KTH Electrical Engineering

Compensating for Unreliable Communication Links in Networked Control Systems

ERIK HENRIKSSON

Licentiate Thesis
Stockholm, Sweden 2009

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

TRITA-EE 2009:036
ISSN 1653-5146
ISBN 978-91-7415-414-6

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i reglerteknik fredagen den 9 oktober 2009, klockan 10:15 i sal Q21, Kungliga Tekniska högskolan, Osquldas väg 6, Stockholm.

Tryck: Universitetsservice AB

Abstract

Control systems utilizing wireless sensor and actuator networks can be severely affected by the properties of the communication links. Radio fading and interference may cause communication losses and outages in situations when the radio environment is noisy and low transmission power is desirable. This thesis proposes a method to compensate for such unpredictable losses of data in the feedback control loop by introducing a predictive outage compensator (POC). The POC is a filter to be implemented at the receiver sides of networked control systems where it generates artificial samples when data are lost. If the receiver node does not receive the data, the POC suggests a command based on the history of past data. It is shown how to design, tune and implement a POC. Theoretical bounds and simulation results show that a POC can improve the closed-loop control performance under communication losses considerably. We provide a deterministic and a stochastic method to synthesize POCs. Worst-case performance bounds are given that relate the closed-loop performance with the complexity of the compensator. We also show that it is possible to achieve good performance with a low-order implementation based on Hankel norm approximation. Tradeoffs between achievable performance, communication loss length, and POC order are discussed. The results are illustrated on a simulated example of a multiple-tank process. The thesis is concluded by an experimental validation of wireless control of a physical lab process. Here the controller and the physical system are separated geographically and interfaced through a wireless medium. For the remote control we use a hybrid model predictive controller. The results reflect the difficulties in wireless control as well as they highlight the flexibility and possibilities one obtains by using wireless instead of a wired communication medium.

Acknowledgment

Many people deserve to be acknowledged for their part in the work of completing this thesis and deriving the results herein.

First I want to thank my main advisor Karl Henrik Johansson for taking me on as his Ph.D. student. Your enthusiasm is contagious and the never ending stream of visions and ideas you provide is truly inspiring. My co-advisor Henrik Sandberg deserves many thanks for his time and invaluable help in realizing these visions and ideas, for this I am truly grateful. Thank you both for all the support you have given me, this thesis would not have existed without you.

I want to thank all my past and present colleagues at the Automatic Control Lab. Thank you for being nice, sociable and helpful. You all make it a great place to work. Thanks also to Karin Karlsson Eklund and Anneli Ström for running the place. Special thanks go to my office mates Oscar Flärdh and Jonas Mårtensson. I truly enjoy the laughs we share and the discussions we have. The nice atmosphere makes every day at work enjoyable.

I would like to thank Alberto Bemporad for hosting my visit at the University of Siena from September 2006 until May 2007. First as a master's thesis student and later as a visiting Ph.D. student. It was a great experience! Thanks also to Stefano Di Cairano for the collaboration during and after my stay.

I also 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) and the European Commission, through the SOCRADES and HYCON projects, for the financial support that made this work possible.

Finally the warmest thanks goes to all my friends and especially my family for all the support you have given me over the years. Without your support this thesis would never have been written!

Erik Henriksson
Stockholm, September 2009

Contents

Chapter 1

Introduction

The introduction of wireless communication technology has led to that we in today's society are becoming more and more connected to, and exchange more and more information with, each other. A large number of interacting mobile embedded systems influence every day aspect of our lives. Our mobile phones and personal computers exchange information over cellular and wireless local area networks. Wireless appliances and systems are being used in offices and homes, but also more and more in large-scale industrial control systems such as in utility infrastructures and transportation networks.

In the near future it is believed that these wireless monitoring and control systems will represent a large portion of all traffic over wireless networks. There are some important shortcomings of today's wireless systems that need to be overcome to enable this second wireless revolution. This thesis discusses one such aspect, which has to do with the imperfections of wireless communication.

In this chapter we first give an introduction to wireless control in industry and some of the current issues. We then proceed to give motivating examples for the use of wireless control in industry as well as for the contribution of the thesis. After this the problem formulation is given, followed by a section on related work. The chapter is concluded with the thesis outline and a presentation of a selection of notation and abbreviations.

1.1 Wireless Control in Industry

Wired communication networks have been commonly used in distributed control systems since the seventies (Samad et al., 2007). The recent advances in low-power wireless radio and sensor technologies have enabled the engineering of a new type of networked sensing and control systems, which are now being tested and evaluated in industry for automation and process control (SOCRADES, Integrated Project, EU Sixth Framework Programme).

There are several benefits of introducing wireless networking in industrial control applications. They can be summarized as follows:

- *Cost:* Wireless links lead to reduced wiring, which constitutes a substantial part of the development cost for many industrial plants due to the high price of copper wires. Wireless technology also has the potential to reduce the installation cost, since hardware installation for a wireless network is limited to some routers and gateways.

- *Flexibility:* With wireless links there are fewer physical design limitations, and it is easier to move the existing equipment as well as installing new. Thanks to mobility and fast reconfiguration, new and better designs can be exploited in system development and operation.

- *Reliability:* Connectors and wires lead to many faults in industrial control systems, partly because of cable wear and tear. Wireless devices consequently have a potential to reducing the downtime for these systems.

The use of wireless technology in feedback control loops raises new challenges. The network medium introduces uncertainties such as packet loss, transmission delay etc. 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 these wireless control systems (Antsaklis and Baillieul, 2007).

The communication protocol can have severe impact on the closed-loop control performance. For example, some communication protocols guarantee the delivery of a message, but on the other hand give high delay variability. Other protocols provide a less reliable communication, but ensure better delay characteristics. For wireless networks, packet losses typically vary heavily with the radio conditions, so if the environment is changing or the nodes are mobile, the control system needs to handle varying network conditions. Recent standardization efforts include the WirelessHART (HART Communication Foundation, 2007) and the ISA100 standards (International Society of Automation, 2009) wireless network communication protocols designed for process automation applications.

1.2 Motivating Examples

To illustrate the use of wireless control and some of the problems that the wireless network introduces, we give examples. The first is a motivation for the use of wireless control in industry. The second gives an intuitive explanation to the problem of loosing data packets in the communication, and the importance of how these losses are distributed in time. The final example illustrates the impact on the control system performance when the system is subject to packet losses.

Example 1.1

A motivating scenario for the contribution of this thesis is the control of the floatation tank process in an mineral processing plant, parts of which are shown in Figure 1.1, at Boliden in Sweden. This system is being investigated within the SOCRADES project (SOCRADES, Integrated Project, EU Sixth Framework Programme). It consists of four tanks in series and is today controlled using four individual PI-controllers (Stenlund and Medvedev, 2002). The control objective is to maintain stable levels in all four flotation tanks as well as compensating for fluctuations in inflow and for load disturbances. For this process we are interested in replacing the wired level and flow sensors with wireless sensors. In doing so, it is desirable not to have to change the overall control structure or not even the control parameter tuning. We look instead for a solution where some additional compensation is done at the actuator node, but no other changes are needed in the closed-loop system.

When a system is controlled over a wireless network the imperfections in this network will 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. This problem 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 the following example.

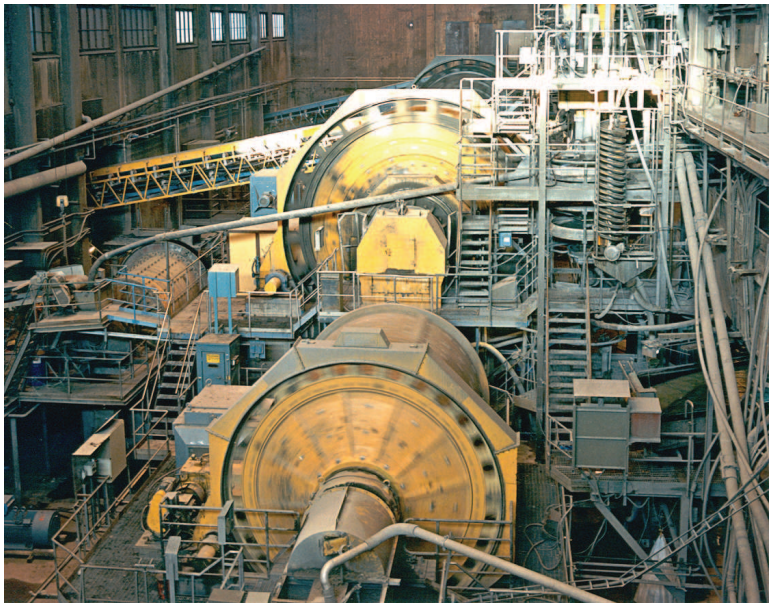


Figure 1.1: Ore crush at the Boliden plant in which the ore is crushed before entering into the flotation tank process. (Copyright: Boliden)

Example 1.2

Let us say that you are driving your car. While driving you base your decisions on what you see around you. If you close your eyes you need to drive based on what you have seen previously and your predictions. Since you are the controller when driving, keeping your eyes closed is similar to loosing sensor data packets in a control system. If you close your eyes, it will only take a few seconds before you have a very limited idea of what is happening around you and after a few more seconds you would probably stop your car. Still, you have no problem at all with keeping your eyes closed for several minutes during a normal 1 h drive, the total time they are closed because you are blinking. In fact, you will blink your eyes around 15 times a minute and the average blink will have a duration of about 200 ms, so during a 1 h drive you will keep your eyes closed for about 3 minutes (Caffier et al., 2003). It is clearly much worse to loose all measurements for say 30 s in a row than it is to loose 6 times as many measurements spread out over one hour. That is, it is not the total amount of lost data over a time interval that is important, but instead how long time it was since the last received information.

The example shows that the distribution of the communication loss is an important parameter when it comes to assessing how losses will affect the control system

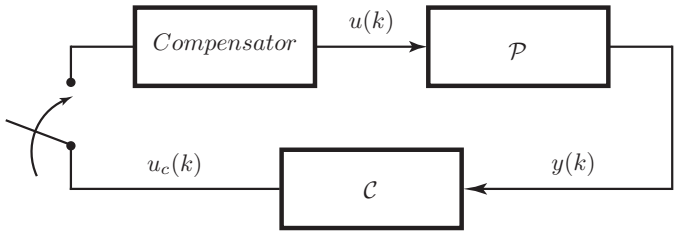


Figure 1.2: Networked Control System controlled under packet loss.

performance. It also highlights that depending on the packet loss profile we will have to take different kinds of actions to reduce the impact of the losses on the system behavior. If the system is subject to sparse and short burst of losses we can probably use the same method as we do when we drive to overcome them, *i.e.*, assume that things stay the same during the loss period. If the losses instead come in longer bursts, like when keeping your eyes close for a few seconds, we have to use a more elaborate method or device to overcome them.

To illustrate how this reasoning reflect the behavior of a control system under losses we study the following example where we instead of loosing sensor data as in Example 1.2 loose data between controller and process.

Example 1.3

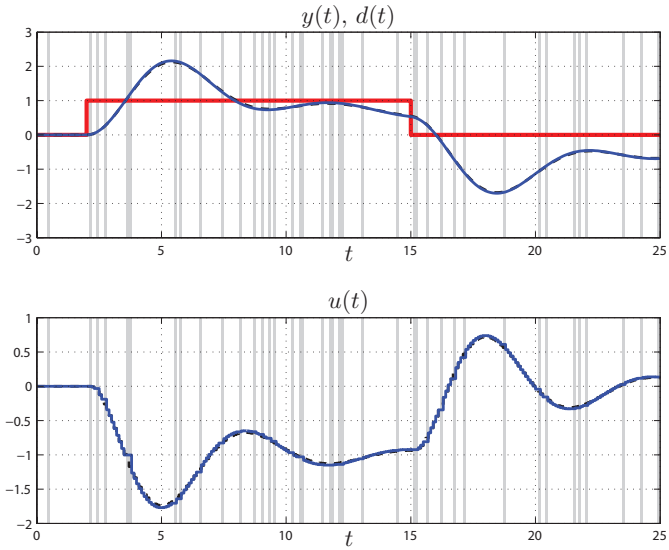
Consider the networked control system in Figure 1.2 and let the switch represents that packets between controller and process can be lost. The process $\mathcal{P}$ is given by

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

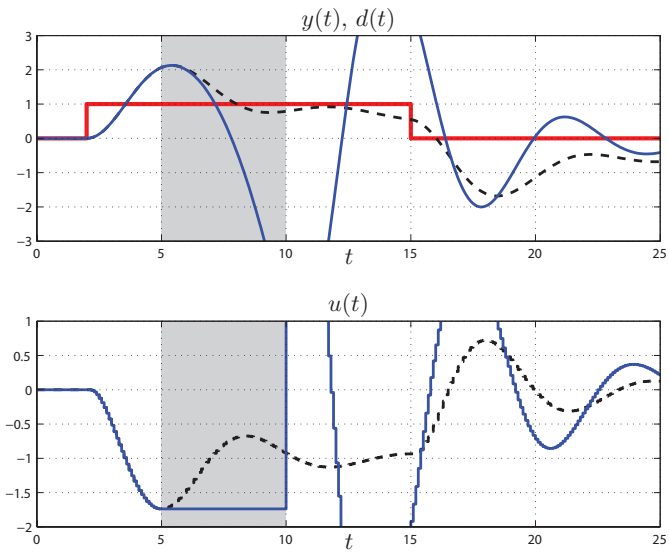
and the controller $\mathcal{C}$ is given by as sampled version of

$$C(s) = \frac{20s^2 + 12s + 1}{s^2 + 2s} \frac{10}{s + 100}$$

sampled with $T_s = 0.1\,s$. If a packet is lost, *i.e.*, the switch is open we use the 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 this system we simulate it on two different loss profiles. Both with 20 % average packet loss over time. Studying the results in Figure 1.3(a) we see that for short and sparse losses the method to hold the last known value works well. Instead looking at Figure 1.3(b) it becomes evident that when losses are grouped into longer connected period a more advanced compensation scheme is needed.



(a) Sparse and short losses.



(b) Connected period of loss.

Figure 1.3: Comparison of system output behavior (solid) with the output of the system without losses (dashed) under disturbance d (solid light) and under 20 % packet loss (greyed)

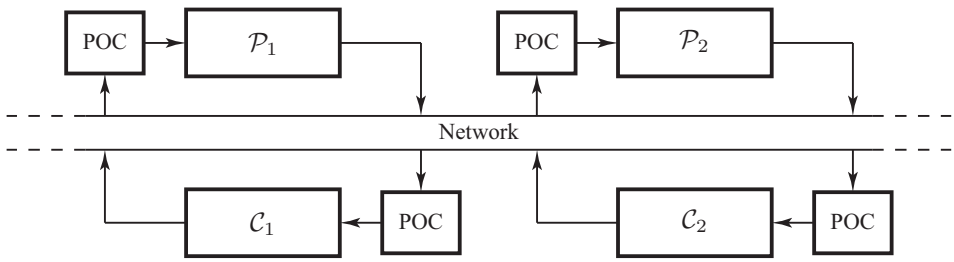


Figure 1.4: Illustration of the POC placement to compensate for outages in a networked control system.

