheme. It is shown that the event-triggered implementation, in stationarity, is able to keep the state in an explicitly computable set given by a disturbance bound and the event threshold. Simulation results underline the effectiveness of the proposed scheme in terms of reducing the communication and computational effort while guaranteeing a desired performance.

The chapter is based on the following publication:

- D. Lehmann, E. Henriksson, and K. H. Johansson. Event-triggered model predictive control of discrete-time linear systems subject to disturbances. In *Proceedings of European Control Conference*, Zurich, Switzerland, 2013

Chapter 6: Model Predictive Control based on Wireless Sensor Feedback

The design and experimental validation of a control system with both wireless sensor and actuator links is presented. The control system is designed for, and the experiments are performed on, a laboratory process which consists of a transport belt where moving parts equipped with wireless sensors are heated by four infrared lamps. The process is actuated by moving the transport belt and by switching the heating lamps on or off. The switching property gives interesting hybrid dynamics in the process, which are handled using a hybrid model predictive control.

The chapter is based on the following publications:

- A. Bemporad, S. Di Cairano, E. Henriksson, and K. H. Johansson. Hybrid model predictive control based on wireless sensor feedback: an experimental study. *International Journal of Robust and Nonlinear Control*, 20(2):209–225, 2010a
- A. Bemporad, S. Di Cairano, E. Henriksson, and K. H. Johansson. Hybrid model predictive control based on wireless sensor feedback: an experimental study. In *Proceedings of IEEE Conference on Decision and Control*, New Orleans, LA, USA, 2007
- E. Henriksson. Hybrid model predictive control based on wireless sensor feedback. Master's thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2007

Chapter 7: Conclusions

The thesis is concluded with a brief summary and discussion of the main results. Following that we discuss interesting problems for future research efforts within the area.

Contributions by the Author

The scientific contribution of the thesis is mainly the author's own work. The results presented in Chapter 3 have been derived in cooperation with the author's supervisors. The results presented in Chapter 4 have been derived in cooperation with D. E. Quevedo and the author's supervisors. The results presented in Chapter 5 is joint work between D. Lehmann and the author, under the supervision of K. H. Johansson. The results presented in Chapter 6 are mainly the results of the authors master's thesis together with additional experiments made by S. Di Cairano.

1.6 Notation and Abbreviations

A selection of notation and abbreviations is presented below.

Notation

$\mathbb{R}$	Set of real numbers.
$\mathbb{R}^+$	Set of non-negative real numbers, $\mathbb{R}^+ = \{\, r \mid 0 \leq r,\ r \in \mathbb{R}\,\}$.
$\mathbb{R}^n$	Set of real-valued column-vectors with n elements.
$\mathbb{N}$	Set of natural numbers, $\mathbb{N} = \{0, 1,\ 2,\ 3,\ \ldots\,\}$.
$\mathbb{N}^+$	Set of positive natural numbers, $\mathbb{N}^+ = \{1,\ 2,\ 3,\ \ldots\,\}$.
$\mathbb{Z}$	Set of integers, $\mathbb{Z} = \{\ \ldots\ ,\ -2,\ -1,\ 0,\ 1,\ 2,\ \ldots\,\}$.
$\hat{x}(k\|\ell)$	Estimate of $x(k)$ based on measurements up until time ℓ.
q	Forward shift operator: $qu(k) = u(k+1)$.
$\mathbf{E}x$	Expected value of x.
$\lambda(A)$	Set of eigenvalues of the matrix A.
$0 < Q$	The matrix Q is positive definite: $0 < v^TQv$, $\forall\, v \in \mathbb{R}^n$
$0 \leq Q$	The matrix Q is positive semi-definite: $0 \leq v^TQv$, $\forall\, v \in \mathbb{R}^n$
$\|v\|$	$\|v\| = \sqrt{v^Tv}$, $v \in \mathbb{R}^n$.
$\\|v\\|_Q$	$\\|v\\|_Q = \sqrt{v^TQv}$, $v \in \mathbb{R}^n$.
$\\|x\\|_2$	$\\|x\\|_2 = \sqrt{\sum_{i=-\infty}^{\infty} \|x(i)\|^2}$, $x(i) \in \mathbb{R}^n$.
ℓ_2	Hilbert space of all x such that $\\|x\\|_2 < \infty$.
$\\|H\\|$	If H is a system: $\\|H\\| = \sup_{u\neq 0} \frac{\\|Hu\\|_2}{\\|u\\|_2}$.
	If H is a matrix: $\\|H\\| = \sup_{u\neq 0} \frac{\|Hu\|}{\|u\|}$.
Γ_H	Hankel-operator of the system H.
$\sigma_i(H)$	ith Hankel singular value of H, sorted in decreasing order.

Abbreviations

AAK	Adamjan-Arov-Krein
DAQ	Data Acquisition
PID	Proportional-Integral-Derivative
POC	Predictive Outage Compensator
PRR	Packet Reception Rate
MIMO	Multiple-Input Multiple-Output
MLD	Mixed Logical Dynamical
MPC	Model Predictive Control

Chapter 2

Background

This chapter gives a short background to wireless networked control systems by presenting related work. First an introduction to the area is given, then wireless networks for control are briefly discussed together with their impact on control performance and possible remedies. Then control laws utilizing adaptive inter sampling times are discussed. The chapter is concluded with a discussion on communication-aware control.

2.1 Wireless Networked Control Systems

A wireless networked control system is a control system in which actuators, sensors and controllers are connected through and communicate over a wireless network, as illustrated in Figure 2.1. The interest in these systems is increasing as the introduction of a wireless medium in the control loop enables new system architectures and designs (Samad et al., 2007; Samad and Annaswamy, 2011). However, several problems arise when feedback control measurements are sent over a wireless medium (Willig et al., 2002; Åkerberg et al., 2011; Agrawal et al., 2014).

The wireless network is typically a shared resource where control loops communicating over it may have to co-exist with other applications, such as monitoring

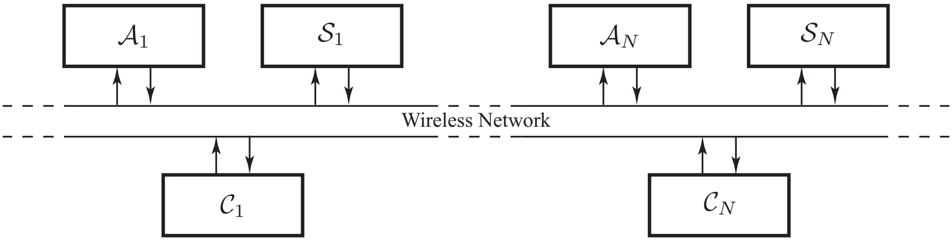


Figure 2.1: Wireless networked control system with actuators $\mathcal{A}$, sensors $\mathcal{S}$ and controllers $\mathcal{C}$ interfaced through the wireless network.

and supervisory systems. This puts requirements on the network as well as the control algorithms, in order to guarantee stability and performance of the overall control system. The need for interaction between control and communication in the design of wireless networked systems was raised in (Kumar, 2001) and has since been very active research area (Bushnell, 2001; Antsaklis and Baillieul, 2004, 2007; Bemporad et al., 2010b).

2.2 Wireless Networks for Control

When information is sent over a wireless channel it is subject to a wide range of imperfections of the network. They are due to variations in radio conditions, because of moving objects, interference, et cetera. Typical scenarios in industrial control settings are reported in (Willig et al., 2002; Agrawal et al., 2014). These imperfections can cause packets sent over the network to be delayed or lost.

The traditional way to compensate for disturbances and variations in the communication channel is to apply various feedback schemes on suitable layers in the communication stack, see textbooks (Tse and Viswanath, 2005; Goldsmith, 2005; Karl and Willig, 2005; Molisch, 2010). Examples of such compensation schemes include power control, *e.g.*, (Chiang et al., 2008), automatic repeat request, *e.g.*, (Molisch, 2010), forward error correcting codes, *e.g.*, (MacWilliams and Sloane, 2003), et cetera. This thesis instead considers how compensation can be made on the application layer, by adapting the control loop. In the networked control systems literature there is an extensive collection of results to overcome the unreliable nature of the network (Bushnell, 2001; Antsaklis and Baillieul, 2004, 2007; Bemporad et al., 2010b). Many results concerning communication packet losses make assumptions on distributions (Sinopoli et al., 2004; Hespanha et al., 2007; Schenato et al., 2007; Gupta and Martins, 2008) or rate limitations (Nair et al., 2007). These models are often hard to validate in practice (Willig et al., 2002). Work to analyze stability of networked control systems under network imperfections has been given much attention, see, *e.g.*, (Heemels et al., 2010; Antunes et al., 2010, 2011, 2012c; Donkers et al., 2012).

Designing protocols suitable for control is another natural way to compensate for unreliable networks as it aims at improving the communication quality instead of compensating for poor communication performance (Al-Karaki and Kamal, 2004; Bachir et al., 2010). Protocol design suitable for control over wireless networks is a large research field of its own and will therefore not be treated further in this thesis. For further reading on protocol design and networking the reader is referred to the book (Karl and Willig, 2005) and papers (Akyildiz and Kasimoglu, 2004; Rozell and Johnson, 2007; Willig, 2008; Ploennigs et al., 2010).

2.3 Compensating for Unreliable Communication

The wireless network medium gives rise to problems due to the unreliability and losses in the communication link. If these artifacts of the network are not properly compensated for, they can have a large impact on the system performance. The approaches to overcome the unreliable nature of the network can be coarsely grouped into two categories: cross-layer methods that explicitly take the network properties into account in the control algorithm and methods that make assumptions about the network but does not utilize any cross-layer signaling. The latter methods enable the use of slightly modified control design methods present in the literature. In (Ling and Lemmon, 2003) the optimal compensation for dropped feedback measurements is posed as a constrained regulator problem. Issues regarding stabilization of systems using smart actuators for a given drop probability are given in (Gupta and Martins, 2008). In (Sinopoli et al., 2004) Kalman filtering under independent and Bernoulli distributed losses is considered, showing how loss probability and system dynamics relate to the expected estimation error covariance. Further, in (Schenato et al., 2007) it is shown that for systems under independent and Bernoulli distributed losses the separation principle hold, provided that successful transmissions are acknowledged. The optimal controller is derived as a linear function of the states and bounds are given on the maximum tolerable loss probability.

The approach taken in Chapter 3 is to add a predictive outage compensator at the receiver side of the network to compensate for the losses therein. An advantage of this approach is that no modifications needs to be made to the existing control structure, facilitating the modular design of networked control systems proposed in (Årzén et al., 2007). Common methods to compensate for packet loss are to hold the last known value or apply an a priori decided constant. A comparison between the two is given in (Schenato, 2009). For work on more advanced compensation schemes, see, *e.g.*, (Gommans et al., 2013). Predictive methods to overcome problems with packet loss have been extensively used in various networked control settings, *e.g.*, (Bemporad, 1998; Quevedo et al., 2008; Scattolini, 2009; Schutter and Scattolini, 2011). In the context of industrial process control, (Heidarinejad et al., 2011) proposes a distributed predictive controller subject to packet losses in the communication. The approach is based on a stabilizing controller which communicates with and coordinates several sub-controllers for increased performance. Packet losses are mitigated by adapting the control actions. In (Liu et al., 2012) the moving horizon estimation problem is investigated under Bernoulli distributed packet losses. In (Kazempour and Ghaisari, 2013) it is proposed to use sensor to sensor communication to overcome random packet losses.

2.4 Adaptive Sampling Strategies

The development of control strategies for wireless automation has become a large area of research in which, up until recently, most efforts have been made under the assumption of periodic communication (Antsaklis and Baillieul, 2007). However, the idea of adaptive sampling is receiving increased attention. The efforts within this area may coarsely be divided into the two paradigms of event- and self-triggered control. In event-triggered control, *e.g.*, (Årzén, 1999; Åström and Bernhardsson, 1999, 2002; Tabuada and Wang, 2006; Tabuada, 2007; Heemels et al., 2008; Wang and Lemmon, 2008, 2011a; Mazo Jr. and Tabuada, 2008, 2009, 2011; Lunze and Lehmann, 2010; Heemels et al., 2013), the sensor continuously monitors the process state and generates a sample when the state violates some predefined condition. Self-triggered control, *e.g.*, (Velasco et al., 2003; Lemmon et al., 2007; Wang and Lemmon, 2009; Anta and Tabuada, 2008, 2009a,b, 2010a; Mazo Jr. and Tabuada, 2008; Mazo Jr. et al., 2009, 2010), uses a model of the system to predict when a new sample needs to be taken in order to fulfill some pre-defined event-condition. A possible advantage of event- over self-triggered control is that the continuous monitoring of the state guarantees that a sample is drawn as soon as the event-condition is violated, thus resulting in an appropriate control action. The self-triggered controller instead operates in open-loop between samples. This could potentially be a problem as disturbances to the process between samples cannot be attenuated. This problem may however be avoided by good choices of the inter sampling times. The possible advantage of self- over event-triggered control is that the transmission time of sensor packets is known a priori and hence we may schedule them, enabling sensors and transmitters to be put to sleep in-between samples and thereby save energy. A control algorithm using self-triggered control is presented in Chapter 4 and an algorithm using event-triggered control is presented in Chapter 5.

2.5 Co-Design of Communication and Control

The research area of joint design of control and communication is currently very active (Hespanha et al., 2007). A view of the convergence of control and communication is given in (Graham and Kumar, 2003) and in (Liu and Goldsmith, 2004) a framework for integrated communication and control design is given. In the context of event-triggered control, (Molin and Hirche, 2009) propose a joint optimization of control and communication, solved using dynamic programming, placing a communication scheduler in the sensor. In (Antunes et al., 2012a,b; Heemels et al., 2013) the control law and event-condition are co-designed to match performance of periodic control using a lower communication rate, in (Heemels and Donkers, 2013) this is extended to decentralized systems.

The use of predictive control in this setting is also gaining popularity (Casavola et al., 2006; Tang and de Silva, 2006; Liu et al., 2006; Zhao et al., 2008; Quevedo and Nešić, 2012). In (Quevedo et al., 2003), predictive methods and vector quantization are used to reduce the controller to actuator communication in multiple input systems. In (Lješnjanin et al., 2014) model predictive control (MPC) is used to design multiple actuator link scheduling and control signals. The idea of using MPC under event-based sampling, as proposed in Chapter 5, is also under development (Muñoz de la Peña and Christofides, 2008; Varutti et al., 2009, 2010; Sijs et al., 2010; Eqtami et al., 2011; Bernardini and Bemporad, 2012), as well as MPC under asynchronous measurements (Liu et al., 2010). In, (Muñoz de la Peña and Christofides, 2008; Varutti et al., 2009, 2010; Eqtami et al., 2011) continuous-time nonlinear systems affected by exogenous disturbances (Varutti et al., 2010; Eqtami et al., 2011) or network delays (Muñoz de la Peña and Christofides, 2008; Varutti et al., 2009) are considered. In these papers the feasibility and convergence of the MPC has been proven by means of a Lyapunov based analysis. The focus of (Sijs et al., 2010) lies in combining an event-triggered state estimator and MPC. It has been shown that the resulting MPC closed-loop system is input-to-state stable with respect to the estimation error.

The problem of joint design of a self-triggering rule and the appropriate control signal using MPC, as addressed in Chapter 4, has been less studied than its event-triggered counterpart. In (Barradas Berglind et al., 2012) an approach relying on an exhaustive search which utilizes sub-optimal solutions giving the control policy and a corresponding self-triggering policy is presented. In (Eqtami et al., 2013) it is suggested that a portion of the open-loop trajectory produced by the MPC should be applied to the process. The time between re-optimizations is then decided via a self-triggering approach.

The concept of minimum-attention control from the seminal work (Brockett, 1997) is also gaining popularity in the networked control systems community (Anta and Tabuada, 2010b; Donkers et al., 2011; Wang and Lemmon, 2011b). Here the aim is to design control laws which maximize time between control actions while guaranteeing certain performance. Likewise has the concept of anytime control gained attention (Greco et al., 2011; Quevedo and Gupta, 2013), here it is suggested that control tasks are executed when computation and communication resources are available.

Chapter 3

Predictive Outage Compensation

The problem considered in this chapter is how to compensate for communication outages in networked control systems by placing devices at the receiver sides of the network. We call these devices predictive outage compensators (POCs), see Figure 3.1. The POCs are designed to compensate for losses in the network, during which sensor data do not reach the controller node or control commands do not reach the actuator. The POC does this by suggesting replacement commands in the event of an outage. The introduction of the POC does not require any modifications to the existing control design.

The general idea is to monitor the received signal and use a signal model to extrapolate the signal, in the event of a communication outage. The POC listens to the received signal. If the signal is received, the POC passes it forward and updates its own internal states using the received data. In the case that no signal is received, the POC uses its internal model to extrapolate the signal based on previously received data. The proposed POC is a generalization of the communication outage compensation algorithms used today, such as holding the last known value

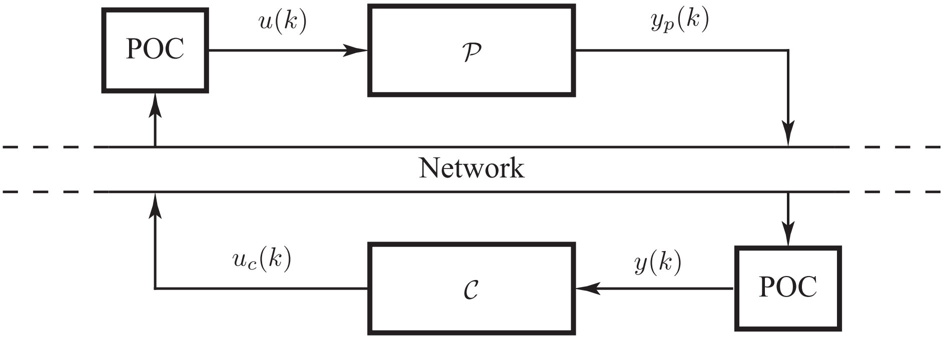


Figure 3.1: Illustration of the POC placement to compensate for communication outages in networked control systems.

or applying constant outputs, and is related to a generalized hold function, *e.g.*, (Sun et al., 1993). Obviously, a POC has a limitation on how efficient it can be for long periods of outages. An important result of the chapter is to build tools to understand how these limitations affect the applicability to real systems.

The main contribution of the chapter is to show that a POC can be based on an optimal outage prediction, under certain assumptions. Theoretical bounds on the prediction errors are derived. An optimal POC can in general be of high order, so a method to derive a low-order implementation based on Hankel-norm approximation is presented. Trade-offs between achievable performance, outage length, and POC order are discussed. Simulations and experiments, performed on a two-tank process controlled over a wireless network, show that both the optimal POC and its reduced-order counterpart can considerably improve the closed-loop control performance under communication outages.

The rest of the chapter is outlined as follows. Section 3.1 details the POC and its operation. In Section 3.2 a stochastically optimal method to synthesize the POC is given, together with prediction error bounds. Section 3.3 contains methods to reduce the complexity of the POC, by means of optimal model-order reduction. Examples on how to synthesize the proposed POC, as well as a simulation study of its performance is given in Section 3.4. The POCs' practical applicability is demonstrated by an experimental implementation and evaluation given in Section 3.5. Finally, the chapter is summarized in Section 3.6.

3.1 Predictive Outage Compensation

The problem we consider is controlling a linear process $\mathcal{P}$ over a communication network with sporadic outages as in Figure 3.1. The process $\mathcal{P}$ is given by

$$\mathcal{P}\begin{cases} x_p(k+1) = A_p x_p(k) + B_u u(k) + B_d d(k) \\ y_p(k) = C_p x_p(k) + v(k), \end{cases}$$

where $d(k)$ is an unknown disturbance and $v(k)$ is measurement noise. The system is controlled by the controller $\mathcal{C}$

$$\mathcal{C}\begin{cases} x_c(k+1) = A_c x_c(k) + B_c e(k) \\ u_c(k) = C_c x_c(k) + D_c e(k) \\ e(k) = r(k) - y(k). \end{cases}$$

where $r(k)$ is the user defined reference on the process output. When there is no communication outage, the applied control is $u(k) = u_c(k)$ and the observation is $y(k) = y_p(k)$. When the communication from the controller to the process is lost and we have an outage, a replacement control command $u(k) = \hat{u}(k)$ is applied by the POC on the process side. If the outage instead happens in the communication from the process to the controller, a replacement observation $y(k) = \hat{y}(k)$ is applied by the POC on the controller side.

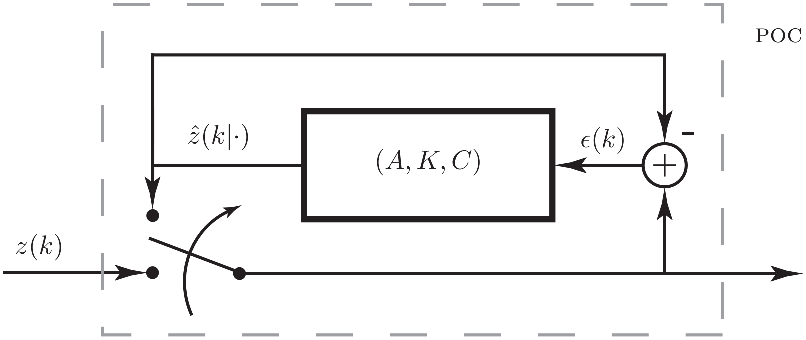


Figure 3.2: At an outage the POC generates predictions of the lost signal.

3.1.1 POC

The POC on the actuator side should choose $\hat{u}(k)$ such that $\Delta u(k) = u_c(k) - \hat{u}(k)$ is as small as possible, and the POC on the controller side should choose $\hat{y}(k)$ such that $\Delta y(k) = y_p(k) - \hat{y}(k)$ is as small as possible. As these two problems are similar, we will from now on consider the transmitted signal, *i.e.*, $y_p(k)$ or $u_c(k)$, to be contained in the signal $z(k)$ together with the reference signal $r(k)$, *e.g.* $z(k) = [u_c^T(k)\ r^T(k)]^T$, with the corresponding estimate denoted by $\hat{z}(k)$.

Let the POC take the state-space form

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + K\epsilon(k) \\ \hat{z}(k|k-1) &= C\hat{x}(k), \end{aligned} \tag{3.1}$$

where $\epsilon(k) = z(k) - \hat{z}(k|k-1) = z(k) - C\hat{x}(k)$ is the one-step-ahead prediction error of the POC, and $\hat{z}(k|k-1)$ is the predicted value of $z(k)$ given measurements up to $k-1$. The matrices (A, K, C) are design parameters, how they should be chosen is discussed further in Section 3.2. A standing assumption in the chapter, is that $A - KC$ is a Schur matrix so that (3.1) is an asymptotically stable system.

The operation of the POC is illustrated in Figure 3.2 where the network is represented by the switch. When the transmitted signal $z(k)$ is received, the prediction error $\epsilon(k) = z(k) - \hat{z}(k|k-1)$ is used to update the POC according to (3.1). When communication is lost, say at time k', we set $\epsilon(k) = 0$ and compute the predicted signal for $k > k'$ as

$$\begin{aligned} \hat{x}(k+1|k') &= A\hat{x}(k|k') \\ \hat{z}(k|k') &= C\hat{x}(k|k'). \end{aligned}$$

3.1.2 Problem Formulation

The proposed POC framework suggest a freedom to choose the realization (A, K, C), as long as $A - KC$ is a Schur matrix. As illustrated by Figure 3.3, there are significant improvements to be made by using a well-designed POC compared to a more naive approach. Good design choices are those making $\Delta z(k) = z(k) - \hat{z}(k)$ small. In this chapter, we show how to choose a POC to minimize $\mathbf{E}|\Delta z(k)|^2$. As the optimal POC may be of high order, we develop methods to reduce its model order, while at the same time taking its performance into account. This may be important if the POC is to be implemented in embedded systems linked to the controller or actuator.

3.2 Synthesis

Let us now consider a stochastic method on how to find optimal (A, K, C) in (3.1) for multiple-input multiple-output (MIMO) systems affected by stochastic disturbances and measurement noise. To characterize the optimal (A, K, C), let us assume that $d(k)$ is colored noise given by

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_w w(k) \\ d(k) &= C_d x_d(k) + D_w w(k), \end{aligned} \tag{3.2}$$

$r(k)$ is colored noise given by

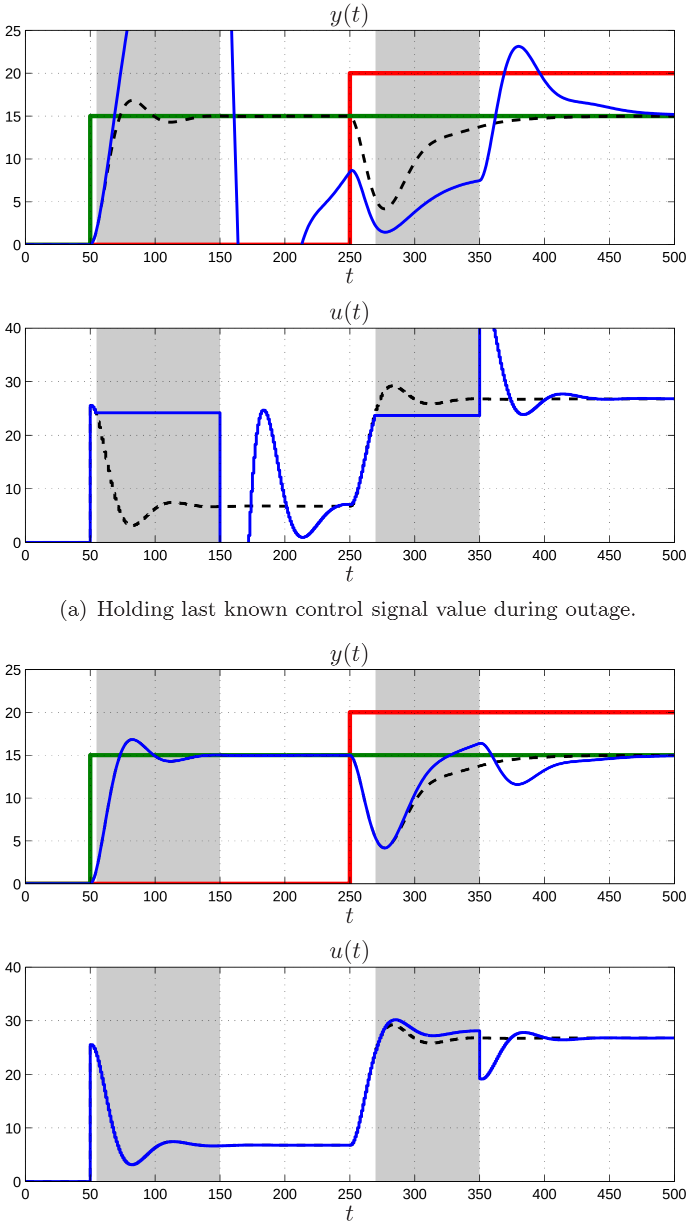
$$\begin{aligned} x_\rho(k+1) &= A_\rho x_\rho(k) + B_\rho \rho(k) \\ r(k) &= C_\rho x_\rho(k) + D_\rho \rho(k), \end{aligned} \tag{3.3}$$

and that the driving signals $w(k)$ and $\rho(k)$, as well as the measurement noise $v(k)$, are white Gaussian stochastic processes with

$$\mathbf{E}\begin{bmatrix} w(k) \\ v(k) \\ \rho(k) \end{bmatrix} = 0, \quad \mathbf{E}\begin{bmatrix} w(k) \\ v(k) \\ \rho(k) \end{bmatrix}\begin{bmatrix} w(l) \\ v(l) \\ \rho(l) \end{bmatrix}^T = \begin{bmatrix} R_w & R_{wv} & R_{w\rho} \\ R_{wv}^T & R_v & R_{v\rho} \\ R_{w\rho}^T & R_{v\rho}^T & R_\rho \end{bmatrix}\delta_{kl}.$$

When there is no outage, the entire closed-loop system evolves as

$$\begin{aligned} \begin{bmatrix} x_p(k+1) \\ x_c(k+1) \\ x_d(k+1) \\ x_\rho(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} A_p - B_u D_c C_p & B_u C_c & B_d C_d & B_u D_c C_\rho \\ -B_c C_p & A_c & 0 & B_c C_\rho \\ 0 & 0 & A_d & 0 \\ 0 & 0 & 0 & A_\rho \end{bmatrix}}_{A_{cl}} \underbrace{\begin{bmatrix} x_p(k) \\ x_c(k) \\ x_d(k) \\ x_\rho(k) \end{bmatrix}}_{x(k)} \\ &+ \underbrace{\begin{bmatrix} B_d D_w & -B_u D_c & B_u D_c D_\rho \\ 0 & -B_c & B_c D_\rho \\ B_w & 0 & 0 \\ 0 & 0 & B_\rho \end{bmatrix}}_{N} \begin{bmatrix} w(k) \\ v(k) \\ \rho(k) \end{bmatrix} \end{aligned} \tag{3.4a}$$



(a) Holding last known control signal value during outage.

(b) Extrapolating control signal using a well-designed POC.

Figure 3.3: Illustration of POC and hold behavior (blue) with nominal behavior (black) under disturbance d (red), reference r (green) and outage (grey area).

$$\underbrace{\begin{bmatrix} y(k) \\ u(k) \\ r(k) \end{bmatrix}}_{z(k)} = \underbrace{\begin{bmatrix} C_p & 0 & 0 & 0 \\ -D_c C_p & C_c & 0 & D_c C_\rho \\ 0 & 0 & 0 & C_\rho \end{bmatrix}}_{C_{cl}} \begin{bmatrix} x_p(k) \\ x_c(k) \\ x_d(k) \\ x_\rho(k) \end{bmatrix} + \underbrace{\begin{bmatrix} v(k) \\ -D_c v(k) + D_c D_\rho \rho(k) \\ D_\rho \rho(k) \end{bmatrix}}_{n(k)}. \tag{3.4b}$$

The steady-state optimal estimator of the state $x(k)$ in (3.4) using measurements $z(k)$ is the Kalman filter:

$$\hat{x}(k+1|k) = A_{cl}\hat{x}(k|k-1) + K_{cl}\Big[z(k) - C_{cl}\hat{x}(k|k-1)\Big], \tag{3.5}$$

where

$$\begin{aligned} K_{cl} &= \big(A_{cl}PC_{cl}^T + NR_{12}\big)\big(C_{cl}PC_{cl}^T + R_2\big)^{-1} \\ P &= A_{cl}PA_{cl}^T + NR_1N^T \\ &\quad - \big(A_{cl}PC_{cl}^T + NR_{12}\big)\big(C_{cl}PC_{cl}^T + R_2\big)^{-1}\big(A_{cl}PC_{cl}^T + NR_{12}\big)^T, \end{aligned} \tag{3.6}$$

with

$$R_1 = \mathbf{E}\begin{bmatrix} w(k) \\ v(k) \\ \rho(k) \end{bmatrix}\begin{bmatrix} w(k) \\ v(k) \\ \rho(k) \end{bmatrix}^T = \begin{bmatrix} R_w & R_{wv} & R_{w\rho} \\ R_{wv}^T & R_v & R_{v\rho} \\ R_{w\rho}^T & R_{v\rho}^T & R_\rho \end{bmatrix},$$

$$R_2 = \mathbf{E}\Big[\ n(k)\ \Big]\Big[\ n(k)\ \Big]^T = R_n$$

and

$$R_{12} = \mathbf{E}\begin{bmatrix} w(k) \\ v(k) \\ \rho(k) \end{bmatrix}\Big[\ n(k)\ \Big]^T = \begin{bmatrix} R_{wn} \\ R_{vn} \\ R_{\rho n} \end{bmatrix},$$

see (Anderson and Moore, 2005). The optimal one-step-ahead prediction of $z(k)$ is $\hat{z}(k|k-1) = C_{cl}\hat{x}(k|k-1)$. Note that the Kalman filter (3.5) has the structure of the POC (3.1), and that optimal predictions of $z(k)$ based on measurements up until $k' \leqslant k$ are generated by

$$\begin{aligned} \hat{x}(k+1|k') &= A_{cl}\hat{x}(k|k') \\ \hat{z}(k|k') &= C_{cl}\hat{x}(k|k'), \end{aligned}$$

where the prediction $\hat{x}(k'+1|k')$ is given by (3.5). We summarize this derivation in the following proposition.

Proposition 3.2.1. *The* POC *minimizing* $\mathbf{E}|\Delta z(k)|^2$ *is given by (3.1) with*

$$A = A_{cl}, \quad K = K_{cl}, \quad C = C_{cl}.$$

Remark 3.2.2. Note that we consider the problem over an infinite time horizon. This is a good assumption if the communication outages are infrequent and not too short. If the outages are frequent, this assumption is not valid and the optimal filter gain K_{cl} should be time varying, *cf.*, (Anderson and Moore, 2005).

It is easy to characterize the statistics of the prediction error $\Delta z(k)$ of the POC synthesized using this method. The Kalman filter gives unbiased estimates, and thus $\mathbf{E}\Delta z(k) = 0$ for all $k > k'$. To compute the variance $\mathbf{E}|\Delta z(k)|^2$, we need the covariance of the state estimation error. Assuming that the Kalman filter has been in operation for a long time before the outage at k', the covariance of $\Delta x(k)$ is given by the solution to the Riccati equation in (3.6):

$$\mathbf{E}\Delta x(k)\Delta x(k)^T = P,$$

where $\Delta x(k) = x(k) - \hat{x}(k|k-1)$. The variance of the one-step-ahead prediction error is $\mathbf{E}|\epsilon(k)|^2 = C_{cl}PC_{cl}^T + R_2$. When an outage occurs, the covariance of the state estimation and prediction error evolve for $k > k'$ as

$$\begin{aligned} P(k+1) &= AP(k)A^T + NR_1N^T, \quad P(k') = P, \\ \mathbf{E}|\Delta z(k)|^2 &= C_{cl}P(k)C_{cl}^T + R_2. \end{aligned}$$

How much effect this prediction error has on the process depends on its dynamics. If $\mathcal{P}$ is an unstable process, even a small error $\Delta z(k)$ can harm the process since it is in open-loop during outage.

3.3 Complexity Analysis and Reduction

The optimal POC is given by a filter of order equal to the sum of the process order, the controller order, the disturbance model order and the reference model order, as shown in previous section. In practice it is often desirable to have a low-order POC, so that it can be implemented through embedded software in the actuator or controller, *cf.*, Figure 3.1. It is thus important to know if there exists a POC of low order with similar performance as the optimal one. Here, model order reduction using the Hankel-norm is shown to be a suitable mathematical tool.

Let ℓ_2 denote the Hilbert space of square-summable signals, *i.e.*, signals u with finite norm $\|u\|_2 := \sqrt{\sum_{i=-\infty}^{\infty} |u(i)|^2}$. Then let us represent the POC by a linear operator $\hat{z} = H\epsilon$ on ℓ_2, realized by

$$H \begin{cases} \hat{x}(k+1) = A\hat{x}(k) + K\epsilon(k), \\ \hat{z}(k|k-1) = C\hat{x}(k), \ \hat{x}(k) \in \mathbb{R}^n, \end{cases} \tag{3.7}$$

where $\epsilon(k) = z(k) - \hat{z}(k|k-1)$. Let us denote a reduced-order POC by a linear operator $\hat{z}_r = H_r \epsilon_r$ on ℓ_2, with order $r < n$, realized by

$$H_r \begin{cases} \hat{x}_r(k+1) = A_r \hat{x}_r(k) + K_r \epsilon_r(k), \\ \hat{z}_r(k|k-1) = C_r \hat{x}_r(k), \;\; \hat{x}_r(k) \in \mathbb{R}^r, \end{cases} \tag{3.8}$$

where $\epsilon_r(k) = z(k) - \hat{z}_r(k|k-1)$.

3.3.1 Hankel-norm Approximation

Assume that an outage occurs at $k' = 0$ and that it has a nontrivial duration $\ell > 0$. The reduced-order POC should produce an outage prediction $\hat{z}_r(k|k')$ that is close to $\hat{z}(k|k')$. Since the predictions only will be applied to the system from k' to $k' + \ell$, we are only interested in making the difference small for $k' < k \leq k' + \ell$. Introducing the time-projection operator P_+ as

$$\begin{aligned} P_+ z &= P_+(\ldots, z(2), z(1), z(0), z(-1), z(-2), \ldots) \\ &= (\ldots, z(2), z(1), 0, 0, 0, \ldots), \end{aligned}$$

we formalize this requirement using the operator notation in (3.7) and (3.8) as making

$$\|P_+ \hat{z} - P_+ \hat{z}_r\|_2 = \|P_+ H \epsilon - P_+ H_r \epsilon_r\|_2 \tag{3.9}$$

small.

Recall that when the POC is in feedback, *i.e.*, there is no outage, the prediction error is given by $\epsilon = z - \hat{z} = z - H\epsilon$ which, re-arranged, becomes $\epsilon = (I + H)^{-1} z$. To get a closed-form expression for ϵ we introduce another time-projection operator P_- as

$$\begin{aligned} P_- \epsilon &= P_-(\ldots, \epsilon(2), \epsilon(1), \epsilon(0), \epsilon(-1), \epsilon(-2), \ldots) \\ &= (\ldots, 0, 0, \epsilon(0), \epsilon(-1), \epsilon(-2), \ldots), \end{aligned}$$

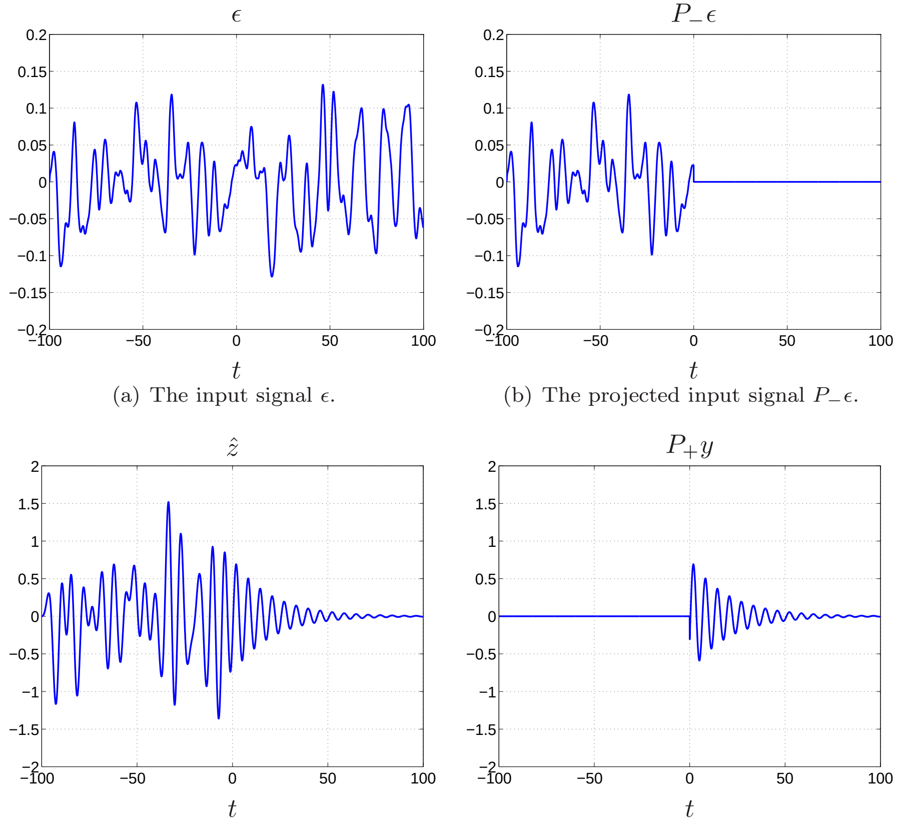
which we can use to write $\epsilon = P_-(I + H)^{-1} z$. Since the same argument holds for the reduced-order POC we can now write (3.9) as

$$\|P_+ H \epsilon - P_+ H_r \epsilon_r\|_2 = \|P_+ H P_- (I + H)^{-1} z - P_+ H_r P_- (I + H_r)^{-1} z\|_2. \tag{3.10}$$

We note that the operator $P_+ H P_-$ is the Hankel-operator Γ_H of H (Zhou et al., 1996), *i.e.*, the past-input to future-output restriction of the map of H, which is illustrated in Figure 3.4.

Since we want to make (3.10) small for any input sequence $z(k)$, we consider

$$\sup_{z \neq 0} \frac{\|\Gamma_H (I + H)^{-1} z - \Gamma_{H_r} (I + H_r)^{-1} z\|_2}{\|z\|_2} = \|\Gamma_H (I + H)^{-1} - \Gamma_{H_r} (I + H_r)^{-1} \;\|, \tag{3.11}$$



(a) The input signal ϵ.

(b) The projected input signal $P_-\epsilon$.

(c) The output $\hat{z}$ of the system H when driven by the control signal $P_-\epsilon$, *i.e.*, $\hat{z} = HP_-\epsilon$.

(d) The projected output $P_+\hat{z}$ of the system H when driven by the control signal $P_-\epsilon$, *i.e.*, $P_+\hat{z} = P_+HP_-\epsilon = \Gamma_H\epsilon$.

Figure 3.4: Illustration of the input-output map of the Hankel-operator Γ_H.

where $\|\cdot\|$ is the induced ℓ_2-norm.

Model-order reduction (Obinata and Anderson, 2001) cannot be immediately applied to (3.11) since the operators Γ_H and Γ_{H_r} are weighted by $(I+H)^{-1}$ and $(I+H_r)^{-1}$, respectively. Instead, we note that the following upper bound holds:

$$\begin{aligned}
&\|\Gamma_H(I+H)^{-1} - \Gamma_{H_r}(I+H_r)^{-1}\| \\
&= \|\Gamma_H(I+H_r)^{-1} - \Gamma_{H_r}(I+H_r)^{-1} + \Gamma_H(I+H)^{-1} - \Gamma_H(I+H_r)^{-1}\| \qquad (3.12) \\
&\leq \|(\Gamma_H - \Gamma_{H_r})(I+H_r)^{-1}\| + \|\Gamma_H\big((I+H)^{-1} - (I+H_r)^{-1}\big)\|.
\end{aligned}$$

We proceed by minimizing the first term of this bound. It turns out, that it is possible to use that solution to bound also the second term, and thus to bound the error criterion (3.11). Note that the rank of the Hankel-operator is equal to the

McMillan degree of the corresponding system, *i.e.*, rank $\Gamma_H = n$ if (A, K, C) is a minimal realization. To make the first term of the upper bound (3.12) small, we propose to solve the problem

$$\min_{\operatorname{rank}\Gamma_{H_r}\leq r}\ \sup_{\|\epsilon_r\|_2\leq 1} \|(\Gamma_H - \Gamma_{H_r})\epsilon_r\|_2 =: \gamma_1(r),$$

where $\epsilon_r = (I + H_r)^{-1}z$. This can be solved using the AAK-lemma (Adamjan et al., 1971; Glover, 1984). In particular, it is well-known that

$$\gamma_1(r) = \sigma_{r+1}(H),$$

where $\sigma_i(H)$, $i = 1, \ldots, n$, are the Hankel singular values of the linear operator H, sorted in decreasing order. The Hankel singular values can be used to determine a suitable approximation order r. Methods for computing a state-space realization (A_r, K_r, C_r) of the optimal H_r^* are available, see (Glover, 1984; Gu, 2005).

Assume now that we choose an optimal Hankel-norm approximation H_r^* of H as the reduced-order POC. What can we then say about the size of the second term of the bound (3.12)? We have that

$$(I + H)^{-1} - (I + H_r)^{-1} = (I + H)^{-1}(H_r - H)(I + H_r)^{-1},$$

and as has been shown in (Glover, 1984; Gu, 2005), there is an optimal Hankel-norm approximation H_r^* such that

$$\|H_r^* - H\| \leq \sum_{i=r+1}^{n} \sigma_i(H).$$

An upper estimate of the second term in (3.12) is therefore

$$\left\|\Gamma_H\big((I + H)^{-1} - (I + H_r)^{-1}\big)z\right\|_2 = \|\Gamma_H(I + H)^{-1}(H_r - H)\epsilon_r\|_2 \leq \gamma_2(r)\|\epsilon_r\|_2,$$

where

$$\gamma_2(r) = \sigma_1(H)\,\|(I + H)^{-1}\| \sum_{i=r+1}^{n} \sigma_i(H).$$

Here we have used that the induced norm of Γ_H is equal to $\sigma_1(H)$.

We summarize the above derivations in the following proposition.

Proposition 3.3.1. *Suppose the* POC H *in (3.7) is stable and choose the reduced-order* POC H_r *in (3.8) to be the optimal Hankel-norm approximation* H_r^* *of* H*. Then during an outage the difference between* $\hat{z}$ *and* $\hat{z}_r$ *is bounded by*

$$\|P_+(\hat{z} - \hat{z}_r)\|_2 = \|\Gamma_H\epsilon - \Gamma_{H_r^*}\epsilon_r\|_2 \leq \gamma(r)\|\epsilon_r\|_2,$$

for any input $z \in \ell_2$ *where*

$$\gamma(r) = \gamma_1(r) + \gamma_2(r) = \sigma_{r+1}(H) + \sigma_1(H)\|(I + H)^{-1}\| \sum_{i=r+1}^{n} \sigma_i(H).$$

Remark 3.3.2. Proposition 3.3.1 shows that if $\sigma_i(H)$, $i = r+1, \ldots, n$, are small, then H_r^* is guaranteed to work well as a reduced-order POC. The bound can be used as follows: A user of the reduced-order POC can compute $\|\epsilon_r\|_2$, since this is the energy of the one-step ahead prediction error and reference deviations, which are fed into H_r^*. If $\|\epsilon_r\|_2$ is small, it means that the prediction is accurate. If then an outage occurs, we can be certain that the outage predictions $\hat{z}_r$ do not deviate from the full-order prediction $\hat{z}$ more than $\gamma(r)\|\epsilon_r\|_2$.

Remark 3.3.3. One restriction in Proposition 3.3.1 is that H must be stable, which would not be the case if the POC has been derived using the optimal, method with unstable modes in the disturbance model (3.2) or in the reference model (3.3). The unstable case can be handled by making a stable – anti-stable decomposition of $H = H_s + H_u$, and then approximate the stable part H_s as above. The unstable term H_u can then be added to the approximation $H_{s,r}^*$.

Remark 3.3.4. A POC holding the last received signal as a one-step-ahead prediction is in fact a reduced-order H_r realized by

$$A_r = 1, \; K_r = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; C_r = 1.$$

3.3.2 Time-Weighted Hankel-norm Approximation

In the above analysis, we try to make $\|P_+(\hat{z} - \hat{z}_r)\|_2$ small, *i.e.*, we try to make the total energy of the reduction error during an outage small. Since the outages are assumed to be of finite but unknown length, we are mostly interested in keeping the error small during the first part of the outage. To achieve this we may add an exponential decay weight, λ^k, $0 < \lambda \leq 1$, in the minimization criterion. We minimize $\|P_+(\hat{z}^\lambda - \hat{z}_r^\lambda)\|_2$ where $\hat{z}^\lambda(k) = \lambda^k \hat{z}(k)$ and $\hat{z}_r^\lambda(k) = \lambda^k \hat{z}_r(k)$. In order to do this we must find the operator H^λ and input signal ϵ^λ generating $\hat{z}^\lambda = H^\lambda \epsilon^\lambda$ from

$$H \begin{cases} \hat{x}(k+1) = A\hat{x}(k) + K\epsilon(k) \\ \hat{z}(k|k-1) = C\hat{x}(k). \end{cases}$$

We do this by defining the new states $\hat{x}^\lambda(k) = \lambda^k \hat{x}(k)$, the new input $\epsilon^\lambda(k) = \lambda^k \epsilon(k)$ and inserting them into H. This gives us

$$H^\lambda \begin{cases} \hat{x}^\lambda(k+1) = \lambda A \hat{x}^\lambda(k) + \lambda K \epsilon^\lambda(k) \\ \hat{z}^\lambda(k) = C\hat{x}^\lambda(k). \end{cases}$$

We may now find H_r^λ making $\|P_+(\hat{z}^\lambda - \hat{z}_r^\lambda)\|_2 = \|P_+(H^\lambda P_- \epsilon^\lambda - H_r^\lambda P_- \epsilon_r^\lambda)\|_2$ small, as suggested in Proposition 3.3.1. This gives us the reduced-order system

$$H_r^\lambda \begin{cases} \hat{x}_r^\lambda(k+1) = A_r^\lambda \hat{x}_r^\lambda(k) + K_r^\lambda \epsilon_r^\lambda(k) \\ \hat{z}_r^\lambda(k) = C_r \hat{x}_r^\lambda(k). \end{cases}$$

or equivalently

$$H_r \begin{cases} \hat{x}_r(k+1) = \dfrac{1}{\lambda} A_r^\lambda \hat{x}_r(k) + \dfrac{1}{\lambda} K_r^\lambda \epsilon_r(k) \\ \hat{z}_r(k) = C_r \hat{x}_r(k). \end{cases}$$

Remark 3.3.5. Clearly the choice of λ will affect the performance of the reduced-order system. One could think of several ways of choosing λ but a simple heuristic is to choose $\lambda = e^{-1/k_{\max}}$, so the weight is equal to e^{-1} after $k_{\max}$ steps into the outage.

Remark 3.3.6. The choice of λ may affect the stability. If λ is chosen small enough, H^λ is asymptotically stable independent of the poles of H. Even though H_r^λ is stable if H^λ is stable, H_r may be open-loop unstable if λ is chosen too small.

3.4 Simulation Evaluation

To illustrate the POC we perform a simulation study where the POC is part of a process control setup. It is shown how to construct a POC using the methodology developed in Section 3.2 and how to derive a reduced-order implementation using the method presented in Section 3.3. The POC is simulated on a scenario, comparing it with nominal closed-loop behavior and the method to compensate for outages by holding the last received signal.

3.4.1 Simulation Setup

The process to be controlled is the tank process depicted in Figure 3.5, consisting of two identical tanks connected in series. The control objective is to keep the level x_2 in the lowest tank around a reference trajectory despite load disturbances d entering the system. The manipulated variable is the voltage u to the pump.

The process corresponds to the real system considered in the next section. The individual tanks are modelled using mass balance and Bernoulli's law. The tanks have cross sectional area $A = 15.5\,\text{cm}^2$, outlet hole area $a = 0.13\,\text{cm}^2$ and the gravitational acceleration is $g = 9.8\,\text{m/s}^2$. A scaled linearized process model around the equilibrium $x_i^0 = 10.7\,\text{cm}$ and $u^0 = 5.9\,\text{V}$ is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} K_s \\ 0 \end{bmatrix} \frac{u_{\max}}{100} \Big(u - d \Big)$$

$$y = \begin{bmatrix} 0 & 100/h_{\max} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v$$

where $\tau = \frac{A}{a}\sqrt{2x_i^0/g} = 18.1\,\text{s}$, $K_s = 0.24\,\text{cm/Vs}$, $h_{\max} = 30\,\text{cm}$ and $u_{\max} = 15\,\text{V}$. The process is sampled with period $T_s = 1\,\text{s}$ and controlled with the discrete-time

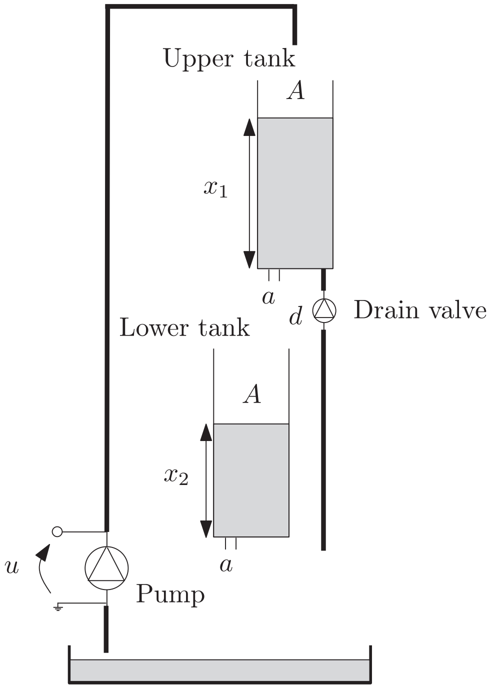


Figure 3.5: The controlled tank process $\mathcal{P}$.

PID-controller $\mathcal{C}$ given by

$$C(z) = K_P + K_I \frac{1}{z-1} + K_D \frac{z-1}{T_f(z-1)+1},$$
$$K_P = 1.33,\ K_I = 0.04,\ K_D = 4.58,\ T_f = 12.26.$$

The closed-loop system has the nominal performance shown in Figure 3.6. At time $t = 50$ s a step is made in the reference from $r = 0$ to $r = 15$. At time $t = 250$ s a step disturbance $d = 20$ occurs, simulating opening the drainage valve in the upper tank. The controller is efficient both in tracking the reference, as well as attenuating the disturbance. The question is then how the system behaves if communication between the controller and the actuator is lost.

A common way to compensate for outages is to use a hold function, *i.e.*, using the last known value of the control signal $u_c(k)$ as the one-step-ahead prediction. Simulating the system using this predictor under the same scenario as above, with the addition of two communication outages between the controller and actuator, one between $t = 55$ s and $t = 100$ s and one between $t = 270$ s and $t = 350$ s, we get the response shown in Figure 3.7. The tracking is in this case poor, compared to the nominal closed-loop behavior.

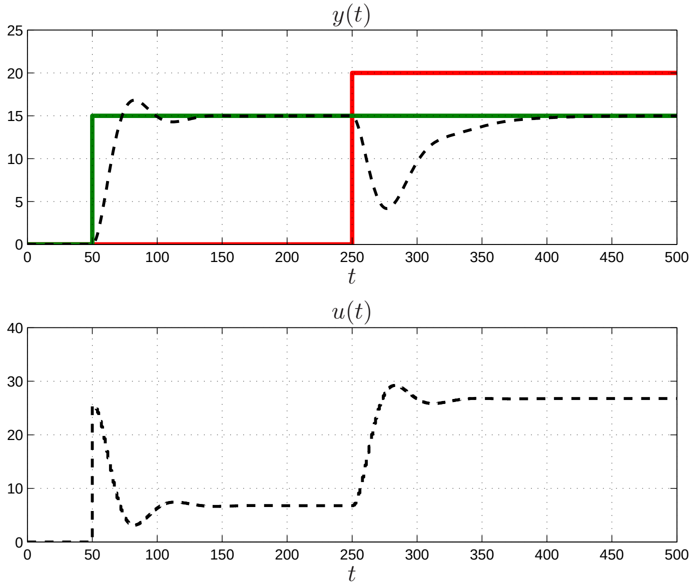


Figure 3.6: Nominal closed-loop behavior (black) under disturbance d (red) and reference r (green).

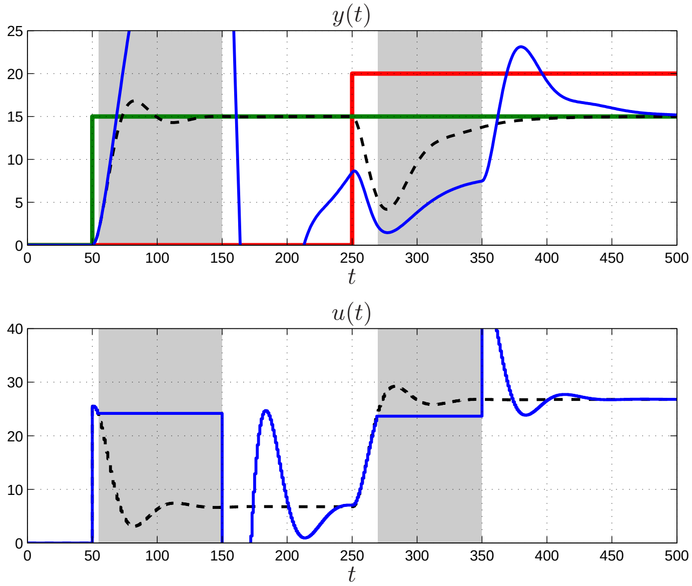


Figure 3.7: Comparison of hold behavior (blue) with nominal behavior (black) under disturbance d (red), reference r (green) and outage (grey area).

3.4.2 Optimal POC

To synthesize an optimal POC, we need stochastic disturbance and reference models that describe d and r well enough. For the current scenario, the disturbance model

$$\begin{aligned} x_d(k+1) &= x_d(k) + w(k) \\ d(k) &= x_d(k) \\ \mathbf{E}w(k)^2 &= 14.9 \end{aligned} \tag{3.13}$$

and reference model

$$\begin{aligned} x_\rho(k+1) &= x_\rho(k) + \rho(k) \\ r(k) &= x_\rho(k) \\ \mathbf{E}\rho(k)^2 &= 1. \end{aligned} \tag{3.14}$$

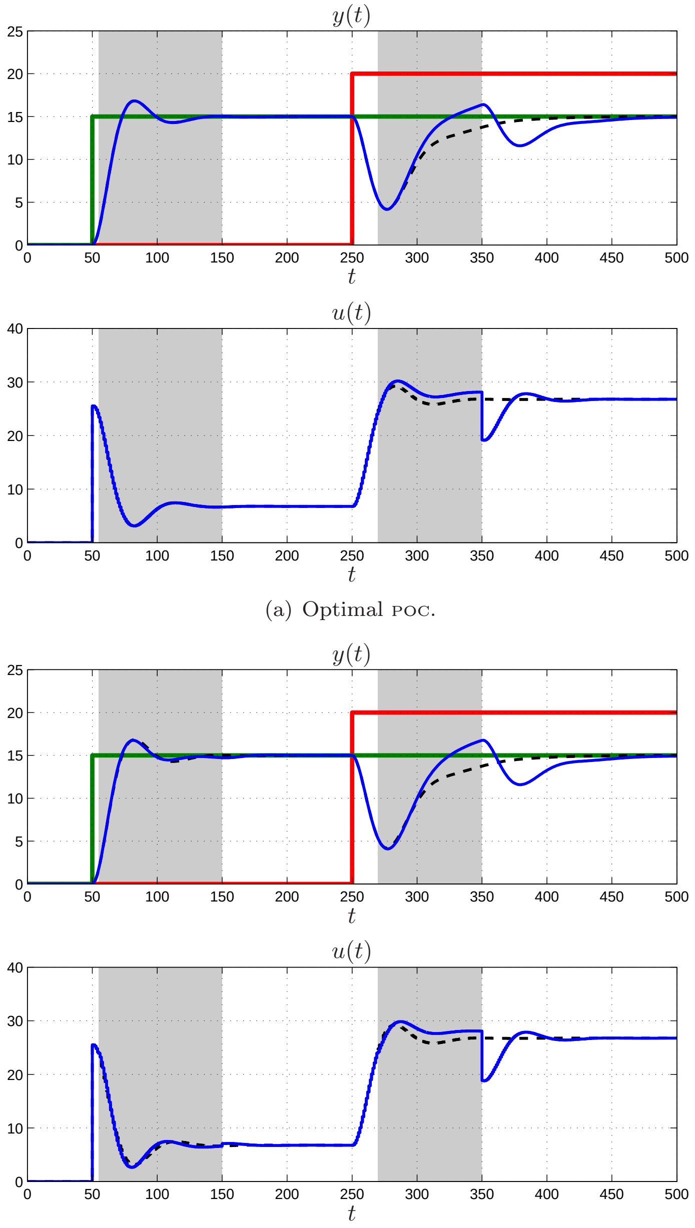
are used. We may now combine the models for the process, controller, disturbance and reference as in (3.4), getting a closed-loop system with McMillan degree $n = n_p + n_c + n_d + n_\rho = 6$. The corresponding optimal POC H when the POC is placed in the actuator is given by (3.5) with

$$C_{cl} = \begin{bmatrix} -D_cC_p & C_c & 0 & D_cC_\rho \\ 0 & 0 & 0 & C_\rho \end{bmatrix}, \; R_1 = \begin{bmatrix} R_w & 0 & 0 \\ 0 & R_v & 0 \\ 0 & 0 & R_\rho \end{bmatrix},$$

$$R_2 = F \begin{bmatrix} R_v & 0 \\ 0 & R_\rho \end{bmatrix} F^T, \; R_{12} = \begin{bmatrix} 0 & 0 \\ R_v & 0 \\ 0 & R_\rho \end{bmatrix} F, \; F = \begin{bmatrix} -D_c & D_cD_\rho \\ 0 & D_\rho \end{bmatrix},$$

$$R_w = 14.1, \; R_v = 4.4, \; R_\rho = 1.$$

Simulating the system with the POC H one get the behavior shown in Figure 3.8(a). At time $t = 0\,\mathrm{s}$ the optimal POC is initialized to the same state as the true system and the prediction is perfect. When the reference changes, the system states start to diverge, as a result so does the estimation error. Effectively what now happens is that the filter in the POC starts to estimate the reference r via the internal model. When the outage occurs, the estimate has clearly almost fully converged as the prediction is close to perfect. When the disturbance acts on the system, the POC has to estimate that as well. Then when communication between the controller and actuator is lost, the POC starts to evolve in open-loop, predicting control signals. If the state estimate has converged before the outage the prediction will be perfect, as long as the disturbance does not change under the outage. However, if the estimate has not fully converged, as is the case in this example, the prediction starts to diverge. Still, one can observe that the prediction error is small, resulting in a small deviation in the output compared to the nominal case.



(a) Optimal POC.

(b) Reduced-order POC with $r = 2$.

Figure 3.8: Comparison of POC behavior (blue) with nominal behavior (black) under disturbance d (red), reference r (green) and outage (grey area).

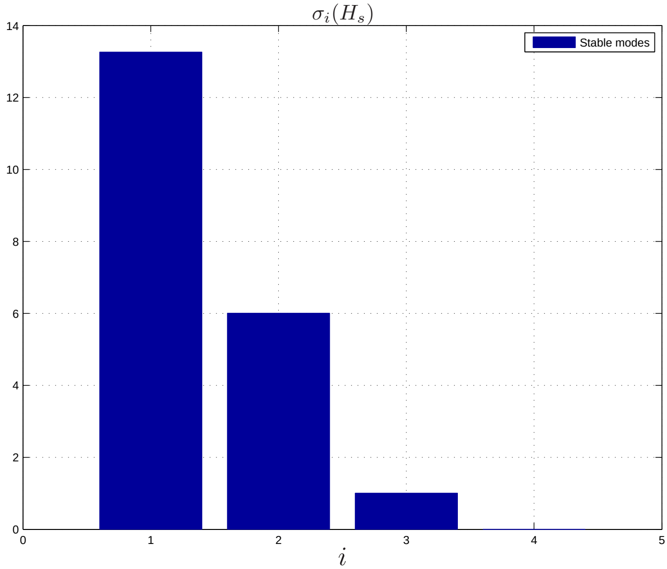


Figure 3.9: Hankel singular values of the stable modes of the POC.

3.4.3 Hankel-norm Approximation

It is desirable to have a POC of low order and therefore a reduced-order approximation of the optimal POC H is desired. First it is observed that the disturbance model (3.13) as well as the reference model (3.14) contains integrations, so H is not asymptotically stable. To handle this a stable – anti-stable decomposition of H is made as $H = H_s + H_u$ and reduction is made on the stable part H_s only, see Remark 3.3.3. To determine a proper reduction order r, the singular values of H_s, shown in Figure 3.9, are studied. There is a significant drop between $\sigma_2(H_s)$ and $\sigma_3(H_s)$, indicating that a good choice of the reduction order r is to choose $r = 2$. Performing optimal Hankel-norm approximation on H_s of order r one gets $H^*_{s,r}$ and the reduced-order POC as $H^*_r = H^*_{s,r} + H_u$ of order $r + 2$ since H_u contains the integrator states from the disturbance model and the reference model.

Evaluating the reduced-order POC on the same simulation scenario as before, one get the result in Figure 3.8(b). The states associated with the reference model accommodate the errors due to the model reduction. When communication is lost the first time, the predicted control signal diverges slightly from the nominal value. When communication is lost the second time, the prediction error is as expected larger than for the optimal POC, which is due both to the model approximation error and that the estimator has not fully converged when the outage occurs. However, the output tracking performance for the reduced-order POC is almost identical to the optimal POC.

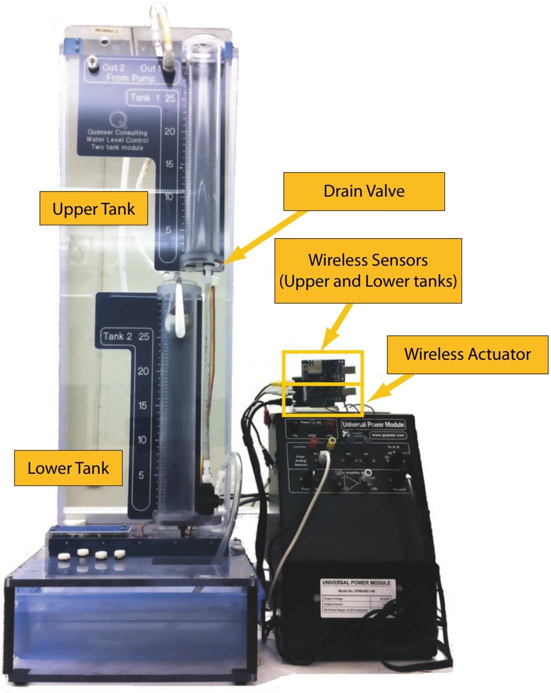


Figure 3.10: The controlled tank process $\mathcal{P}$ from Quanser.

3.5 Experimental Evaluation

In this section, we study the practical applicability of the POC via experimental evaluations. The physical process is shown in Figure 3.10 and schematically described in Figure 3.5. The pressure sensors measuring the tank levels are wired to a wireless device which converts the measurements into tank levels and transmits them to the remote controller. The controller then computes the appropriate control action and in turn transmits it to the wireless actuator. The receiver in the actuator converts this command into an actual voltage applied to the pump.

3.5.1 Experimental Setup

For the experiments we, as in Section 3.4, place the POC between the controller and actuator. To control the process we use the controller

$$\begin{aligned} x_c(k+1) &= \begin{bmatrix} 1 & 0 \\ 0 & 0.75 \end{bmatrix} x_c(k) + \begin{bmatrix} 1 \\ -0.25 \end{bmatrix} e(k) \\ u(k) &= \begin{bmatrix} 0.03 & 0.26 \end{bmatrix} x_c(k) + 1.65 \cdot e(k) \\ e(k) &= r(k) - y(k). \end{aligned}$$

The POC is then synthesized based on this controller, in the way described in Section 3.4.2, using the same models for process, reference and disturbance.

The scenario for the experiments is as follows:

- At $t = 0\,\mathrm{s}$ the experiments starts with $r = 40$.
- At $t = 100\,\mathrm{s}$ there is a step change in the reference to $r = 53$.
- At $t = 230\,\mathrm{s}$ a drainage valve in the upper tank is opened, allowing water to flow directly into the reservoir, thus causing a disturbance.
- At $t = 280\,\mathrm{s}$ the drainage valve is closed.
- At $t = 500\,\mathrm{s}$ the experiment is ended.