1.3 Problem Formulation

As seen in Example 1.2 and 1.3 subsequent losses of packets in the network during which sensor data do not reach the controller node and/or control commands do not reach the actuator are hard to handle. We denote such periods of subsequent losses as *outages*.

The problem considered in this thesis is how to compensate for such 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 1.4. The POCs are designed to overcome losses in the network 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 and by this it gives the desired modular properties described in Example 1.1. Obviously, a POC has a limitation on how efficient it can be for long periods of outages. An important result of the thesis is to build tools to understand how these limitations affect the applicability to real systems.

Further, a case study of wireless control for a physical lab process with interesting hybrid dynamics is considered. The process is controlled over both wireless sensor and actuator links using a remotely placed hybrid model predictive control (MPC).

1.4 Contributing Papers

This thesis is based on the following publications:

- Chapters 2–5:

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

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

- Chapter 6:

  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*, 2009. To Appear.

  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 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, Royal Institute of Technology (KTH), Stockholm, Sweden, 2007.

The scientific contribution of the thesis is mainly the author's own work. The results presented in Chapters 2–5 have been derived in cooperation with the author's supervisors. 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.

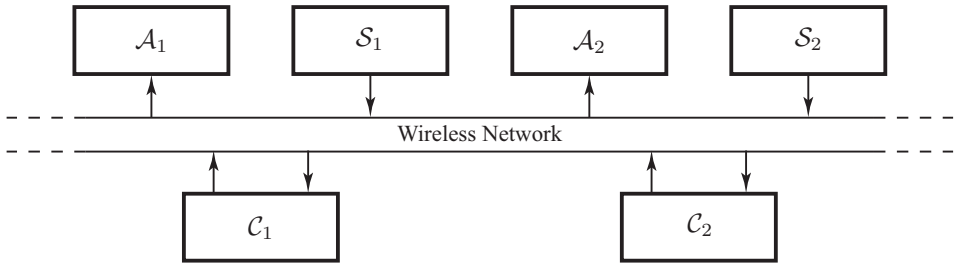


Figure 1.5: Wireless networked control system with actuator, sensor and control devices interfaced through the network.

1.5 Related Work

This section gives a short background to wireless networked control. First an introduction to the area is given, then wireless networks are briefly discussed together with their impact on control performance. The section is concluded with a discussion on how unreliable links can be compensated for.

1.5.1 Wireless Networked Control Systems

A wireless networked control system is a control system in which actuators, sensors and controllers are connected and communicate over a wireless network, as illustrated in Figure 1.5. The use of wireless networks in the control system introduce a wide range of advantages but also problems which need to be addressed.

Control over wireless networks is a young research area without mature theory or tools, but with a lot of current activity (Bushnell, 2001; Antsaklis and Baillieul, 2004, 2007). The need for interaction between control and communication in the design of wireless networks was raised in (Kumar, 2001). Open research problems in the area of control using wireless sensors networks include choice of architectures and modular design and implementation (Sinopoli et al., 2003; Årzén et al., 2007). A cross-layer framework for the joint design of wireless networks and distributed controllers is attempting (Liu and Goldsmith, 2004), although care needs to be taken to avoid undesirable interactions (Kawadia and Kumar, 2005).

The problems introduced by the network in control systems can be briefly summarized as follows:

- *Delay:* Packets in the network are delayed because of transmission delays and medium access delays. The problem of delay in control system is a well studied topic with a vast literature, *e.g.*, (Richard, 2003).

- *Rate Limitations:* When several devices share a common network resource the rate at which they can transmit data over the network is limited by the

network capacity. These limitations impose constraints on achievable performance. An overview of feedback control under data rate constraints is given in (Nair et al., 2007).

- *Packet Loss:* The problem which this thesis focuses on is losses of data packet in the communication. These losses originate from overflow in communication buffers and from transmission errors in the physical layer due to shadowing and fading channels. A survey regarding control over networks subject to loss is given in (Hespanha et al., 2007).

1.5.2 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 etc. Typical scenarios in industrial control settings are reported in (Willig et al., 2002). These imperfections can cause packets to be delayed or lost. It is hard to prevent delays and losses from occurring, and it is difficult to provide accurate stochastic models for them. Stationary models commonly used in the literature on networked control can be hard to justify in practice (Hespanha et al., 2007; Schenato et al., 2007). Further reading on the area of wireless communications, see the textbooks (Tse and Viswanath, 2005; Goldsmith, 2005).

A commonly used and simple way to model losses in wireless networks is to assume that packet losses are independent and identically distributed (i.i.d.) according to a Bernoulli distribution, so that a packet is successfully transmitted with probability p and lost with probability $1-p$, independent of previous packets. Another common method to model losses is the Gilbert model (Gilbert, 1960) in which packets are lost according to a two-state Markov chain, illustrated in Figure 1.6. The transitions between the states where the packet is lost and where the packet is received are governed by probabilities p_1 and p_2 according to Figure 1.6. If the present packet was received the next will be received with probability p_1 and lost with probability $1-p_2$ and similarly if the present packet is lost. How p_1 and p_2 are chosen governs how the packet loss is distributed in time. The advantage of the Gilbert model is that it, in contrast to the Bernoulli model, captures the fact that packet losses in real systems typically come in bursts (Willig et al., 2002). The approach taken in this thesis is to assume that all losses occur in the form of outages without any detailed model of the probability of loss.

Designing protocols suitable for control is a very natural way to compensate for unreliable networks and their impact on control performance since the approach aims at solving the problem at the root, *i.e.*, improve the communication quality instead of compensating for poor communication performance in the control design. Protocol design suitable for control over wireless networks is a large research field of its own and will therefore not be treated further. For further reading on protocol design and networking the reader is referred to the recent survey (Willig, 2008).

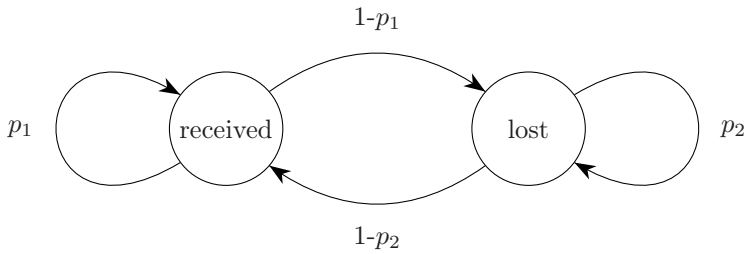


Figure 1.6: The Gilbert model.

1.5.3 Compensating for Unreliable Communication Links

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. Several approaches to overcome the unreliable nature of the network are present in the literature (Bushnell, 2001; Antsaklis and Baillieul, 2004, 2007). They 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. These latter methods enable the use of slightly modified control design methods present in the literature. There are recent contributions to the area of joint communication and control design, a recent survey of the area is given in (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 (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 i.i.d 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 i.i.d. Bernoulli losses the separation principle hold, provided that successful transmissions are acknowledged. The optimal LQG controller is derived as a linear function of the state and bounds are given on the maximum tolerable loss probability.

The approach taken in this thesis is to add a compensation device 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 and that the control design can be made without taking the network properties into account. In doing so we can facilitate the use of a modular design of networked

control systems as proposed in (Årzén et al., 2007). Two common choices of the mentioned compensation devices, which are ubiquitous in the literature, are the hold and zero compensators. In the hold compensator the last received data is applied when a packet is lost. In the zero compensator an a priori decided constant, often zero, is used as a prediction of the lost command. A performance comparison between the two is given in (Schenato, 2009). The main contributions of this thesis can be interpreted as a formalization and generalization of these methods. Instead of holding the last command or applying a constant, the POC suggests a command based on the history of past data. Predictive control has been extensively used in various networked control settings *e.g.*, (Bemporad, 1998; Quevedo et al., 2008) but we believe that our study on the complexity and synthesis of the POC is new.

1.6 Outline

The rest of the thesis is outlined as follows. The POC is presented in Chapter 2. Chapter 3 gives two methods to synthesize a POC. The complexity of a POC is investigated in Chapter 4, which also presents methods to reduce the complexity by means of model order reduction. The synthesis methods are then evaluated through simulations in Chapter 5. Chapter 6 presents a case study of control of a hybrid process over a wireless medium. Finally, Chapter 7 summarizes the thesis and presents directions for future research.

1.7 Notation and Abbreviations

The author has tried to use standard notation in the thesis. A selection of notation and abbreviations is presented below.

Notation

$\mathbb{R}$	Set of real numbers.
$\mathbb{N}$	Set of natural numbers, $\mathbb{N} = \{1,\ 2,\ 3,\ \dots\}$.
$\hat{x}(k\|\ell)$	Estimate of $x(k)$ based on measurement up until time ℓ.
$\|v\|$	$\|v\| = \sqrt{v^T v}$, $v \in \mathbb{R}^n$.
$\|v\|_1$	$\|v\|_1 = \sum_{i=1}^{n} \|v_i\|$, $v \in \mathbb{R}^n$.
$\|x\|_2$	$\|x\|_2 = \sqrt{\sum_{i=-\infty}^{\infty} \|x(i)\|^2}$.
ℓ_2	Hilbert space of all x s.t. $\|x\|_2 < \infty$.
$\|G\|$	Induced ℓ_2-norm: $\|G\| = \sup_{u \neq 0} \frac{\|Gu\|_2}{\|u\|_2}$.
Γ_G	Hankel operator of the system G.
$\sigma_i(G)$	ith Hankel singular value of G.
q	Forward shift operator: $qu(k) = u(k+1)$.
$\mathbf{E}x$	Expected value of x.

Abbreviations

DAQ	Data Acquisition
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LAN	Local Area Network
LMI	Linear Matrix Inequality
MAC	Medium Access Control
MIMO	Multiple-Input Multiple-Output
MIP	Mixed Integer Program
MLD	Mixed Logical Dynamical
MPC	Model Predictive Control
POC	Predictive Outage Compensator
PRR	Packet Reception Rate
SISO	Single-Input Single-Output
TCP	Transmission Control Protocol
USB	Universal Serial Bus
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

Chapter 2

Predictive Outage Compensation

Consider the networked control system in Figure 2.1, which shows how actuators, sensors and controllers are being connected through a wireless network. Sensors and controllers use a medium access control (MAC) protocol to decide when to transmit sensor and control data over the network. These data are received by the controllers and actuators. At the input to each such device there is a predictive outage compensator (POC), which is a filter that can generate artificial samples during outage. If the MAC and POC are working appropriately, they allow us to abstract away the details of the network in the control design.

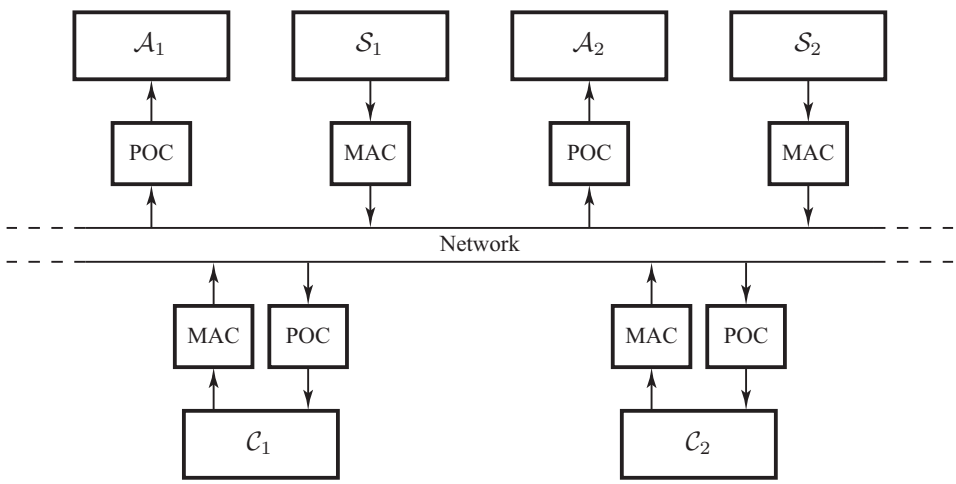


Figure 2.1: Actuator, sensor and control devices are interfaced to the network through MAC and POC protocols

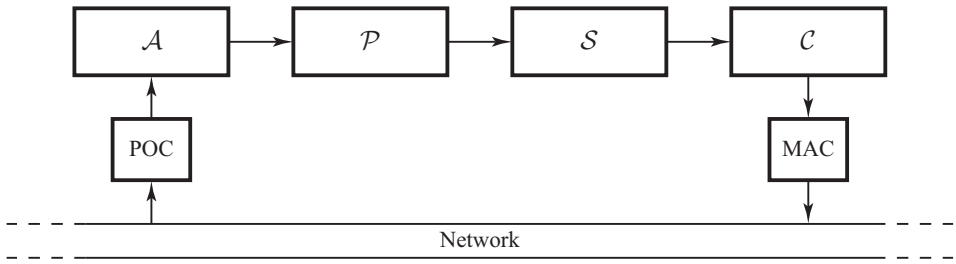


Figure 2.2: Control setup showing how actuator, process, sensor and controller are interfaced to the network through MAC and POC protocols

In this chapter we will study the particular control setup shown in Figure 2.2. The MAC and the network are not studied further in the thesis. Instead the focus is on how to construct the POC when placed as in Figure 2.2.

The proposed POC is a generalization of the communication outage compensation algorithms used today such as holding the last known value or applying constant outputs, and is related to a generalized hold function (*e.g.*, (Sun et al., 1993)). The general idea is to monitor the control signal and use a signal model to extrapolate the signal in the event of a communication outage. The POC listens to the control signal sent from the controller. If the signal is received, the POC passes the control signal forward to the actuator and updates its own internal states using the received signal. In the case that no control action is received the POC uses its internal model to extrapolate the control signal based on the signal model and previously received data.

The chapter is outlined as follows. First the system is described and the considered problem is detailed. After this the form of the POC is given, followed by the proposed procedure for commissioning and running it. Finally the chapter is summarized.

2.1 System Description

The problem we will consider is controlling the linear plant $\mathcal{P}$ over a communication network with sporadic outages. The plant $\mathcal{P}$ is given by

$$\mathcal{P}\begin{cases} x_p(k+1) = A_p x_p(k) + B_u u(k) + B_d d(k) \\ y(k) = C_p x_p(k) + v(k), \end{cases} \tag{2.1}$$

where $d(k)$ and $v(k)$ are process and measurement noise, respectively. When there is no communication outage, the control $u(k) = u_c(k)$ is given by the controller $\mathcal{C}$

$$\mathcal{C}\begin{cases} x_c(k+1) = A_c x_c(k) + B_c y(k) \\ u_c(k) = C_c x_c(k). \end{cases} \tag{2.2}$$

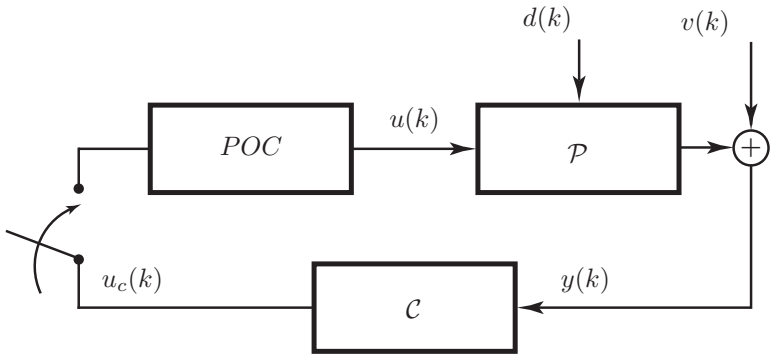


Figure 2.3: The control setup studied where the MAC and Network are abstracted by a switch turning communication on or off.