Running the above scenario on our process without any outage we get the response shown in Figure 3.11. To examine how the system performs under outage we study two different outage scenarios, one during the reference step and one during the disturbance step. For both these cases we compare the behavior of our proposed POC and a hold compensator.

3.5.2 Results

In our first two experiments we study the effect of an outage during the reference step. The outage occurs at $t = 105\,\mathrm{s}$, *i.e.*, very short after the reference change, and lasts for 100 s. The result is shown in Figure 3.12. It is evident that the optimal POC is superior to the hold function, as it is able to predict the lost signal for a substantial outage.

For our second two experiments, we study the effect of an outage during a step disturbance. Here the outage occurs at $t = 250\,\mathrm{s}$ and lasts for 100 s. The result of these experiments are shown in Figure 3.13. As seen the performance of the hold function and the optimal POC is very similar. This is due to several factors. First of all, the outage occurs shortly after the disturbance starts to act on the system, resulting in that the POC estimate has not yet converged. This explains the initial prediction error by the POC. Secondly the disturbance changes during the

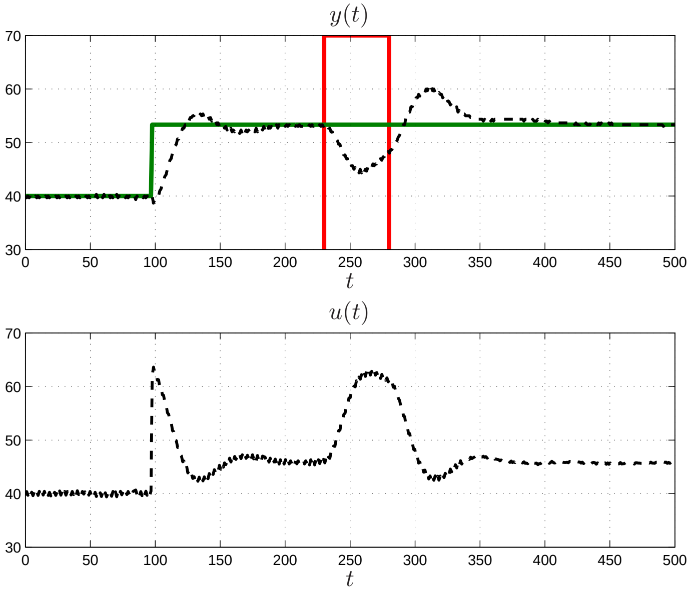
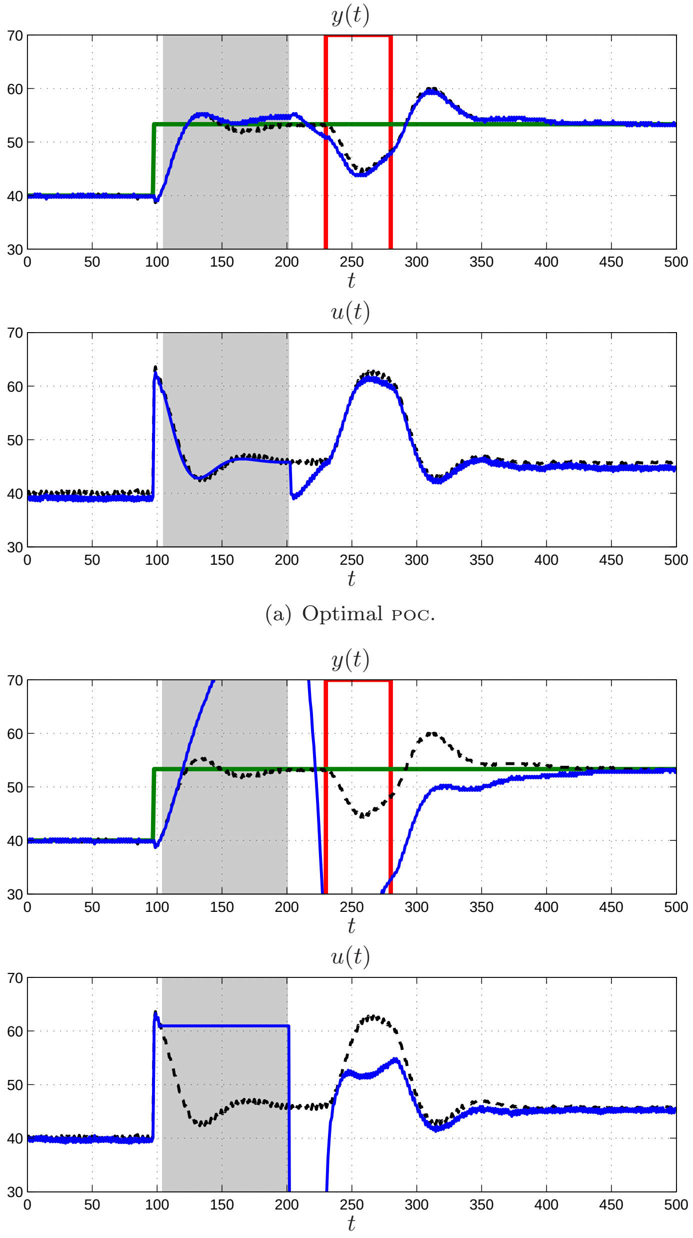


Figure 3.11: Nominal closed-loop behavior (black) under disturbance d (red) and reference r (green).

outage. This is an intrinsic problem for any predictive compensation algorithm, as this change is not visible to the predictor which operates in open-loop during the outage. Hence, the POC will continue to extrapolate the control signal assuming that the disturbance has not changed.

3.6 Summary

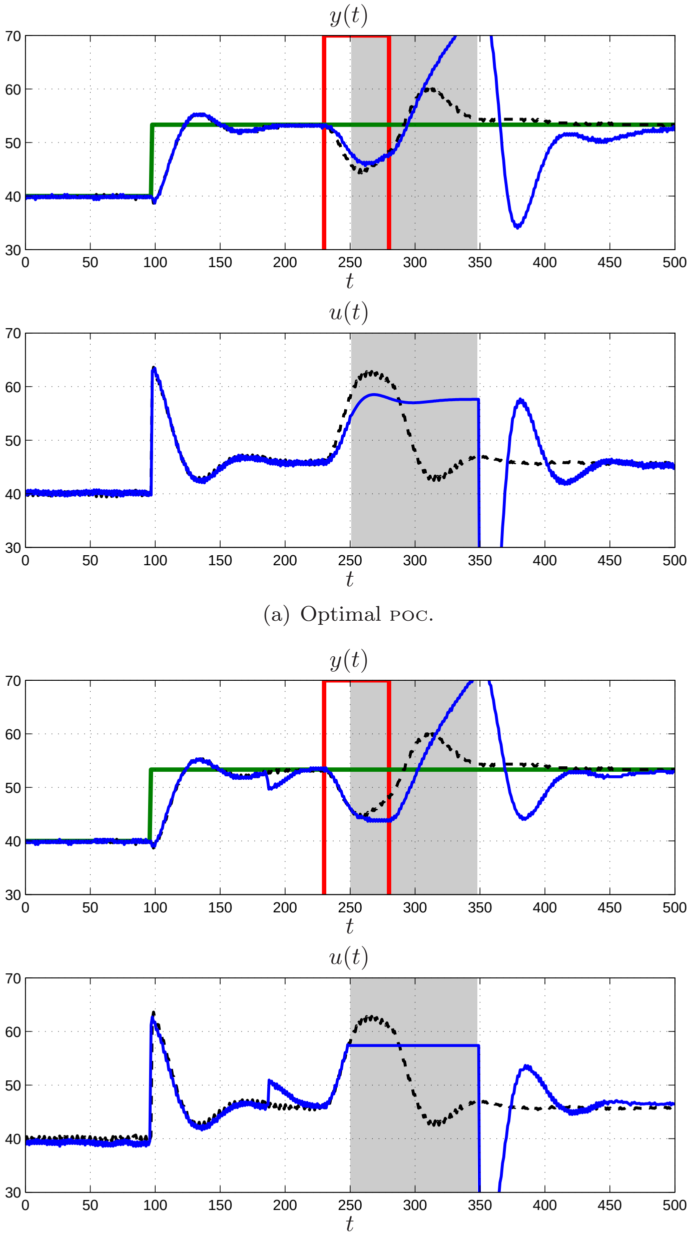
We presented a new methodology to compensate for communication losses in networked control systems. The proposed POC was shown to give significantly improved performance compared to previously used compensation schemes. In particular, we derived a method to synthesize a POC for MIMO systems affected by stochastic disturbances and noise. Prediction error bounds were presented. Methods were also developed to reduce the complexity of a POC by means of Hankel-norm approximation. A priori approximation error bounds for this reduction methods were presented. Finally, POCs were demonstrated on a simulated tank system as well as in real experiments.



(a) Optimal POC.

(b) Hold.

Figure 3.12: Comparison of POC and hold behavior (blue) with nominal behavior (black) under disturbance d (red), reference r (green) and outage (grey area).



(a) Optimal POC.

(b) Hold.

Figure 3.13: Comparison of POC and hold behavior (blue) with nominal behavior (black) under disturbance d (red), reference r (green) and outage (grey area).

Chapter 4

Self-Triggered Model Predictive Control

Consider the networked control system in Figure 4.1, which shows how the sensors and the controller are connected through a wireless network. The wireless network is controlled by a *Network Manager* which allocates medium access to the sensors and triggers their transmissions. This setup is motivated by current industry standards, *e.g.*, the WirelessHART protocol (Hart Communication Foundation, 2007), which utilizes this structure for wireless control in process industry. Here the triggering is in turn generated by the controller which, in addition to computing the appropriate control action, dynamically determines the time of the next sample by a self-triggering approach. In doing so, the controller gives varying attention to the different loops depending on their state, while trying to communicate few samples. To achieve this, the controller must, for every loop, trade control performance against inter sampling time and give a quantitative measure of the resulting performance. The main contribution of the chapter is to show that a self-triggering controller can be derived using a receding horizon control formulation where the predicted cost is used to jointly determine what control signal to be applied as well as the time of the next sampling instant. Using this model predictive control (mpc) formulation it is possible to guarantee a minimum and a maximum time between samples. Initially a single-loop system is considered. This is later extended to the multiple-loop case, which can be analyzed with additional constraints on the communication pattern.

The outline of the chapter is as follows. In Section 4.1 the self-triggered network scheduling and control problem is defined and formulated as a receding horizon control problem. Section 4.2 presents the open-loop optimal control problem for the single-loop case, to be solved by the receding horizon controller, together with its optimal solution. Section 4.3 presents the single-loop receding horizon control algorithm in further detail and gives conditions for when it is stabilizing. The results are then extended to the multiple-loop case in Section 4.4, where conditions for stability and conflict-free transmissions are given. The proposed method is explained and evaluated on simulated examples in Section 4.5. The results are then summarized in Section 4.6.

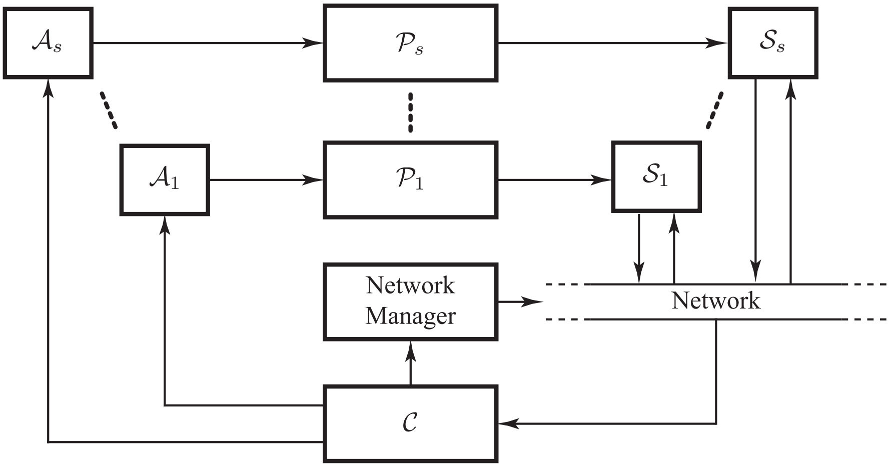


Figure 4.1: Actuators $\mathcal{A}$ and processes $\mathcal{P}$ are wired to the controller $\mathcal{C}$ while the sensors $\mathcal{S}$ communicate over a wireless network, which in turn is coordinated by the *Network Manager*.

4.1 Problem Formulation

We consider the problem of controlling $s \geq 1$ processes $\mathcal{P}_1$ through $\mathcal{P}_s$ over a shared communication network as in Figure 4.1. The processes are controlled by the controller $\mathcal{C}$ which computes the appropriate control action and schedule for each process. Each process $\mathcal{P}_\ell$ is given by a linear time-invariant system

$$\begin{aligned} x_\ell(k+1) &= A_\ell x_\ell(k) + B_\ell u_\ell(k), \\ x_\ell(k) &\in \mathbb{R}^{n_\ell},\ u_\ell(k) \in \mathbb{R}^{m_\ell}. \end{aligned} \tag{4.1}$$

The controller works in the following way: When sensor $\mathcal{S}_\ell$ at time $k = k_\ell$ transmits a sample $x_\ell(k_\ell)$ to the controller $\mathcal{C}$, it computes the control signal $u_\ell(k_\ell)$ and sends it to the actuator $\mathcal{A}_\ell$. The actuator in turn will apply this control signal to the process until it receives a new value from the controller. Jointly with deciding $u_\ell(k_\ell)$ the controller also decides how many discrete time steps, say $I_\ell(k_\ell)$, it will wait before it needs to change the control signal the next time. This value $I_\ell(k_\ell)$ is sent to the *Network Manager* which will schedule the sensor $\mathcal{S}_\ell$ to send a new sample at time $k = k_\ell + I_\ell(k_\ell)$. To guarantee conflict-free transmissions on the network, no two sensors are allowed to transmit at the same time. Hence, when deciding the time to wait $I_\ell(k_\ell)$, the controller must make sure that no other sensor $\mathcal{S}_q$, $q \neq \ell$, already is scheduled for transmission at time $k = k_\ell + I_\ell(k_\ell)$.

We propose that the controller $\mathcal{C}$ should be implemented as a receding horizon controller which for an individual loop ℓ at every sampling instant $k = k_\ell$ solves an open-loop optimal control problem. It does so by minimizing the infinite-horizon quadratic cost function

$$\sum_{l=0}^{\infty}\left(\|x_\ell(k_\ell+l)\|^2_{Q_\ell}+\|u_\ell(k_\ell+l)\|^2_{R_\ell}\right)$$

subject to the user defined weights Q_ℓ and R_ℓ, while taking system dynamics, control performance, inter sample time and overall network schedulability into consideration.

4.2 Open-Loop Optimal Control

For pedagogical ease we will in this section study the case when we control a single process on the network allowing us to drop the loop-index ℓ. The process we control has dynamics

$$x(k+1)=Ax(k)+Bu(k),\ x(k)\in\mathbb{R}^n,\ u(k)\in\mathbb{R}^m \tag{4.2}$$

and the open-loop cost function we propose the controller to minimize at every sampling instant is

$$J(x(k),i,\mathcal{U})=\frac{\alpha}{i}+\sum_{l=0}^{\infty}\left(\|x(k+l)\|^2_Q+\|u(k+l)\|^2_R\right), \tag{4.3}$$

where $\alpha\in\mathbb{R}^+$, Q and R are design variables. Further, $0<Q$ and $0<R$ are symmetric matrices of appropriate dimensions. We optimize this cost over the constraint that the control sequence $\mathcal{U}=\{u(k),u(k+1),\ldots\}$ should follow the specific shape illustrated in Figure 4.2, for some fixed period $p\in\mathbb{N}^+$. That is, the number of discrete time units $i=I(k)$ to wait before taking the next sample $x(k+i)$, as well as the levels in the control sequence $\mathcal{U}$ are free variables over which we optimize. Note that neither the state nor the control values have a constrained magnitude.

The shape of the control trajectory $\mathcal{U}$ is motivated by the following idea. We will use a slight modification of regular receding horizon control where we at time instant k sample the system and use this sample $x(k)$ to compute the predicted, with respect to the constraints, optimal control trajectory $\mathcal{U}$. We will then apply this sequence from time k until time instant $k+i$ when we will take a new sample $x(k+i)$ and redo the optimization. By this method we get a joint optimization of the control signal to be applied as well as the time to the next sampling instant. By the choice of constraints on the shape of $\mathcal{U}$ we will get the shape of the predicted control signal to coincide with the one we will actually apply, up until time $k+i$. The reason for letting the system be controlled by a control signal with period p after this is that we hope for the receding horizon algorithm to converge to this rate. In Section 4.3 we will later provide methods for choosing p so that this happens. The reason for wanting convergence to a down-sampled control is that we want the system to be sampled at a slow rate when it has reached steady state, at the same time as we want it to be sampled faster during the transients.

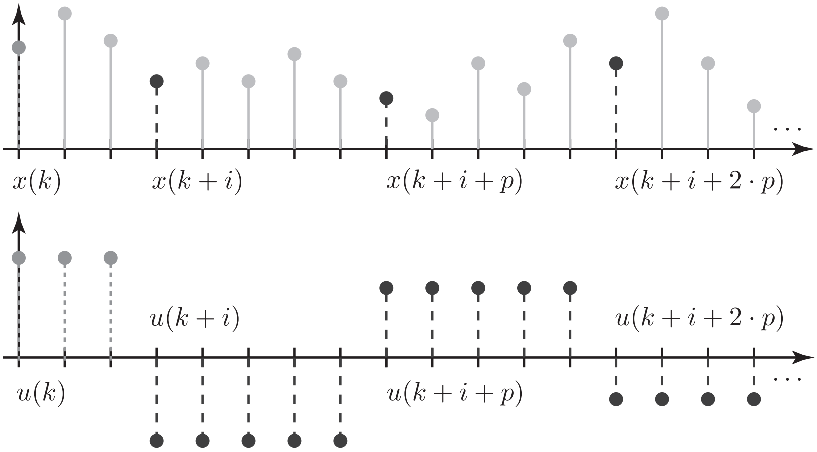


Figure 4.2: The prediction horizon. Here typical signal predictions are shown with $I(k) = 3$ and $p = 5$.

Mathematically we may formulate the cost (4.3) including the constraints as

$$\begin{aligned} J(x(k), i, \mathcal{U}(i)) &= \frac{\alpha}{i} + \sum_{l=0}^{i-1} \Big(\|x(k+l)\|_Q^2 + \|u(k)\|_R^2 \Big) \\ &\quad + \sum_{r=0}^{\infty} \left(\sum_{l=0}^{p-1} \Big(\|x(k+i+r\cdot p+l)\|_Q^2 + \|u(k+i+r\cdot p)\|_R^2 \Big) \right) \end{aligned} \tag{4.4}$$

where $\mathcal{U}(i) = \{u(k), u(k+i), u(k+i+\eta\cdot p), \ldots\}$, $\eta \in \mathbb{N}^+$, are the decision variables over which we optimize. The term α/i reflects the cost of sampling. We use this cost to weight the cost of sampling against the classical quadratic control performance cost. For a given $x(k)$, choosing a large α will force i to be larger and hence give longer inter sampling times. By the construction of the cost we may first choose Q and R to get the desired control performance and then tune α to get the desired sampling behavior. One could imagine a more general cost of sampling, here however we found α/i sufficient.

4.2.1 Cost Function Minimization

Having defined the open-loop cost (4.4) we proceed by computing its optimal value. We start by noticing that, even though we have a joint optimization problem, we may state it as

$$\underset{i}{\text{minimize}} \Big(\underset{\mathcal{U}(i)}{\text{minimize}}\ J(x(k), i, \mathcal{U}(i)) \Big). \tag{4.5}$$

We will use this separation and start by solving the inner problem, that of minimizing $J(x(k), i, \mathcal{U}(i))$ for a given value of i. In order to derive the solution, and for future reference, we need to define some variables.

Definition 4.2.1. *We define notation for the lifted model as*

$$A^{(i)} = A^i, \quad B^{(i)} = \sum_{q=0}^{i-1} A^q B.$$

and notation for the generalized weighting matrices associated to (4.4) as

$$\begin{aligned} Q^{(i)} &= Q^{(i-1)} + {A^{(i-1)}}^T Q A^{(i-1)} \\ R^{(i)} &= R^{(i-1)} + {B^{(i-1)}}^T Q B^{(i-1)} + R \\ N^{(i)} &= N^{(i-1)} + {A^{(i-1)}}^T Q B^{(i-1)} \end{aligned}$$

where $Q^{(1)} = Q$, $R^{(1)} = R$ and $N^{(1)} = 0$.

Using Definition 4.2.1 it is straightforward to show the following lemma.

Lemma 4.2.2. *It holds that*

$$\sum_{l=0}^{i-1} \Big(\|x(k+l)\|_Q^2 + \|u(k)\|_R^2 \Big) \\ = x(k)^T Q^{(i)} x(k) + u(k)^T R^{(i)} u(k) + 2x(k)^T N^{(i)} u(k)$$

and $x(k+i) = A^{(i)} x(k) + B^{(i)} u(k)$.

Lemma 4.2.3. *Assume that $0 < Q$, $0 < R$ and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Then*

$$\min_{\mathcal{U}(i)} \sum_{r=0}^{\infty} \left(\sum_{l=0}^{p-1} \Big(\|x(k+i+r\cdot p+l)\|_Q^2 + \|u(k+i+r\cdot p)\|_R^2 \Big) \right) = \|x(k+i)\|_{P^{(p)}}^2$$

where

$$\begin{aligned} P^{(p)} &= Q^{(p)} + {A^{(p)}}^T P^{(p)} A^{(p)} - \big({A^{(p)}}^T P^{(p)} B^{(p)} + N^{(p)}\big) L^{(p)} \\ L^{(p)} &= \big(R^{(p)} + {B^{(p)}}^T P^{(p)} B^{(p)}\big)^{-1} \big({A^{(p)}}^T P^{(p)} B^{(p)} + N^{(p)}\big)^T \end{aligned} \tag{4.6}$$

and the minimizing control signal characterizing $\mathcal{U}(i)$ is given by

$$u(k+i+r\cdot p) = -L^{(p)} x(k+i+r\cdot p).$$

Proof. Following Lemma 4.2.2 the problem is equivalent to

$$\min_{\mathcal{U}(i)} \sum_{r=0}^{\infty} \Big(x(k+i+r\cdot p)^T Q^{(p)} x(k+i+r\cdot p) \\ + u(k+i+r\cdot p)^T R^{(p)} u(k+i+r\cdot p) \\ + 2x(k+i+r\cdot p)^T N^{(p)} u(k+i+r\cdot p) \Big)$$

with

$$x(k+i+(r+1)\cdot p) = A^{(p)} x(k+i+r\cdot p) + B^{(p)} u(k+i+r\cdot p).$$

This problem has the known optimal solution, see e.g. (Bertsekas, 1995), $\|x(k+i)\|^2_{P^{(p)}}$. Where $P^{(p)}$ is given by the Riccati equation (4.6), which has a solution provided that $0 < R^{(p)}$, implied by $0 < R$, and $0 < Q^{(p)}$, implied by $0 < Q$, and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Exactly what is stated in the lemma. □

Using the above results we may formulate the main result of this section as follows.

Theorem 4.2.4. *Assume that $0 < Q$, $0 < R$ and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Then*

$$\min_{\mathcal{U}(i)} J(x(k), i, \mathcal{U}(i)) = \frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}} \tag{4.7}$$

where

$$\begin{aligned} P^{(i)} &= Q^{(i)} + A^{(i)T} P^{(p)} A^{(i)} - \big(A^{(i)T} P^{(p)} B^{(i)} + N^{(i)}\big) L^{(i)} \\ L^{(i)} &= \big(R^{(i)} + B^{(i)T} P^{(p)} B^{(i)}\big)^{-1} \big(A^{(i)T} P^{(p)} B^{(i)} + N^{(i)}\big)^T \end{aligned} \tag{4.8}$$

and $P^{(p)}$ is given by Lemma 4.2.3. Denoting the vector of all ones in $\mathbb{R}^n$ as $\mathbf{1}_n$, the minimizing control signal sequence is given by

$$\mathcal{U}^* = \{-L^{(i)} x(k) \mathbf{1}_i^T, -L^{(p)} x(k+i+r\cdot p)\mathbf{1}_p^T, \ldots\},\ r \in \mathbb{N}$$

where also $L^{(p)}$ is given by Lemma 4.2.3.

Proof. From the theorem we have that $0 < Q$, $0 < R$ and that the pair $(A^{(p)}, B^{(p)})$ is controllable. Thus we may use Lemma 4.2.3 to express the cost (4.4) as

$$J(x(k), i, \mathcal{U}(i)) = \frac{\alpha}{i} + \|x(k+i)\|^2_{P^{(p)}} + \sum_{l=0}^{i-1} \Big(\|x(k+l)\|^2_Q + \|u(k)\|^2_R \Big).$$

Now applying Lemma 4.2.2 we get

$$J(x(k), i, \mathcal{U}(i)) = \frac{\alpha}{i} + \|x(k+i)\|^2_{P^{(p)}} \\ + x(k)^T Q^{(i)} x(k) + u(k)^T R^{(i)} u(k) + 2x(k)^T N^{(i)} u(k)$$

with $x(k+i) = A^{(i)}x(k) + B^{(i)}u(k)$. Minimizing $J(x(k), i, \mathcal{U}(i))$ now becomes a finite horizon optimal control problem with one prediction step into the future. This problem has the well defined solution (4.7), see *e.g.*, (Bertsekas, 1995), given by iterating the Riccati equation (4.8). □

Now, getting back to the original problem (4.5). Provided that the assumptions of Theorem 4.2.4 hold, we may apply it giving that

$$\underset{i,\mathcal{U}(i)}{\text{minimize}}\ J(x(k), i, \mathcal{U}(i)) = \underset{i}{\text{minimize}}\ \left\{\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}}\right\}.$$

Unfortunately the authors are not aware of any method to solve this problem in general. If however i is restricted to a known *finite* set $\mathcal{I}^0 \subset \mathbb{N}^+$ we may find the optimal value within this set for a given value of $x(k)$ by simply evaluating $\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}}$ $\forall i \in \mathcal{I}^0$ and by this obtaining the i which gives the lowest value of the cost. This procedure gives the optimum of (4.5). Note that the computational complexity of finding the optimum is not necessarily high, as we may compute $P^{(i)}$ $\forall i \in \mathcal{I}^0$ off-line prior to execution.

4.3 Single-Loop Self-Triggered MPC

We have defined the open-loop cost which we propose the receding horizon controller to minimize at every sampling instant when controlling the single process (4.2) over the network. Further we have calculated its optimal value as a function of the time interval i to wait until the next sample and the corresponding minimizing control signal to be applied. Now we continue with formulating the receding horizon implementation in further detail.

We will assume that $0 < Q$, $0 < R$ and that the down-sampled pair $(A^{(p)}, B^{(p)})$ is controllable. Let us also assume that the *finite* and *non-empty* set $\mathcal{I}^0 \subset \mathbb{N}^+$ is given and let $\gamma = \max \mathcal{I}^0$. From this we may use Theorem 4.2.4 to compute the pairs $(P^{(i)}, L^{(i)})$ $\forall i \in \mathcal{I}^0$ and formulate our proposed single-loop receding horizon control algorithm as in Algorithm 4.1.

Remark 4.3.1. Note that the control signal value is sent to the actuator at the same time as the controller requests the scheduling of the next sample by the sensor.

Remark 4.3.2. The proposed algorithm guarantees a minimum and a maximum inter sampling time. The minimum time is 1 time step in the time scale of the underlying process (4.2) and the maximum inter sampling time is γ time steps. This implies that there is some minimum attention to every loop independent of the predicted evolution of the process.

Remark 4.3.3. Even though we are working in uniformly sampled discrete-time the state is not sampled at every time instant k. Instead the set of samples of the

Algorithm 4.1 Single-Loop Self-Triggered MPC

```
1. At time $k = k'$ the sample $x(k')$ is transmitted by the sensor to the controller.
2. Using $x(k')$ the controller computes
       $I(k') = \arg\min_{i \in \mathcal{I}^0} \frac{\alpha}{i} + \|x(k')\|^2_{P^{(i)}}$,
       $u(k') = -L^{(I(k'))} x(k')$.
3. a) The controller sends $u(k')$ to the actuator which applies $u(k) = u(k')$ to (4.2) until $k = k' + I(k')$.
   b) The network manager is requested to schedule the sensor to transmit a new sample at time $k = k' + I(k')$.
```

state actually taken is given by the set $\mathcal{D}$, which assuming that the first sample is taken at $k = 0$, is given by

$$\mathcal{D} = \{x(0), x(I(0)), x(I(I(0))), \dots\}. \tag{4.9}$$

4.3.1 Stability

Having established and detailed our single-loop receding horizon control law we continue with giving conditions for when it is stabilizing. Letting $\lambda(A)$ denote the set of eigenvalues to A and $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$ we first state the following controllability conditions.

Lemma 4.3.4. *The system $(A^{(i)}, B^{(i)})$ is controllable if and only if the pair (A, B) is controllable and A has no eigenvalue $\lambda \in \lambda(A)$ such that $\lambda \neq 1$ and $\lambda^i = 1$.*

Proof. The proof is based on adaptation of Lemma 3.4.1 and Theorem 4 in (Sontag, 1998) to the problem of down-sampling. Let $\mathbf{R}(\cdot)$ be the reachability matrix. For controllability we require

$$\begin{aligned}\mathbf{R}\big(A^{(i)}, B^{(i)}\big) &= \Big[\sum_{q=0}^{i-1} A^q B, \dots, A^{(n-1)i} \sum_{q=0}^{i-1} A^q B\Big] \\ &= \Big(\sum_{q=0}^{i-1} A^q\Big)\Big[B, \dots, A^{(n-1)i} B\Big] = \Big(\sum_{q=0}^{i-1} A^q\Big)\mathbf{R}\big(A^i, B\big)\end{aligned}$$

to have full rank, i.e., for both $\sum_{q=0}^{i-1} A^q$ and $\mathbf{R}\big(A^i, B\big)$ to have full rank.

We start by showing the former. According to the Spectral Mapping Theorem the eigenvalues of $f(A) = \sum_{q=0}^{i-1} A^q$ will be $f(\lambda)$ with $\lambda \in \lambda(A)$. For $f(A)$ to have full rank we require $f(\lambda) \neq 0$, $\forall \lambda \in \lambda(A)$. Since

$$f(1) = i, \quad f(\lambda) = \sum_{q=0}^{i-1} \lambda^q = \frac{\lambda^i - 1}{\lambda - 1}, \quad \lambda \neq 1$$

this is fulfilled if and only if $\lambda^i \neq 1$ $\forall \lambda \in \lambda(A)$ such that $\lambda \neq 1$, precisely what is stated in the lemma.

Moving on, assume that $\mathbf{R}\big(A^i, B\big)$ does not have full rank. Which according to the *Hautus Lemma* in (Sontag, 1998) is equivalent to that $[\lambda^i I - A^i, B]$ is rank deficient, implying $\exists\, v \neq 0$ such that for some λ we have $v^T[\lambda^i I - A^i, B] = 0$ implying $v^T \lambda^i - v^T A^i = 0$ and $v^T B = 0$. Thus for $\mathbf{R}\big(A^i, B\big)$ to be rank deficient there must exist a left eigenvector v^T to A^i such that $v^T B = 0$. Now let e^T be a left eigenvector to A which, since $e^T A^i = e^T \lambda^i$, also is a left eigenvector to A^i. Thus for $\mathbf{R}\big(A^i, B\big)$ to be rank deficient there must $\exists\, e$ such that $e^T B = 0$. Since $\mathbf{R}\big(A, B\big)$ has full rank, by the statement in the lemma, we have that for any (non zero) left eigenvector e^T it holds that $e^T[\lambda^i I - A^i, B] \neq 0$ implying $e^T B \neq 0$ which contradicts the assumption that $\mathbf{R}\big(A^i, B\big)$ is rank deficient. □

Using the above, and our previous results, we may now give conditions for when the proposed receding horizon control algorithm is stabilizing.