When the communication is lost and we have an outage a replacement control command $u(k) = \hat{u}(k)$ is applied.

From now on we will use an abstraction of the control setup presented in Figure 2.2 where the MAC and the network in the communication link are represented by a switch turning communication on or off, so that the command from the controller is either received or lost. The abstracted setup is shown in Figure 2.3.

When the switch in Figure 2.3 is open, *i.e.*, communication is lost, the POC will generate the replacement command $\hat{u}(k)$ and apply it to the process so that $u(k) = \hat{u}(k)$ during the outage. The process will then evolve according to

$$\begin{aligned} x_{p,out}(k+1) &= A_p x_{p,out}(k) + B_u \hat{u}(k) + B_d d(k) \\ y_{out}(k) &= C_p x_{p,out}(k) + v(k). \end{aligned}$$

The effect of the outage on the plant becomes

$$\Delta x_p(k+1) = A_p \Delta x_p(k) + B \Delta u(k), \tag{2.3}$$

where $\Delta u(k) = u_c(k) - \hat{u}(k)$ and $\Delta x_p(k) = x_p(k) - x_{p,out}(k)$. To minimize the effect of the outage, *i.e.*, make $\Delta x_p(k)$ small, we would like to make $\Delta u(k)$ as small as possible. Note that $d(k)$ is assumed independent of the outage.

Remark 2.1.1. Keeping $\Delta u(k)$ small is an indirect way of keeping $\Delta x_p(k)$ small. If one views (2.3) as a system with input $\Delta u(k)$ and state $\Delta x_p(k)$ it becomes clear that the optimal open loop trajectory of $\Delta u(k)$ minimizing $\|\Delta x_p(k)\|_2$ is not guaranteed to be small at all. For example if $\Delta x_p(k) = 0$ the optimal strategy is to keep $\Delta u(k) = 0$ but for $\Delta x_p(k) \neq 0$, $\Delta u(k)$ has a distinct curve shape. Although for a reasonably well behaved system with stable A_p, a small $\Delta u(k)$ should keep $\Delta x_p(k)$ small. On the other hand, if A_p is unstable $\Delta x_p(k)$ will start to grow even for small $\Delta u(k)$.

The reason for working with $\Delta u(k)$ is twofold. First of all it is a simpler problem to make $\Delta u(k)$ small than to make $\Delta x_p(k)$ small. Secondly the problem of making $\Delta u(k)$ small can be solved using only local information available at the POC.

Remark 2.1.2. When communication is lost we loose the feedback in the control system and hence $\hat{u}(k)$ must be generated by an open-loop controller. Unstable processes $\mathcal{P}$ are therefore inherently difficult to handle regardless of which method is used to generate $\hat{u}(k)$.

2.2 Predictive Outage Compensator

Let us now formulate the problem and give the structure of the POC. As mentioned previously the problem we want to solve is how to choose $\hat{u}(k)$ in the switching policy

$$u(k) = \begin{cases} u_c(k) & \text{Command from controller received} \\ \hat{u}(k) & \text{Command from controller lost} \end{cases} \tag{2.4}$$

such that $\Delta u(k) = u_c(k) - \hat{u}(k)$ is made as small as possible.

We assume that the POC takes the state-space form

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + K\epsilon(k) \\ \hat{u}(k|k-1) &= C\hat{x}(k), \end{aligned} \tag{2.5}$$

where $\epsilon(k) = u_c(k) - \hat{u}(k|k-1) = u_c(k) - C\hat{x}(k)$ is the one-step-ahead prediction error of the POC, and $\hat{u}(k|k-1)$ is the predicted value of $u_c(k)$ given measurements up to $k-1$. If $u_c(k)$ is lost $\epsilon(k)$ can not be computed, instead we assume that the prediction is correct giving $\epsilon(k) = 0$. The matrices (A, K, C) are design parameters, how they should be chosen is discussed further in Chapter 3. However, a standing assumption in the thesis is that $A - KC$ is a Schur matrix so that (2.5) is an asymptotically stable system.

Connecting back to Figure 2.3, the operation of the POC is illustrated in Figure 2.4 where the network is represented by the switch as in (2.4). The method for commissioning the POC is given next.

2.3 Predictive Compensation Procedure

We now consider the procedure used when commissioning the proposed POC. The work flow is illustrated in Figure 2.5 which shows the different steps in commissioning and running the algorithm. Theses steps will be described in detail next.

2.3.1 Initialization

When first commissioning the POC one needs to make some initial design decisions. The first is to decide the model order of the POC, *i.e.*, the number of states in

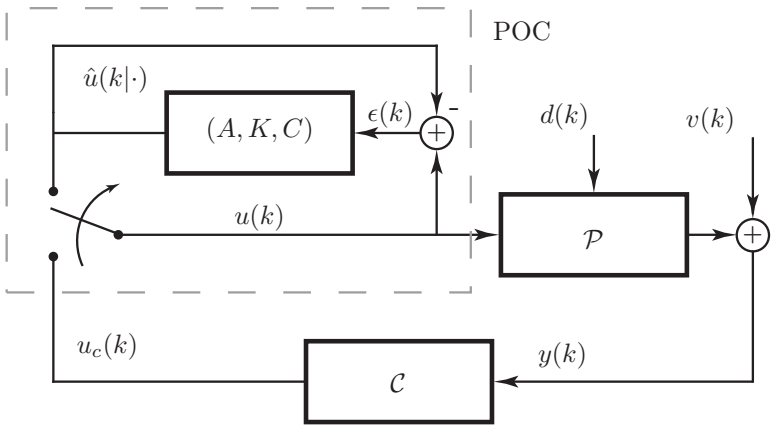


Figure 2.4: The Predictive Outage Compensator.

$\hat{x}(k)$ in (2.5). The needed complexity is set by the desired performance of the system and what kind of deviations that are tolerable during an outage. For a given performance criterion the main underlying factors that govern the needed complexity are the expected outage lengths and how fast the underlying system is. A limiting factor on the complexity is of course also the available computational power. In general a lower complexity is possible if higher deviations can be allowed or if the packet loss pattern is sparse with short periods of loss. Methods to analyze the needed complexity are treated in Chapter 4. Another decision that needs to be taken during the initialization phase is the methodology for detecting packet losses.

2.3.2 Compensator Tuning

The next stage is to tune the POC by choosing values for the matrices (A, K, C) according to the decided model order. How (A, K, C) should be chosen is of course dependent on the system dynamics and what type of disturbances the system is subject to. Different methods for finding (A, K, C) for a given system under a given type of disturbance are treated further in Chapter 3. Chapter 3 also gives bounds which relate the quality of the tuning and the tolerable outage length.

2.3.3 Control Monitoring

This step is the core part of the predictive outage compensation procedure. The POC is in this mode when the communication is working and the control signal $u_c(k)$ has been received from the controller. In this mode the POC compares the received control signal with its own estimate $\hat{u}(k|k-1)$. The difference between the true and predicted value is computed as the prediction error signal

$$\epsilon(k) = u_c(k) - \hat{u}(k|k-1) = u_c(k) - C\hat{x}(k).$$

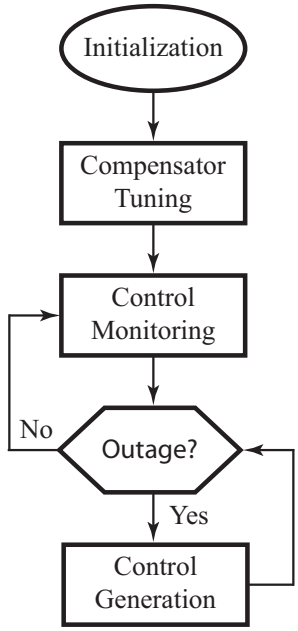


Figure 2.5: Flow diagram describing the POC

The new information in $\epsilon(k)$ is then used to update the POCs internal states as

$$\hat{x}(k+1) = A\hat{x}(k) + K\epsilon(k)$$

from which an estimate of the control signal $u_c(k+1)$ can be computed as $\hat{u}(k+1|k) = C\hat{x}(k+1)$.

2.3.4 Outage Detection

This is the communication monitoring part of the POC where it listens to the communication channel for the control signal from the controller. If the control signal $u_c(k)$ is received the POC goes into the control monitoring mode and updates its internal states. If the control signal is not received the POC enters the control generation mode.

2.3.5 Control Generation

When communication is lost and we no longer receive any control signals we can not compute $\epsilon(k) = u_c(k) - \hat{u}(k|k-1)$, since $u_c(k)$ is not known. Instead when an outage occurs, say at time k', we set $\epsilon(k) = 0$ and let the POC switch to "prediction

mode" and compute the predicted input to the plant for $k > k'$ as

$$\begin{aligned} \hat{x}(k+1|k') &= A\hat{x}(k|k') \\ \hat{u}(k|k') &= C\hat{x}(k|k'). \end{aligned} \tag{2.6}$$

How large $\Delta u(k) = u_c(k) - \hat{u}(k|k')$ in (2.3) becomes of course depend on how the realization (A, K, C) is chosen and the realization of the noises $d(k)$ and $v(k)$. For the methods to synthesize (A, K, C) given in Chapter 3 bounds on $\Delta u(k)$ are given as functions of the closed-loop system dynamics and noise realization.

2.4 Summary

In this chapter we have introduced the POC and the system description. Further we have introduced a general form of POC in (2.5). Finally the procedure for commissioning and running the POC has been described.

Chapter 3

Synthesis

Within the proposed predictive outage compensation framework presented in Chapter 2 we have the freedom to choose the realization of the POC, namely (A, K, C), arbitrarily. Clearly there will for a given system exist better and worse choices of this realization. In this chapter we will present two methods to synthesize a POC: one deterministic and one stochastic. The deterministic method is convenient for SISO systems affected by a slowly varying deterministic disturbance and the stochastic method is practical for MIMO systems affected by stochastic disturbances and measurement noise.

The chapter starts by reintroducing the central notation in Chapter 2. Then we proceed to present the deterministic synthesis method and the corresponding worst-case error bound. After this the stochastic synthesis method is presented together with the corresponding optimal error bound. The chapter is concluded by a short summary.

3.1 Introduction

For convenience we start by reintroducing the equations from Chapter 2 that govern the system and the POC.

First, the process $\mathcal{P}$ is given by

$$\mathcal{P} \begin{cases} x_p(k+1) = A_p x_p(k) + B_u u(k) + B_d d(k) \\ y(k) = C_p x_p(k) + v(k), \end{cases} \tag{3.1}$$

where $d(k)$ and $v(k)$ are process and measurement noise, respectively. Due to packet losses in the controller to process communication the control signal $u(k)$ in $\mathcal{P}$ is given by the switching policy

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

where $u_c(k)$ is given by the controller $\mathcal{C}$

$$\mathcal{C}\begin{cases} x_c(k+1) = A_c x_c(k) + B_c y(k) \\ u_c(k) = C_c x_c(k). \end{cases} \tag{3.2}$$

and $\hat{u}(k)$ is given by the POC

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + K\epsilon(k) \\ \hat{u}(k|k-1) &= C\hat{x}(k). \end{aligned} \tag{3.3}$$

When communication between the controller and the process is working the POC estimate is updated with $\epsilon(k) = u_c(k) - \hat{u}(k|k-1)$. When communication is lost we set $\epsilon(k) = 0$ and let the POC evolve in open loop as

$$\begin{aligned} \hat{x}(k+1|k') &= A\hat{x}(k|k') \\ \hat{u}(k|k') &= C\hat{x}(k|k') \end{aligned} \tag{3.4}$$

where k' is the time instant of the last known packet.

We now move on to characterizing the deterministic and stochastic synthesis methods.

3.2 Deterministic Synthesis

Let us proceed by characterizing the deterministic synthesis method to find (A, K, C) for SISO systems affected by a slowly varying deterministic disturbance. The derivation is based on transfer functions but the resulting POC can be translated to the state-space form (3.3) and explicit expressions for (A, K, C) are given.

Following the notation in Section 3.1 we will in the derivation assume that the control signal in closed loop is generated by the transfer function from $d(k)$ to $u_c(k)$ as

$$u_c(k) = G^0(q)d(k) = \frac{E^0(q)}{F^0(q)}d(k) \tag{3.5}$$

where q denotes the one step forward shift operator and $E^0(q)$ and $F^0(q)$ are polynomials of degree n_{E0} and n_{F0}. The effect of the measurement noise is assumed to be incorporated in the disturbance $d(k)$ so that $v(k) \equiv 0$.

In reality the transfer function $G^0(q)$ can not be assumed to be known, instead it must be identified. This can be done based on for example conventional system identification (Ljung and Söderström, 1983) or physical modelling. The identification problem is not treated further in the thesis, instead it is assumed that an estimate $G(q)$ of $G^0(q)$ is available. Given the estimate $G(q)$ the control signal estimate is given by

$$\hat{u}(k) = G(q)\hat{d}(k) = \frac{E(q)}{F(q)}\hat{d}(k) \tag{3.6}$$

where $E(q)$ and $F(q)$ are polynomials of degree n_E and n_F respectively and $\hat{d}(k)$ can be interpreted as a virtual disturbance, or as our guess of the real $d(k)$.

3.2.1 Studied Scenario

In the studied scenario it is assumed that the disturbance $d(k)$ is a load disturbance entering the system at the control signal. The output from the process $\mathcal{P}$, see (3.1), is then given by

$$y(k) = P(q)\Big(u(k) + d(k)\Big).$$

The output from the controller $\mathcal{C}$, see (3.2), is given by

$$u_c(k) = -C(q)y(k).$$

By this we can conclude that when communication is working, *i.e.*, the loop is closed, we have the system output (3.7)

$$y(k) = \frac{P(q)}{1 + P(q)C(q)}d(k) \tag{3.7}$$

and the controller signal (3.8)

$$u_c(k) = -\frac{P(q)C(q)}{1 + P(q)C(q)}d(k) = G^0(q)d(k). \tag{3.8}$$

We see that $G^0(q) = -T(q)$, where $T(q)$ is the complementary sensitivity function for the closed loop system. That is, given a reasonable control design, $G^0(q)$ should be a well-behaved function with lowpass characteristics. By comparing (3.5) and (3.6) one can see that for a good choice of $G(q)$ and $\hat{d}(k)$ we should be able to get similar or same behavior of the system output when communication is lost as when it is functioning.