Theorem 4.3.5. *Assume $0 < Q$, $0 < R$ and that (A, B) is controllable. If we choose $i \in \mathcal{I}^0 \subset \mathbb{N}^+$ and $p = p^*$ given by*

$$p^* = \max\{i | i \in \mathcal{I}^0, \forall \lambda \in \lambda(A)\ \lambda^i \neq 1 \textit{ if } \lambda \neq 1\} \tag{4.10}$$

and apply Algorithm 4.1, then

$$\frac{\alpha}{\gamma} \leq \lim_{k \to \infty} \min_{i \in \mathcal{I}^0} \left(\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}} \right) \leq \frac{\alpha}{\epsilon} \left(\frac{1}{p^*} - (1 - \epsilon) \frac{1}{\gamma} \right),$$

where ϵ is the largest value in the interval $(0, 1]$ which $\forall i \in \mathcal{I}^0$ fulfills

$$\big(A^{(i)} - B^{(i)} L^{(i)}\big)^T P^{(p^*)} \big(A^{(i)} - B^{(i)} L^{(i)}\big) \leq (1 - \epsilon) P^{(i)}.$$

Proof. By assumption (A, B) is controllable. Together with the choice of p^* this, via Lemma 4.3.4, implies that $(A^{(p^*)}, B^{(p^*)})$ is controllable. Further let $\hat{x}(k'|k)$ denote an estimate of $x(k')$, given all available measurements up until time k. Defining

$$\|\hat{x}(k|k)\|^2_{S^{(i)}} \triangleq \sum_{l=0}^{i-1} \Big(\|\hat{x}(k + l|k)\|^2_Q + \|\hat{u}(k|k)\|^2_R \Big) \tag{4.11}$$

we may, since by assumption $0 < Q$ and $0 < R$, use Lemma 4.2.3 and Theorem 4.2.4 to express V_k, the optimal value of the cost (4.4) at the current sampling instant

k, as

$$\begin{aligned} V_k \triangleq & \min_{i\in\mathcal{I}^0,\hat{u}(k|k)} J(x(k),i,\hat{u}(k|k)) \\ = & \min_{i\in\mathcal{I}^0} \frac{\alpha}{i} + \|\hat{x}(k+i|k)\|^2_{P(p^*)} + \|\hat{x}(k|k)\|^2_{S(i)} \\ = & \min_{i\in\mathcal{I}^0} \frac{\alpha}{i} + \|\hat{x}(k|k)\|^2_{P(i)}. \end{aligned} \tag{4.12}$$

We will use V_k as a Lyapunov-like function. Assume that V_{k+i} is the optimal cost at the next sampling instant $k+i$. Again using Theorem 4.2.4 we may express it as

$$\begin{aligned} V_{k+i} \triangleq & \min_{j\in\mathcal{I}^0,\hat{u}(k+i|k+i)} J(x(k+i),j,\hat{u}(k+i|k+i)) \\ \leq & \min_{\hat{u}(k+i|k+i)} J(x(k+i),j=p^*,\hat{u}(k+i|k+i)) \\ = & \frac{\alpha}{p^*} + \|\hat{x}(k+i|k+i)\|^2_{P(p^*)} = \frac{\alpha}{p^*} + \|\hat{x}(k+i|k)\|^2_{P(p^*)}. \end{aligned} \tag{4.13}$$

Where the inequality comes from the fact that choosing $j = p^*$ is sub-optimal. Taking the difference we get

$$V_{k+i} - V_k \leq \frac{\alpha}{p^*} - \frac{\alpha}{i} - \|\hat{x}(k|k)\|^2_{S(i)}$$

which in general is not decreasing. However we may use the following idea to bound this difference: Assume that there $\exists\,\epsilon \in (0,1]$ and $\beta \in \mathbb{R}^+$ such that we may write

$$V_{k+i} - V_k \leq -\epsilon V_k + \beta.$$

As this should hold for all sampling instances $\mathcal{D}$ given in (4.9), we have that at l sampling instances into the future, which happens at let's say time $k+l'$, we have that

$$V_{k+l'} \leq (1-\epsilon)^l \cdot V_k + \beta \cdot \sum_{r=0}^{l-i} (1-\epsilon)^l.$$

Since $\epsilon \in (0,1]$ this is equivalent to

$$V_{k+l'} \leq (1-\epsilon)^l \cdot V_k + \beta \cdot \frac{1-(1-\epsilon)^l}{1-(1-\epsilon)},$$

which as $l \to \infty$ gives us an upper bound on the cost function, $V_{k+l'} \leq \beta/\epsilon$. Applying this idea on our setup we should fulfill

$$\frac{\alpha}{p^*} - \frac{\alpha}{i} - \|\hat{x}(k|k)\|_{S(i)} \leq -\epsilon\frac{\alpha}{i} - \epsilon\|\hat{x}(k+i|k)\|^2_{P(p^*)} - \epsilon\|\hat{x}(k|k)\|^2_{S(i)} + \beta.$$

Choosing $\beta = \alpha/p^* - (1-\epsilon)\alpha/\gamma$ we have fulfillment if

$$\epsilon\Big(\|\hat{x}(k+i|k)\|^2_{P(p^*)} + \|\hat{x}(k|k)\|^2_{S(i)}\Big) \leq \|\hat{x}(k|k)\|^2_{S(i)}. \tag{4.14}$$

Clearly there $\exists \epsilon \in (0,1]$ such that the above relation is fulfilled if $0 < \|\hat{x}(k|k)\|^2_{S^{(i)}}$. If $\|\hat{x}(k|k)\|^2_{S^{(i)}} = 0$ we must, following the definition (4.11) and the assumption $0 < Q$, have that $\hat{x}(k|k) = 0$ and $\hat{x}(k+i|k) = 0$ and hence the relation is fulfilled also in this case. Using the final step in (4.12) we may express (4.14) in easily computable quantities giving the condition

$$\|\hat{x}(k+i|k)\|^2_{P^{(p^*)}} \leq (1-\epsilon)\|\hat{x}(k|k)\|^2_{P^{(i)}}.$$

As this should hold $\forall x$ and $\forall i \in \mathcal{I}^0$ we must fulfill

$$\big(A^{(i)} - B^{(i)}L^{(i)}\big)^T P^{(p^*)} \big(A^{(i)} - B^{(i)}L^{(i)}\big) \leq (1-\epsilon)P^{(i)}$$

which is stated in the theorem. Summing up we have

$$V_{k+l'} \leq \frac{\alpha}{\epsilon}\left(\frac{1}{p^*} - (1-\epsilon)\frac{1}{\gamma}\right)$$

which is minimized by maximizing ϵ. From the definition of the cost (4.3) we may also conclude that $\alpha/\gamma \leq V_{k+l'}$. With

$$V_{k+l'} = \min_{i \in \mathcal{I}^0}\left(\frac{\alpha}{i} + \|\hat{x}(k+l'|k+l')\|^2_{P^{(i)}}\right),$$

we may conclude that

$$\frac{\alpha}{\gamma} \leq \lim_{k\to\infty}\min_{i\in\mathcal{I}^0}\left(\frac{\alpha}{i} + \|\hat{x}(k|k)\|^2_{P^{(i)}}\right) \leq \frac{\alpha}{\epsilon}\left(\frac{1}{p^*} - (1-\epsilon)\frac{1}{\gamma}\right).$$

□

Remark 4.3.6. The bound given in Theorem 4.3.5 scales linearly with the choice of α.

Assumption 4.3.7. *Assume that $\nexists \lambda \in \lambda(A)$ except possibly $\lambda = 1$ such that $|\lambda| = 1$ and the complex argument $\angle\lambda = \frac{2\pi}{\gamma}\cdot n$ for some $n \in \mathbb{N}^+$.*

Lemma 4.3.8. *Let Assumption 4.3.7 hold, then $p^* = \gamma$.*

Proof. From (4.10) it is clear that $p^* = \gamma$ if $\nexists \lambda \in \lambda(A)$ except $\lambda = 1$ such that $\lambda^\gamma = 1$. In polar coordinates we have that $\lambda = |\lambda|\exp(j\cdot\angle\lambda)$ implying $\lambda^\gamma = |\lambda|^\gamma \exp(j\cdot\gamma\cdot\angle\lambda) = 1$ may only be fulfilled if $|\lambda| = 1$ and $\angle\lambda = \frac{2\pi}{\gamma}\cdot n$ for some $n \in \mathbb{N}^+$, which contradicts Assumption 4.3.7. □

Corollary 4.3.9. *Assume $0 < Q$, $0 < R$ and that (A,B) is controllable. Further assume that either Assumption 4.3.7 holds or $\alpha = 0$. If we choose $i \in \mathcal{I}^0 \subset \mathbb{N}^+$ and $p = p^*$ given by (4.10) and apply Algorithm 4.1, then*

$$\lim_{k\to\infty} x(k) = 0.$$

Proof. From Theorem 4.3.5 we have that as $k \to \infty$

$$\frac{\alpha}{\gamma} \leq \min_{i \in \mathcal{I}^0} \left(\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}} \right) \leq \frac{\alpha}{\epsilon} \left(\frac{1}{p^*} - (1-\epsilon) \frac{1}{\gamma} \right)$$

which as $p^* = \gamma$, given by Lemma 4.3.8, simplifies to

$$\frac{\alpha}{\gamma} \leq \min_{i \in \mathcal{I}^0} \left(\frac{\alpha}{i} + \|x(k)\|^2_{P^{(i)}} \right) \leq \frac{\alpha}{\gamma}$$

independent of ϵ. Implying that as $k \to \infty$ we have that $i = \gamma$ and $\|x(k)\|^2_{P^{(i)}} = 0$, since $0 < P^{(i)}$ provided $0 < Q$ this implies $x(k) = 0$ independent of α. In the case $\alpha = 0$ the bound from Theorem 4.3.5 simplifies to that as $k \to \infty$,

$$0 \leq \min_{i \in \mathcal{I}^0} \|x(k)\|^2_{P^{(i)}} \leq 0$$

and hence for the optimal i we have $\|x(k)\|^2_{P^{(i)}} = 0$ implying $x(k) = 0$ as above. □

From the above results we may note the following.

Remark 4.3.10. If the assumptions of Theorem 4.3.5 hold, Corollary 4.3.9 will hold except in the extremely rare case that the underlying system (A, B) becomes uncontrollable under down-sampling by a factor γ, see Lemma 4.3.4. If it does not hold one may use Lemma 4.3.8 to re-design $\mathcal{I}^0$ giving a new value on γ which recovers the case $p^* = \gamma$, so that it will hold.

4.4 Multiple-Loop Self-Triggered MPC

Having detailed the controller for the single-loop case and given conditions for when it is stabilizing we now continue with extending to the multiple-loop case when we control multiple loops on the network, as described in Figure 4.1. The idea is that the controller $\mathcal{C}$ now will run s such single-loop controllers described in Algorithm 4.1 in parallel, one for each process $\mathcal{P}_\ell$, $\ell \in \mathbb{L} = \{1, 2, \dots, s\}$, controlled over the network. To guarantee conflict-free communication on the network the controller $\mathcal{C}$ will, at the same time, centrally coordinate the transmissions of the different loops.

4.4.1 Open-Loop Optimal Control

We start by extending the results in Section 4.2 to the multiple-loop case. The cost function we propose the controller to minimize at every sampling instant for loop ℓ is then

$$\begin{aligned} J_\ell(x_\ell(k), i, \mathcal{U}_\ell(i)) = \frac{\alpha_\ell}{i} &+ \sum_{l=0}^{i-1} \left(\|x_\ell(k+l)\|^2_{Q_\ell} + \|u_\ell(k)\|^2_{R_\ell} \right) \\ &+ \sum_{r=0}^{\infty} \left(\sum_{l=0}^{p_\ell - 1} \left(\|x_\ell(k+i+r \cdot p_\ell + l)\|^2_{Q_\ell} + \|u_\ell(k+i+r \cdot p_\ell)\|^2_{R_\ell} \right) \right), \end{aligned}$$

derived in the same way as (4.4) now with $\alpha_\ell \in \mathbb{R}^+$, $0 < Q_\ell$ $0 < R_\ell$ and period $p_\ell \in \mathbb{N}^+$ specific for the control of process $\mathcal{P}_\ell$ given by (4.1). From this we can state the following.

Definition 4.4.1. *We define the notation in the multiple-loop case following Definition 4.2.1. For a matrix E_ℓ, e.g., A_ℓ and Q_ℓ, we denote $\big(E_\ell\big)^{(i)}$ by $E_\ell^{(i)}$.*

Theorem 4.4.2. *Assume that $0 < Q_\ell$, $0 < R_\ell$ and that the pair $(A_\ell^{(p)}, B_\ell^{(p)})$ is controllable. Then*

$$\min_{\mathcal{U}_\ell(i)} J_\ell(x_\ell(k), i, \mathcal{U}_\ell(i)) = \frac{\alpha_\ell}{i} + \|x_\ell(k)\|^2_{P_\ell^{(i)}},$$

where

$$\begin{aligned} P_\ell^{(i)} &= Q_\ell^{(i)} + A_\ell^{(i)^T} P_\ell^{(p_\ell)} A_\ell^{(i)} - \big(A_\ell^{(i)^T} P_\ell^{(p_\ell)} B_\ell^{(i)} + N_\ell^{(i)}\big) L_\ell^{(i)}, \\ L_\ell^{(i)} &= \big(R_\ell^{(i)} + B_\ell^{(i)^T} P_\ell^{(p_\ell)} B_\ell^{(i)}\big)^{-1} \big(A_\ell^{(i)^T} P_\ell^{(p_\ell)} B_\ell^{(i)} + N_\ell^{(i)}\big)^T, \end{aligned}$$

and

$$\begin{aligned} P_\ell^{(p_\ell)} &= Q_\ell^{(p_\ell)} + A_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} A_\ell^{(p_\ell)} - \big(A_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} B_\ell^{(p_\ell)} + N_\ell^{(p_\ell)}\big) L_\ell^{(p_\ell)}, \\ L_\ell^{(p_\ell)} &= \big(R_\ell^{(p_\ell)} + B_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} B_\ell^{(p_\ell)}\big)^{-1} \big(A_\ell^{(p_\ell)^T} P_\ell^{(p_\ell)} B_\ell^{(p_\ell)} + N_\ell^{(p_\ell)}\big)^T. \end{aligned}$$

Denoting the vector of all ones in $\mathbb{R}^n$ as $\mathbf{1}_n$, the minimizing control signal sequence is given by

$$\mathcal{U}_\ell^* = \{-L_\ell^{(i)} x_\ell(k) \mathbf{1}_i^T, -L_\ell^{(p_\ell)} x_\ell(k + i + r \cdot p_\ell) \mathbf{1}_{p_\ell}^T, \ldots\},\ r \in \mathbb{N}.$$

Proof. Application of Theorem 4.2.4 on the individual loops. □

4.4.2 Receding Horizon Control Law

To formulate our multiple-loop receding horizon control law we will re-use the results in Section 4.3 and apply them on a per loop basis.

For each process $\mathcal{P}_\ell$ with dynamics given in (4.1) let the weights $\alpha_\ell \in \mathbb{R}^+$, $0 < Q_\ell$ $0 < R_\ell$ and period $p_\ell \in \mathbb{N}^+$ specific to the process be defined. If we further define the *finite* set $\mathcal{I}_\ell^0 \subset \mathbb{N}^+$ for loop ℓ we may apply Theorem 4.4.2 to compute the pairs $(P_\ell^{(i)}, L_\ell^{(i)})$ $\forall i \in \mathcal{I}_\ell^0$. Provided of course that the pair $(A_\ell^{(p_\ell)}, A_\ell^{(p_\ell)})$ is controllable.

As discussed in Section 4.1 the controller must when choosing i take into consideration what transmission times other loops have reserved as well as overall network schedulability. Hence at time $k = k_\ell$ loop ℓ is restricted to choose $i \in \mathcal{I}_\ell(k_\ell) \subseteq \mathcal{I}_\ell^0$ where $\mathcal{I}_\ell(k_\ell)$ contains the feasible values of i which gives collisions-free scheduling of the network. How $\mathcal{I}_\ell(k_\ell)$ should be constructed when multiple loops are present on the network is discussed further later in this section.

We may now continue with formulating our control law for controlling multiple processes over the network as in Algorithm 4.2, to be executed whenever a sample is received by the controller.

Algorithm 4.2 Multiple-Loop Self-Triggered MPC

1. At time $k = k_\ell$ the sample $x_\ell(k_\ell)$ of process $\mathcal{P}_\ell$ is transmitted by the sensor $\mathcal{S}_\ell$ to the controller $\mathcal{C}$.
2. The controller $\mathcal{C}$ constructs $\mathcal{I}_\ell(k_\ell)$.
3. Using $x_\ell(k_\ell)$ the controller $\mathcal{C}$ computes
$$\begin{aligned} I_\ell(k_\ell) &= \arg \min_{i \in \mathcal{I}_\ell(k_\ell)} \frac{\alpha_\ell}{i} + \|x_\ell(k_\ell)\|^2_{P_\ell^{(i)}}, \\ u_\ell(k_\ell) &= -L_\ell^{(I(k_\ell))} x_\ell(k_\ell). \end{aligned}$$
4. a) The controller $\mathcal{C}$ sends $u_\ell(k_\ell)$ to the actuator $\mathcal{A}_\ell$ which applies $u_\ell(k) = u_\ell(k_\ell)$ to (4.1) until $k = k_\ell + I_\ell(k_\ell)$.

   b) The *Network Manager* is requested to schedule the sensor $\mathcal{S}_\ell$ to transmit a new sample at time $k = k_\ell + I_\ell(k_\ell)$.

When the controller is initialized at time $k = 0$ it is assumed that the controller has knowledge of the state $x_\ell(0)$ for all processes $\mathcal{P}_\ell$ controlled over the network. It will then execute Algorithm 4.2 entering at step 2, in the order of increasing loop-index ℓ.

4.4.3 Schedulability

What remains to be detailed in the multiple-loop receding horizon control law is a mechanism for loop ℓ to choose $\mathcal{I}_\ell(k_\ell)$ to achieve collision-free scheduling. We now continue with giving conditions for when this holds.

First we note that when using Theorem 4.4.2 we make the implicit assumption that it is possible to apply the corresponding optimal control signal sequence $\mathcal{U}_\ell^*$. For this to be possible we must be able to measure the state $x_\ell(k)$ at the future time instances

$$S_\ell(k_\ell) = \{k_\ell + I_\ell(k_\ell),\ k_\ell + I_\ell(k_\ell) + p_\ell,\ k_\ell + I_\ell(k_\ell) + 2 \cdot p_\ell,\ \dots\}. \tag{4.15}$$

Hence this sampling pattern must be reserved for use by sensor $\mathcal{S}_\ell$. We state the following to give conditions for when this is possible.

Lemma 4.4.3. *Let loop ℓ choose its set $\mathcal{I}_\ell(k_\ell)$ of feasible times to wait until the next sample to be*

$$\mathcal{I}_\ell(k_\ell) = \{i \in \mathcal{I}_\ell^0 | i \neq k_q^{next} - k_\ell + n \cdot p_q - m \cdot p_\ell,\ m, n \in \mathbb{N},\ q \in \mathbb{L} \setminus \{\ell\}\}$$

where k_q^{next} is the next transmission time of sensor $\mathcal{S}_q$. Then it is possible to reserve the needed sampling pattern $S_\ell(k_\ell)$ in (4.15) at time $k = k_\ell$.

Proof. When loop q was last sampled at time $k_q < k_\ell$ it was optimized over $\mathcal{I}_q(k_q)$ and found the optimal feasible time until the next sample $I_q(k_q)$. The loop then reserved the infinite sequence

$$S_q(k_q) = \{k_q + I_q(k_q),\ k_q + I_q(k) + p_q,\ k_q + I_q(k_q) + 2 \cdot p_q,\ \dots\}.$$

When loop ℓ now should choose $I_\ell(k_\ell)$ it must be able to reserve (4.15). To ensure that this is true, it must choose $I_\ell(k_\ell)$ so that $S_\ell(k_\ell) \cap S_q(k_q) = \emptyset$, $\forall\, q \in \mathbb{L} \setminus \{\ell\}$. This holds if we have that

$$\begin{aligned} &k_\ell + I_\ell(k_\ell) + m \cdot p_\ell \neq k_q + I_q(k_q) + n \cdot p_q, \\ &\forall\, m, n \in \mathbb{N},\ \forall\, q \in \mathbb{L} \setminus \{\ell\}. \end{aligned}$$

Simplifying the above condition we get conditions on the feasible values of $I_\ell(k_\ell)$

$$\begin{aligned} I_\ell(k_\ell) \neq \Big(k_q + I_q(k_q)\Big) - k_\ell + n \cdot p_q - m \cdot p_\ell, \\ \forall\, m, n \in \mathbb{N},\ \forall\, q \in \mathbb{L} \setminus \{\ell\}. \end{aligned}$$

Noticing that $\Big(k_q + I_q(k_q)\Big)$ is the next transmission of loop q we denote it k_q^{next}, giving the statement in the lemma. □

Constructing $\mathcal{I}_\ell(k_\ell)$ as above we are not guaranteed that $\mathcal{I}_\ell(k_\ell) \neq \emptyset$. To guarantee this we make the following assumption.

Assumption 4.4.4. *Assume that for every loop ℓ on the network $\mathcal{I}_\ell^0 = \mathcal{I}^0$ and $p_\ell = p$. Further assume that $\mathbb{L} \subseteq \mathcal{I}^0$ and $\max \mathbb{L} \leq p$.*

Theorem 4.4.5. *Let Assumption 4.4.4 hold. If every loop ℓ chooses*

$$\mathcal{I}_\ell(k_\ell) = \{i \in \mathcal{I}^0 | i \neq k_q^{next} - k_\ell + r \cdot p,\ r \in \mathbb{Z},\ q \in \mathbb{L} \setminus \{\ell\}\},$$

all transmissions on the network will be conflict-free and it will always be possible to reserve the needed sampling pattern $S_\ell(k_\ell)$ in (4.15).

Proof. Assuming $p_\ell = p$ and $\mathcal{I}_\ell^0 = \mathcal{I}^0$ for all loops Lemma 4.4.3 gives $\mathcal{I}_\ell(k_\ell)$ as stated in the theorem. Since by assumption $\mathbb{L} \subseteq \mathcal{I}^0$ we know that

$$\mathcal{I}_\ell(k_\ell) \supseteq \{i \in \mathbb{L} | i \neq k_q^{\text{next}} - k_\ell + r \cdot p,\ r \in \mathbb{Z},\ q \in \mathbb{L} \setminus \{\ell\}\}.$$

Further, since $\max \mathbb{L} \leq p$ we know that the set

$$\{i \in \mathbb{L} | i = k_q^{\text{next}} - k_\ell + r \cdot p,\ r \in \mathbb{Z}\}$$

contains at most one element for a given loop q. Thus,

$$\{i \in \mathbb{L} | i \neq k_q^{\text{next}} - k_\ell + r \cdot p,\ r \in \mathbb{Z},\ q \in \mathbb{L} \setminus \{\ell\}\}$$

contains at least one element as $q \in \mathbb{L} \setminus \{\ell\} \subset \mathbb{L}$. Hence $\mathcal{I}_\ell(k_\ell)$ always contains at least one element $\forall\, k_\ell$, and thus there always exists a feasible time to wait. □

Remark 4.4.6. The result in Lemma 4.4.3 requires the reservation of an infinite sequence. This is no longer required in Theorem 4.4.5 as all loops cooperate when choosing the set of feasible times to wait. In fact loop ℓ only needs to know the current time k_ℓ, the period p and the times when the other loops will transmit next $k_q^{\text{next}}\ \forall\, q \in \mathbb{L} \setminus \{\ell\}$ in order to find $\mathcal{I}_\ell(k_\ell)$.

Remark 4.4.7. If Assumption 4.4.4 holds and every loop on the network chooses $\mathcal{I}_\ell(k_\ell)$ according to Theorem 4.4.5, then it is guaranteed that at time k_ℓ we can reserve (4.15) and that no other loop can make conflicting reservations. Hence at time $k_\ell + I_\ell(k_\ell)$ the sequence

$$S_\ell(k_\ell + I_\ell(k_\ell)) = \{\ k_\ell + I_\ell(k_\ell) + p,\ k_\ell + I_\ell(k_\ell) + 2 \cdot p,\ k_\ell + I_\ell(k_\ell) + 3 \cdot p,\ \dots\ \}$$

is guaranteed to be available. Thus $p \in \mathcal{I}_\ell(k_\ell + I_\ell(k_\ell))$.

4.4.4 Stability

We continue with giving conditions for when the multiple-loop receding horizon control law described in Algorithm 4.2 is stabilizing. Extending the theory developed Section 4.3 to the multiple-loop case we may state the following.

Theorem 4.4.8. *Assume $0 < Q_\ell$, $0 < R_\ell$ and that (A_ℓ, B_ℓ) is controllable. Further let Assumption 4.4.4 hold. If we then choose $i \in \mathcal{I}_\ell(k) \subseteq \mathcal{I}^0 \subset \mathbb{N}^+$, with $\mathcal{I}_\ell(k)$ chosen as in Theorem 4.4.5, and $p = p^*$ given by*

$$p^* = \max\{i | i \in \mathcal{I}^0, \forall\, \ell\ \forall \lambda \in \lambda(A_\ell)\ \lambda^i \neq 1 \text{ if } \lambda \neq 1\}, \tag{4.16}$$

and apply Algorithm 4.2, then as $k \to \infty$

$$\frac{\alpha_\ell}{\gamma} \leq \min_{i \in \mathcal{I}_\ell(k)} \left(\frac{\alpha_\ell}{i} + \|x_\ell(k)\|^2_{P_\ell^{(i)}} \right) \leq \frac{\alpha_\ell}{\epsilon_\ell} \left(\frac{1}{p^*} - (1 - \epsilon_\ell) \frac{1}{\gamma} \right),$$

where $\gamma = \max \mathcal{I}^0$ and ϵ_ℓ is the largest value in the interval $(0, 1]$ which $\forall i \in \mathcal{I}^0$ fulfills

$$\left(A_\ell^{(i)} - B_\ell^{(i)} L_\ell^{(i)}\right)^T P_\ell^{(p^*)} \left(A_\ell^{(i)} - B_\ell^{(i)} L_\ell^{(i)}\right) \leq (1 - \epsilon_\ell) P_\ell^{(i)}.$$

Proof. Direct application of Theorem 4.3.5 on each loop. The corresponding proof carries through as the choice of $\mathcal{I}_\ell(k)$ together with the remaining assumptions guarantees that we may apply Theorem 4.4.2 on the feasible values of i in every step, thus the expression for V_k in (4.12) always exists. The critical step is that the upper bound on V_{k+i} in (4.13) must exist, *i.e.*, the choice $j = p^*$ must be feasible. This is also guaranteed by the choice of $\mathcal{I}_\ell(k)$, see Remark 4.4.7. □

Corollary 4.4.9. *Assume $0 < Q_\ell$, $0 < R_\ell$ and that (A_ℓ, B_ℓ) is controllable. Further, let Assumption 4.4.4 hold. In addition, let $\mathcal{I}^0$ be chosen so that the resulting $\gamma = \max \mathcal{I}^0$ guarantees that Assumption 4.3.7 holds for every loop ℓ or alternatively let $\alpha_\ell = 0$ for every loop ℓ. If we then choose $i \in \mathcal{I}_\ell(k) \subseteq \mathcal{I}^0 \subset \mathbb{N}^+$, with $\mathcal{I}_\ell(k)$ chosen as in Theorem 4.4.5, and $p = p^*$ given by (4.16) and apply Algorithm 4.2 it holds that*

$$\lim_{k\to\infty} x_\ell(k) = 0.$$

Proof. The proof follows from the results in Theorem 4.4.8 analogous to the proof of Corollary 4.3.9. □

4.5 Simulation Results

To illustrate the proposed theory we now continue with giving simulations. First we show how the control law works when one single loop is controlled over the network and focus on the loop specific mechanisms of the controller. Secondly we illustrate how the controller works when several loops are present on the network and focus on how the controller allocates network access to different loops.

4.5.1 Single-Loop

Let us exemplify and discuss how the controller handles the control performance versus communication rate trade-off in an individual loop. We do this by studying the case with a single system on the network. The system we study is the single integrator system which we discretize using sample and hold with sampling time $T_s = 1\,\mathrm{s}$ giving us $x(k+1) = Ax(k) + Bu(k)$ with $(A, B) = (1, 1)$. Since we want the resulting self-triggered MPC described in Algorithm 4.1 to be stabilizing we need to make sure that our design fulfills the conditions of Theorem 4.3.5. If we further want it to be asymptotically stabilizing, we in addition need it to fulfill the conditions of Corollary 4.3.9.

The design procedure is then as follows: First we note that the system (A, B) is controllable. The next step is to decide the weights $0 < Q$ and $0 < R$ in the quadratic cost function (4.3). This is done in the same way as in classical linear quadratic control, see *e.g.*, (Maciejowski, 2002). Here we for simplicity choose $Q = 1$ and $R = 1$. We note that the system only has the eigenvalue $\lambda = 1$, fulfilling Assumption 4.3.7, so that (4.10) in Theorem 4.3.5 gives $p^* = \max \mathcal{I}^0$. Hence Corollary 4.3.9, and thus Theorem 4.3.5, will hold for every choice of $\mathcal{I}^0$. This means that we may choose the elements in $\mathcal{I}^0$, *i.e.*, the possible down-sampling rates, freely. A natural way to choose them is to decide on a maximum allowed down-sampling rate and then choose $\mathcal{I}^0$ to contain all rates from 1 up to this number. Let's say that we here want the system to be sampled at least every $5 \cdot T_s$ s, then a good choice is $\mathcal{I}^0 = \{1, 2, 3, 4, 5\}$, giving $p^* = \max \mathcal{I}^0 = 5$.

Now having guaranteed that the conditions of Theorem 4.3.5 and Corollary 4.3.9 hold we have also guaranteed that the conditions of Theorem 4.2.4 are fulfilled.

i	$A^{(i)}$	$B^{(i)}$	$Q^{(i)}$	$R^{(i)}$	$N^{(i)}$	$L^{(i)}$	$P^{(i)}$	$V^{(i)}(x)$
1	1	1	1	1	0	0.70	1.70	$\alpha/1 + P^{(1)} \cdot x^2$
2	1	2	2	3	1	0.46	1.73	$\alpha/2 + P^{(2)} \cdot x^2$
3	1	3	3	8	3	0.35	1.89	$\alpha/3 + P^{(3)} \cdot x^2$
4	1	4	4	18	6	0.28	2.08	$\alpha/4 + P^{(4)} \cdot x^2$
$5{=}p^*$	1	5	5	35	10	0.23	2.30	$\alpha/5 + P^{(5)} \cdot x^2$

Table 4.3: The pre-computed control laws with related cost functions and intermediate variables.

Hence we may use it to compute the state-feedback gains and cost function matrices that are used in Algorithm 4.1. The results from these computations are shown in Table 4.3, together with some of the intermediate variables from Definition 4.2.1.

We see that the cost functions are quadratic functions in the state x where the coefficients $P^{(i)}$ are functions of Q and R. We also see that the cost to sample α/i enters linearly and as we change it we will change the offset level of the curves and thereby their values related to each other. However it will not affect the state-feedback gains. A graphical illustration of the cost functions in Table 4.3, for the choice $\alpha = 0.2$, is shown in Figure 4.4 together with the curve $I(k) = \arg\min_{\mathcal{I}^0} V^{(i)}(x(k))$, *i.e.*, the index of the cost function which has the lowest value for a given state $x(k)$. This is the partitioning of the state space that the self-triggered MPC controller will use to choose which of the state-feedback gains to apply and how long to wait before sampling again.

Applying our self-triggered MPC described in Algorithm 4.1 using the results in Table 4.3 to our integrator system when initialized in $x(0) = 2$ we get the response shown in Figure 4.5. Note here that the system will converge to the fixed sampling rate p^* as the state converges. It may now appear as it is sufficient to use periodic control and sample the system every $p^* \cdot T_s$ s to get good control performance. To compare the performance of this periodic sampling strategy with the self-triggered strategy above we apply the control which minimizes the same cost function (4.3) as above with the exception that the system now may only be sampled every $p^* \cdot T_s$ s. This is in fact the same as using the receding horizon control above while restricting the controller to choose $i = p^*$ every time. The resulting simulations are shown in Figure 4.6(a). As seen, there is a large degradation of the performance in the transient while the stationary behavior is almost the same. By this we can conclude that it is not sufficient to sample the system every $p^* \cdot T_s$ s if we want to achieve the same transient performance as with the self-triggered sampling.

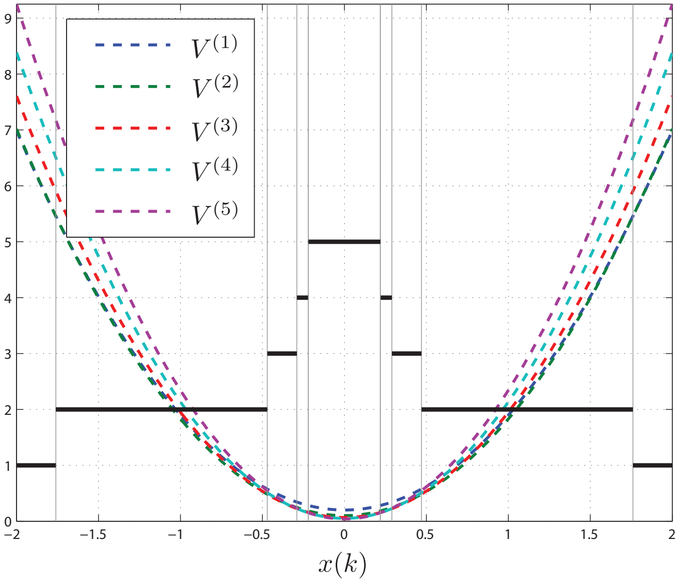


Figure 4.4: The cost functions $V^{(i)}(x(k))$ (dashed) together with the partitioning of the state space and the time to wait $I(k) = \arg\min_{\mathcal{I}^0} V^{(i)}(x(k))$ (solid).

In the initial transient response the self-triggered MPC controller sampled after one time instant. This indicates that there is performance to gain by sampling every time instant. To investigate this we apply the control which minimizes the same cost function (4.3), now with the exception that the system may be sampled every T_s s, *i.e.*, classical unconstrained linear quadratic control. Now simulating the system we get the response shown in Figure 4.6(b). As expected we get slightly better transient performance in this case compared to our self-triggered sampling scheme, it is however comparable. Note, however, that this improvement comes at the cost of a drastically increased communication need, which may not be suitable for systems where multiple loops share the same wireless medium.