3.2.2 Compensation Procedure

First we note that the model $G(q)$ may differ from the true system $G^0(q)$. It will turn out that it is useful to define an equivalent disturbance $\hat{d}^*(k)$ as

$$u_c(k) = G(q)\hat{d}^*(k) = \frac{E(q)}{F(q)}\hat{d}^*(k), \tag{3.9}$$

i.e., the value of $\hat{d}(k)$ making $\hat{u}(k) = u_c(k)$. As long as communication is working the POC in this method uses the inverted model $G^{-1}(q)$ and the received control signal $u_c(k)$ to filter out an estimate of $\hat{d}^*(k)$ as

$$\hat{d}^*(k) = G^{-1}(q)u_c(k) = \frac{F(q)}{E(q)}u_c(k)$$

or more precisely shifted as $q^{-n_F}E(q)\hat{d}^*(k) = q^{-n_F}F(q)u_c(k)$. Which after rearranging gives

$$\begin{aligned} &e_0\hat{d}^*(k + (n_E - n_F)) = \\ &\quad = u_c(k) + \ldots + f_{n_F}u_c(k - n_F) - e_1\hat{d}^*(k - n_F - 1 + n_E) - \ldots - e_{n_E}\hat{d}^*(k - n_F) \end{aligned}$$

where e_i and f_i are the coefficients of $E(q)$ and $F(q)$ respectively. As seen, what one can actually estimate is $\hat{d}^*(k)$ shifted back $n_F - n_E$ which is the relative degree of the POC.

When an outage occurs an estimate of $\hat{d}^*(k)$ is used to drive the POC filter. Since we can no longer estimate $\hat{d}^*(k)$ using $u_c(k)$ we have to decide on another method. In this thesis we propose to use the method to hold the last known $\hat{d}^*(k)$ which we denote by $\bar{d}$. This is motivated by that if $G(q)$ is close to $G^0(q)$ and $d(k)$ is slowly varying, then $\hat{d}^*(k)$ should also be slowly varying, and that estimating a slowly varying signal by a constant is reasonable. Another policy is to set $\bar{d} = 0$, this is reasonable if for example $\hat{d}^*(k)$ is zero mean white noise. The POC in prediction mode is then given by

$$\hat{u}(k) = \frac{E(q)}{F(q)}\bar{d} \tag{3.10}$$

again shifted as $q^{-n_F}F(q)\hat{u}(k) = q^{-n_F}E(q)\bar{d}$, which in the same way as before gives

$$\hat{u}(k) = \Big(e_0 + \ldots + e_{n_E}\Big)\bar{d} - f_1\hat{u}(k-1) - \ldots - f_{n_F}\hat{u}(k-n_F).$$

If the true value for u_c is known for any time instant in the recursion, the true value is used instead of the predicted $\hat{u}$.

3.2.3 State-Space Realization

In order to describe the POC derived above on the POC state-space form in (3.3) and compute prediction error bounds, we need to realize (3.5) and (3.6) on state-space form. To do this we first introduce the signals

$$p(k) := q^{1-n_{F0}}E^0(q)d(k), \quad \hat{p}(k) := q^{1-n_F}E(q)\hat{d}(k)$$

which we use to re-define (3.5) and (3.6) in the following way:

$$u_c(k) = \frac{q^{n_{F0}-1}}{F^0(q)}p(k) \tag{3.11}$$

$$\hat{u}(k) = \frac{q^{n_F-1}}{F(q)}\hat{p}(k) \tag{3.12}$$

where $\deg F = n_F$ and $\deg F^0 = n_{F0}$. The signal $p(k)$ can be thought of as a driving noise signal that contains everything in $u_c(k+1)$ that cannot be explained by a linear combination of $u_c(k), \ldots, u_c(k-n_{F0}+1)$. The signal $\hat{p}(k)$ has a similar interpretation for $\hat{u}(k)$.

The system (3.11) can be realized in the state-space form

$$\begin{aligned} z(k) &= c(F^0)z(k-1) + Hp(k-1) \\ u_c(k) &= H^T z(k), \end{aligned} \tag{3.13}$$

where $c(F^0)$ is a companion matrix of the polynomial $F^0(q) = q^{n_{F0}} + f_1^0 q^{n_{F0}-1} + \ldots + f_{n_{F0}}^0$,

$$c(F^0) = \begin{pmatrix} -f_1^0 & -f_2^0 & \ldots & -f_{n_{F0}}^0 \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix} \in \mathbb{R}^{n_{F0} \times n_{F0}}$$

and

$$H = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n_{F0}}, \quad z(k) = \begin{pmatrix} u_c(k) \\ u_c(k-1) \\ \vdots \\ u_c(k-n_{F0}+1) \end{pmatrix} \in \mathbb{R}^{n_{F0}}.$$

The model (3.12) can be realized in the same form using a companion matrix $c(F)$, as will be seen next. If $\deg F < \deg F^0$, we can define the polynomial coefficients $f_{n_F+1} = \ldots = f_{n_{F0}} = 0$ so that $c(F) \in \mathbb{R}^{n_{F0} \times n_{F0}}$, and both models have the same state dimension.

3.2.4 Computing A, K and C

The deterministically synthesized POC can, using the model (3.12) and the companion matrix $c(F)$, be written on predictor form as

$$\begin{aligned} \hat{z}(k|k-1) &= c(F)\hat{z}(k-1|k-1) + H\hat{p}(k-1|k-1) \\ \hat{z}(k|k) &= \hat{z}(k|k-1) + K_z\epsilon(k) \\ \hat{u}(k|k-1) &= H^T\hat{z}(k|k-1) \\ \hat{u}(k|k) &= H^T\hat{z}(k|k) \end{aligned} \tag{3.14}$$

where $\epsilon(k) = u_c(k) - \hat{u}(k|k-1)$ and

$$\hat{z}(k|k) = \begin{pmatrix} \hat{u}(k|k) \\ \hat{u}(k-1|k-1) \\ \vdots \\ \hat{u}(k-n_F+1|k-n_F+1) \end{pmatrix}.$$

As mentioned in Section 3.2.2 the true value of the control signal is used in the prediction recursion provided it is known, *i.e.*, we require $\hat{u}(k|k) = u_c(k)$ provided the feedback loop is closed. That is we require

$$\begin{aligned} \hat{u}(k|k) &= H^T\hat{z}(k|k) = \hat{u}(k|k-1) + H^T K_z\epsilon(k) \\ &= \left(1 - H^T K_z\right)\hat{u}(k|k-1) + H^T K_z u_c(k) \equiv u_c(k) \end{aligned} \tag{3.15}$$

which is fulfilled for $K_z = H$ making $H^T K_z = H^T H = 1$.

Remaining to be decided is how $\hat{p}(k-1|k-1)$ in (3.14) should be chosen. First we define $\hat{p}(k-1|k)$ as the signal containing all the information in $u_c(k)$ that can not be explained by the linear combination $H^T c(F)\hat{z}(k-1|k-1)$, *i.e.*, by the regression of past estimates $\hat{u}$. That is, in feedback

$$\hat{p}(k-1|k) := u_c(k) - H^T c(F)\hat{z}(k-1|k-1)$$

Which following the notation in (3.9) means that $\hat{p}(k-1|k) = q^{-N_{F^0}} E(q)\hat{d}^*(k)$.

When communication is lost $\hat{p}(k-1|k)$ is not computable and instead we apply $\hat{p}(k-1|k-1)$, *i.e.*, an estimate of $\hat{p}(k-1|k)$ given measurements up to $k-1$. If we use the policy to hold the last disturbance estimate as described in Section 3.2.2 the prediction equation is given by

$$\begin{aligned}\hat{p}(k-1|k) &= \hat{p}(k-1|k-1) + K_p\epsilon(k)\\ \hat{p}(k|k) &= \hat{p}(k-1|k).\end{aligned}$$

For this measurement update to comply with the definition of $\hat{p}(k-1|k)$ we require

$$\begin{aligned}\hat{p}(k-1|k) &= \hat{p}(k-1|k-1) + K_p\epsilon(k)\\ &= \hat{p}(k-1|k-1) + K_p u_c(k) - K_p\hat{u}(k|k-1)\\ &= \hat{p}(k-1|k-1) + K_p u_c(k)\\ &\quad - K_p\Big(H^T c(F)\hat{z}(k-1|k-1) + H^T H\hat{p}(k-1|k-1)\Big)\\ &= \Big(1 - K_p\Big)\hat{p}(k-1|k-1) + K_p\Big(u_c(k) - H^T c(F)\hat{z}(k-1|k-1)\Big)\\ &\equiv u_c(k) - H^T c(F)\hat{z}(k-1|k-1)\end{aligned}$$

which is fulfilled for $K_p = 1$.

We summarize the above derivations of $\hat{z}(k|k)$ and $\hat{z}(k|k)$ in the deterministic synthesis method in the following proposition.

Proposition 3.2.1. *The POC synthesized using the deterministic method can be realized on the form (3.3) as*

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

where

$$\hat{x}(k+1) = \begin{pmatrix}\hat{z}(k|k)\\ \hat{p}(k|k)\end{pmatrix}, \quad A = \begin{pmatrix}c(F) & H\\ 0 & 1\end{pmatrix},$$

$$K = \begin{pmatrix}H\\ 1\end{pmatrix}, \quad C = \begin{pmatrix}H^T c(F) & 1\end{pmatrix}$$

Remark 3.2.2. Note that

$$A - KC = \begin{pmatrix} \Big(I - HH^T\Big)c(F) & 0 \\ -H^T c(F) & 0 \end{pmatrix}$$

and that

$$\Big(I - HH^T\Big)c(F) = \begin{pmatrix} 0 & 0 \\ I & 0 \end{pmatrix}$$

making

$$\lambda_i\Big(A - KC\Big) = 0, \quad \forall i$$

Hence, in feedback, $\hat{z}(k|k) = z(k)$ regardless of how $F(q)$ in (3.12) is chosen.

Remark 3.2.3. If instead we choose the alternative policy to set the disturbance estimate to zero, $\bar{d} = 0$, in outage, also described in Section 3.2.2, we get the following formulation. First we have to re-define $\hat{p}(k|k)$ as the prediction equation is given by

$$\hat{p}(k|k) = 0 \cdot \hat{z}(k-1|k-1) + 0 \cdot \hat{p}(k-1|k-1) + 0 \cdot \epsilon(k) \equiv 0$$

(A, K, C) in Proposition 3.2.1 are now instead given as

$$A = \begin{pmatrix} c(F) & H \\ 0 & 0 \end{pmatrix}, \quad K = \begin{pmatrix} H \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} H^T c(F) & 1 \end{pmatrix}.$$

We note that this is not a minimal realization. A minimal realization is obtained by instead choosing (A, K, C) as

$$A = c(F), \quad K = H, \quad C = H^T c(F).$$

3.2.5 Examples of POCs

Let us consider the two most commonly used outage compensation methods and how they relate to the deterministic synthesis method.

Example 3.1

A common version of outage compensation is to keep applying the last received control command when no command new is received. That is $\hat{u}(k|k') = u_c(k')$ in outage, where $u_c(k')$ is the last known command. We note that this method can be modelled in the deterministic synthesis framework by

$$\begin{aligned} \hat{z}(k|k) &= 1 \cdot \hat{z}(k-1|k-1) + 1 \cdot \hat{p}(k-1|k-1) + 1 \cdot \epsilon(k) \\ \hat{p}(k|k) &= 0 \cdot \hat{z}(k-1|k-1) + 0 \cdot \hat{p}(k-1|k-1) + 0 \cdot \epsilon(k) \equiv 0 \\ \hat{u}(k|k-1) &= 1 \cdot \hat{z}(k-1|k-1) + 1 \cdot \hat{p}(k-1|k-1). \end{aligned}$$

This is a POC designed as described in Remark 3.2.3 here with $c(F) = 1$ and $H = 1$ giving $A = K = C = 1$. For future reference we denote this realization as the **hold POC**.

Example 3.2
The simplest version of outage compensation is to use an a priori decided constant command if no new control signal arrives. A common choice of this constant is zero, giving $\hat{u}(k|k-1) = 0$ during the outage. In the deterministic synthesis framework this means

$$\begin{aligned}\hat{z}(k|k) &= 0 \cdot \hat{z}(k-1|k-1) + 1 \cdot \hat{p}(k-1|k-1) + 1 \cdot \epsilon(k)\\ \hat{p}(k|k) &= 0 \cdot \hat{z}(k-1|k-1) + 0 \cdot \hat{p}(k-1|k-1) + 0 \cdot \epsilon(k) \equiv 0\\ \hat{u}(k|k-1) &= 0 \cdot \hat{z}(k-1|k-1) + 1 \cdot \hat{p}(k-1|k-1),\end{aligned}$$

This is also a POC designed as described in Remark 3.2.3. Here we have $c(F) = 0$ and $H = 1$ giving $A = 0$, $K = 1$ and $C = 1 \cdot 0 = 0$. Since $C = 0$ the POC input-output relation is equivalently realized by $A = K = C = 0$. We will for future reference denote this realization as the **zero POC**.

3.3 Worst-Case Prediction Error Bounds

We have now derived the deterministic POC synthesis method and expressed how the resulting POC is realized on the form (3.3). We will proceed by computing deterministic worst-case bounds on the prediction error $|u_c(k) - \hat{u}(k|k')|$ in outage, where k' denotes the time at which the last command was received. The section is concluded with an example illustrating the worst-case bound for a POC synthesized using the deterministic method and how this bound relate to bounds for the hold and zero POCs, as a function of the outage length.

In order to bound the error between the ideal input from the nominal model, $u_c(k)$, and the input from the POC, $\hat{u}(k|k')$ we are going to use the following lemma.

Lemma 3.3.1. Consider the linear time-invariant input-output model

$$\delta(k) = \sum_{j=k_0}^{k} \gamma(k-j)\rho(j), \quad k \geq k_0, \tag{3.16}$$

with impulse response $\gamma(j)$. It holds that

$$|\delta(k)| \leqslant \left(\sum_{j=0}^{k-k_0} |\gamma(j)| \right) \max_{k_0 \leqslant j \leqslant k} |\rho(j)|.$$

For bounded input over the interval $[k_0, k_f]$, the maximum output over the same interval is bounded by

$$\max_{k_0 \leqslant k \leqslant k_f} |\delta(k)| \leqslant \left(\sum_{j=0}^{k_f - k_0} |\gamma(j)| \right) \max_{k_0 \leqslant k \leqslant k_f} |\rho(k)|.$$

Both bounds are tight, *i.e.*, there is an input $\rho(k)$ that achieves equality.

Proof. The proof is an application of Theorem 27.2 in (Rugh, 1996). Taking the norm of (3.16), applying the triangle inequality and using the submultiplicative property of the norm we get

$$\begin{aligned} |\delta(k)| = \left| \sum_{j=k_0}^{k} \gamma(k-j)\rho(j) \right| &\leqslant \sum_{j=k_0}^{k} |\gamma(k-j)\rho(j)| \\ &\leqslant \sum_{j=k_0}^{k} |\gamma(k-j)||\rho(j)|, \quad k \geq k_0, \end{aligned}$$

then replacing $|\rho(j)|$ by its maximum and using the fact that this is a constant value we have

$$\begin{aligned} |\delta(k)| &\leqslant \sum_{j=k_0}^{k} |\gamma(k-j)||\rho(j)| \leqslant \sum_{j=k_0}^{k} \left(|\gamma(k-j)| \max_{k_0 \leqslant j \leqslant k} |\rho(j)| \right) \\ &= \left(\sum_{j=k_0}^{k} |\gamma(k-j)| \right) \max_{k_0 \leqslant j \leqslant k} |\rho(j)|, \quad k \geq k_0. \end{aligned}$$

Clearly an upper bound on $|\delta(k)|$, $k \geq k_0$ is an upper bound on $\max_{k_0 \leqslant k} |\delta(k)|$. Hence

$$\max_{k_0 \leqslant k} |\delta(k)| \leqslant \left(\sum_{j=k_0}^{k} |\gamma(k-j)| \right) \max_{k_0 \leqslant j \leqslant k} |\rho(j)|,$$

If we upper bound k by k_f so that $k_0 \leqslant k \leqslant k_f$ it holds that

$$\begin{aligned} \max_{k_0 \leqslant k \leqslant k_f} |\delta(k)| &\leqslant \left(\sum_{j=k_0}^{k_f} |\gamma(k_f - j)| \right) \max_{k_0 \leqslant j \leqslant k_f} |\rho(j)| \\ &= \left(\sum_{j=0}^{k_f - k_0} |\gamma(j)| \right) \max_{k_0 \leqslant k \leqslant k_f} |\rho(k)| \end{aligned}$$

□

To compute the bound we will also need the following assumption.

Assumption 3.3.1. We assume that the outage occurs at time $k = 0$, without loss of generality. By this we mean that the feedback is lost, *i.e.*, $u_c(k)$ is not known, for k in the time interval $k \in [1, k_f]$. Further we assume that the system has been in feedback for $k \in [-n_{F^0} + 1, 0]$ so that $u_c(k)$ is known for all k in this interval.

The prediction error can now be computed as described in the following proposition.

Proposition 3.3.2. *Under Assumption 3.3.1 and by the property ensured in (3.15) it holds that* $\hat{z}(0|0) = z(0)$. *The signals* $\hat{u}(k) = \hat{u}(k|0)$ *and* $u_c(k)$ *when* $k \in [0, k_f]$ *are given by*

$$u_c(k) = H^T c(F^0)^k z(0) + \sum_{j=0}^{k-1} H^T c(F^0)^{k-j-1} H p(j) \tag{3.17a}$$

$$\hat{u}(k) = H^T c(F)^k z(0) + \sum_{j=0}^{k-1} H^T c(F)^{k-j-1} H \hat{p}(j) \tag{3.17b}$$

so the error between ideal and actual input in outage mode is given by

$$\begin{aligned} u_c(k) - \hat{u}(k) &= H^T [c(F^0)^k - c(F)^k] z(0) \\ &+ \sum_{j=0}^{k-1} H^T c(F^0)^{k-j-1} H p(j) - \sum_{j=0}^{k-1} H^T c(F)^{k-j-1} H \hat{p}(j). \end{aligned} \tag{3.18}$$

In order to derive simple expressions for the error bounds, the following assumptions are made. They should be relatively easy to verify for a given system.

Assumption 3.3.3. It is assumed that both (3.11) and (3.12) are exponentially stable, *i.e.*, there are constants $c_0 > 0$, $c > 0$, $0 \leq \lambda_0 \leq 1$, $0 \leq \lambda \leq 1$ such that

$$\|H^T c(F^0)^k\|_1 \leq c_0 \lambda_0^k, \quad \|H^T c(F)^k\|_1 \leq c\lambda^k$$

Where $\|\cdot\|_1$ is the 1-norm of a vector (sum of magnitude of elements). Furthermore, we assume the actual input, the disturbance and the disturbance estimate are bounded so that

$$\begin{aligned} |u_c(k)| \leq \rho_u, \quad |p(k)| \leq \rho_p, \quad \forall k \\ |\hat{p}(k)| \leq \rho_{\hat{p}}, \quad k \geq 0 \end{aligned}$$

The constants λ and λ_0 are measures of how fast the systems are. Using the error model (3.18) and the assumptions we are going to analyze the error behavior. A simple example is also given at the end of this section.

3.3.1 Nominal Deterministically Synthesized POC

Let us denote the deterministically synthesized POC with $F(q) = F^0(q)$ the nominal deterministically synthesized POC. The full error model (3.18) then reduces to

$$u_c(k) - \hat{u}(k) = \sum_{j=0}^{k-1} H^T c (F^0)^{k-j-1} H \Big(p(j) - \hat{p}(j) \Big).$$

From this the worst-case prediction bounds for the nominal deterministically synthesized POC can now be computed by applying the triangle inequality and Lemma 3.3.1 to the error giving

$$\begin{aligned} |u_c(k) - \hat{u}(k)| &\leq \Big(\rho_p + \rho_{\hat{p}} \Big) \sum_{j=0}^{k-1} |H^T c (F^0)^{k-j-1} H| \\ &\leq \Big(\rho_p + \rho_{\hat{p}} \Big) c_0 \sum_{j=0}^{k-1} \lambda_0^{k-j-1} = \Big(\rho_p + \rho_{\hat{p}} \Big) c_0 \frac{1 - \lambda_0^k}{1 - \lambda_0} =: \Gamma_0(\rho_p, \rho_{\hat{p}}, k) \end{aligned}$$

The bound Γ_0 converges exponentially fast at a rate λ_0 to

$$\Big(\rho_p + \rho_{\hat{p}} \Big) \frac{c_0}{1 - \lambda_0}.$$

3.3.2 General Deterministically Synthesized POC

Now let us compute the worst-case prediction bounds for the general deterministically synthesized POC,

$$\hat{u}(k+1) = -f_1 \hat{u}(k) - \ldots - f_{n_F} \hat{u}(k - n_F + 1) + \hat{p}(k),$$

in order to compute the bound we make the following assumption.

Assumption 3.3.4. Assume there are constants $c_1 \geq c_1'$ and $1 \geq \lambda_1 \geq \lambda_1' \geq 0$ such that

$$\| H^T [c(F^0)^k - c(F)^k] \|_1 \leq c_1 \lambda_1^k - c_1' (\lambda_1')^k$$

Conservative choices that work under Assumption 3.3.3 are $c_1 = c + c_0$, $\lambda_1 = \max\{\lambda, \lambda_0\}$, and $c_1' = 0$.

The worst-error bound can now be computed by applying the assumptions, the

triangle inequality and Lemma 3.3.1 to the full error model (3.18). We now have

$$\begin{aligned}|u_c(k) - \hat{u}(k)| &\leq \rho_u \|H^T[c(F^0)^k - c(F)^k]\|_1 \\ &\quad + \rho_p \sum_{j=0}^{k-1} |H^T c(F^0)^{k-j-1} H| + \rho_{\hat{p}} \sum_{j=0}^{k-1} |H^T c(F)^{k-j-1} H| \\ &\leq \rho_u (c_1 \lambda_1^k - c_1'(\lambda_1')^k) + \rho_p c_0 \sum_{j=0}^{k-1} \lambda_0^{k-j-1} + \rho_{\hat{p}} c \sum_{j=0}^{k-1} \lambda^{k-j-1} \\ &= \rho_u (c_1 \lambda_1^k - c_1'(\lambda_1')^k) + \rho_p c_0 \frac{1-\lambda_0^k}{1-\lambda_0} + \rho_{\hat{p}} c \frac{1-\lambda^k}{1-\lambda} =: \Gamma_1(\rho_u, \rho_p, \rho_{\hat{p}}, k)\end{aligned}$$

The bound Γ_1 converges exponentially fast at a rate λ_1 to

$$\frac{\rho_p c_0}{1-\lambda_0} + \frac{\rho_{\hat{p}} c}{1-\lambda}$$

Remark 3.3.5. To make the worst-case bounds Γ_0 and Γ_1 small, it is clear that it is best to use a zero policy for $\hat{p}(k)$, *i.e.*, $\rho_{\hat{p}} = 0$. It is important to remember that this is a strict worst-case analysis that assumes that we have no knowledge whatsoever of $p(k)$. If we have knowledge of how quickly $p(k)$ evolves, then it can be very beneficial to choose a nonzero $\hat{p}(k)$ to counteract it, as shall be seen in Chapter 5.

3.3.3 Hold POC

We move on to the hold POC introduced in Example 3.1. First let us recall that here $\hat{u}(k) = \hat{u}(k-1)$. Hence, if there is an outage at $k = 0$, we have $\hat{u}(k) = u_c(0)$ for $k \geq 0$. Since $u_c(0) = H^T z(0)$ the error model (3.18) reduces to

$$u_c(k) - \hat{u}(k) = H^T[c(F^0)^k - I]z(0) + \sum_{j=0}^{k-1} H^T c(F^0)^{k-j-1} H p(j),$$

In order to bound the error we make the following assumptions.

Assumption 3.3.6. Assume there are constants $c_2 \geq c_2'$ such that

$$\|H^T[c(F^0)^k - I]\|_1 \leq c_2 - c_2' \lambda_0^k$$

Conservative choices for c_2, c_2' that work under Assumption 3.3.3 are $c_2 = 1 + c_0$ and $c_2' = 0$.

Applying the assumption, the triangle inequality, and Lemma 3.3.1, we have

$$\begin{aligned}|u_c(k) - \hat{u}(k)| &\leq \rho_u \|H^T[c(F^0)^k - I]\|_1 + \rho_p \sum_{j=0}^{k-1} |H^T c(F^0)^{k-j-1} H|, \\ &\leq \rho_u(c_2 - c_2'\lambda_0^k) + \rho_p c_0 \sum_{j=0}^{k-1} \lambda_0^{k-j-1} \\ &= \rho_u(c_2 - c_2'\lambda_0^k) + \rho_p c_0 \frac{1-\lambda_0^k}{1-\lambda_0} =: \Gamma_2(\rho_u, \rho_p, k).\end{aligned}$$

which bounds the error. The bound Γ_2 converges exponentially fast at a rate λ_0. to

$$\rho_u c_2 + \frac{\rho_p c_0}{1-\lambda_0}.$$

3.3.4 Zero POC

We continue by computing the worst-case bound for the zero POC, introduced in Example 3.2, where the input is simply set to zero in outage mode, *i.e.*, $\hat{u}(k) = 0$, $k > 0$. The error model (3.18) then reduces to

$$u_c(k) - \hat{u}(k) = H^T c(F^0)^k z(0) + \sum_{j=0}^{k-1} H^T c(F^0)^{k-j-1} H p(j),$$

for $k > 0$. Applying the triangle inequality and Lemma 3.3.1, we have

$$\begin{aligned}|u_c(k) - \hat{u}(k)| &\leq \rho_u \|H^T c(F^0)^k]\|_1 + \rho_p \sum_{j=0}^{k-1} |H^T c(F^0)^{k-j-1} H| \\ &\leq \rho_u c_0 \lambda_0^k + \rho_p c_0 \sum_{j=0}^{k-1} \lambda_0^{k-j-1} \\ &= \rho_u c_0 \lambda_0^k + \rho_p c_0 \frac{1-\lambda_0^k}{1-\lambda_0} =: \Gamma_3(\rho_u, \rho_p, k).\end{aligned}$$

which bounds the error. The bound Γ_3 converges exponentially fast with rate λ_0 to

$$\frac{\rho_p c_0}{1-\lambda_0}.$$

3.3.5 Worst-Case Bound Example

We will now illustrate the worst-case bounds using a simple example. As described in Section 3.2.1 the transfer function from disturbance $d(k)$ to control $u_c(k)$ is given by the negated complementary sensitivity function. Let us assume that the feedback control system is operating well and designed so that it can be described by a first-order system (in continuous time)

$$-\omega_b/(s+\omega_b),$$

with bandwidth ω_b. A zero-order-hold sampled realization with sampling period h is

$$u_c(k+1) = \lambda_0 u_c(k) + p(k),$$

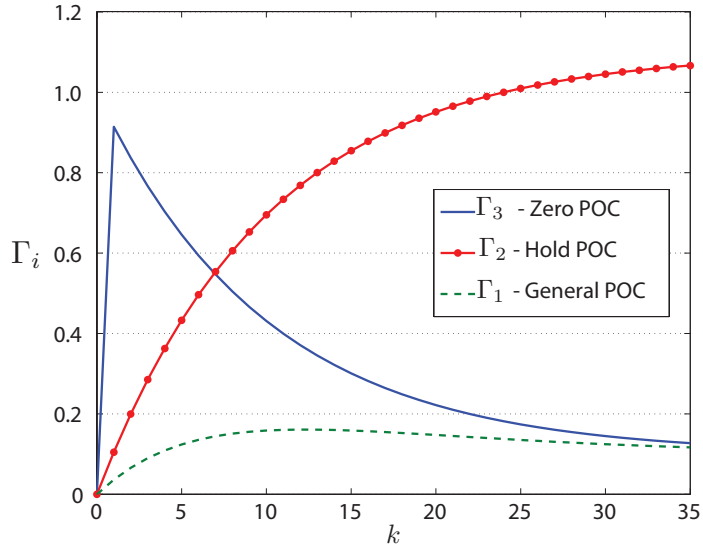
where $\lambda_0 = e^{-\omega_b h}$, and $p(k) = (e^{-\omega_b h} - 1)d(k)$. As POC, let us use

$$\hat{u}(k+1) = \lambda \hat{u}(k) + \hat{p}(k),$$

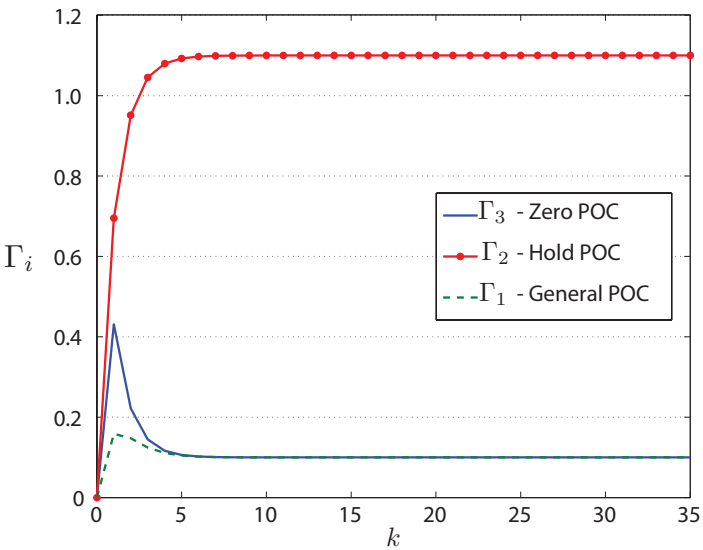
where $\lambda = e^{-1.3\omega_b h}$. This means that we have overestimated the actual bandwidth of the system with 30% in the POC. The other constants in Assumption 3.3.3 are $c_0 = c = 1$, and we assume that the signal $u_c(k)$ must be smaller than one, $\rho_u = 1$, and that the disturbance $d(k)$ are smaller than 0.1. This means that $\rho_p = 0.1(1-e^{-\omega_b h})$. Based on the discussion in Remark 3.3.5, we also choose the zero policy for $\hat{p}(k)$, *i.e.*, $\rho_{\hat{p}} = 0$. It is also easy to verify that the constants in Assumption 3.3.6 and 3.3.4 can be chosen as $c_2 = c_2' = 1$, $c_1 = c_1' = 1$, $\lambda_1 = \lambda_0$, and $\lambda_1' = \lambda$.