From the above, we may conclude that our self-triggered MPC combines the low communication rate in stationarity of the slow periodic controller with the quick transient response of the fast periodic sampling. In fact we may, using our method, recover the transient behavior of fast periodic sampling at the communication cost of one extra sample compared to slow periodic sampling. The reason for this is that the fast sampling rate only is needed in the transient while we in stationarity can obtain sufficient performance with a lower rate.

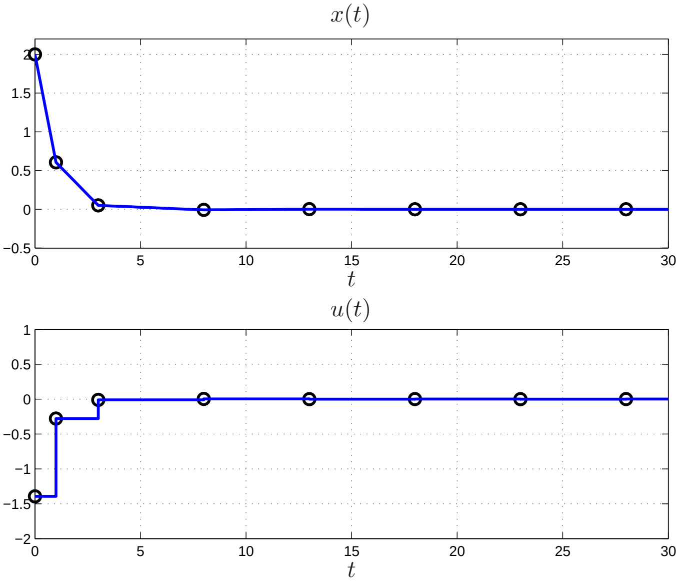


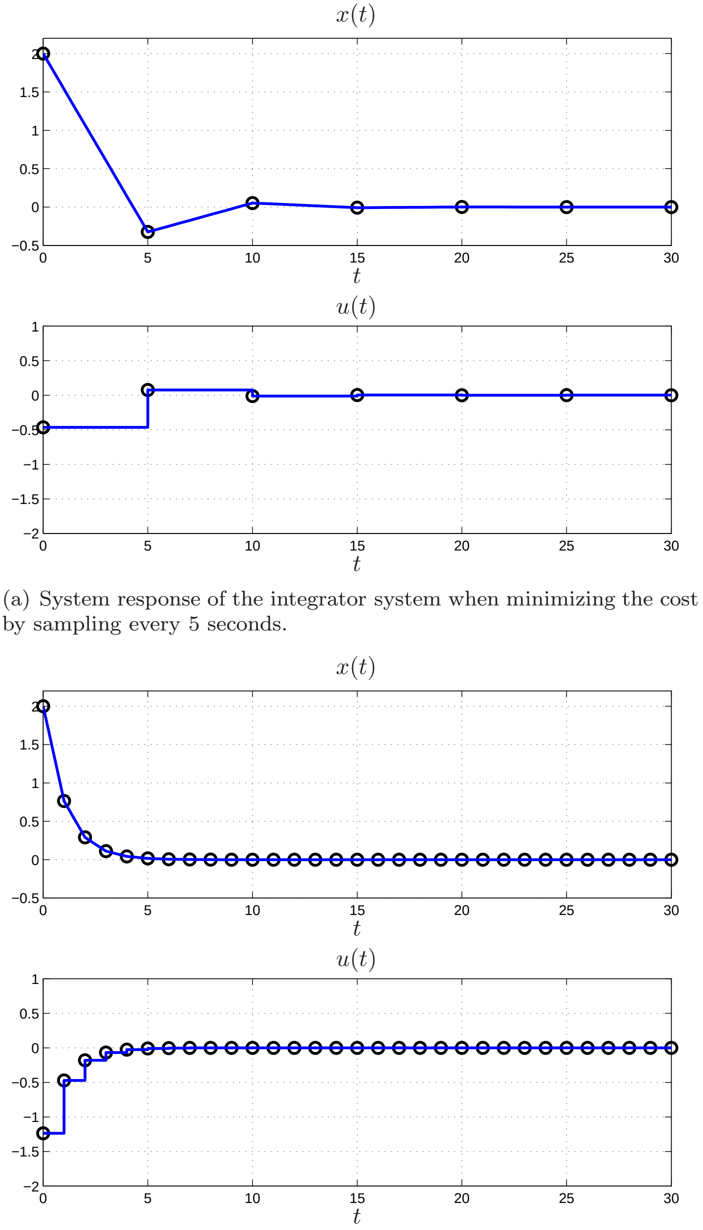
Figure 4.5: System response of the integrator system when minimizing the cost by using the proposed single-loop self-triggered MPC. System response is shown in blue with sampling instances encircled in black.

4.5.2 Multiple-Loop

We now continue with performing a simulation study where we control two systems over the same network. We will keep the integrator system from Section 4.5.1 now denoting it process $\mathcal{P}_1$ with dynamics $x_1(k+1) = A_1x_1(k) + B_1u_1(k)$ with $(A_1, B_1) = (1, 1)$ as before. In addition we will the control process $\mathcal{P}_2$ which is a double integrator system which we discretize using sample and hold with sampling time $T_s = 1\,\mathrm{s}$ giving

$$\underbrace{\begin{pmatrix} x_2^1(k+1) \\ x_2^2(k+1) \end{pmatrix}}_{x_2(k+1)} = \underbrace{\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}}_{A_2} \underbrace{\begin{pmatrix} x_2^1(k) \\ x_2^2(k) \end{pmatrix}}_{x_2(k)} + \underbrace{\begin{pmatrix} 1 \\ 0.5 \end{pmatrix}}_{B_2} u_2(k).$$

We wish to control these processes using our proposed multiple-loop self-triggered MPC described in Algorithm 4.2. As we wish to stabilize these systems we start by checking the conditions of Theorem 4.4.8 and Corollary 4.4.9. First we may easily verify that both the pairs (A_1, B_1) and (A_2, B_2) are controllable. To use the stability results we need Assumption 4.4.4 to hold, implying that we must choose $p_1 = p_2 = p$, $\mathcal{I}_1^0 = \mathcal{I}_2^0 = \mathcal{I}^0$ and choose $\mathcal{I}^0$ such that $\{1, 2\} \in \mathcal{I}^0$ and $2 \leq p$. For reasons of performance we wish to guarantee that the systems are sampled at



(a) System response of the integrator system when minimizing the cost by sampling every 5 seconds.

(b) System response of the integrator system when minimizing the cost by sampling every second.

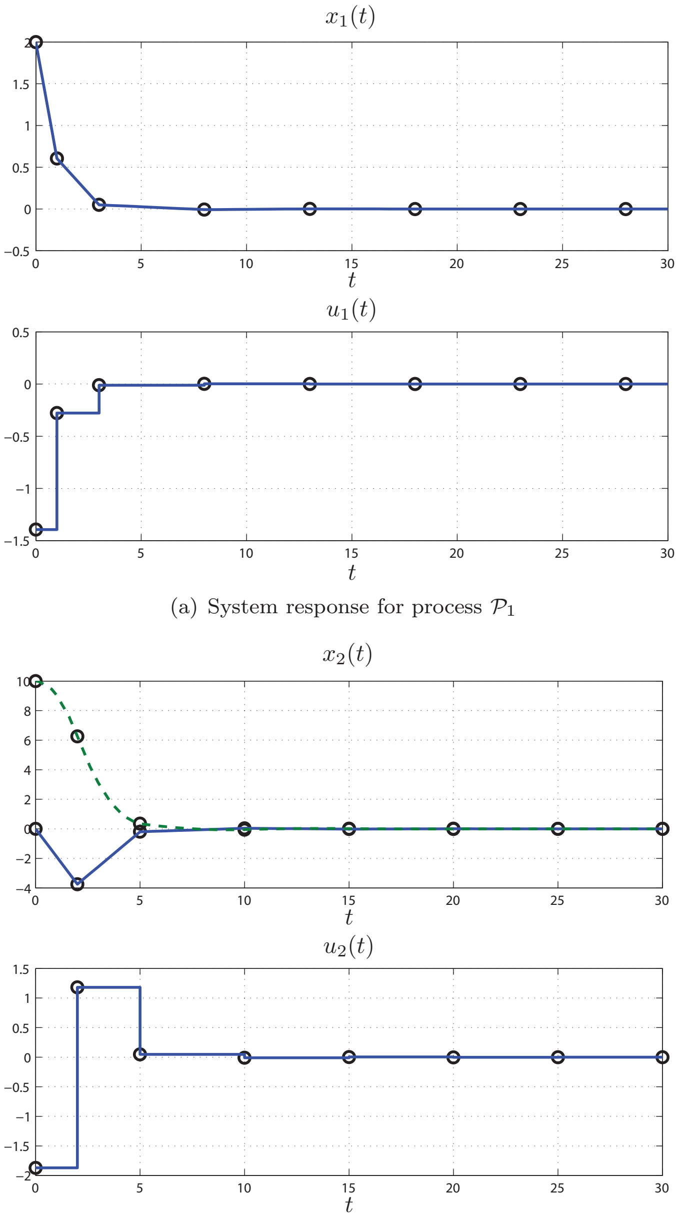
Figure 4.6: Comparing control performance for different periodic sampling policies. System response is shown in blue with sampling instances encircled in black.

least every $5 \cdot T_s$ s and therefore choose $\mathcal{I}_1^0 = \mathcal{I}_2^0 = \mathcal{I}^0 = \{1, 2, 3, 4, 5\}$ fulfilling the requirement above. We also note that $\lambda(A_1) = \{1\}$ and $\lambda(A_2) = \{1, 1\}$ and that hence both system fulfill Assumption 4.3.7 for this choice of $\mathcal{I}^0$, implying that (4.16) in Theorem 4.4.8 gives $p^* = \max \mathcal{I}^0 = 5$. Thus choosing $p = p^*$ as stated in Theorem 4.4.8 results in that Assumption 4.4.4 holds. What now remains to be decided are the weights α_ℓ, Q_ℓ and R_ℓ.

For the integrator process $\mathcal{P}_1$ we keep the same tuning as in Section 4.5.1 with $Q_1 = R_1 = 1$. Having decided Q_1, R_1, $\mathcal{I}^0$ and p^* we use Theorem 4.4.2 to compute the needed state-feedback gains and cost function matrices $(P_1^{(i)}, L_1^{(i)})$ $\forall\, i \in \mathcal{I}^0$ needed by Algorithm 4.2. We also keep $\alpha_1 = 0.2$ as it gave a good communication versus performance trade-off. For the double integrator process $\mathcal{P}_2$ the weights are chosen to be $Q_2 = I$ as we consider both states equally important and $R_2 = \frac{1}{10}$ to favor control performance and allow for larger control signals. Having decided Q_2, R_2, $\mathcal{I}^0$ and p^* we may use Theorem 4.4.2 to compute the needed state-feedback gains and cost function matrices $(P_2^{(i)}, L_2^{(i)})$ $\forall\, i \in \mathcal{I}^0$ needed by Algorithm 4.2. The sampling cost is chosen to be $\alpha_2 = 1$, as this gives a good trade-off between control performance and the number of samples.

We have now fulfilled all the assumptions of both Theorem 4.4.8 and Corollary 4.4.9. Hence applying Algorithm 4.2 choosing $\mathcal{I}_\ell(k_\ell)$ according to Theorem 4.4.5 will asymptotically stabilize both process $\mathcal{P}_1$ and $\mathcal{P}_2$. Controlling $\mathcal{P}_1$ and $\mathcal{P}_2$ using our multiple-loop self-triggered MPC described in Algorithm 4.2 with the above designed tuning we get the result shown in Figure 4.7. As expected, the behavior of the controller illustrated in Section 4.5.1 carries through also to the case when we have multiple loops on the network. In fact comparing Figure 4.5 showing how the controller handles process $\mathcal{P}_1$ when controlling it by itself on the network and Figure 4.7(a) which shows how $\mathcal{P}_1$ is handled in the multiple-loop case we see that they are the same. Further we see that, as expected, in stationarity the two loops controlling process $\mathcal{P}_1$ and $\mathcal{P}_2$ both converge to the sampling rate p^*.

As mentioned previously the controller uses the mechanism in Theorem 4.4.5 to choose the set of feasible times to wait until the next sample. In Figure 4.8 we can see how the resulting sets $\mathcal{I}_\ell(k_\ell)$ look in detail. At time $k = 0$ loop 1 gets to run Algorithm 4.2 first. As sensor $\mathcal{S}_2$ is not scheduled for any transmissions yet $\mathcal{I}_1(0) = \mathcal{I}^0$ from which the controller chooses $I_1(0) = 1$. Then loop 2 gets to run Algorithm 4.2 at time $k = 0$. As sensor $\mathcal{S}_1$ now is scheduled for transmission at time $k = 0 + I_1(0) = 1$, Theorem 4.4.5 gives $\mathcal{I}_2(0) = \mathcal{I}^0 \setminus \{1\}$. From which the controller chooses $I_2(0) = 2$. The process is then repeated every time a sample is transmitted to the controller, giving the result in Figure 4.8. As seen both the set $\mathcal{I}_\ell(\cdot)$ and the optimal time to wait $I_\ell(\cdot)$ converges to some fixed value as the state of the corresponding process $\mathcal{P}_\ell$ converges to zero.



(a) System response for process $\mathcal{P}_1$

(b) System response for process $\mathcal{P}_2$

Figure 4.7: The processes $\mathcal{P}_1$ and $\mathcal{P}_2$ controlled and scheduled on the same network using the proposed multiple-loop self-triggered MPC. System response is shown in blue and green with sampling instances encircled in black.

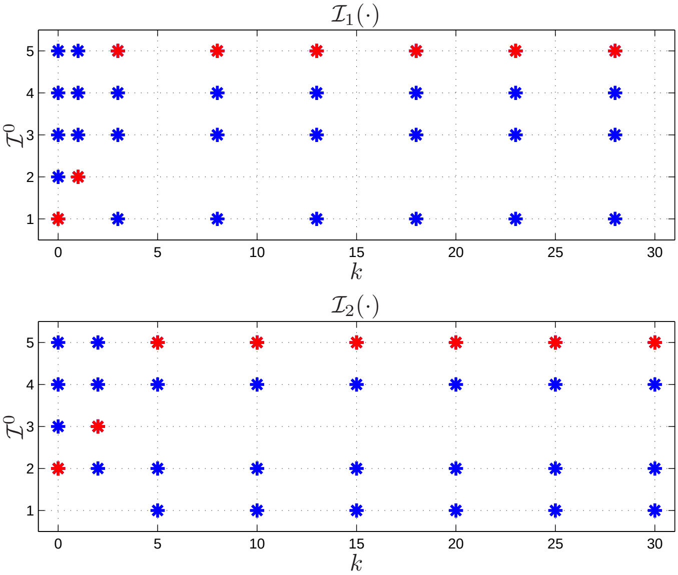


Figure 4.8: The sets $\mathcal{I}_\ell(\cdot)$ of feasible times to wait until the next sample for loop 1 and loop 2 (starred). The optimal time to wait $I_\ell(\cdot)$ is highlighted in red.

4.6 Summary

We have studied joint design of control and adaptive scheduling of multiple loops, and have presented a method which at every sampling instant computes the optimal control signal to be applied as well as the optimal time to wait before taking the next sample. It is shown that this control law may be realized using MPC and computed explicitly. The controller is also shown to be stabilizing under mild assumptions. Simulation results show that the use of the presented control law may help reducing the required amount of communication without almost any loss of performance compared to fast periodic sampling.

In the multiple-loop case we have also presented an algorithm for guaranteeing conflict-free transmissions. It is shown that under mild assumptions there always exists a feasible schedule for the network. In addition both the multiple-loop self-triggered MPC and the corresponding scheduling algorithm scales linearly in the number of loops.

Further it is worth noticing that the developed framework is not limited to just varying the time to the next sample i as in Figure 4.2. One could imagine optimizing over several of the initial inter sample times before reverting into sampling with period p. An interesting topic for future research is to further investigate the complexity and possible performance increase for such an extended formulation.

Chapter 5

Event-Triggered Model Predictive Control

This chapter presents an approach to event-triggered model predictive control (MPC) for discrete-time linear systems subject to input and state constraints as well as exogenous disturbances. Stability properties are derived by evaluating the difference between the event-triggered implementation and the conventional time-triggered scheme. It is shown that the event-triggered implementation, in stationarity, is able to keep the state in an explicitly computable set given by a disturbance bound and the event threshold. Simulation results underline the effectiveness of the proposed scheme in terms of reducing the communication and computational effort while guaranteeing a desired performance.

The outline of the chapter is as follows. In Section 5.1 the event-triggered control problem is stated and formulated as a receding horizon control problem. In Section 5.2 the corresponding open-loop optimal control problem is defined and solved. Section 5.3 describes the operation of the event-triggered MPC in further detail, and provides analysis of the event mechanism. Section 5.4 states the corresponding time-triggered MPC studied for comparison. Stability results for the two compared MPC schemes are given in Section 5.5. In Section 5.6 simulation results are presented. Finally, the chapter is summarized in Section 5.7.

5.1 Problem Formulation

We consider the problem of controlling the process $\mathcal{P}$ given by the linear time-invariant system

$$\begin{aligned} &x(k+1) = Ax(k) + Bu(k) + Ed(k), \\ &x(k) \in \mathbb{R}^n,\ u(k) \in \mathbb{R}^m,\ d(k) \in \mathbb{R}^r, \end{aligned} \tag{5.1}$$

where $d(k)$ is an unknown disturbance, over a communication network as in Figure 5.1. The process is controlled by the event-triggered controller $\mathcal{C}$. The controller works in the following way: When an event is triggered in the sensor $\mathcal{S}$ at time $k_\ell = k$, the sensor transmits the sample $x(k_\ell)$ to the controller $\mathcal{C}$. This sample is used by the controller to, together with an internal model of the process, compute

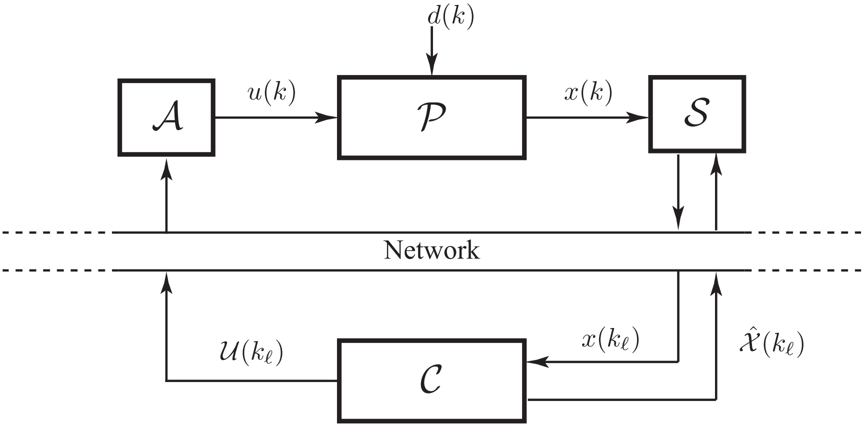


Figure 5.1: The event-triggered control loop.

an open-loop control sequence

$$\mathcal{U}(k_\ell) = \{\hat{u}(k_\ell|k_\ell), \hat{u}(k_\ell + 1|k_\ell), \ldots, \hat{u}(k_\ell + N - 1|k_\ell)\}$$

of predicted future appropriate control actions. The controller simultaneously also computes the corresponding sequence of the predicted values of the states

$$\hat{\mathcal{X}}(k_\ell) = \{\hat{x}(k_\ell|k_\ell), \hat{x}(k_\ell + 1|k_\ell), \ldots, \hat{x}(k_\ell + N|k_\ell)\}.$$

The control sequence $\mathcal{U}(k_\ell)$ is then sent to the actuator $\mathcal{A}$, which will use it as a play-out buffer and apply the control signal $u(k) = \hat{u}(k|k_\ell)$ to the process. The sequence of predicted states $\hat{\mathcal{X}}(k_\ell)$ is sent to the sensor which will monitor the process state $x(k)$ and compute the prediction error $|x(k) - \hat{x}(k|k_\ell)|$ at every time instant k. Whenever this value exceeds some predefined threshold, or if the sequence of predictions runs empty, the sensor will generate an event, and transmit the state $x(k_{\ell+1})$ to the controller. The controller will then compute and send updated trajectories $\mathcal{U}(k_{\ell+1})$ and $\hat{\mathcal{X}}(k_{\ell+1})$ to the actuator and sensor respectively.

Using this control mechanism the network is only utilized for transmission of information at event times k_ℓ. In addition the control law only needs to be re-evaluated at these event times. This saves communication bandwidth as well as computational effort. Provided that the control law is well-designed, so that events are only rarely generated, this gives a sparser communication pattern than traditional control algorithms, which communicate at every time instant k. We propose that the controller $\mathcal{C}$ should be implemented as a receding horizon controller that solves a finite-horizon open-loop optimal control problem at every sampling instant k_ℓ.

5.2 Open-Loop Optimal Control

The open-loop optimal control problem we propose for the controller to solve at every sampling instant k_ℓ is

$$\underset{\mathcal{U}(k_\ell)}{\text{minimize}}\ J\big(x(k_\ell),\mathcal{U}(k_\ell)\big) \tag{5.2}$$

with

$$\begin{aligned} J\big(x(k_\ell),\mathcal{U}(k_\ell)\big) &= \|\hat{x}(k_\ell+N|k_\ell)\|^2_{Q_N} \\ &\quad + \sum_{l=0}^{N-1}\Big(\|\hat{x}(k_\ell+l|k_\ell)\|^2_Q + \|\hat{u}(k_\ell+l|k_\ell)\|^2_R\Big), \end{aligned} \tag{5.3}$$

where the weighting matrices Q_N, Q and R as well as the horizon length $N \in \mathbb{N}^+$ are design variables. Further $0 \le Q_N$, $0 \le Q$ and $0 < R$ are symmetric matrices of appropriate dimensions. While solving (5.2) over

$$\mathcal{U}(k_\ell) = \{\hat{u}(k_\ell|k_\ell), \hat{u}(k_\ell+1|k_\ell), \dots, \hat{u}(k_\ell+N-1|k_\ell)\}, \tag{5.4}$$

the controller must satisfy the process model

$$\begin{aligned} \hat{x}(k_\ell+l+1|k_\ell) &= A\hat{x}(k_\ell+l|k_\ell) + B\hat{u}(k_\ell+l|k_\ell), \\ \hat{x}(k_\ell|k_\ell) &= x(k_\ell), \end{aligned} \tag{5.5}$$

and the constraints

$$x \in \mathbf{X} \subseteq \mathbb{R}^n, \quad u \in \mathbf{U} \subseteq \mathbb{R}^m. \tag{5.6}$$

Assumption 5.2.1. *The disturbance $d(k)$ belongs to $\mathbf{D}$, where $\mathbf{D} \subset \mathbb{R}^r$ is a compact set. Further, the optimization problem* (5.2)–(5.6) *is feasible at every sampling instant k_ℓ.*

Remark 5.2.2. If $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{D}$ are compact and convex polyhedra, the set of initial conditions $\mathbf{X}_0 \subseteq \mathbf{X}$, such that (5.2)–(5.6) is feasible for all $x(k_\ell) \in \mathbf{X}_0$, is computable. Further, $\mathbf{X}_0$ is a compact convex polyhedron. Conditions for feasibility at every sampling instant can be derived. See (Kerrigan, 2000; Kerrigan and Maciejowski, 2001) for details.

Remark 5.2.3. If $\mathbf{D}$ is compact and $\mathbf{X} = \mathbb{R}^n$, then the set of feasible initial conditions is $\mathbf{X}_0 = \mathbb{R}^n$. Further, the optimization problem (5.2)–(5.6) is feasible at every sampling instant k_ℓ (Kerrigan, 2000; Kerrigan and Maciejowski, 2001).

Let Assumption 5.2.1 hold. Then the optimal solution,

$$\mathcal{U}^*(k_\ell) = \arg\min_{\mathcal{U}(k_\ell)}\ J\big(x(k_\ell),\mathcal{U}(k_\ell)\big),$$

exists and we may characterize it by the non-linear operator G as

$$G\big(\hat{x}(k_\ell+l|k_\ell)\big) \triangleq \hat{u}^*(k_\ell+l|k_\ell), \quad l \in \{0,1,\dots,N-1\}.$$

Definition 5.2.4. *The constraints are said to be inactive if $\hat{x}(k+l|k)$ and $\hat{u}(k+l|k)$ belong to the interior of their constraint sets,* i.e.,

$$\hat{x}(k+l|k) \in \operatorname{int}(\mathbf{X}), \quad G\big(\hat{x}(k+l|k)\big) \in \operatorname{int}(\mathbf{U}), \quad l \in \{0,1,\ldots,N-1\}.$$

We may now characterize the solution of our open-loop optimization problem.

Lemma 5.2.5. *Let the constraints be inactive. Then it holds that*

$$G\big(\hat{x}(k+l|k)\big) = K_l \hat{x}(k+l|k), \quad l \in \{0,1,\ldots,N-1\}.$$

where

$$\begin{aligned} P_l &= A^T P_{l+1} A + Q + A^T P_{l+1} B K_l, \quad P_N = Q_N \\ K_l &= -(B^T P_{l+1} B + R)^{-1} B^T P_{l+1} A. \end{aligned}$$

Proof. See (Bertsekas, 1995). □

Lemma 5.2.6. *Let the constraints be inactive. Further let Q_N satisfy the Riccati Equation*

$$Q_N = A^T Q_N A + Q - A^T Q_N B (B^T Q_N B + R)^{-1} B^T Q_N A.$$

Then it holds that

$$G\big(\hat{x}(k+l|k)\big) = K \hat{x}(k+l|k), \quad l \in \{0,1,\ldots,N-1\}$$

where

$$K = -(B^T Q_N B + R)^{-1} B^T Q_N.$$

Further it holds that $A+BK$ is Schur.

Proof. See (Bertsekas, 1995). □

5.3 Event-Triggered MPC

Information is sent over the feedback link only if the event condition, which is discussed next, is satisfied. The time instants at which this happens are denoted by k_ℓ, where $\ell \in \mathbb{N}$ is the event counter. In the following it is assumed that the first event $\ell = 0$ occurs at time $k_0 = 0$.

An event is generated at time $k_{\ell+1}$ whenever either the difference between the process state $x(k)$ and the state $\hat{x}(k|k_\ell)$ predicted by the MPC exceeds a certain threshold, or the prediction horizon N has expired, *i.e.*, when

$$|x(k) - \hat{x}(k|k_\ell)| \geq \bar{e} \text{ or } k \geq k_\ell + N, \tag{5.7}$$

where $\bar{e} \geq 0$ is the threshold parameter chosen by design.

From this we may formulate our event-triggered MPC controller as in Algorithm 5.1. The memory variable k_{latest} is the time instant when the latest event occurred.

Algorithm 5.1 Event-Triggered MPC

```
$k := 0$
while $k < \infty$ do
  if $|x(k) - \hat{x}(k|k_{\text{latest}})| \geq \bar{e}$ or $k \geq k_{\text{latest}} + N$ or $k = 0$ then
  1.      The sample $x(k)$ of process $\mathcal{P}$ is transmitted by the sensor $\mathcal{S}$ to the
          controller $\mathcal{C}$.

  2.      Using $x(k)$ the controller $\mathcal{C}$ computes

              $\mathcal{U}^*(k) = \arg\min_{\mathcal{U}(k)} J\big(x(k), \mathcal{U}(k)\big)$

          and the resulting predicted state trajectory $\hat{\mathcal{X}}(k)$.

  3.      The time instant of the latest event $k_{\text{latest}}$ is assigned the value of the
          current time, i.e., $k_{\text{latest}} := k$.

  4.  a)      The controller $\mathcal{C}$ sends $\mathcal{U}^*(k_{\text{latest}})$ to the actuator $\mathcal{A}$ which applies
              $u(k) = \hat{u}^*(k|k_{\text{latest}}) = \hat{u}^*(k|k)$ to the process $\mathcal{P}$ given by (5.1).

      b)      The controller $\mathcal{C}$ sends $\hat{\mathcal{X}}(k_{\text{latest}})$ to the actuator $\mathcal{A}$ which now
              uses these predictions to check the event condition.
  else
    The actuator $\mathcal{A}$ applies $u(k) = \hat{u}^*(k|k_{\text{latest}})$, contained in the latest received
    $\mathcal{U}^*(k_{\text{latest}})$, to the process $\mathcal{P}$ given by (5.1).
  end if
  $k := k + 1$
end while
```

Introducing a change of variable $k = k_\ell + l$ with $l \in \{0, 1, \dots, N\}$ the prediction error between two event times is given by

$$e(l, k_\ell) = x(k_\ell + l) - \hat{x}(k_\ell + l|k_\ell). \tag{5.8}$$

Letting $d_{\max}$ be the worst disturbance over all times,

$$d_{\max} = \max_{d \in \mathbf{D}} |d|,$$

we may bound the error according to the following theorem.

Theorem 5.3.1. *The prediction error $e(l, k_\ell)$ in (5.8) is bounded as*

$$|e(l, k_\ell)| \leq e_{\max}, \ \forall k_\ell, \ l \in \{0, 1, \dots, N\}$$

where

$$e_{\max} = \|A\|\bar{e} + \|E\|d_{\max}.$$

Proof. The evolution of the prediction error is given by

$$\begin{aligned} e(l+1,k_\ell) &= x(k_\ell+l+1) - \hat{x}(k_\ell+l+1|k_\ell) \\ &= Ax(k_\ell+l) + Bu(k_\ell+l) + Ed(k_\ell+l) - A\hat{x}(k_\ell+l|k_\ell) - Bu(k_\ell+l) \\ &= Ae(l,k_\ell) + Ed(k_\ell+l), \end{aligned}$$

with $e(0,k_\ell) = 0$ according to (5.5) and (5.8).

Consider a given time instant $k' = k_\ell + l'$ such that no event is generated, *i.e.*, $|e(l',k_\ell)| < \bar{e}$. There always exists such a k' as $l' = 0$ is allowed. Now assume that $d(k_\ell + l')$ is such that the error at the next time instant $|e(l'+1,k_\ell)| \geq \bar{e}$, so that an event is generated. Then largest value this error could take is

$$\begin{aligned} \max_{d(k_\ell+l')} |e(l'+1,k_\ell)| &= \max_{d(k_\ell+l')} |Ae(l',k_\ell) + Ed(k_\ell+l')| \\ &\leq \max_{d(k_\ell+l')} |Ae(l',k_\ell)| + \max_{d(k_\ell+l')} |Ed(k_\ell+l')| \\ &\leq \|A\|\bar{e} + \|E\|d_{\max} = e_{\max}. \end{aligned}$$

□

5.4 Time-Triggered MPC

For comparison with our proposed event-triggered MPC we also study the behavior of classical time-triggered MPC, whose operation is described in Figure 5.2. As seen in the figure, the state is here communicated every time instant k. Letting the time-triggered MPC solve the same open-loop optimal control problem as our proposed event-triggered MPC gives Algorithm 5.2.

Algorithm 5.2 Time-Triggered MPC

```
$k := 0$
while $k < \infty$ do
  1. The sample $x(k)$ of process $\mathcal{P}$ is transmitted by the sensor $\mathcal{S}$ to the
     controller $\mathcal{C}_T$.
  2. Using $x(k)$ the controller $\mathcal{C}_T$ computes
        $\mathcal{U}^*(k) = \arg\min_{\mathcal{U}(k)} J\big(x(k),\mathcal{U}(k)\big).$
  3. The controller $\mathcal{C}_T$ sends $\hat{u}^*(k|k)$ to the actuator $\mathcal{A}$ which applies $u(k) =$
     $\hat{u}^*(k|k)$ to the process $\mathcal{P}$ given by (5.1).
  4. $k := k + 1$
end while
```

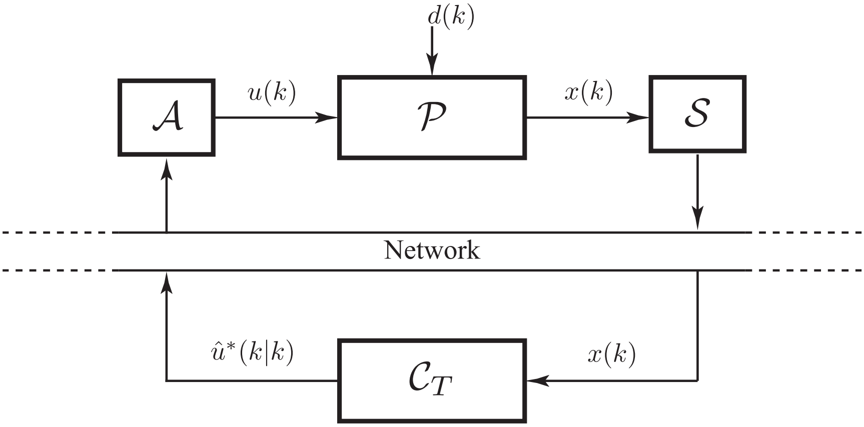


Figure 5.2: The time-triggered control loop.

5.5 Stability Analysis

We continue with studying the stability of both the time-triggered MPC and our proposed event-triggered MPC. It turns out that we may use the results from the time-triggered MPC to characterize our event-triggered scheme.

5.5.1 Time-Triggered MPC

Theorem 5.5.1. *Let Assumption 5.2.1 hold. Further let there exist K such that*

1. $\lim_{k\to\infty} \left|G\big(\hat{x}(k|k)\big) - K\hat{x}(k|k)\right| = 0$
2. *$A + BK$ is Schur.*

Then, applying Algorithm 5.2 stabilizes the closed-loop system. Further it holds that

$$\lim_{k\to\infty} |x(k)| \le r_t(d_{\max}),$$

where

$$r_t(d_{\max}) = \sum_{i=0}^{\infty} \left\| \big(A+BK\big)^i E \right\| d_{\max}. \tag{5.9}$$

Proof. According to Assumption 5.2.1 the problem is feasible and the controller is able to keep the system within its specified constraints. With $u(k) = G\big(\hat{x}(k|k)\big)$ and $\hat{x}(k|k) = x(k)$ at each sampling instant k, the closed-loop system is given by

$$\begin{aligned} x(k+1) &= Ax(k) + BG\big(\hat{x}(k|k)\big) + Ed(k) \\ &= Ax(k) + BK\hat{x}(k|k) + Ed(k) + B\Big(G\big(\hat{x}(k|k)\big) - K\hat{x}(k|k)\Big) \\ &= \big(A+BK\big)x(k) + Ed(k) + B\Big(G\big(\hat{x}(k|k)\big) - K\hat{x}(k|k)\Big). \end{aligned}$$

As $k \to \infty$, the state $x(k)$ is given by

$$\lim_{k\to\infty} |x(k)| = \lim_{k\to\infty} \left| \bar{A}^k x_0 + \sum_{j=0}^{k-1} \bar{A}^{k-1-j} Ed(j) + \sum_{j=0}^{k-1} \bar{A}^{k-1-j} B\Delta(j,j) \right|$$

where $\bar{A} = A + BK$ and $\Delta(j,j) = G\big(\hat{x}(j|j)\big) - K\hat{x}(j|j)$. By assumption we have that $\lim_{k\to\infty} \left\|\bar{A}^k\right\| = 0$ and $\lim_{k\to\infty} |\Delta(k,k)| = 0$. Using this we have

$$\lim_{k\to\infty} |x(k)| \leq \lim_{k\to\infty} \sum_{j=0}^{k-1} \left|\bar{A}^{k-1-j} Ed(j)\right| \leq \lim_{k\to\infty} \sum_{j=0}^{k-1} \left\|\bar{A}^{k-1-j} E\right\| d_{\max}.$$

With the change of variables $i = k - 1 - j$, this gives

$$\lim_{k\to\infty} |x(k)| \leq \sum_{i=0}^{\infty} \left\|\bar{A}^i E\right\| d_{\max},$$

as stated in the theorem. □

Remark 5.5.2. If Algorithm 5.2 is able to drive the system into a region where the constraints are inactive, Lemma 5.2.5 implies that the first requirement on K in Theorem 5.5.1 is guaranteed to hold, with $K = K_0$. However, it does not guarantee that $A + BK$ is Schur. This property may be inferred by restricting Q_N to fulfill certain properties, *cf.*, Lemma 5.2.6.

Remark 5.5.3. For robust MPC the reader is referred to (Rossiter et al., 1998; Chisci et al., 2001; Marruedo et al., 2002; Mayne et al., 2005; Kim et al., 2006; Trodden and Richards, 2006; Limon et al., 2010), where the convergence analysis is primarily carried out by showing that a Lyapunov function decreases over time despite the influence of unknown disturbances.

5.5.2 Event-Triggered MPC

Theorem 5.5.4. *Let Assumption 5.2.1 hold. Further let there exist K such that*

1. $\lim_{k\to\infty} \left|G\big(\hat{x}(k|k_\ell)\big) - K\hat{x}(k|k_\ell)\right| = 0$
2. *$A + BK$ is Schur.*

Then, applying Algorithm 5.1 stabilizes the closed-loop system. Further it holds that

$$\lim_{k\to\infty} |x(k)| \leq r_e(d_{\max}, \bar{e}),$$

where

$$\begin{aligned} r_e(d_{\max}, \bar{e}) &= \sum_{i=0}^{\infty} \left\| \big(A + BK\big)^i E \right\| d_{\max} + \sum_{i=0}^{\infty} \left\| \big(A + BK\big)^i BK \right\| e_{\max} \\ &= r_t(d_{\max}) + r_\delta(d_{\max}, \bar{e}). \end{aligned}$$

Proof. According to Assumption 5.2.1 the problem is feasible and the controller is able to keep the system within its specified constraints. The closed-loop system is given by

$$\begin{aligned} x(k+1) &= Ax(k) + BG\big(\hat{x}(k|k_\ell)\big) + Ed(k) \\ &= Ax(k) + BK\hat{x}(k|k_\ell) + Ed(k) + B\Big(G\big(\hat{x}(k|k_\ell)\big) - K\hat{x}(k|k_\ell)\Big). \end{aligned}$$

Using the change of variables $k = k_\ell + l$ and the prediction error (5.8), this may be written as

$$x(k+1) = \big(A + BK\big)x(k) + Ed(k) + B\Big(G\big(\hat{x}(k|k_\ell)\big) - K\hat{x}(k|k_\ell)\Big) - BKe(l, k_\ell).$$

As $k \to \infty$, the state $x(k)$ is given by

$$\begin{aligned} \lim_{k\to\infty} |x(k)| = \lim_{k\to\infty} \Bigg| &\bar{A}^k x_0 + \sum_{j=0}^{k-1} \bar{A}^{k-1-j} Ed(j) \\ &+ \sum_{j=0}^{k-1} \bar{A}^{k-1-j} B\Delta(j, k_\ell) - \sum_{j=0}^{k-1} \bar{A}^{k-1-j} BKe(j - k_\ell, k_\ell) \Bigg| \end{aligned}$$

where $\bar{A} = A + BK$ and $\Delta(j, k_\ell) = G\big(\hat{x}(j|k_\ell)\big) - K\hat{x}(j|k_\ell)$. By assumption we have that $\lim_{k\to\infty} \left\|\bar{A}^k\right\| = 0$ and $\lim_{k\to\infty} |\Delta(j, k_\ell)| = 0$. Using this we have

$$\lim_{k\to\infty} |x(k)| \leq \lim_{k\to\infty} \sum_{j=0}^{k-1} \left|\bar{A}^{k-1-j} Ed(j)\right| + \lim_{k\to\infty} \sum_{j=0}^{k-1} \left|\bar{A}^{k-1-j} BKe(j - k_\ell, k_\ell)\right|.$$

By using Theorem 5.3.1, this can be bounded by

$$\lim_{k\to\infty} |x(k)| \leq \lim_{k\to\infty} \sum_{j=0}^{k-1} \left\|\bar{A}^{k-1-j} E\right\| d_{\max} + \lim_{k\to\infty} \sum_{j=0}^{k-1} \left\|\bar{A}^{k-1-j} BK\right\| e_{\max}.$$

With the change of variables $i = k - 1 - j$, this gives

$$\lim_{k\to\infty} |x(k)| \leq \sum_{i=0}^{\infty} \left\|\bar{A}^i E\right\| d_{\max} + \sum_{i=0}^{\infty} \left\|\bar{A}^i BK\right\| e_{\max},$$

as stated in the theorem. □

Remark 5.5.5. If Algorithm 5.1 is able to drive the system into a region where the constraints are inactive, Lemma 5.2.6 gives a method to choose Q_N which results in a K fulfilling the requirements in Theorem 5.5.4.

Remark 5.5.6. Theorem 5.5.4 shows how the event-triggered MPC approximates the time-triggered implementation through the parameter $e_{\max}$ depending on the event threshold $\bar{e}$ and the maximal disturbance $d_{\max}$. By increasing $\bar{e}$ the event condition (5.7) is trigged less often leading to a lower communication rate, however the bound $r_\delta(d_{\max}, \bar{e})$ is increased. By lowering $\bar{e}$ the event condition is trigged more often leading to a higher communication rate, resulting in a lowering of the bound $r_\delta(d_{\max}, \bar{e})$.

Corollary 5.5.7. *Let the assumptions of Theorem 5.5.4 hold. Further let*

$$|Ed(k)| \geq \bar{e}, \ \forall k.$$

Then, applying Algorithm 5.1 stabilizes the closed-loop system. Further it holds that

$$\lim_{k\to\infty} |x(k)| \leq r_t(d_{\max})$$

where $r_t(d_{\max})$ *is given by* (5.9) *in Theorem 5.5.1.*

Proof. Assume that an event has been generated at time k_ℓ and hence $\hat{x}(k_\ell|k_\ell) = x(k_\ell)$. A new event is detected at the next time instant $k_\ell + 1$ if

$$|e(1, k_\ell)| = |x(k_\ell + 1) - \hat{x}(k_\ell + 1|k_\ell)| = |Ed(k_\ell)| \geq \bar{e}$$

holds. If $d(k_\ell)$ satisfies this condition for all $k_\ell \in \mathbb{N}$, an event is generated at each time instant k. Therefore, the event-triggered implementation in this case applies $u(k) = G\big(\hat{x}(k|k)\big)$ to the process (5.1), just as the time-triggered MPC does. Thus recovering the same bound on $x(k)$. □

Remark 5.5.8. The result in Corollary 5.5.7 shows that making the event threshold small enough with respect to the disturbance $d(k)$ results in that the event-triggered MPC recovers the performance and behavior of the time-triggered MPC. It follows from the proof that the event-triggered MPC in this case gives the same controls as the time-triggered MPC.

Remark 5.5.9. Methods to analyze the feasibility and convergence properties of event-triggered MPC subject to exogenous disturbances can be found in (Eqtami et al., 2011; Bernardini and Bemporad, 2012).

5.5.3 Relations to Model-Based Event-Triggered Control

Without constraints, the optimal control problem (5.2)–(5.6) is equivalent to a state-feedback controller which can be explicitly determined, see Lemma 5.2.5.

By incorporating the MPC component on the actuator node this method shows some interesting relations to existing event-triggered control schemes denoted model-based event-triggered control, see (Lunze and Lehmann, 2010; Garcia and Antsaklis, 2011) for the continuous-time case and (Grüne et al., 2010) for the discrete-time scenario. In fact, when considering no constraints the input provided by the MPC is the same as the input provided by a time-triggered state-feedback loop when the controller matrix K has been obtained from the optimal control problem (5.2)–(5.6).

5.6 Simulation Evaluation

To illustrate the proposed theory we proceed with a simulation study. We study the control of the process

$$x(k+1) = \begin{pmatrix} 1 & -0.5 \\ 0.5 & 0 \end{pmatrix} x(k) + \begin{pmatrix} 0.5 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0.25 \\ 0 \end{pmatrix} d(k), \quad x(0) = x_0$$

subject to the constraints

$$-2 \leq u \leq 2, \qquad -10 \leq x_i \leq 10,\ i = 1, 2.$$

The prediction horizon is chosen to be $N = 10$ with the weights

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R = 0.1.$$

From this we may use Lemma 5.2.6 to compute a suitable final state weight Q_N and corresponding state-feedback gain K resulting in

$$Q_N = \begin{pmatrix} 1.554 & -0.151 \\ -0.151 & 1.080 \end{pmatrix}, \quad K = \begin{pmatrix} 1.513 & -0.795 \end{pmatrix}.$$

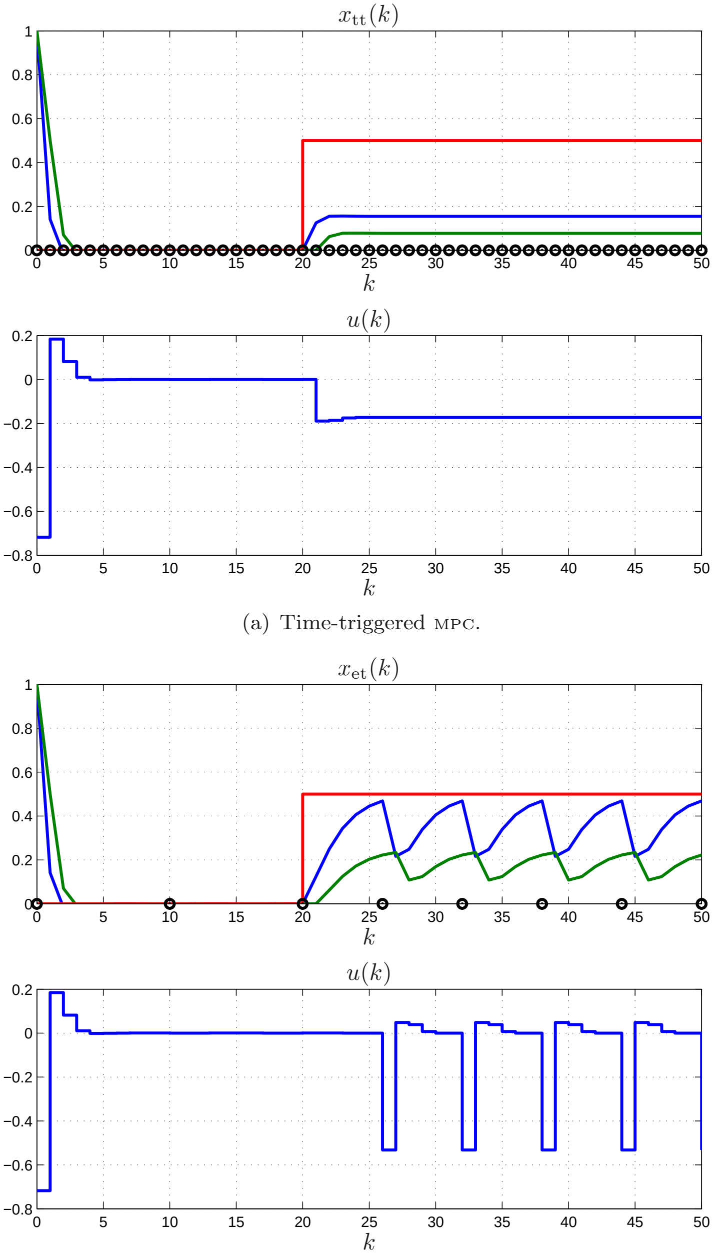
The event threshold for the event-triggered implementation is set to $\bar{e} = 0.5$. Considering these parameters the assumptions in Theorem 5.5.1 and Theorem 5.5.4 are satisfied.

5.6.1 Process subject to Small Disturbances

First we study the case when the system is subject to a small disturbance

$$d(k) = \begin{cases} 0, & \text{for } 0 \leq k \leq 20, \\ 0.5, & \text{for } k > 20. \end{cases}$$

The resulting behavior for the time-triggered MPC is given in Figure 5.3(a) and for the event-triggered MPC in Figure 5.3(b). For $k \leq 20$ the disturbance is zero and, hence, the state-trajectory $x_{\text{et}}(k)$ of the event-triggered scheme and the state-trajectory $x_{\text{tt}}(k)$ of the time-triggered scheme coincide as the state predicted by the event-triggered MPC and the state measured by the time-triggered MPC are the same. The two events generated in the event-triggered scenario during this period, at $k_1 = 10$ and $k_2 = 20$, are due to that the prediction horizon of the event-triggered MPC expires. At $k = 20$, the magnitude of the disturbance changes and the following events are caused by the deviation of the predicted state versus the measured state. As seen in Figure 5.4 this occurs every sixth time step.



(a) Time-triggered MPC.

(b) Event-triggered MPC.

Figure 5.3: Behavior of event-triggered and time-triggered MPC for a small disturbance. States (blue and green), disturbance (red) and event times (black).

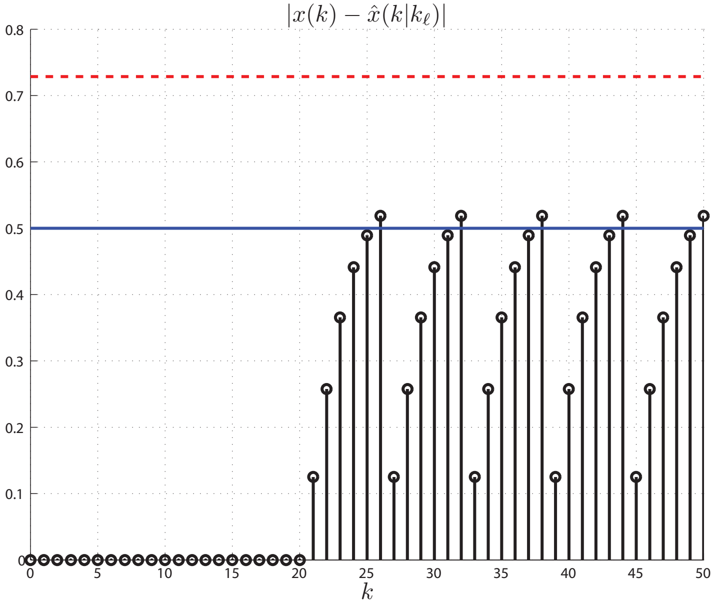


Figure 5.4: The prediction error $|x(k) - \hat{x}(k|k_\ell)|$ (black), the event threshold $\bar{e}$ (blue) and the prediction error bound $e_{\max}$ (red).

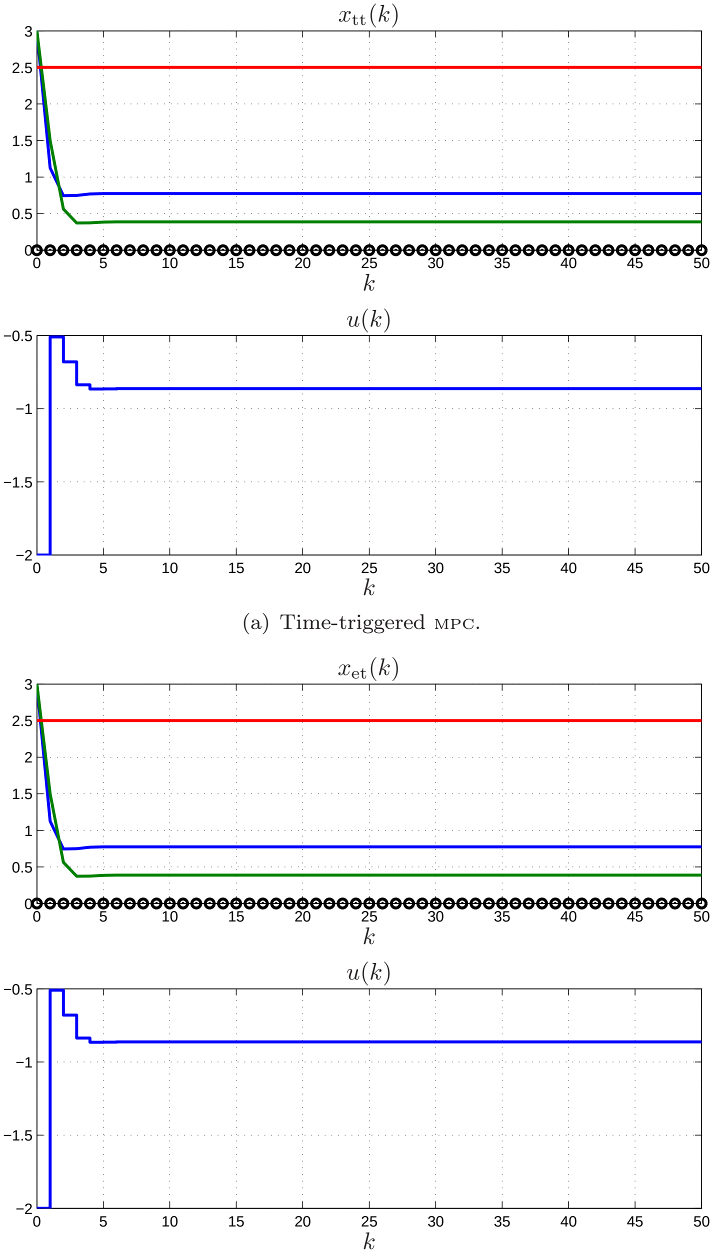
As seen in Figure 5.4 and expected from Theorem 5.5.4 we, for large k, have that the upper bound on $|x_{\text{et}}(k)|$, corresponding to the event-triggered MPC, is larger than the upper bound on $|x_{\text{tt}}(k)|$, corresponding to the time-triggered MPC. Here one should keep in mind that the event-triggered algorithm only uses 17% of the communication and computational resources compared to the time-triggered MPC, as it only communicates every sixth time instant.

5.6.2 Process subject to Large Disturbances

We continue our study with the case where the system is subject to a large disturbance

$$d(k) = 2.5, \forall\, k \;\Rightarrow\; |Ed(k)| = 0.625 \geq \bar{e}, \forall\, k.$$

The resulting behavior when controlling the process using time-triggered MPC is given in Figure 5.5(a), and the behavior using event-triggered MPC is given in Figure 5.5(b). As expected from Corollary 5.5.7 they give the same performance, as the event-triggered MPC communicates and re-computes the control law every time instant k.



(a) Time-triggered MPC.

(b) Event-triggered MPC.

Figure 5.5: Behavior of event-triggered and time-triggered MPC for a large disturbance. States (blue and green), disturbance (red) and event times (black).

5.7 Summary

The chapter investigated the stationary behavior of event-triggered MPC and evaluated the difference to a conventional time-triggered implementation. It was shown how the event-triggered approach is affected by the event condition as well as the disturbance magnitude. The analytical results were illustrated by simulations.

Chapter 6

Model Predictive Control based on Wireless Sensor Feedback

Here the design and experimental validation of a control system with both wireless sensor and actuator links is presented. The control system is designed for, and the experiments are performed on, a laboratory process built at the University of Siena, Italy. The process, shown in Figure 6.1, consists of a transport belt where moving parts equipped with wireless sensors are heated by four infrared lamps.

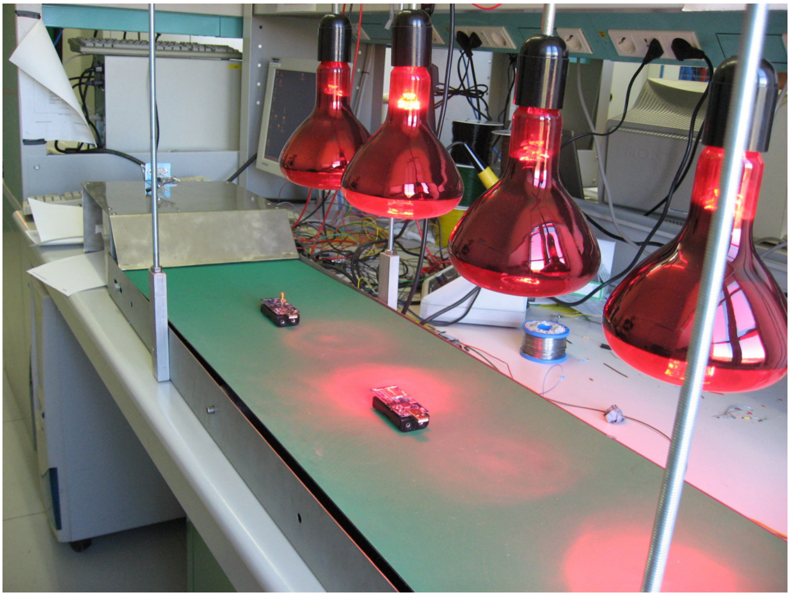


Figure 6.1: The controlled laboratory process.

The studied process is motivated by heating processes in the plastic and printing industry, where one wants to move parts over a transport belt and at the same time have them follow a specific temperature profile.

The process is actuated by moving the transport belt and by switching the heating lamps on or off. This switching property introduces interesting hybrid dynamics in the process, which we will handle using hybrid model predictive control (MPC), introduced in (Bemporad and Morari, 1999). The reason for using hybrid MPC is that it explicitly takes the hybrid nature of the system into account as well as handles physical constraints on states and inputs.

Since MPC is computationally intensive, the amount of computational power required can not be assumed to be available close to the process. Instead a wireless control structure will support the de-localization of the MPC to a remote computer able to handle the computations. Both the measurements from the process to the MPC and the control signals from the MPC to the process will be transmitted over wireless links. For this, a particular control systems architecture will be used.

The chapter is outlined as follows. First the process is described in further detail and a model is developed to be used for the MPC design. After that, the control system architecture is presented, together with the control design. Following that, the details regarding the physical implementation are presented. The chapter is then concluded by simulations and results from experiments on the physical control system.

6.1 Process Description and Modelling

We start by describing the laboratory process, and derive a control and estimation oriented hybrid dynamical model of the same.

6.1.1 Physical Process

The main components of the process, whose schematics are shown in Figure 6.2, are the belt actuated by a motor equipped with an angular encoder, four heating lamps placed over the belt and a mote placed on the belt, emulating the part to be heat treated. The heating lamps are placed in a row and two on–off switches are available to actuate them. The first switch controls lamps 1 and 3, the second switch, lamps 2 and 4. The lamps are grouped to reduce the complexity of the model and of the control algorithm. The mote is a temperature sensor equipped with a radio device able to transmit its temperature reading.

To derive a dynamic model of the process, experiments where performed which showed that the system is governed by the differential equations

$$\begin{aligned} \dot{T}_1 &= -\alpha\big(T_1 - T_{ss}(p, u_1, u_2)\big), \\ \dot{T}_2 &= -\beta\big(T_2 - T_1\big), \\ \dot{p} &= \gamma(u_c), \end{aligned} \tag{6.1}$$

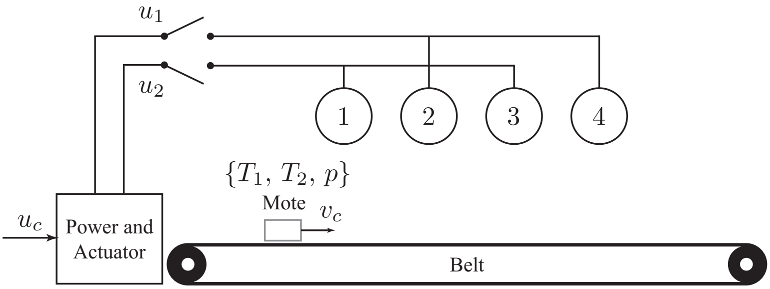


Figure 6.2: Schematics of the process.

where $T_1 \in \mathbb{R}$ is interpreted as the sensor casing temperature, $T_2 \in \mathbb{R}$ is interpreted as the sensor temperature and $p \in \mathbb{R}$ is the position of the mote on the belt. The control inputs are $u_c \in \mathbb{R}$, $u_1, u_2 \in \{0, 1\}$, affecting the system through the static nonlinearity $T_{ss} : \mathbb{R}^3 \to \mathbb{R}$. The parameters $\alpha, \beta > 0$ are physical constants, identified from the experimental data. The continuous signal $v_c = \gamma(u_c)$ corresponds to the mote velocity, which is obtained through a static nonlinear mapping $\gamma(\cdot)$ of the control command. As regards the discrete input signals, $u_1 = 0$ when the lamps 1 and 3 are off and $u_1 = 1$ when they are on. The signal u_2 governs the lamps 2 and 4 in the same way. $T_{ss}(p, u_1, u_2)$ is the steady-state temperature of the sensor casing at position p with the lamps switches as (u_1, u_2) and is given by

$$T_{ss}(p, u_1, u_2) = f_1(p)u_1 + f_2(p)u_2 + T_{amb}, \tag{6.2}$$

where $T_{amb} \in \mathbb{R}$ is the ambient temperature and $f_i(p) : \mathbb{R} \to \mathbb{R}$, $i \in \{1, 2\}$ describe the increase in steady-state temperature at position p obtained by turning on the ith switch.

6.1.2 Hybrid Model

In order to use hybrid MPC, as described in (Bemporad and Morari, 1999), we need to approximate the continuous-time model (6.1) and the nonlinearity T_{ss} in (6.2) by a piecewise affine hybrid model. To do this we introduce an auxiliary variable χ to model a piecewise affine approximation of $(T_{ss} - T_{amb})$. First we partition $\mathbb{R}$ into ℓ intervals $\{I_1, I_2, \dots, I_\ell\}$ and approximate f_i, $i = 1, 2$ in (6.2), by the functions

$$\begin{aligned} \chi_i(p(t)) =& \begin{cases} K^i_j p(t) + h^i_j & \text{if } u_i = 1, p \in I_j, j = 1, \dots, \ell \\ 0 & \text{otherwise,} \end{cases} \\ & i = 1, 2, \\ \chi(p(t)) =& \chi_1(p(t)) + \chi_2(p(t)). \end{aligned} \tag{6.3}$$

The notation $\chi(p(t))$ is used to highlight that χ depends on the position p, which changes in time. For simplicity of notation we will from now on use the notation

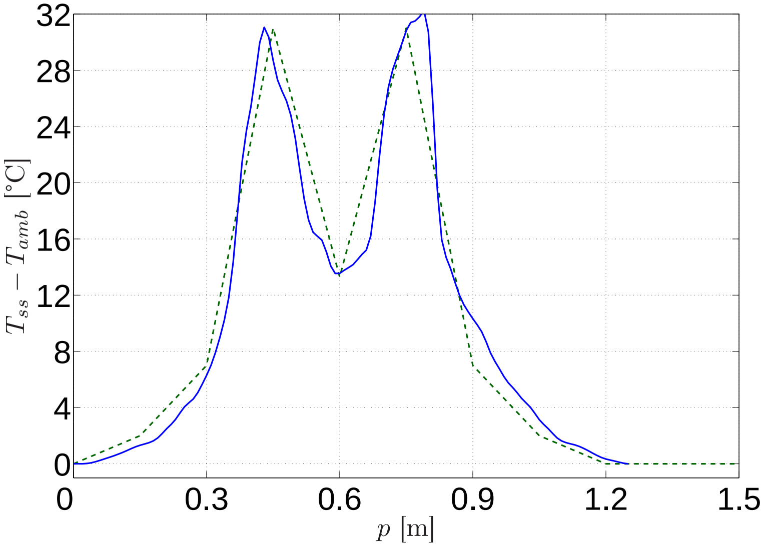


Figure 6.3: Measured $T_{ss} - T_{amb}$ (solid) and its piecewise affine approximation (dashed).

$\chi(t)$ instead. The effect of T_{amb} will be introduced later as a measured disturbance. The nonlinear function $(T_{ss} - T_{amb})$ and its approximation is shown in Figure 6.3.

The continuous-time model is sampled with period $T_s = 250\,\text{ms}$ giving the following discrete-time system

$$x(k+1) = \underbrace{\begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\Phi} x(k) + \underbrace{\begin{pmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & b_{32} \end{pmatrix}}_{\Gamma} \begin{pmatrix} \chi(k) \\ v_c(k) \end{pmatrix},$$

$$y(k) = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{C} x(k),$$

where $x = (T_1, T_2, p)^T$, $\chi(k)$ is a sampled version of (6.3) and the belt velocity $v_c = \gamma(u_c)$ is used as system input.

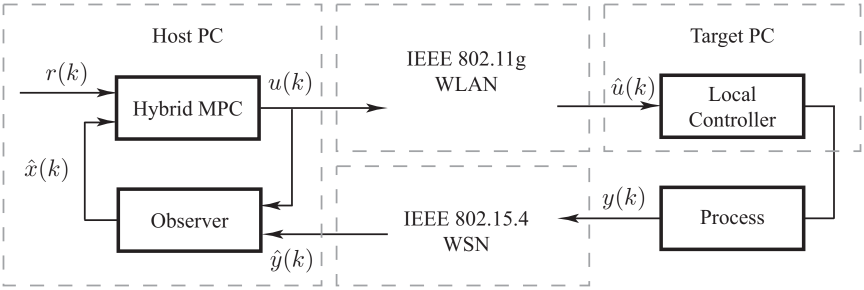


Figure 6.4: The wireless control architecture.

6.2 Control System Architecture

To be able to place the MPC computations in a computer located away from the process we use the control system architecture shown in Figure 6.4. The solid boxes are the functional blocks while the dashed boxes show the physical platforms on which they are implemented. Next we describe the architecture further.

6.2.1 Control System

The architecture described in Figure 6.4 is called the reference governor approach, see (Bemporad et al., 1997; Gilbert and Kolmanovsky, 1999), and has previously been studied in the context of unreliable network links in (Bemporad, 1998). As seen, this is a cascade type of control architecture where the process is actuated by a local controller at the process. The local controller in turn receives its reference from the remotely executed hybrid MPC which computes the optimal input commands, based on the estimated system states received from the observer. These optimal commands are then sent over a wireless channel to the local controller.

By this the computational power required to solve the optimization problem of finding the optimal inputs for the desired performance is moved away from the process to a powerful computer located at a base station. The local controller is computationally light and embedded in, or placed close to, the actuator where it performs low level control tasks.

6.2.2 Wireless Networks

Also shown in Figure 6.4 are the two networks used to support the de-localization of the remote MPC from the process site. Measurements are sent from the process to the observer over a wireless sensor network implemented on the network standard IEEE 802.15.4, while the commands from the MPC to the local controller are communicated over WLAN implemented on IEEE 802.11g.

We model these networks by switches turning communication on or off, so that data sent over the network is either received or lost. Following the notation in Figure 6.4 and letting ε denote void or ”no data” we get

$$\hat{u}(k) = \begin{cases} u(k) & \text{Command from controller received} \\ \varepsilon & \text{Command from controller lost} \end{cases}$$

$$\hat{y}(k) = \begin{cases} y(k) & \text{Command from sensor received} \\ \varepsilon & \text{Command from sensor lost} \end{cases}$$

where $u(k)$ and $y(k)$ are the outputs from the MPC and process respectively and $\hat{u}(k)$ and $\hat{y}(k)$ are the control command respectively the sensor value received after transmission.

6.2.3 Compensating for Packet Losses

To overcome packet losses in the wireless transmission, the system implements two different methods. If a command from the MPC to the local controller is lost, the local controller applies a hold mechanism giving $\hat{u}(k) = \hat{u}(k-1)$. In the case that a sensor packet is lost the observer, see Section 6.3.3, will evolve in open-loop to predict the states of the system, much similar to the behavior of the POC described in Chapter 3.