We plot the error bounds Γ_1, Γ_2, and Γ_3 as functions of outage time k in Figure 3.1. In Figure 3.1(a), the feedback control system is slow with $\omega_b h = 0.1$ and in Figure 3.1(b) the system is fast with $\omega_b h = 1.0$. As can be seen, for the slow system, upper bound for the general POC is the smallest for all times, whereas the hold POC bound is smaller than the zero POC for outages shorter than seven samples. If the bandwidth is increased with a factor 10, then the hold POC bound is by far largest for all times. The reason is that the system is capable of very fast changes, and to hold a constant input can quickly push the system in the wrong direction. Also in the fast case is the general POC gives the smallest bound for all times, even though the zero POC does quite well also.

The example seems to indicate that a general POC can do much better than the traditional hold and zero POCs both for slow and fast systems, even though the model had a parameter error of 30%.



(a) Low bandwidth, $\omega_b h = 0.1$.



(b) High bandwidth, $\omega_b h = 1.0$.

Figure 3.1: The worst-case bounds for systems with different bandwidth.

3.4 Stochastic Synthesis

Let us now consider a stochastic method on how to find optimal, in a minimum variance sense, (A, K, C) in (3.3), for MIMO systems affected by stochastic disturbances and measurement noise. To characterize the optimal (A, K, C) we need to know the models of $\mathcal{P}$, in (3.1), and $\mathcal{C}$, in (3.2) and know a stochastic model of the noise. Even if this is not always the case, it is still interesting to characterize the optimal solution since it can be used for comparison with other solutions.

Again using the notation in Section 3.1, let us first assume that $d(k)$ is colored noise given by

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_w w(k) \\ d(k) &= C_d x_d(k) + D_w w(k), \end{aligned} \tag{3.19}$$

and that the measurement noise $v(k)$ is Gaussian and white. Let

$$\mathbf{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} = 0, \quad \mathbf{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix}\begin{bmatrix} w(l) \\ v(l) \end{bmatrix}^T = R\delta_{kl},$$

be the expected value and covariance of the noises. When there is no outage, the entire closed-loop system evolves as

$$\begin{aligned} \begin{bmatrix} x_p(k+1) \\ x_c(k+1) \\ x_d(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} A_p & B_u C_c & B_d C_d \\ B_c C_p & A_c & 0 \\ 0 & 0 & A_d \end{bmatrix}}_{A_{cl}} \begin{bmatrix} x_p(k) \\ x_c(k) \\ x_d(k) \end{bmatrix} \\ &\quad + \underbrace{\begin{bmatrix} B_d D_w & 0 \\ 0 & B_c \\ B_w & 0 \end{bmatrix}}_{N} \begin{bmatrix} w(k) \\ v(k) \end{bmatrix} \\ u_c(k) &= \underbrace{\begin{bmatrix} 0 & C_c & 0 \end{bmatrix}}_{C_{cl}} \begin{bmatrix} x_p(k) \\ x_c(k) \\ x_d(k) \end{bmatrix}. \end{aligned} \tag{3.20}$$

The optimal estimator of the state in (3.20) using measurements $u_c(k)$ is the Kalman filter

$$\hat{x}(k+1|k) = A_{cl}\hat{x}(k|k-1) + K_{cl}\big[u_c(k) - C_{cl}\hat{x}(k|k-1)\big], \tag{3.21}$$

where

$$\begin{aligned} K_{cl} &= \big(A_{cl}PC_{cl}^T\big)\big(C_{cl}PC_{cl}^T\big)^{-1} \\ P &= A_{cl}PA_{cl}^T + NRN^T - \big(A_{cl}PC_{cl}^T\big)\big(C_{cl}PC_{cl}^T\big)^{-1}\big(A_{cl}PC_{cl}^T\big)^T, \end{aligned} \tag{3.22}$$

see for example (Anderson and Moore, 2005). The optimal one-step-ahead prediction of $u_c(k)$ is $\hat{u}(k|k-1) = C_{cl}\hat{x}(k|k-1)$. Note that the Kalman filter (3.21) has the structure of the POC (3.3), and that optimal predictions of $u_c(k)$ based on measurements up until $k' \leqslant k$ are generated by

$$\begin{aligned} \hat{x}(k+1|k') &= A_{cl}\hat{x}(k|k') \\ \hat{u}(k|k') &= C_{cl}\hat{x}(k|k'), \end{aligned} \tag{3.23}$$

where the first prediction $\hat{x}(k'+1|k')$ is given by (3.21). Also the optimal predictor (3.23) has the form of a POC in prediction mode (3.4).

Proposition 3.4.1. *The POC synthesized using the optimal stochastic method can be realized on the form (3.3) as*

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

where $\hat{x}(k)$ is equal to the state $\hat{x}(k+1|k)$ of (3.21) and

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

Remark 3.4.2. Note that we consider the problem over an infinite time horizon. This is a good assumption if the communication outages are infrequent. If the time horizon is finite, the optimal filter gain K_{cl} should be time varying, see (Anderson and Moore, 2005).

Next, we compute the prediction error of a POC synthesized using the optimal stochastic method. This error serves as a lower bound on the prediction error of other POCs. We can note that the state dimension of such a POC is large, since it contains the states for the process, controller and noise model. How to approximate it is discussed in Chapter 4.

3.5 Optimal Stochastic Prediction Error Bounds

It is easy to characterize the statistics of the prediction error $\Delta u(k)$ of the POC synthesized using the optimal stochastic method. The Kalman filter gives unbiased estimates and thus $\mathbf{E}\Delta u(k) = 0$ for all $k > k'$. To compute the variance $\mathbf{E}|\Delta u(k)|^2$, we need the covariance of the state estimation error. Assuming that the Kalman filter has been in operation a long time before the outage at k', the covariance of $\Delta x(k)$ is given by the solution to the Riccati equation in (3.22),

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

where $\Delta x(k) = x(k) - \hat{x}(k|k-1)$ and $x(k)$ is the state vector in (3.20). The variance of the one-step-ahead prediction error is $\mathbf{E}|\epsilon(k)|^2 = C_{cl}PC_{cl}^T$. When an outage

occurs, the covariance of the state estimation and prediction error evolve for $k > k'$ as

$$P(k+1) = AP(k)A^T + NRN^T, \quad P(k') = P,$$
$$\mathbf{E}|\Delta u(k)|^2 = C_{cl}P(k)C_{cl}^T.$$

The POC derived using the optimal stochastic method minimizes the variance of the noise $\Delta u(k) = u_c(k) - \hat{u}(k)$. How much effect this prediction error noise has on the plant depends on its dynamics. If $\mathcal{P}$ is an unstable process, even a small error $\Delta u(k)$ can harm the process since it is in open loop during outage.

3.6 Summary

In this chapter we have presented two methods to design the POC, one for SISO systems affected by deterministic disturbances and one for MIMO systems affected by stochastic disturbances. Further we have shown that the commonly used POCs, to hold the last value or use a zero output, are special cases.

For the deterministic method upper bounds on the prediction error were presented together with an example. For the stochastic method optimal prediction error bounds were given in form of a the covariance for the prediction error.

Chapter 4

Complexity Analysis and Reduction

This chapter gives methods to reduce the complexity of the POC using model order reduction. The optimal POC, both in the stochastic and in the deterministic setting in Chapter 3, is given by a filter of order equal to the sum of the process order, the controller order and the disturbance model order. In practice, it is important to know if there exists a POC of low order with similar performance. The optimal POC gives the achievable performance under the given structure. In this chapter, model order reduction using the Hankel norm and switched balanced truncation are shown to be suitable mathematical tools to find out the answer to this question on the existence of a suitable low order POC.

The chapter starts by introducing the needed notation. Then the method for reduction in the Hankel norm is presented, followed by the method based on switched balanced truncation. Finally some commonly used crude approximation methods are given for comparison.

4.1 Reduced-Order POCs

We start by introducing the needed notation. Let us represent the POC by a linear operator $\hat{u} = G\epsilon$ on ℓ_2, where G is realized by

$$G \begin{cases} \hat{x}(k+1) = A\hat{x}(k) + K\epsilon(k), \\ \hat{u}(k|k-1) = C\hat{x}(k),\ \hat{x}(k) \in \mathbb{R}^n, \end{cases} \tag{4.1}$$

where $\epsilon(k) = u_c(k) - \hat{u}(k|k-1)$. As has been discussed in Section 3.4, the dimension n of the state vector $\hat{x}(k)$ can be very large for the optimal POC. This is for example the case if it is synthesized using the stochastic method since K then is the Kalman gain of the system and the state $\hat{x}$ is composed by both the process, controller and noise model states. It is of interest to investigate how the order of the POC can be reduced, and to get an understanding of the performance–complexity trade-off. Let us denote a reduced-order POC by a linear operator $\hat{u}_r = G_r\epsilon_r$ on ℓ_2, with order

$r < n$, realized by

$$G_r \begin{cases} \hat{x}_r(k+1) = A_r\hat{x}_r(k) + K_r\epsilon_r(k), \\ \hat{u}_r(k|k-1) = C_r\hat{x}_r(k),\ \hat{x}_r(k) \in \mathbb{R}^r, \end{cases} \tag{4.2}$$

where $\epsilon_r(k) = u_c(k) - \hat{u}_r(k|k-1)$. Two different methods for generating G_r from a given G are suggested next.

4.2 Hankel Approximation

We will now establish a method to reduce the complexity of G by Hankel norm approximation. The use of Hankel norm is motivated and explained, further reduction error bounds are given.

Assuming, without loss of generality, that the outage occurs at $k' = 0$, we would like the reduced-order POC to produce an outage prediction $\hat{u}_r(k|k')$ that is close to $\hat{u}(k|k')$. Since the predictions only will be applied to the system after the outage at $k' = 0$ we are naturally only interested in making the difference small for $k > 0$. Introducing the time-projection operator P_+ as

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

we formalize this requirement using the operator notation in (4.1) and (4.2) as making

$$\|P_+\hat{u} - P_+\hat{u}_r\|_2 = \|P_+G\epsilon - P_+G_r\epsilon_r\|_2 \tag{4.3}$$

small.

Recall that when the POC is in feedback, *i.e.*, there is no outage, the prediction error is given by $\epsilon = u_c - \hat{u} = u_c - G\epsilon$, which rearranged becomes $\epsilon = (I+G)^{-1}u_c$. During the outage the prediction error can not be computed and instead $\epsilon = 0$ is used, as described in Section 2.2. To get a closed-form expression for ϵ we introduce another time-projection operator P_- as

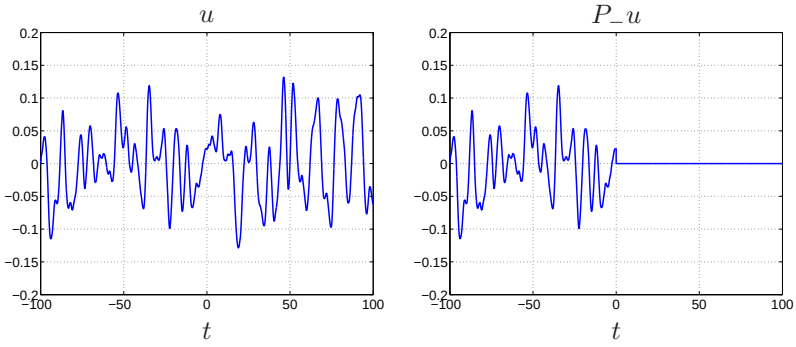
$$\begin{aligned} P_-u &= P_-(\ldots, u(2), u(1), u(0), u(-1), u(-2), \ldots) \\ &= (\ldots, 0, 0, u(0), u(-1), u(-2), \ldots), \end{aligned}$$

which we can use to write $\epsilon = P_-(I+G)^{-1}u_c$. Since the same argument holds for the reduced-order POC we can now write (4.3) as

$$\|P_+G\epsilon - P_+G_r\epsilon_r\|_2 = \|P_+GP_-(I+G)^{-1}u_c - P_+G_rP_-(I+G_r)^{-1}u_c\|_2 \tag{4.4}$$

We note that the operator P_+GP_- is exactly the Hankel operator Γ_G of G, see for example (Zhou et al., 1996), the past-input to future-output map of G. The input-output map of Γ_G is given by

$$\hat{u}(k|0) = \sum_{i=-\infty}^{0} CA^{k-i-1}K\epsilon(i) = (\Gamma_G\epsilon)(k), \quad k > 0,$$



(a) The input signal u.

(b) The projected input signal $P_- u$.

(c) The output y of the system G when driven by the control signal $P_- u$, *i.e.*, $y = GP_- u$.

(d) The projected output $P_+ y$ of the system G when driven by the control signal $P_- u$, *i.e.*, $P_+ y = P_+ GP_- u = \Gamma_G u$.

Figure 4.1: Illustration of the input-output map of the Hankel operator Γ_G of the system G.

which is illustrated in Figure 4.1

Since we want to make (4.4), a thus (4.3) with $\epsilon(k) = \epsilon_r(k) = 0$ for $k > 0$, small for any input sequence $u_c(k)$ we want to make the criterion

$$\sup_{u_c \neq 0} \frac{\|\Gamma_G(I+G)^{-1}u_c - \Gamma_{G_r}(I+G_r)^{-1}u_c\|_2}{\|u_c\|_2} = \|\Gamma_G(I+G)^{-1} - \Gamma_{G_r}(I+G_r)^{-1}\ \| \tag{4.5}$$

small. Here $\|\cdot\|$ is the induced ℓ_2-norm.

Remark 4.2.1. Note that we here try to make $\|P_+(\hat{u} - \hat{u}_r)\|_2$ small, *i.e.*, we try to make the total energy in the reduction error for $k > 0$ small. Since the outages are of finite length, we are arguably mostly interested in keeping the error small during the first part of the outage. An interesting idea to achieve this is to add an exponential decay weight, $alpha^k$, $0 < \alpha \leq 1$ in the minimization criterion and try to minimize $\|P_+(\hat{u} - \hat{u}_r)\alpha^k\|_2$.