6.3 Control System Design

We now move on to synthesize the different parts of the control system and the involved controllers. We also describe the hybrid MPC algorithm.

6.3.1 The Local Controller

The local controller is divided into two parts. The first is a signal conversion which generates the motor commands $\hat{u}_c(k)$ from the commanded belt velocity $\hat{v}_c(k)$ by performing the inversion $\hat{u}_c(k) = \gamma^{-1}\big(\hat{v}_c(k)\big)$. The second part of the controller is a feedback component in the belt motor servo which rejects disturbances caused by varying mass on the belt and variable friction.

6.3.2 The Hybrid MPC

To apply hybrid MPC as we want, the hybrid model developed in Section 6.1.1 needs to be extended by two additional states. The first one is the ambient temperature T_{amb} in (6.2), which is assumed constant. The second additional state is the ”input memory” state x_u, used to weight the acceleration of the belt in the cost. The dynamics of x_u are defined by $x_u(k+1) = v_c(k)$. The acceleration at time k for a given input $v_c(k)$ can then be computed by backward Euler approximation as $\big(v_c(k) - x_u(k)\big)/T_s$. The extended system model becomes

$$
\begin{aligned}
x(k+1) &= \begin{pmatrix} a_{11} & 0 & 0 & 1 & 0 \\ a_{21} & a_{22} & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x(k) + \begin{pmatrix} b_{11}\chi(k) \\ b_{11}\chi(k) \\ b_{32}v_c(k) \\ 0 \\ v_c(k) \end{pmatrix}, \\
y(k) &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} x(k),
\end{aligned}
\tag{6.4}
$$

where $x = (T_1, T_2, p, T_{amb}, x_u)^T$.

In order to apply hybrid MPC, the system model in (6.4) must be formulated as a mixed logical dynamical (MLD) system as described in (Bemporad and Morari, 1999). How this conversion can be made is detailed further in Section 6.4.3. Converting (6.4) one get a MLD system as

$$
\begin{aligned}
x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k), \\
y(k) &= Cx(k), \\
E_2\delta(k) + E_3z(k) &\leqslant E_1u(k) + E_4x(k) + E_5,
\end{aligned}
\tag{6.5}
$$

where $u = (v_c, u_1, u_2)^T \in \mathbb{R} \times \{0,1\}^2$ is the input vector and $z(k) \in \mathbb{R}^{22}$ and $\delta(k) \in \{0,1\}^{10}$ are continuous and binary auxiliary variables, respectively. The auxiliary variables describe the piecewise affine dynamics given by (6.3).

Using this MLD model we can now formulate the hybrid MPC based on the following optimization problem, solved at each time step k,

$$
\begin{aligned}
\min\ & J(\{u(k+n), \delta(k+n|k), z(k+n|k)\}_0^{N-1}, x(k)) \triangleq \\
& q_\rho\rho^2 + \sum_{n=0}^{N-1} \Bigg(q_z \left(\frac{1}{T_s}\right)^2 \Big(v_c(k+n) - x_u(k+n|k)\Big)^2 \\
& \qquad + q_{v_c}v_c(k+n)^2 + \|y(k+n|k) - y_r\|_{Q_y} \Bigg)
\end{aligned}
\tag{6.6}
$$

subject to (6.5) and

$$
\begin{aligned}
\begin{pmatrix} 20 \\ 20 \\ 0 \end{pmatrix} \leqslant \begin{pmatrix} T_1(k+n|k) \\ T_2(k+n|k) \\ p(k+n|k) \end{pmatrix} &\leqslant \begin{pmatrix} 50 \\ 50 \\ 1.2 \end{pmatrix}, & n &= 1, \dots, N \\
-0.1 \ \leqslant\ v_c(k+n|k) &\leqslant\ 0.1, & n &= 0, \dots, N-1 \\
u_1(k+n|k),\ u_2(k+n|k) &\in\ \{0,1\}, & n &= 0, \dots, N-1,
\end{aligned}
$$

where distances are expressed in m, velocities in m/s and temperatures in $^\circ C$.

The tuning parameters of the hybrid MPC (6.6) are chosen as

$$N = 4, \quad q_\rho = 10^3, \quad q_{v_c} = 2, \quad q_z = 1, \quad Q_y = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.6 \end{pmatrix}$$

according to the following rationales: We want to track the position and the temperature reference and at the same time keep the state in a predefined "safe" set, that excludes high and low temperatures and excessive velocities. The acceleration and velocity of the belt should be low in order to reduce power consumption and avoid violent dynamics that cause wear.

The reference on the belt velocity v_c is set to 0, favoring light actuation of the belt. The output reference profile $y_r \in \mathbb{R}^2$ defines the desired behavior of the system. The length of the horizon N affects the performance of the controller. A longer horizon gives a smoother behavior and a shorter one gives a more aggressive controller. A longer horizon also gives a more complex optimization problem, hence the prediction horizon N is chosen by trading off between the performance and the available computational power.

The hybrid MPC executes the following operations at each time step k:

1. The system output $\hat{y}(k)$ is measured and the state estimate $\hat{x}(k)$ is computed;
2. The optimal control problem (6.6) is solved with $x(k|k) = \hat{x}(k)$;
3. The first optimal input $u^*(k)$ is applied to the system as the current control $u(k)$.

6.3.3 Observer

Since only the belt position p and the sensor temperature T_2 are measurable we need an observer to estimate the system states. To observe the states we use a reduced order nonlinear Luenberger observer for simplicity. It is given by

$$\begin{gathered} \hat{x}(k+1|k+1) = \Phi\hat{x}(k|k) + \xi(k) + K\big[\hat{y}(k+1) - C\big(\Phi\hat{x}(k|k) + \xi(k)\big)\big], \\ \xi(k) = \begin{pmatrix} b_{11}\chi(k|k) \\ b_{21}\chi(k|k) \\ T_s v_c(k) \end{pmatrix}, \quad K = \begin{pmatrix} k_{11} & k_{12} \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \end{gathered} \tag{6.7}$$

where $k_{11} = 5$, $k_{22} = 0$.

If the packet at time k is lost, the estimation evolves in open-loop so that (6.7) becomes $\hat{x}(k+1|k+1) = \Phi\hat{x}(k|k) + \xi(k)$. This is much like the behavior of the POC in Chapter 3 which also updates its internal states if the measurement is received and evolves in open-loop if it is lost.

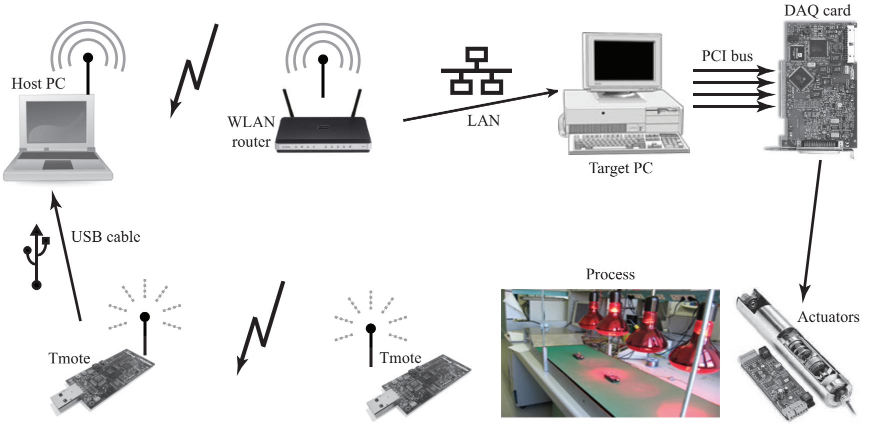


Figure 6.5: The hardware architecture.

6.4 Implementation

As mentioned earlier, Figure 6.4 shows the control system architecture and present both the functional blocks and the physical platforms which implement them. This section describes the implementation of the system infrastructure further. First the hardware platforms are presented and after that we present the software running on them. Finally the controller implementation is discussed.

6.4.1 Hardware Architecture

The hardware architecture of the system is shown in Figure 6.5. The MPC and the observer runs in the Host PC, which is a 1.2 GHz Pentium laptop, equipped with an integrated IEEE 802.11g WLAN card. The local controller runs in the Target PC, which is a Pentium 133 MHz. To enable communication with the Host PC, the Target PC is connected via ethernet LAN to a WLAN router. To interface the Target PC with the process, a data acquisition (DAQ) board from National Instruments is used. The process belt is moved using a belt roller with an encapsulated servo motor. An angular encoder on the belt measures the velocity. The lamps are controlled using two relays, one for each pair of lamps, to turn on and off their supply currents. The encoder and all the actuators are connected to the Target PC through the DAQ-board. The motes moving on the belt are Tmote Sky wireless sensors from Moteiv (Moteiv Corporation, 2007) equipped with temperature sensors, a low-power 8 MHz 16-bit microprocessor and a IEEE 802.15.4 radio transceiver. The mote placed on the belt measures its own temperature and communicates it to another mote connected to the USB-port of the Host PC.

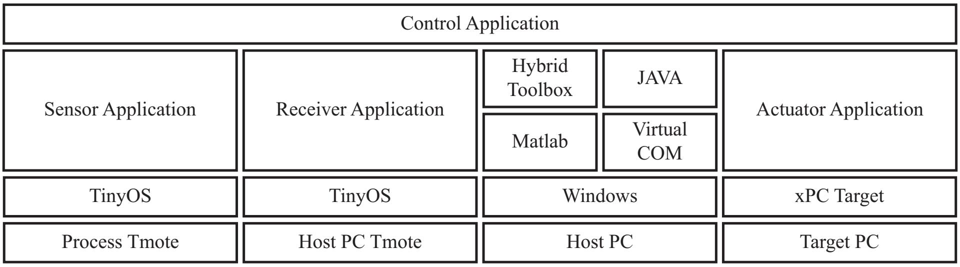


Figure 6.6: The software architecture.

6.4.2 Software Architecture

The software architecture of the system is shown in Figure 6.6. The control application consists of a distributed implementation over four platforms: Two of these are implemented on Tmote Sky motes and two are implemented on PCs.

The Host PC runs Microsoft Windows XP. On top of this, it runs MATLAB 7.1, and the HYBRID TOOLBOX v1.1.0 (Bemporad, 2003) for running the MPC. The underlying optimization software used in the execution of the MPC is CPLEX 9.0 (ILOG, Inc., 2004). Concurrently to the MPC, the Host PC runs a Virtual COM software, which reads the USB-port of the Host PC Tmote and abstracts it as a virtual RS-232 COM-port. This virtual COM port is in turn read by a Java application which presents the data in a suitable MATLAB format. We simply denote the software abstraction of the Host PC as *Host*.

The Target PC runs xPC-TARGET real-time kernel (The MathWorks Inc., 2000), with an application developed in SIMULINK and compiled with Real-Time Workshop. The xPC-TARGET toolbox provides a transparent way to use a standard PC, in our case the Target PC, as a micro-controller. It also provides a hardware abstraction for TCP/IP communication with the *Host* as well as an abstraction towards the DAQ-board. The full software abstraction of the Target PC from the WLAN router to the DAQ-board is referred to as the *Target*.

Both the Tmote Sky mote on the belt and the Tmote Sky mote connected to the Host PC are running TinyOS with custom applications. The Tmote on the belt is running a sensor application software, which samples the onboard temperature sensor and sends the data to the Host PC Tmote. The Host PC Tmote runs a receiver application software which listens to these packets and forwards them to the USB-port on the *Host*.

6.4.3 Controller Implementation

The hybrid MPC is implemented on the *Host* within the HYBRID TOOLBOX for MATLAB. The system model (6.4) is written in HYSDEL (Torrisi and Bemporad, 2004) and automatically converted by the associated compiler into the MLD system (6.5).

The optimal control problem (6.6) is formulated using the HYBRID TOOLBOX and included into a SIMULINK model as an S-function. The resulting optimization problem consists of 141 optimization variables, 93 continuous and 48 binary, and 585 mixed-integer linear inequalities. The average time required to solve the optimization problem using the optimization software CPLEX is 17 ms, with a worst-case computation time of about 125 ms. After the control command has been computed, it is sent to the *Target* via the wireless TCP/IP link.

From a functional point of view, the *Target* and the motor electronics implement the local controller. The Target PC computes the required motor input voltage to track the belt velocity commands received from the remote controller. To compensate for packet losses in the *Host* to *Target* link it also implements a hold function, which holds the last known commands and applies them if no new commands are received. Further, it integrates the encoder signal to generate the position measurements sent to the *Host*. The feedback component of the local controller is a servo-controller implemented in the motor electronics.

6.5 Experimental Results

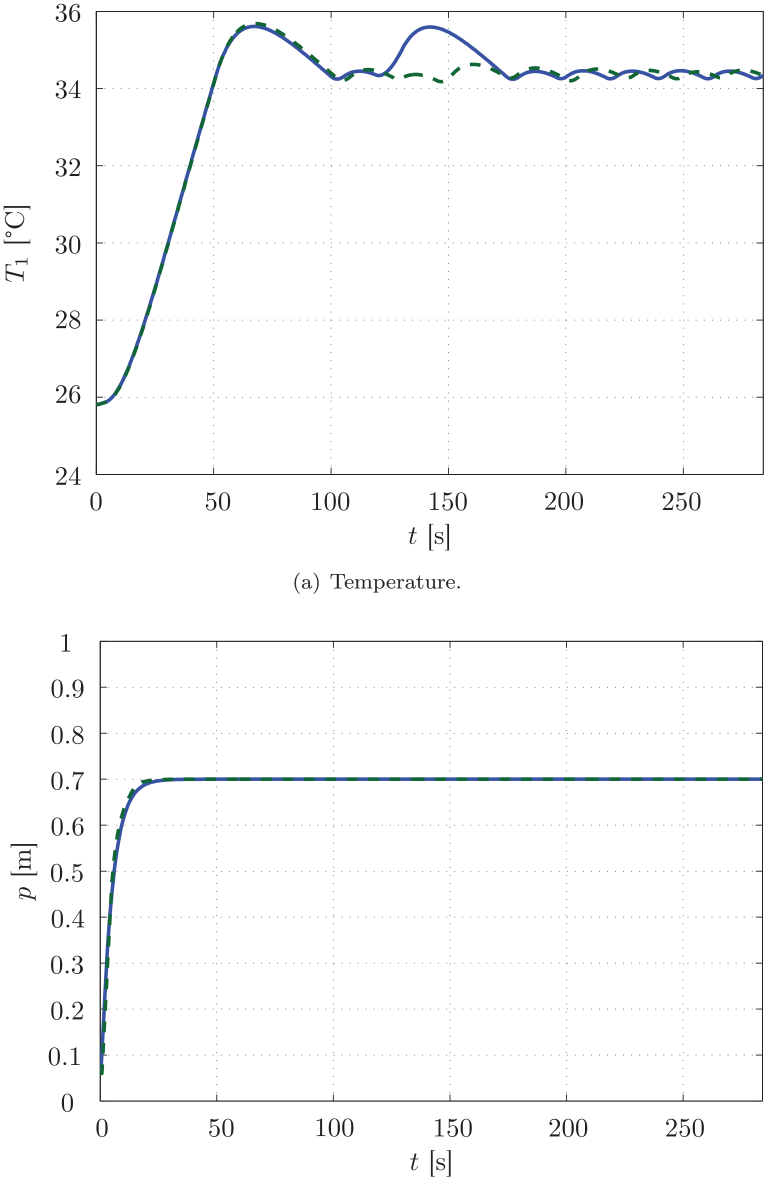
In this section we present experimental results of the process with the hybrid MPC designed in Section 6.3.2. The experiments aim at evaluating the performance of the control architecture and the impact of the wireless communication on the system behavior.

First we analyze the behavior with respect to data losses in the communication link between the MPC and the local controller, denoted the forward channel, where the input commands are sent over the WLAN network. Then we look at data losses in the sensor to MPC link, denoted the backward or feedback channel, where the measurements are sent over the wireless sensor network.

6.5.1 Losses in the MPC to Local Controller Communication Link

For the forward link case, we use a constant reference $y_r = [35,\ 0.7]^T$ for the temperature and position respectively. Since the WLAN network used is very reliable, data losses are introduced on purpose by discarding packets according to a data-loss profile obtained from a sensor network. In this way we are able to evaluate the effects of using a less reliable network than the WLAN.

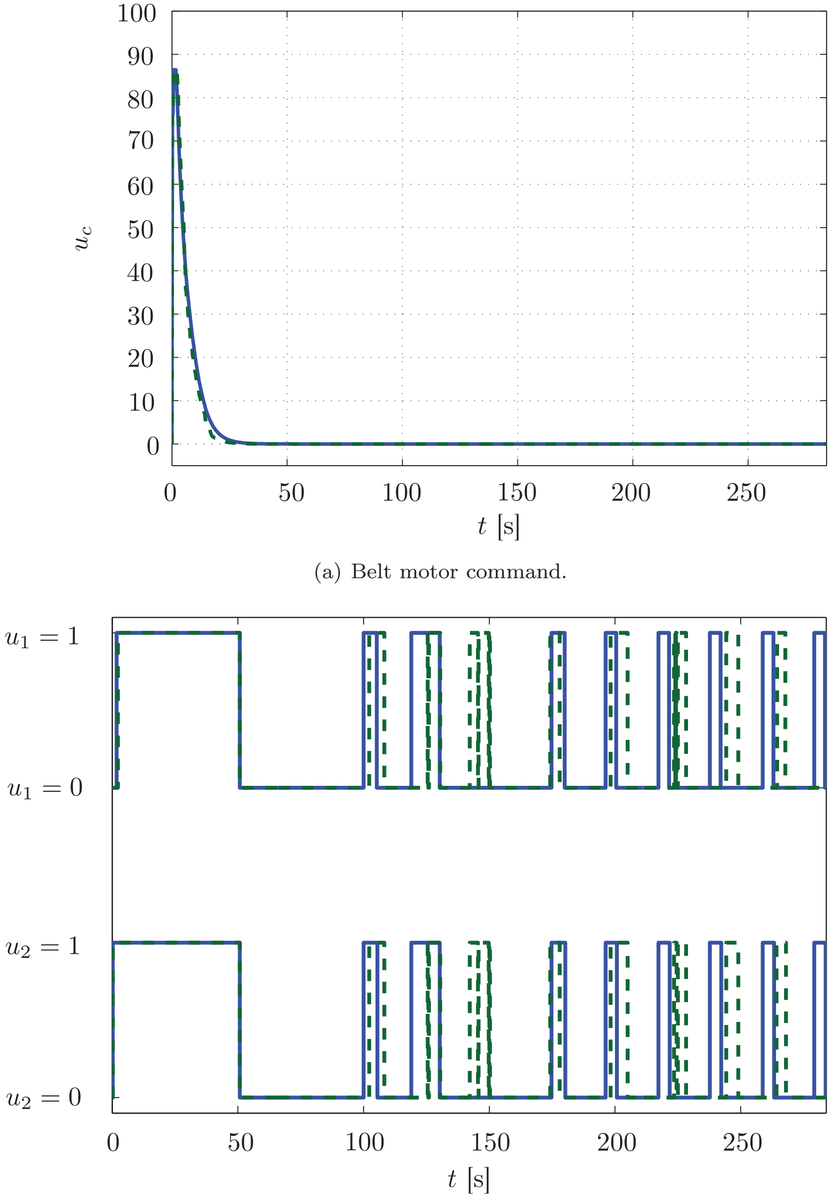
Figure 6.7 and Figure 6.8 show the results when the system is simulated as described above. The deviation from the behavior without losses that occurs around $t = 150\,\mathrm{s}$ is due to a massive packet drop burst. The position is not affected by the drops since the input u_c is in steady state, *i.e.*, constant. Hence the backup control command is equal to the control action of the system without losses, since the policy is to hold the last known value if the present is lost. Performing the same experiment on the true system with the same packet loss profile one gets the results in Figure 6.9 and Figure 6.10. Besides the effect of the packet loss, errors are now introduced by external noise and modelling imperfections. In particular, the input behavior is more aggressive due to the piecewise affine approximation of (6.2). As a consequence the controller keeps switching the lamps on and off and the temperature chatters around the equilibrium in a limit cycle. However, the results are still close to the behavior without losses.



(a) Temperature.

(b) Position.

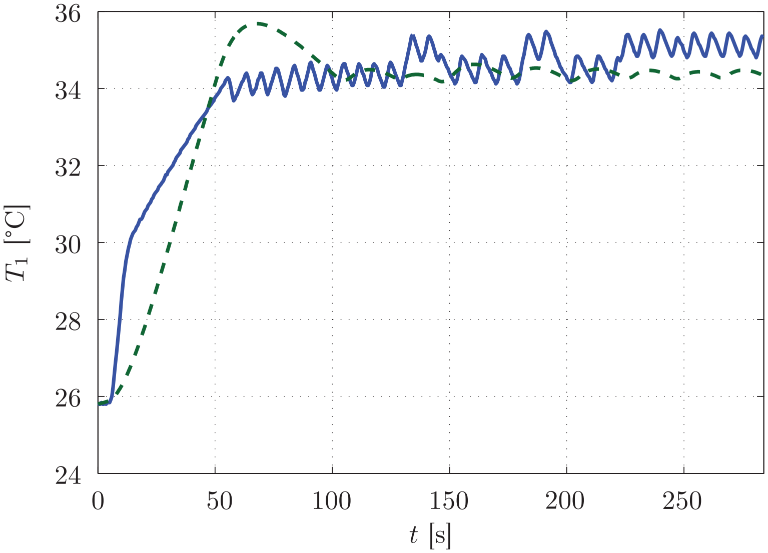
Figure 6.7: Simulations with losses in the forward channel: Simulated behavior with losses (solid) and simulated behavior without losses (dashed).



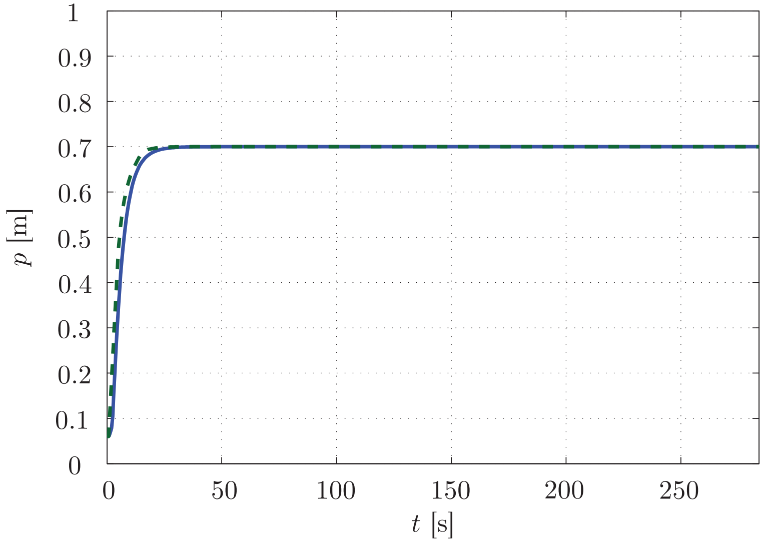
(a) Belt motor command.

(b) Lamp commands.

Figure 6.8: Simulations with losses in the forward channel: Simulated behavior with losses (solid) and simulated behavior without losses (dashed).

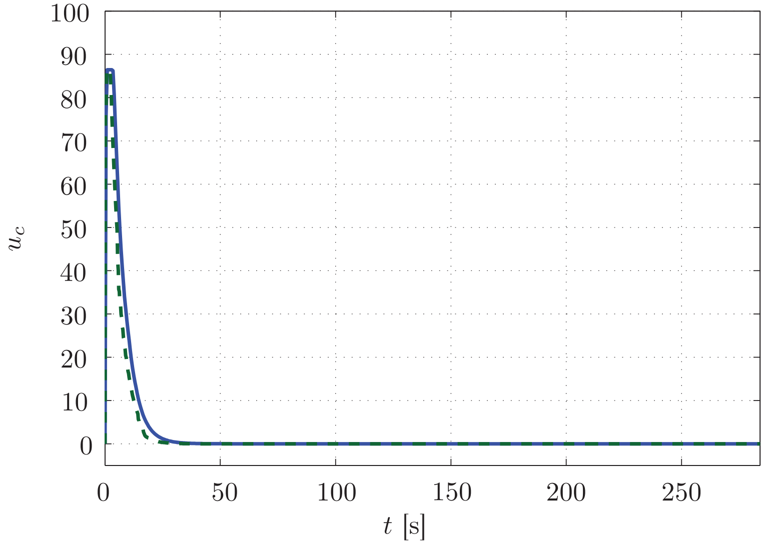


(a) Temperature.

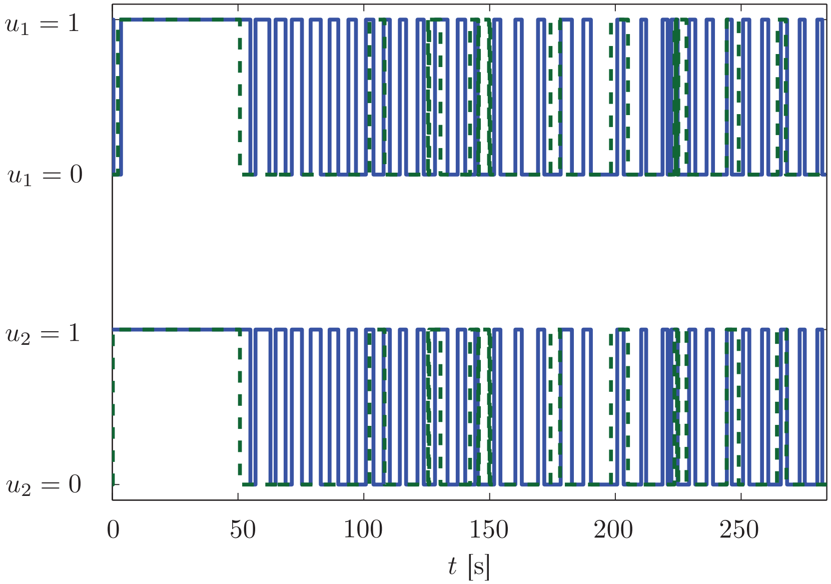


(b) Position.

Figure 6.9: Experiments with losses in the forward channel: Experimental behavior with losses (solid) and simulated behavior without losses (dashed).



(a) Belt motor command.



(b) Lamp commands.

Figure 6.10: Experiments with losses in the forward channel: Experimental behavior with losses (solid) and simulated behavior without losses (dashed).

6.5.2 Losses in the Sensor to MPC Communication Link

Now let us consider the case where losses occur in the feedback channel between the sensor and the MPC. In these experiments, the temperature reference is a square wave with maximum 42 °C, minimum 38 °C and frequency 3 mHz. The position reference is also a square wave with maximum 0.9 m, minimum 0.5 m and frequency 10 mHz. The initial position is 0 m and the initial temperature is the ambient temperature.

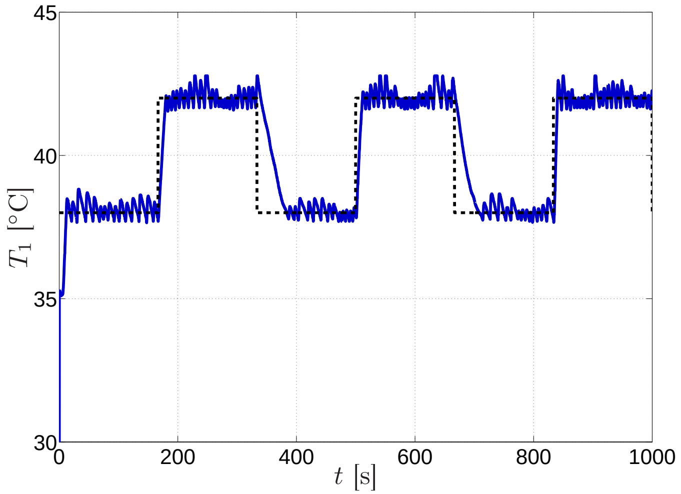
To induce packet loss, an extra sensor node is used to disturb the communication of the sensor on the belt by sending large amounts of data into the network. Even though the base station is able to discard the data sent by this extra sensor, the extra traffic and processing required will cause packet loss. If a measurement is lost, the process state estimate is updated by letting the state observer evolve in open-loop, as described in Section 6.3.

Two experiments are performed, one with low packet loss and one with high packet loss. In the low loss case the packet loss is induced as described above. In the high packet loss case the sensor antenna is also covered with aluminium foil, disrupting the radio signals, to increase the number of lost packets.

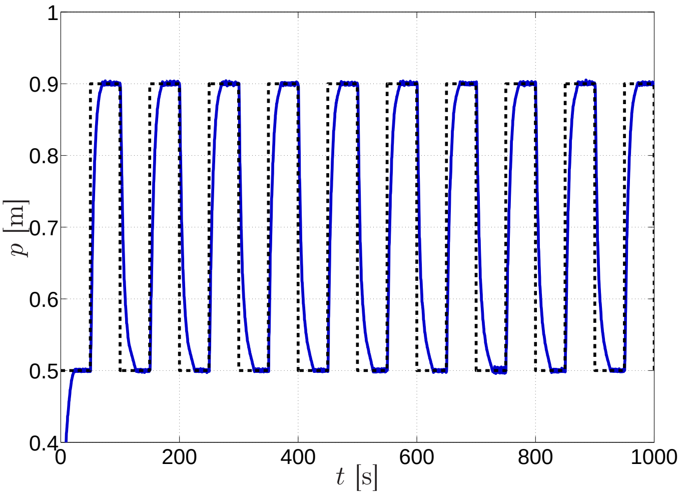
Low Loss Scenario

We first study the low loss scenario. Figure 6.11 shows the output as perceived by the MPC. That is, it shows the measurement from the sensor if it is received, otherwise it shows the prediction from the observer. The control signal computed by the MPC and sent to the local controller is shown in Figure 6.12. As seen the temperature reference tracking is quite good with small oscillations around the set-point. These are due to the discrete nature of the lamp switching. The tracking of the position reference is even better. The reason for this is that the position measurements are sent through the more reliable WLAN network. In fact, no measurements are lost in this link during the experiment.

The sensor network communications performance is shown in Figure 6.13. Here the temperature measurements received from the sensor are shown together with the packet reception rate (PRR). The PRR(t) is computed as the ratio between the number of received and the number of sent packets during the time interval $[t - 15\,s,\ t + 15\,s]$. Here the network is shown to be relatively reliable with only 7.9% of total number of measurements lost.

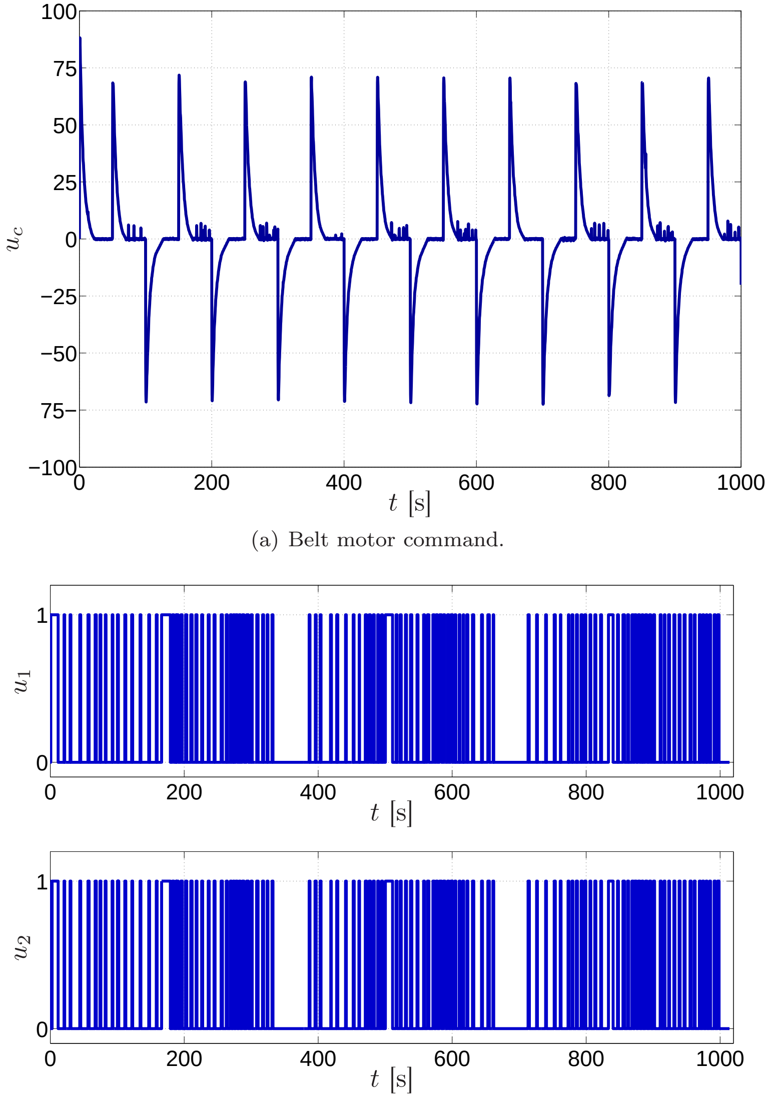


(a) Temperature: In case the measurement is lost the predicted value is shown.



(b) Position.