Model order reduction is a well-studied topic, and there are many methods available, see for example the book (Obinata and Anderson, 2001). The problem of making (4.5) small does not fit clearly to any of these methods, however, since the operators Γ_G and Γ_{G_r} are weighted differently by $(I+G)^{-1}$ and $(I+G_r)^{-1}$, respectively. To be able to solve the problem, we note that the following bound holds:

$$\begin{aligned}
&\|\Gamma_G(I+G)^{-1} - \Gamma_{G_r}(I+G_r)^{-1}\| \\
&\quad = \|\Gamma_G(I+G_r)^{-1} - \Gamma_{G_r}(I+G_r)^{-1} + \Gamma_G(I+G)^{-1} - \Gamma_G(I+G_r)^{-1}\| \\
&\quad \le \|(\Gamma_G - \Gamma_{G_r})(I+G_r)^{-1}\| + \|\Gamma_G\big((I+G)^{-1} - (I+G_r)^{-1}\big)\|. \quad (4.6)
\end{aligned}$$

We will proceed by minimizing the first term of this upper bound. It turns out that it is then possible to use that solution to bound the second term, and thus to bound the error criterion (4.5).

Note that the rank of the Hankel operator is equal to the McMillan degree of the corresponding system, if (A, K, C) is a minimal realization, *i.e.*, rank $\Gamma_G = n$. To make the first term in the upper bound (4.6) small, we therefore propose to solve the problem

$$\min_{\text{rank}\Gamma_{G_r} \le r} \sup_{\|\epsilon_r\|_2 \le 1} \|(\Gamma_G - \Gamma_{G_r})\epsilon_r\|_2 =: \gamma_1(r), \quad (4.7)$$

where $\epsilon_r = (I+G_r)^{-1}u_c$. The reason is that the approximation problem (4.7) can be solved using the AAK-lemma, see (Adamjan et al., 1971; Glover, 1984). In particular, it is well-known that

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

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

Assume now we choose an optimal Hankel approximation G_r^* of G as the reduced POC. What can we then say about the size of the second term in the bound (4.6)? We have that

$$(I+G)\Big((I+G)^{-1} - (I+G_r)^{-1}\Big)(I+G_r) = (I+G_r) - (I+G) = G_r - G$$

giving

$$(I+G)^{-1} - (I+G_r)^{-1} = (I+G)^{-1}(G_r - G)(I+G_r)^{-1},$$

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

$$\|G_r^* - G\| \le \sum_{i=r+1}^{n} \sigma_i(G). \quad (4.8)$$

An upper estimate of the second term is therefore

$$\left\|\Gamma_G\Big((I+G)^{-1}-(I+G_r^*)^{-1}\Big)u_c\right\|_2 = \|\Gamma_G(I+G)^{-1}(G_r^*-G)\epsilon_r\|_2 \le \gamma_2(r)\|\epsilon_r\|_2 \tag{4.9}$$

where

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

Here we have used that the induced norm of Γ_G is equal to $\sigma_1(G)$. This bound is expected to be quite conservative, since we have used the submultiplicative property of the induced norm and the upper bound (4.8) which is derived using the triangle inequality. The reason this bound is used is that the author is not aware of any method to compute a better bound for the induced norm $\|\Gamma_G(I+G)^{-1}(G_r^*-G)\|$.

We summarize the above results in the following proposition.

Proposition 4.2.2. *Suppose the system (4.2) is chosen as an optimal Hankel approximation G_r^* of the stable system G in (4.1). Then it holds for any input $u_c \in \ell_2$ that*

$$\|P_+(\hat{u}-\hat{u}_r)\|_2 = \|\Gamma_G\epsilon - \Gamma_{G_r^*}\epsilon_r\|_2 \le \gamma(r)\|\epsilon_r\|_2,$$

where

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

Proposition 4.2.2 shows that if $\sigma_i(G)$, $i = r+1,\dots,n$, are small, then G_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 which is feeded into G_r^*. If $\|\epsilon_r\|_2$ is small, it means that the environment is not very noisy, and the prediction works well. If then an outage occurs, we can be certain that the outage predictions $\hat{u}_r$ do not deviate from the full-order prediction $\hat{u}$ more than $\gamma(r)\|\epsilon_r\|_2$, under the same circumstances.

Remark 4.2.3. One restriction in Proposition 4.2.2 is that G must be stable. This is for example not the case if the POC has been constructed using the optimal stochastic method with unstable modes in the disturbance model (3.19). The stability condition can be handled by making a stable/anti-stable decomposition of G, *i.e.*, $G = G_s + G_u$, and then approximate the stable part G_s as above. The unstable term G_u can then be added to the approximation $G_{s,r}^*$.

Remark 4.2.4. When G has been synthesized using the optimal stochastic method as described in Section 3.4, it intuitively makes sense to make the reduction criteria $\|\Gamma_G - \Gamma_{G_r}\|$ small. The reason is that the correction $\epsilon = (I+G)^{-1}u_c$ that is applied to Γ_G then is an innovation sequence, and is thus white noise. It therefore contains an equal amount of all frequencies and to make an unweighed criterion like $\|\Gamma_G - \Gamma_{G_r}\|$ small is natural.

4.3 Balanced Truncation of Switched Systems

A perhaps more direct approach to obtain a reduced-order POC is to use balanced truncation of linear time-varying systems, see for example (Lall and Beck, 2003; Sandberg and Rantzer, 2004). We proceed to explain reduction using switched balanced truncation and give upper bounds on the reduction error.

A POC G can be modelled by the linear time-varying system

$$\begin{aligned} \hat{x}(k+1) &= A(k)\hat{x}(k) + K(k)u_c(k) \\ \hat{u}(k) &= C\hat{x}(k), \end{aligned} \tag{4.10}$$

where

$$A(k) = \begin{cases} A - KC, & \text{no outage} \\ A, & \text{outage} \end{cases}$$

$$K(k) = \begin{cases} K, & \text{no outage} \\ 0, & \text{outage} \end{cases}$$

The interpretations of $\hat{x}(k)$ and $\hat{u}(k)$ depend on if the system is in outage or not at time k.

The idea behind switched balanced truncation is to use, assuming they exist, a generalized controllability Gramian $P \in \mathbb{R}^{n\times n}$ and a generalized observability Gramian $Q \in \mathbb{R}^{n\times n}$ for (4.10) satisfying the Linear Matrix Inequalities (LMIs)

$$\begin{aligned} A(k)PA(k)^T - P + K(k)K(k)^T \leq 0, \quad \forall k \\ A(k)^T QA(k) - Q + C^T C \leq 0, \quad \forall k \end{aligned} \tag{4.11}$$

to define generalized Hankel singular values $\sigma_i = \sqrt{\lambda_i(PQ)}$. The Gramians are then used to compute a balancing coordinate transformation such that both Gramians become diagonal and equal to $\Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_n\}$, see (Moore, 1981). By truncating the states of the balanced realization of (4.10) that correspond to σ_i, $i = r+1, \ldots, n$, we obtain the reduced-order POC G_r as

$$\begin{aligned} \hat{x}_r(k+1) &= A_r(k)\hat{x}_r(k) + K_r(k)u_c(k) \\ \hat{u}_r(k) &= C_r\hat{x}_r(k). \end{aligned} \tag{4.12}$$

Effectively, (4.11) reduces to four LMIs, and can be solved using standard software.

For details of the method see (Lall and Beck, 2003; Sandberg and Rantzer, 2004). As first shown in (Lall and Beck, 2003) there is a bound on the error $\|\hat{u} - \hat{u}_r\|_2$, as follows.

Proposition 4.3.1. *Suppose (4.11) are feasible and the system (4.12) is chosen as a truncated balanced realization of (4.10). Then it holds for any input $u_c \in \ell_2$ that*

$$\|\hat{u} - \hat{u}_r\|_2 \leq \left(2\sum_{i=r+1}^{n} \sigma_i\right) \|u_c\|_2.$$

Remark 4.3.2. Note that the error bound in Proposition 4.3.1 also bound the difference between $\hat{u}$ and $\hat{u}_r$ when there is no outage. The bound in Proposition 4.2.2 bounds the error in outage only. Arguably we are more interested in the magnitude of the error when outage occurs since it is only then $\hat{u}$ and $\hat{u}_r$ are actually applied to the plant. On the other hand, the bound in Proposition 4.3.1 also holds for arbitrary numbers of consecutive outages.

Remark 4.3.3. Balanced truncation only works when the LMIs (4.11) are feasible, whereas the method in Section 4.2 always works. This feasibility problem is the same as the problem of finding a common Lyapunov function for switched linear systems, see for example (Liberzon, 2003). To characterize POCs that ensure feasibility is an interesting topic for future research. A simple necessary condition is again that G is stable. Unstable G can be dealt with like mentioned in Remark 4.2.3.

4.4 Hold and Zero Approximation

As mentioned previously in Chapter 3 two common ways to implement outage compensation is a to use a hold or a zero POC. Seen in the light of model order reduction these two POCs are in fact crude approximations G_r of G, where G could be any POC. We summarize these observations in the following two remarks.

Remark 4.4.1. The hold POC is the reduced POC G_r obtained by the reduction method which takes any POC G and approximates it with G_r as

$$G_r \begin{cases} \hat{x}_r(k+1) = 1 \cdot \hat{x}_r(k) + 1 \cdot \epsilon_r(k), \\ \hat{u}_r(k|k-1) = 1 \cdot \hat{x}_r(k), \end{cases} \tag{4.13}$$

with $\epsilon_r(k) = u_c(k) - \hat{u}_r(k|k-1)$, *i.e.*, the POC realized by $A_r = K_r = C_r = 1$ giving $\hat{u}_r(k|k-1) = u_c(k')$ where $u_c(k')$ is the last know control signal.

Remark 4.4.2. The zero POC is the reduced POC G_r obtained by the reduction method which takes any POC G and approximates it with G_r as

$$G_r \left\{ \begin{array}{l} \hat{x}_r(k+1) = 0 \cdot \hat{x}_r(k) + 0 \cdot \epsilon_r(k), \\ \hat{u}_r(k|k-1) = 0 \cdot \hat{x}_r(k), \end{array} \right\} \tag{4.14}$$

with $\epsilon_r(k) = u_c(k) - \hat{u}_r(k|k-1)$, *i.e.*, the POC realized by $A_r = K_r = C_r = 0$ giving $\hat{u}_r(k|k-1) = 0$.

4.5 Summary

Two methods to reduce the complexity of the POC were given, one based on Hankel norm approximation and one based on switched balanced truncation. For both methods upper bounds on the resulting reduction error were given.

Further it was shown that the two commonly used POC methods to hold the last value or use a zero output can be seen as crude approximation of a more complex POC.

Chapter 5

Simulation Evaluation

To illustrate the behavior of the POC designs presented in the thesis, under realistic scenarios, this chapter presents a simulation study of a level control system. POCs designed using both the deterministic and the stochastic method are given for various levels of complexity.

The chapter begins by describing the simulated system and the simulation scenario. Then a POC is derived using the nominal deterministic method and simulations are made comparing it with lower-order approximations. After this we simulate the POC given by the optimal stochastic method. Finally, the reduction methods presented in Chapter 4 are applied to the POC given by the optimal stochastic method and comparative simulations are performed.

5.1 System and Scenario

We start by introducing the system and scenario for the simulation example. As the simulated process $\mathcal{P}$ we consider a tank process consisting of five identical tanks connected in series as depicted in Figure 5.1.

This system is motivated by a floatation process in an ore concentrator at Boliden in Sweden, which is being investigated for wireless control within the SOCRADES project (SOCRADES, Integrated Project, EU Sixth Framework Programme). The control objective is to keep the level x_5 in the lowest tank around an equilibrium point despite load disturbances d entering the system. The manipulated variable is the flow u from the pump.

The individual tanks are modelled using mass balance and Bernoulli's law. Assuming that the tanks have cross sectional area $A = 1\,\mathrm{m}^2$, outlet hole area $a = 0.2\,\mathrm{m}^2$ and that the gravitational acceleration is $g = 10\,\mathrm{m/s}^2$, a linearized process model around the equilibrium $x_i^0 = 5\,\mathrm{m}$ and $u^0 = 2\,\mathrm{m}^3/\mathrm{s}$ is given by

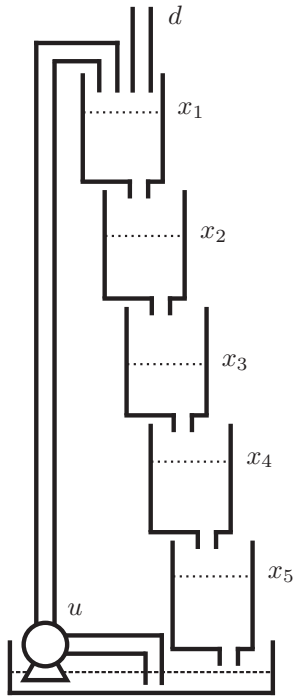


Figure 5.1: The controlled tank process $\mathcal{P}$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \frac{1}{\tau} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Big(\; u + d \; \Big) \tag{5.1}$$

where $\tau = \frac{A}{a}\sqrt{\frac{2x_i^0}{g}} = 5\,\mathrm{s}$. The rise time of the water tank process is about 25 s.

The process is controlled by a controller that has been derived using loop-shaping and is given as

$$C(s) = \frac{s + \omega_I}{s} \Big(\frac{\tau_d s + 1}{\beta \tau_d s + 1} \Big)^5 \frac{10\omega_I}{s + 10\omega_I} \tag{5.2}$$

with the parametrization $\omega_I = 0.2$, $\beta = 0.30$ and $\tau_d = 8.54$. This control design gives a cross-over frequency $\omega_c = 0.22\,\mathrm{rad/s}$ with a phase-margin $\varphi_m = 60^o$, which makes the closed-loop system about five times faster than the open-loop system. The process and controller are sampled with period $T_s = 1\,\mathrm{s}$

The simulation scenario for $0 \leqslant t \leqslant 140$ used when simulating the system is the following:

t = 0 System starts at rest and the disturbance d starts to act on the system.

t = 40 A communication outage occur between controller and actuator so the POC is activated.

t = 80 Communication is restored so the controller drives the system back into rest.

Since the deterministic and stochastic methods are designed for different types of disturbances, the disturbance d will be chosen accordingly. That is, when simulating the POC derived using the deterministic method a piece-wise constant deterministic disturbance will be used and when simulating the POC derived using the stochastic method a stochastic random-walk disturbance will be used.

As a comparison to the POC simulations under the above scenario we will also show the closed-loop response of the system affected by the same disturbance d, *i.e.*, show how the system would have reacted to the disturbance if no outage had occurred. This will be referred to as loss-free behavior and the signal trajectories of this system are referred to as the loss-free trajectories.

5.2 Simulation Evaluation of Deterministic Synthesis

Let us study the POC synthesized using the deterministic method derived in Section 3.2 and simulate it according to the given example and scenario with the disturbance d realized by

$$d(t) = \begin{cases} 0, & t < 20 \\ 1, & 20 \leq t < 60 \\ 0.5, & 60 \leq t. \end{cases}$$

We start by simulating the POC and then compare these results with simulations of the zero and the hold POCs.