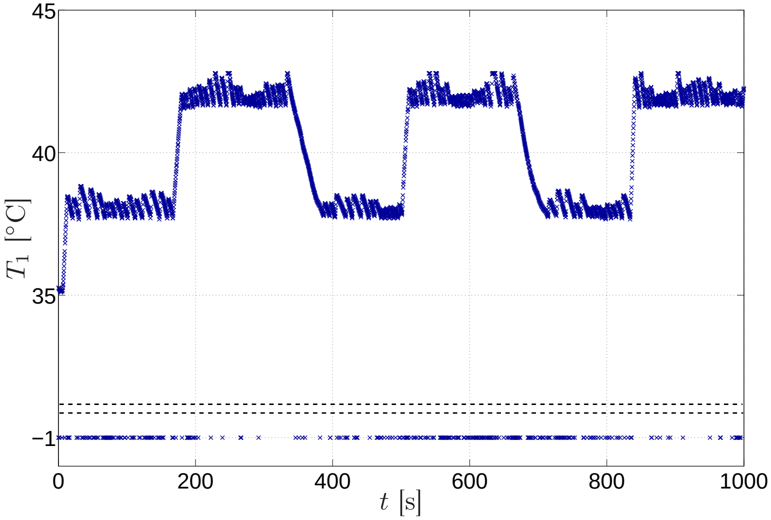
Figure 6.11: Experiments with losses in the feedback channel, low loss case: Temperature and position (solid) and corresponding references (dashed) as perceived by the MPC.



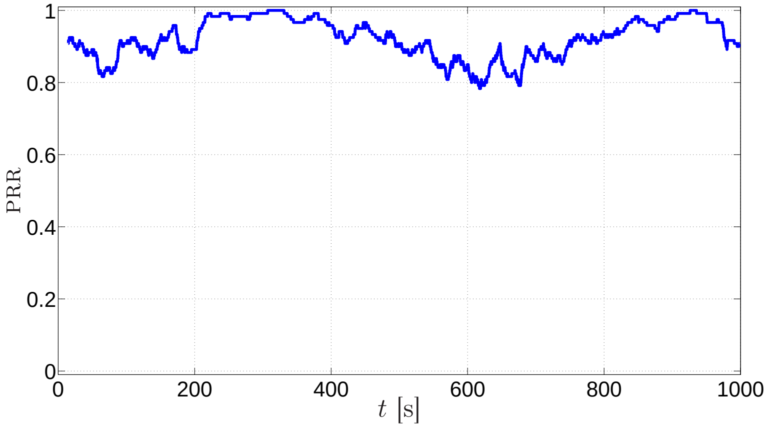
(a) Belt motor command.

(b) Lamp commands.

Figure 6.12: Experiments with losses in the feedback channel, low loss case: Commands issued by the MPC.



(a) Measurements received by the observer. The value −1 indicates the measurement is not received.



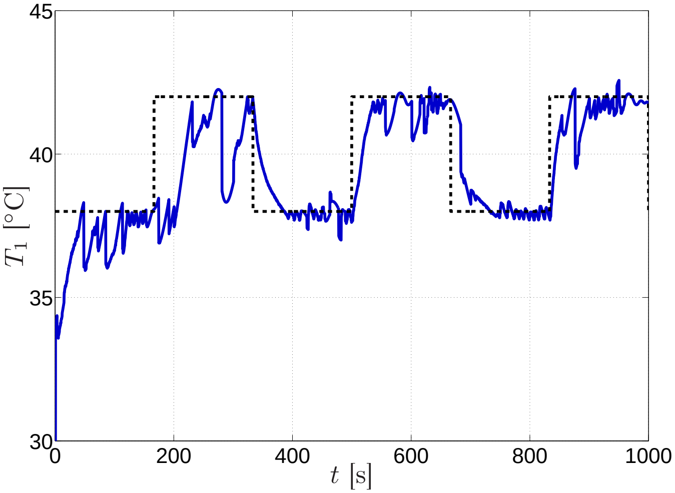
(b) Packet reception rate PRR(t) computed over a moving centered window of 30 s.

Figure 6.13: Experiments with losses in the feedback channel, low loss case: Measurements received from the network.

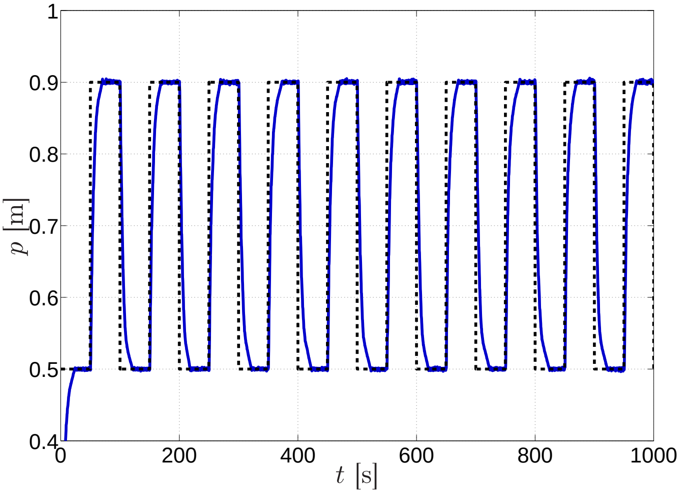
High Loss Scenario

We now move on to the high loss scenario. Similarly as in the low loss scenario Figure 6.14 shows the output as perceived by the MPC. The control signal computed by the MPC and sent to the local controller is shown in Figure 6.15.

Figure 6.16 shows the sensor network communications performance. As seen, the packet reception rate is much lower than before with 62.8% of the temperature measurements lost. This affects the temperature reference tracking as it appears from Figure 6.14. In particular, the abrupt changes in the temperature value seen by the controller reveal long bursts of missing data causing the state estimate to diverge. As a consequence, when a measurement is finally received there is a jump in the estimate.

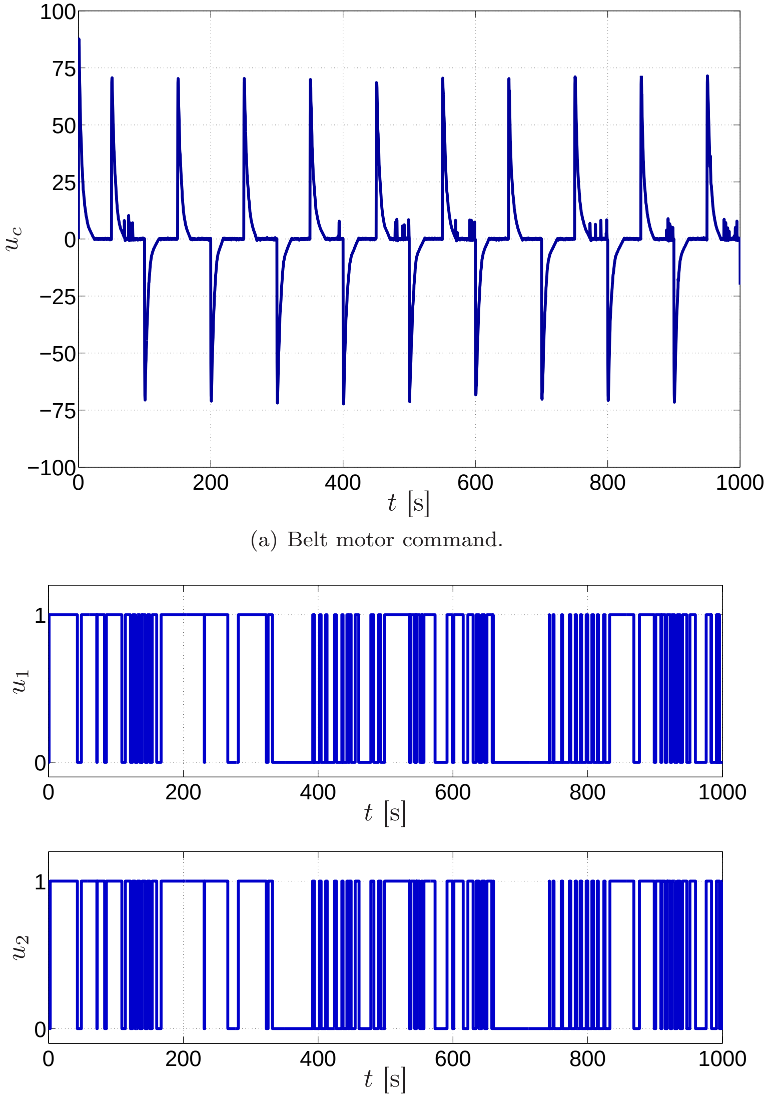


(a) Temperature: In case the measurement is lost the predicted value is shown.



(b) Position.

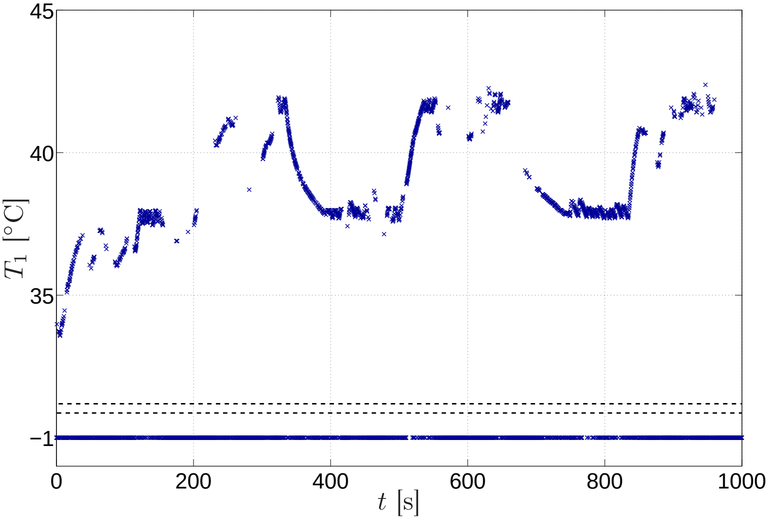
Figure 6.14: Experiments with losses in the feedback channel, high loss case: Temperature and position (solid) and corresponding references (dashed) as perceived by the MPC.



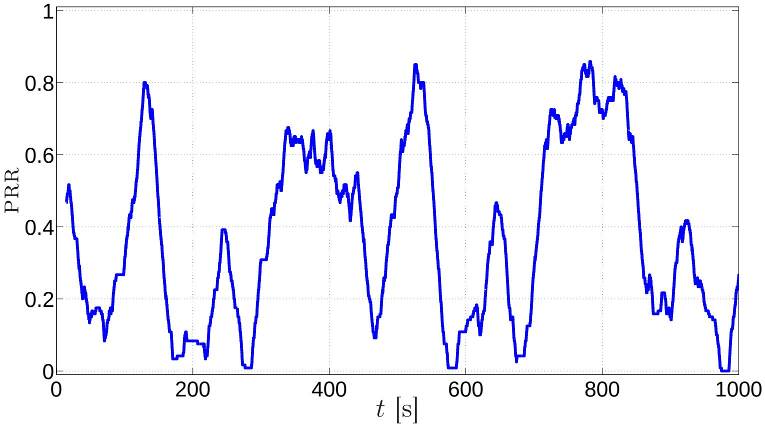
(a) Belt motor command.

(b) Lamp commands.

Figure 6.15: Experiments with losses in the feedback channel, high loss case: Commands issued by the MPC.



(a) Measurements received by the observer. The value −1 indicates the measurement is not received.



(b) Packet reception rate PRR(t) computed over a moving, centered window of 30 s.

Figure 6.16: Experiments with losses in the feedback channel, high loss case: Measurements received from the network.

6.6 Summary

This chapter has presented a hybrid MPC design for a physical process and an experimental demonstration of remote control of the same, over wireless networks. It was shown that data packets dropped in both forward and feedback communication links can be handled with good results using standard hybrid MPC techniques.

Chapter 7

Conclusions

This final chapter concludes the thesis with a brief summary of the main results, followed by some observations and interesting problems for future research.

7.1 Summary

The thesis addressed wireless networked control systems in process industry and how existing control structures and paradigms could be adapted to work better when feedback measurements and actuation commands are sent over a wireless network, subject to packet losses and communication constraints.

In Chapter 3 the predictive outage compensator (poc) was introduced. Simulation and experimental results, performed on a wirelessly controlled two-tank process, showed that the introduction of a poc significantly could improve performance under outage, compared to previously used compensation schemes. Methods were derived to synthesize a poc and corresponding theoretical performance bounds were computed. It was also shown that it is possible to achieve good performance with a low-order implementation based on Hankel-norm approximation. Trade-offs between achievable performance, outage length, and poc order were discussed.

In Chapter 4 a self-triggered model predictive control (mpc) algorithm was proposed and studied. It proved useful for controlling multiple processes on a shared communication network, using jointly optimal control signals and adaptive sampling intervals. The proposed control law was computed explicitly and shown to be stabilizing. It was also shown that, under mild assumptions, it was possible to schedule all processes on the network using the proposed mpc. Simulation results showed that the use of the presented control law may help reduce the amount of communication, without almost any loss of performance compared to periodic sampling.

In Chapter 5 an event-triggered mpc was presented in which the process was controlled using open-loop optimal control sequences, updated at event times. Stability properties under input and state constraints as well as exogenous disturbances were derived. The performance versus communication trade-off for the proposed MPC was evaluated in comparison with classical time-triggered MPC. Simulation results underlined the effectiveness of the proposed scheme in terms of reducing the communication and computational effort while guaranteeing a desired performance.

In Chapter 6 the implementation of a hybrid MPC design for control over wireless networks was presented. It was shown that the setup is easy to tune and that the use of hybrid MPC has several advantages. The most obvious advantage is that it offers the possibility to handle hybrid dynamics in the system such as on–off inputs, as well as enforce constraints on states and inputs in an explicit way. One drawback is that hybrid MPC can be computationally intensive.

7.2 Future Work

Looking into the future, the methods and ideas presented in the thesis have the potential to be developed and refined further. Below we list some observations and possible topics for future research.

The POC in Chapter 3 estimates the full state of the closed-loop system. An interesting topic for future work is to find alternative methods of using this estimate. One could use a separate open-loop controller in outage, for instance. Another problem of both practical and theoretical importance is how to generate controls to minimize the bump in the control signal after a communication outage. These bumps are due to integral-windup in the controller and it is of interest to investigate possible anti-windup and bumpless transfer strategies for networked control systems. Also other implementations of the POC are possible, such as optimization based methods or adaptive techniques.

In the self-triggered MPC developed in Chapter 4 an important idea is the introduction of a cost for sampling. It would be of interest to investigate more functions describing the cost for sampling and how they would affect the behavior of the controller. Another important part of the control design is constraining the shape of the control signal trajectory. In the current setup, only the first sample step is left open to optimize over, while the remaining are kept at a fixed down-sampling rate. One could imagine optimizing over several of the initial inter sample times before reverting into a fixed down-sampling rate. An interesting topic for future research is to further investigate the complexity and possible performance increase for such an extended formulation. The approach also allows extensions to particular shapes of the control commands, such as fixed-size pulses or quantized controls.

In the event-triggered MPC presented in Chapter 5, it would be of interest to investigate how one could introduce a disturbance observer and use the estimated disturbance when solving the open-loop optimal control problem. By such integral action the controller should be able to achieve better performance in the presence of disturbances. However, since the samples are only taken intermittently as decided by the previous optimization, care needs to be taken when designing this observer to avoid wind-up phenomena.

Bibliography

V. M. Adamjan, D. Z. Arov, and M. G. Krein. Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem. *Math. USSR Sbornik*, 15(1):31–73, 1971.

P. Agrawal, A. Ahlén, T. Olofsson, and M. Gidlund. Characterization of long term channel variations in industrial wireless sensor networks. In *Proceedings of IEEE International Conference on Communications*, Sydney, Australia, 2014. To Appear.

J. Åkerberg, M. Gidlund, and M. Björkman. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In *Proceedings of IEEE International Conference on Industrial Informatics*, Lisbon, Portugal, 2011.

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

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

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Dover Publications, 2005.

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

A. Anta and P. Tabuada. On the benefits of relaxing the periodicity assumption for networked control systems over CAN. In *Proccedings of IEEE Real-Time Systems Symposium*, Washington, DC, USA, 2009a.

A. Anta and P. Tabuada. Isochronous manifolds in self-triggered control. In *Proceedings of IEEE Conference on Decision and Control*, Shanghai, P. R. China, 2009b.

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

A. Anta and P. Tabuada. On the minimum attention and anytime attention problems for nonlinear systems. In *Proceedings of IEEE Conference on Decision and Control*, Atlanta, GA, USA, 2010b.

P. Antsaklis and J. Baillieul. Special issue on networked control systems. *IEEE Transaction on Automatic Control*, 49(9):1421–1603, 2004. Guest Editors.

P. Antsaklis and J. Baillieul. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1):1–316, 2007. Guest Editors.

D. Antunes, J. P. Hespanha, and C. Silvestre. Stochastic hybrid systems with renewal transitions. In *Proceedings of American Control Conference*, Baltimore, MD, USA, 2010.

D. Antunes, J. P. Hespanha, and C. Silvestre. Stochastic networked control systems with dynamic protocols. In *Proceedings of American Control Conference*, San Francisco, CA, USA, 2011.

D. Antunes, W. P. M. H. Heemels, J. P. Hespanha, and C. Silvestre. Scheduling measurements and controls over networks - part II: rollout strategies for simultaneous protocol and controller design. In *Proceedings of American Control Conference*, Montreal, Canada, 2012a.

D. Antunes, W. P. M. H. Heemels, and P. Tabuada. Dynamic programming formulation of periodic event-triggered control: performance guarantees and co-design. In *Proceedings of IEEE Conference on Decision and Control*, Maui, HI, USA, 2012b.

D. Antunes, J. P. Hespanha, and C. Silvestre. Volterra integral approach to impulsive renewal systems: application to networked control. *IEEE Transactions on Automatic Control*, 57(3):607–619, 2012c.

K.-E. Årzén. A simple event-based PID controller. In *Proceedings of IFAC World Congress*, Beijing, P. R. China, 1999.

K.-E. Årzén, A. Bicchi, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes. A component-based approach to the design of networked control systems. *European Journal of Control*, 13(2–3):261–279, 2007.

K. J. Åström and B. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of IFAC World Congress*, Beijing, P. R. China, 1999.

K. J. Åström and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 2002.

K. J. Åström and B. Wittenmark. *Computer-Controlled Systems: Theory and Design.* Prentice Hall, third edition, 1997.

A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. MAC essentials for wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 12(2):222–248, 2010.

J. D. J. Barradas Berglind, T. M. P. Gommans, and W. P. M. H. Heemels. Self-triggered MPC for constrained linear systems and quadratic costs. In *Proceedings of IFAC Conference on Nonlinear Model Predictive Control*, Noordwijkerhout, Netherlands, 2012.

A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. In *Proceedings of IEEE Conference on Decision and Control*, Tampa, FL, USA, 1998.

A. Bemporad. *Hybrid Toolbox – User's Guide*, December 2003.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997.

A. Bemporad, S. Di Cairano, E. Henriksson, and K. H. Johansson. Hybrid model predictive control based on wireless sensor feedback: an experimental study. In *Proceedings of IEEE Conference on Decision and Control*, New Orleans, LA, USA, 2007.

A. Bemporad, S. Di Cairano, E. Henriksson, and K. H. Johansson. Hybrid model predictive control based on wireless sensor feedback: an experimental study. *International Journal of Robust and Nonlinear Control*, 20(2):209–225, 2010a.

A. Bemporad, W. P. M. H. Heemels, and M. Johansson. *Networked Control Systems.* Lecture Notes in Control and Information Sciences. Springer-Verlag, 2010b.

D. Bernardini and A. Bemporad. Energy-aware robust model predictive control based on noisy wireless sensors. *Automatica*, 48(1):36–44, 2012.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 1995.

R. W. Brockett. Minimum attention control. In *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, USA, 1997.

L. G. Bushnell. Special section on networks and control. *IEEE Control Systems Magazine*, 21(1):22–185, 2001. Guest Editor.

A. Casavola, E. Mosca, and M. Papini. Predictive teleoperation of constrained dynamic systems via internet-like channels. *IEEE Transactions on Control Systems Technology*, 14(4):681–694, 2006.

M. Chiang, P. Hande, T. Lan, and C. W. Tan. Power control in wireless cellular networks. *Foundations and Trends in Networking*, 2(4):381–533, 2008.

L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37(7):1019–1028, 2001.

I. Cornell. Implementation of a collection tree routing protocol and a predictive outage compensator. Master's thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2012.

M. C. F. Donkers, P. Tabuada, and W. P. M. H. Heemels. On the minimum attention and the anytime attention control problems for linear systems: a linear programming approach. In *Proceedings of IEEE Conference on Decision and Control*, Orlando, FL, USA, 2011.

M. C. F. Donkers, W. P. M. H. Heemels, D. Bernardini, A. Bemporad, and V. Shneer. Stability analysis of stochastic networked control systems. *Automatica*, 48(5):917–925, 2012.

A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. Novel event-triggered strategies for model predictive controllers. In *Proceedings of IEEE Conference on Decision and Control*, Orlando, FL, USA, 2011.

A. Eqtami, S. Heshmati-Alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos. Self-triggered model predictive control for nonholonomic systems. In *Proceedings of European Control Conference*, Zurich, Switzerland, 2013.

E. Garcia and P. J. Antsaklis. Model-based event-triggered control with time-varying network delays. In *Proceedings of IEEE Conference on Decision and Control*, Orlando, FL, USA, 2011.

E. G. Gilbert and I. V. Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control*, 9(15):1117–1141, 1999.

K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their $\mathcal{L}_\infty$-error bounds. *International Journal of Control*, 39(6):1115–1193, 1984.

A. Goldsmith. *Wireless Communications.* Cambridge University Press, 2005.

T. M. P. Gommans, W. P. M. H. Heemels, N. W. Bauer, and N. van de Wouw. Compensation-based control for lossy communication networks. *International Journal of Control*, 86(10):1880–1897, 2013.

S. Graham and P. R. Kumar. The convergence of control, communication, and computation. In *Personal Wireless Communications*, volume 2775 of *Lecture Notes in Computer Science*, pages 458–475. Springer-Verlag, 2003.

L. Greco, D. Fontanelli, and A. Bicchi. Design and stability analysis for anytime control via stochastic scheduling. *IEEE Transactions on Automatic Control*, 56 (3):571–585, 2011.

L. Grüne, S. Jerg, O. Junge, D. Lehmann, J. Lunze, F. Müller, and M. Post. Two complementary approaches to event-based control. *Automatisierungstechnik*, 58 (4):173–182, 2010.

G. Gu. All optimal Hankel-norm approximations and their error bounds in discrete-time. *International Journal of Control*, 78(6):408–423, 2005.

V. Gupta and N. C. Martins. On stability in the presence of analog erasure channels. In *Proceedings of IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

Hart Communication Foundation. WirelessHART, 2007. International Electrotechnical Commission (IEC) standard 62591.

W. P. M. H. Heemels and M. C. F. Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, 2013.

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

W. P. M. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nesic. Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance. *IEEE Transactions on Automatic Control*, 55 (8):1781–1796, 2010.

W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, 2013.

M. Heidarinejad, J. Liu, D. Muñoz de la Peña, J. F. Davis, and P. D. Christofides. Handling communication disruptions in distributed model predictive control. *Journal of Process Control*, 21(1):173–181, 2011.

E. Henriksson. Hybrid model predictive control based on wireless sensor feedback. Master's thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2007.

E. Henriksson, H. Sandberg, and K. H. Johansson. Predictive compensation for communication outages in networked control systems. In *Proceedings of IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

E. Henriksson, H. Sandberg, and K. H. Johansson. Reduced-order predictive outage compensators for networked systems. In *Proceedings of IEEE Conference on Decision and Control*, Shanghai, P.R. China, 2009.

E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. Self-triggered model predictive control for network scheduling and control. In *Proceedings of IFAC International Symposium on Advanced Control of Chemical Processes*, Singapore, 2012.

E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson. Multiple-loop self-triggered model predictive control for network scheduling and control. *IEEE Transactions on Control Systems Technology*, 2013a. Under Review.

E. Henriksson, H. Sandberg, and K. H. Johansson. Predictive outage compensation for networked control systems. *Journal of Process Control*, 2013b. Under Review.

J. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE: Special Issue on Technology of Networked Control Systems*, 95(1):138–162, 2007.

Ilog, Inc. *CPLEX 9.0 User Manual*, 2004.

International Society of Automation. ISA100.11a Wireless Systems for Industrial Automation: Process Control and Related Applications, 2009.

H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks.* Wiley, 2005.

F. Kazempour and J. Ghaisari. Stability analysis of model-based networked distributed control systems. *Journal of Process Control*, 23(3):444–452, 2013.

E. C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control.* PhD thesis, University of Cambridge, 2000.

E. C. Kerrigan and J. M. Maciejowski. Robust feasibility in model predictive control: necessary and sufficient conditions. In *Proceedings of IEEE Conference on Decision and Control*, Orlando, FL, USA, 2001.

J.-S. Kim, T.-W. Yoon, A. Jadbabaie, and C. De Persis. Input-to-state stable finite horizon MPC for neutrally stable linear discrete-time systems with input constraints. *Systems & Control Letters*, 55(4):293–303, 2006.

P. R. Kumar. New technological vistas for systems and control: the example of wireless networks. *IEEE Control Systems Magazine*, 21(1):24–37, 2001.

D. Lehmann, E. Henriksson, and K. H. Johansson. Event-triggered model predictive control of discrete-time linear systems subject to disturbances. In *Proceedings of European Control Conference*, Zurich, Switzerland, 2013.

M. D. Lemmon, T. Chantem, X. S. Hu, and M. Zyskowski. On self-triggered full-information $\mathcal{H}_\infty$ controllers. In *Proceedings of Hybrid Systems: Computation and Control*, Pisa, Italy, 2007.

D. Limon, I. Alvarado, T. Alamo, and E. Camacho. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3):248–260, 2010.

Q. Ling and M. D. Lemmon. Optimal dropout compensation in networked control systems. In *Proceedings of IEEE Conference on Decision and Control*, Maui, HI, USA, 2003.

A. Liu, L. Yu, and W. Zhang. Moving horizon estimation for networked systems with multiple packet dropouts. *Journal of Process Control*, 22(9):1593–1608, 2012.

G. P. Liu, D. Rees J. X. Mu, and S. C. Chai. Design and stability analysis of networked control systems with random communication time delay using the modified MPC. *International Journal of Control*, 79(4):288–297, 2006.

J. Liu, D. Muñoz de la Peña, and P. D. Christofides. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica*, 46(1):52–61, 2010.

X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Proceedings of IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

M. Lješnjanin, D. E. Quevedo, and D. Nešić. Packetized MPC with dynamic scheduling constraints and bounded packet dropouts. *Automatica*, 2014. In Press.

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

J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.

F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 2003. Eleventh impression.

D. L. Marruedo, T. Alamo, and E. F. Camacho. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, NV, USA, 2002.

D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2): 219–224, 2005.

M. Mazo Jr. and P. Tabuada. On event-triggered and self-triggered control over sensor/actuator networks. In *Proceedings of IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

M. Mazo Jr. and P. Tabuada. On the robustness of self-triggered control for sensor/actuator networks. In *Proceedings of IEEE Conference on Decision and Control*, Shanghai, P. R. China, 2009.

M. Mazo Jr. and P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, 56(10): 2456–2461, 2011.

M. Mazo Jr., A. Anta, and P. Tabuada. On self-triggered control for linear systems: guarantees and complexity. In *Proceedings of European Control Conference*, Budapest, Hungary, 2009.

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

P. Mhaskar, Jinfeng Liu, and P. D. Christofides. *Fault-Tolerant Process Control: Methods and Applications*. Springer-Verlag, 2013.

A. Molin and S. Hirche. On LQG joint optimal scheduling and control under communication constraints. In *Proceedings of IEEE Conference on Decision and Control*, Shanghai, P. R. China, 2009.

Andreas F. Molisch. *Wireless Communications*. Wiley, second edition, 2010.

Moteiv Corporation. *Tmote Sky: Datasheet*, March 2007.

D. Muñoz de la Peña and P. D. Christofides. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Transactions on Automatic Control*, 53(9):2076–2089, 2008.

G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans. Feedback control under data rate constraints: an overview. *Proceedings of the IEEE: Special Issue on Technology of Networked Control Systems*, 95(1):108–137, 2007.

P. Neumann. Communication in industrial automation – what is going on? *Control Engineering Practice*, 15(11):1332–1347, 2007. Special Issue on Manufacturing Plant Control: Challenges and Issues.

G. Obinata and B. D. O. Anderson. *Model Reduction for Control System Design*. Springer-Verlag, 2001.

J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch. Comparative study of energy-efficient sampling approaches for wireless control networks. *IEEE Transactions on Industrial Informatics*, 6(3):416–424, 2010.

D. E. Quevedo and V. Gupta. Sequence-based anytime control. *IEEE Transactions on Automatic Control*, 58(2):377–390, 2013.

D. E. Quevedo and D. Nešić. Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses. *Automatica*, 48(8):1803–1811, 2012.

D. E. Quevedo, G. C. Goodwin, and J. S. Welsh. Minimizing down-link traffic in networked control systems via optimal control techniques. In *Proceedings of IEEE Conference on Decision and Control*, Maui, HI, USA, 2003.

D. E. Quevedo, E. I. Silva, and G. C. Goodwin. Control over unreliable networks affected by packet erasures and variable transmission delays. *IEEE Journal on Selected Areas in Communication*, 26(4):672–685, 2008.

J. A. Rossiter, B. Kouvaritakis, and M. J. Rice. A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 34(1):65–73, 1998.

C. J. Rozell and D. H. Johnson. Power scheduling for wireless sensor and actuator networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks*, Cambridge, MA, USA, 2007.

T. Samad and A.M. Annaswamy. The impact of control technology. Technical report, IEEE Control Systems Society, 2011. Editors.

T. Samad, P. McLaughlin, and J. Lu. System architecture for process automation: review and trends. *Journal of Process Control*, 17(3):191–201, 2007.

R. Scattolini. Architectures for distributed and hierarchical model predictive control – a review. *Journal of Process Control*, 19(5):723–731, 2009.

L. Schenato. To zero or to hold control inputs with lossy links? *IEEE Transactions on Automatic Control*, 54(5):1093–1099, 2009.

L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE: Special Issue on Technology of Networked Control Systems*, 95(1):163–187, 2007.

B. De Schutter and R. Scattolini. Special issue on hierarchical and distributed model predictive control. *Journal of Process Control*, 21(5):684–815, 2011. Guest Editors.

J. Sijs, M. Lazar, and W. P. M. H. Heemels. On integration of event-based estimation and robust MPC in a feedback loop. In *Proceedings of Hybrid Systems: Computation and control*, Stockholm, Sweden, 2010.

B. Sinopoli, L. Schenato, M. Franceschetti, K. Polla, M. Jordan, and S. Sastry. Kalman filtering with intermittent observation. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, 2004.

E. D. Sontag. *Mathematical Control Theory. Deterministic Finite-Dimensional Systems*, volume 6 of *Texts in Applied Mathematics.* Springer-Verlag, New York, second edition, 1998.

W. Sun, K. M. Nagpal, and P. P. Khargonekar. $\mathcal{H}_\infty$ control and filtering for sampled-data systems. *IEEE Transactions on Automatic Control*, 38(8):1162–1175, 1993.

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

P. Tabuada and X. Wang. Preliminary results on state-triggered scheduling of stabilizing control tasks. In *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006.

P. L. Tang and C. W. de Silva. Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control. *IEEE Transactions on Control Systems Technology*, 14(4):707–718, 2006.

The MathWorks Inc. *xPC Target - For Use with Real-Time Workshop*, 2000. User's Guide – Version 1.1.

F. D. Torrisi and A. Bemporad. HYSDEL — A tool for generating computational hybrid models. *IEEE Transactions Control Systems Technology*, 12(2):235–249, 2004.

P. Trodden and A. Richards. Robust distributed model predictive control using tubes. In *Proceedings of American Control Conference*, Minneapolis, MN, USA, 2006.

D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.

P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen. Event-based model predictive control for networked control systems. In *Proceedings of IEEE Conference on Decision and Control*, Shanghai, P. R. China, 2009.

P. Varutti, T. Faulwasser, B. Kern, M. Kögel, and R. Findeisen. Event-based reduced-attention predictive control for nonlinear uncertain systems. In *Proceedings of IEEE International Symposium on Computer-Aided Control System Design*, Yokohama, Japan, 2010.

M. Velasco, J. Fuertes, and P. Martí. The self triggered task model for real-time control systems. In *Proceedings of IEEE Real-Time Systems Symposium*, Cancun, Mexico, 2003.

J. Wallander. Implementation of a wireless control system: network routing and unreliable communication link compensation. Master's thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, 2012.

X. Wang and M. D. Lemmon. Event design in event-triggered feedback control systems. In *Proceedings of IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

X. Wang and M. D. Lemmon. Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability. *IEEE Transactions on Automatic Control*, 54(3):452–467, 2009.

X. Wang and M. D. Lemmon. Event-triggering in distributed networked control systems. *IEEE Transactions on Automatic Control*, 56(3):586–601, 2011a.

X. Wang and M. D. Lemmon. Minimum attention controllers for event-triggered feedback systems. In *Proceedings of IEEE Conference on Decision and Control*, Orlando, FL, USA, 2011b.

A. Willig. Recent and emerging topics in wireless industrial communication: a selection. *IEEE Transactions on Industrial Informatics*, 4(2):102–124, 2008.

A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. *IEEE Transactions on Industrial Electronics*, 49(6):1265–1282, 2002.

Y-B. Zhao, G. P. Liu, and D. Rees. Improved predictive control approach to networked control systems. *IET Control Theory and Applications*, 2(8):675–681, 2008.

K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.