5.2.1 Nominal Deterministically Derived POC

The system response with the nominal deterministically derived POC is shown in Figure 5.2. Recall that this POC is obtained when the internal model of the POC is chosen as the true system, *i.e.*, following the notation in Section 3.2 $G(q) = G^0(q)$. The predicted trajectory follows the loss-free trajectory during the outage up until $t = 60$ when the disturbance changes. This is the best we can do since we can only follow the loss-free trajectory that was observed prior to the outage. As the disturbance changes so does the loss-free trajectory. However since communication is lost the POC can not detect this change of the loss-free trajectory since it can not

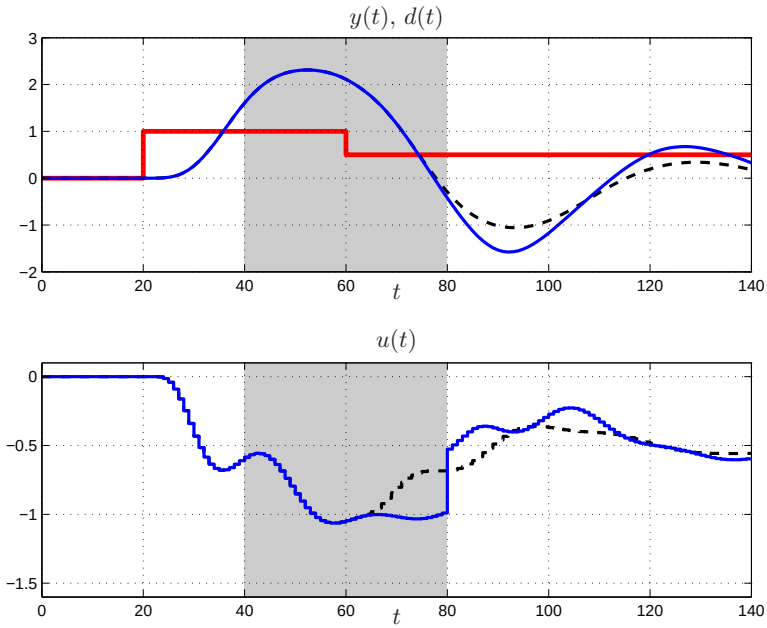


Figure 5.2: Comparison of nominal deterministically synthesized POC behavior (solid dark) with loss-free behavior (dashed) under disturbance d (solid light) and outage (greyed)

measure it. When communication is restored the controller can use a small input correction term to recover the loss-free system behavior. We note that the settling time for the loss-free system and the system using this POC are almost identical.

5.2.2 Hold POC

The hold POC system response can be seen in Figure 5.3(a). The hold POC initially, for the first few samples after the outage, manages to keep the system quite close to the loss-free trajectory. However, after these initial samples the system trajectory starts growing away from the setpoint. When communication is restored the controller needs to use a large control signal to recover the loss-free system behavior.

5.2.3 Zero POC

The response for the system with the zero POC can be seen in Figure 5.3(b). As seen, when the outage occurs, the system output starts to grow rapidly, taking the system far away from the desired setpoint. As a consequence of the large pertur-

bation caused during the outage the controller has to use a large control signal to recover the loss-free system behavior once communication is restored.

The large magnitude of the control signal for the hold POC and the zero POC, once communication is restored, is dependent on two factors. The first and most apparent reason is the fact that the system drifts far away from the setpoint and therefore a large control effort is needed. The second and more subtle reason is that there is a large difference between the control signal that the controller is computing and the one that is actuated. This difference will cause integral windup effects in the controller which also appear in the transient after communication is restored.

The difference in computed and applied control signal due to lost packets can, from the controllers perspective, be seen as a form of "virtual" saturation. As is well know saturation of the control signal causes windup effects, unless it is compensated for. Unfortunately, to apply anti-windup techniques the controller needs to know the shape of the "virtual" saturation, or more precisely which command was applied. This means that the controller needs to be acknowledged if the sent packet has been applied or not. If it was not applied, it needs to know which replacement command that was issued.

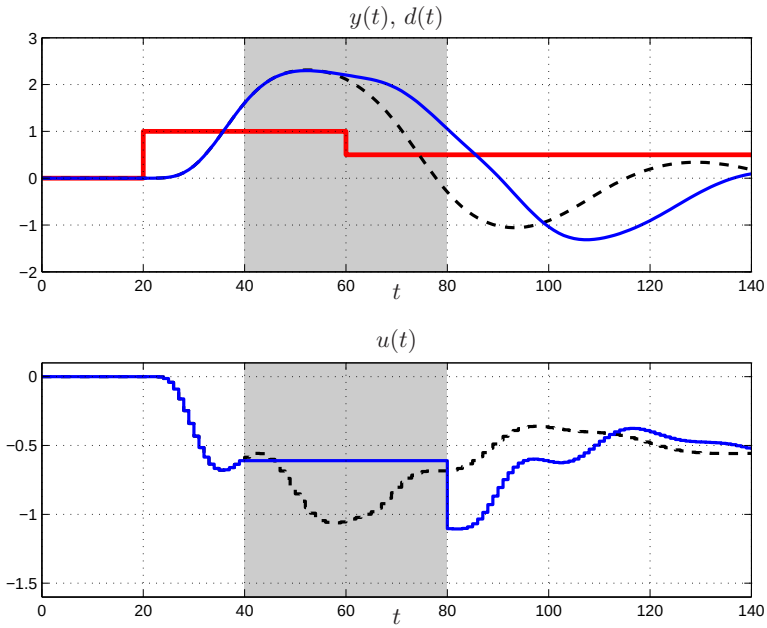
5.3 Simulation Evaluation of Stochastic Synthesis

We now move on to the POC given by the optimal stochastic method derived in Section 3.4. As with the deterministically derived POC we simulate it on the given example and under the given scenario. However, here a sampled realization of the disturbance d is generated by the system

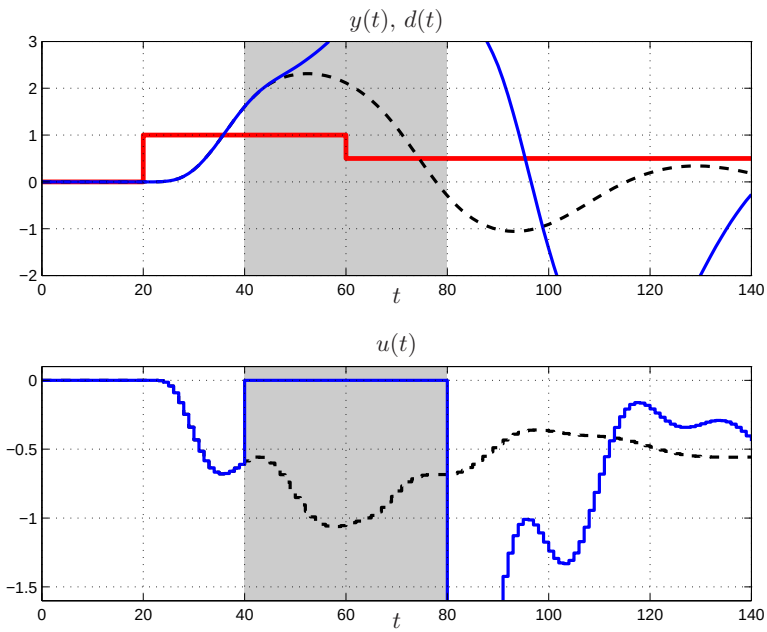
$$\begin{aligned} x_d(k+1) &= x_d(k) + w(k) \\ d(k) &= x_d(k) \\ \mathbf{E}w(k)^2 &= 0.01 \end{aligned} \tag{5.3}$$

Combining the sampled models for the process (5.1), controller (5.2) and disturbance (5.3) following Section 3.4, one get a closed-loop system with McMillan degree $n = n_p + n_c + n_d = 5 + 7 + 1 = 13$. Deriving the POC G for this system using the optimal stochastic method, with G given by (3.21), G will have the same degree. In the following we use $R = \mathrm{diag}([10^{-2}, 10^{-4}])$.

The resulting simulated behavior of the closed-loop system with this POC is shown in Figure 5.4. At time $t = 0$ the POC is initialized to have the same state as the true system and the prediction is perfect. However at the same instant the disturbance starts acting and the system states start to diverge. As a result, so does the estimation error. Effectively what now happens is that the Kalman filter in the POC starts to estimate the variance of the random-walk disturbance d via the internal model. At $t = 40$, communication between the controller and actuator is lost, hence so is the feedback into the predictor, and the POC starts to evolve in open loop predicting the control signal. If the state estimate in the POC has converged the prediction will be perfect, as long as the disturbance does not change



(a) hold POC



(b) zero POC

Figure 5.3: Comparison of hold and zero POC behavior (solid dark) with loss-free behavior (dashed) under disturbance d (solid light) and outage (greyed)

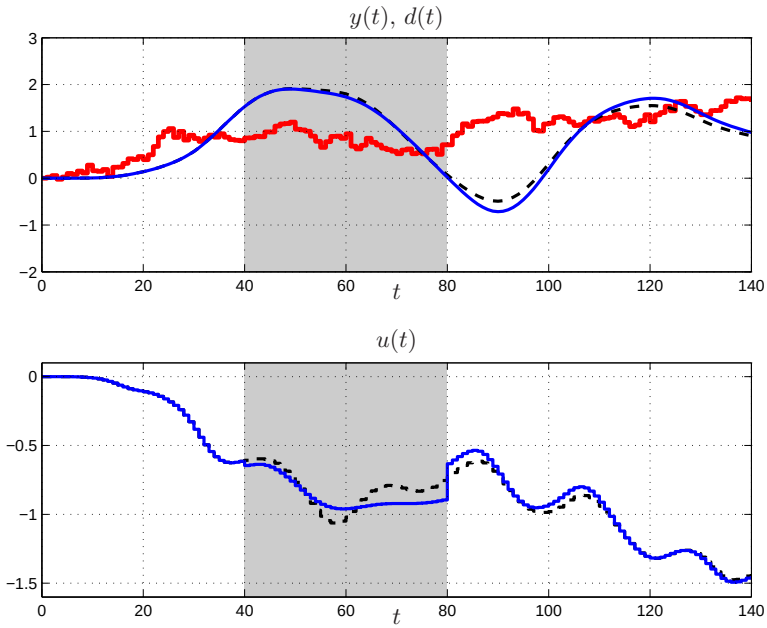


Figure 5.4: Comparison of optimal stochastically synthesized POC behavior (solid) with loss-free behavior (dashed) under disturbance d (dotted) and outage (greyed)

under the outage. However, if the estimate has not fully converged and there is noise, as is the case in this example, the prediction will start to diverge. Still, one can observe that the prediction error is small for the first 15 s resulting in a very small deviation in the output compared to the loss-free case.

5.4 Simulation Evaluation of Reduced Order POCs

Let us now use the POC G, derived using the optimal stochastic method, presented in Section 5.3 to illustrate the reduction methods presented in Chapter 4.

5.4.1 Hankel Approximation

We will now go through the method of approximation in the Hankel norm for the POC G in the given example. We discuss how the approximation order could be chosen and how unstable modes should be handled. Finally simulation examples are given for approximations of various orders.

First we observe that the disturbance model (5.3) in G contains an integration and that G hence is not asymptotically stable. To handle this a stable/anti-stable decomposition of G is made as $G = G_s + G_u$ and reduction is made on the stable

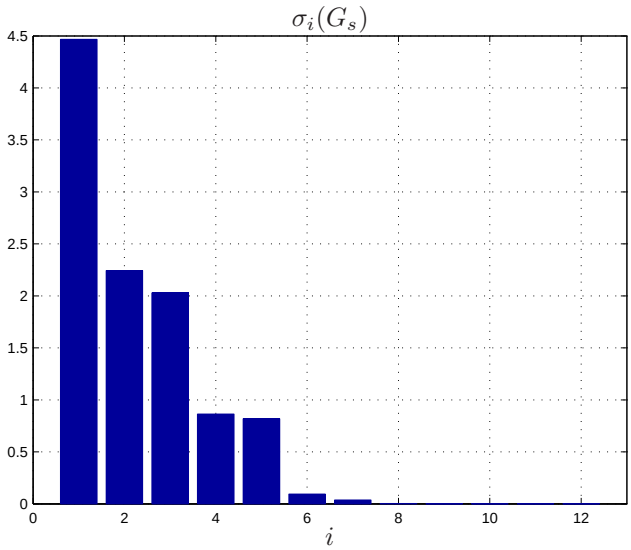


Figure 5.5: Hankel singular values of the stable modes in the optimal stochastically synthesized POC

part G_s only, see Remark 4.2.3. To determine a proper reduction order r the singular values of G_s, shown in Figure 5.5, are studied. As seen there are significant drops between $\sigma_1(G_s)$ and $\sigma_2(G_s)$, between $\sigma_3(G_s)$ and $\sigma_4(G_s)$ and between $\sigma_5(G_s)$ and $\sigma_6(G_s)$, indicating that a good choice of the reduction order r is to choose r in the set $\{1, 3, 5\}$. Performing optimal Hankel norm approximation on G_s of order r one get $G^*_{s,r}$ and the reduced POC as $G^*_r = G^*_{s,r} + G_u$ of order $r + 1$ since G_u only contains the integrator state from the disturbance model.

A quantitative bound $\gamma(r)$ on the reduction is given in Proposition 4.2.2. For the given example the bounds for the suggested choices of r are presented in Table 5.6 together with the true norm of $\|P_+(\hat{u}-\hat{u}_r)\|_2$ and the contribution from the partial bound $\gamma_1(r)$ in (4.7). As seen, in this example, the bound imposed by $\gamma_1(r)$ is close to the true value of $\|P_+(\hat{u} - \hat{u}_r)\|_2$ whereas the upper bound given by $\gamma(r)$ is significantly larger. This is due to the conservative derivations of the component $\gamma_2(r)$ from (4.9).

Evaluating the reduced POCs with order $r = 3$ and $r = 1$ on the same simulation scenario as for stochastic method case, one gets the results in Figure 5.7(b) and Figure 5.7(a).

At $t = 0$ the system starts at rest in the origin and the reduced POCs are initialized accordingly to $x_r(0) = 0$. As before the disturbance d starts to act on the system immediately causing it to diverge and the POCs start estimating the states. The state associated with the disturbance model will now, apart from the distur-

Table 5.6: Comparison of simulated reduction error norm compared to computed reduction error norm bounds for different reduction orders r

r	$\|\epsilon_r\|_2$	$\gamma_1(r)\|\epsilon_r\|_2$	$\gamma(r)\|\epsilon_r\|_2$	$\|P_+(\hat{u}-\hat{u}_r)\|_2$
5	0.32	0.03	0.25	0.02
3	0.29	0.25	2.99	0.26
1	0.35	0.77	11.87	0.39

bance, accommodate the errors due to the model reduction. When communication is lost at $t = 40$ the POCs will as before evolve in open loop predicting the control signal.

Studying the reduced POC of reduction order $r = 3$ in Figure 5.7(a) and comparing it with the optimal stochastically derived POC in Figure 5.4 we see that its prediction error is larger, although the difference is not significant. In fact, the output tracking performance for the reduced POC with $r = 3$ is almost identical to the optimal stochastically derived POC in Figure 5.4.

If we instead study the reduced POC of reduction order $r = 1$ in Figure 5.7(b) we see that the prediction error as expected is larger than for the reduced POC of reduction order $r = 3$, this is due both to the model approximation error and the fact that the estimator has not fully converged when the outage occur.

5.4.2 Hold Approximation

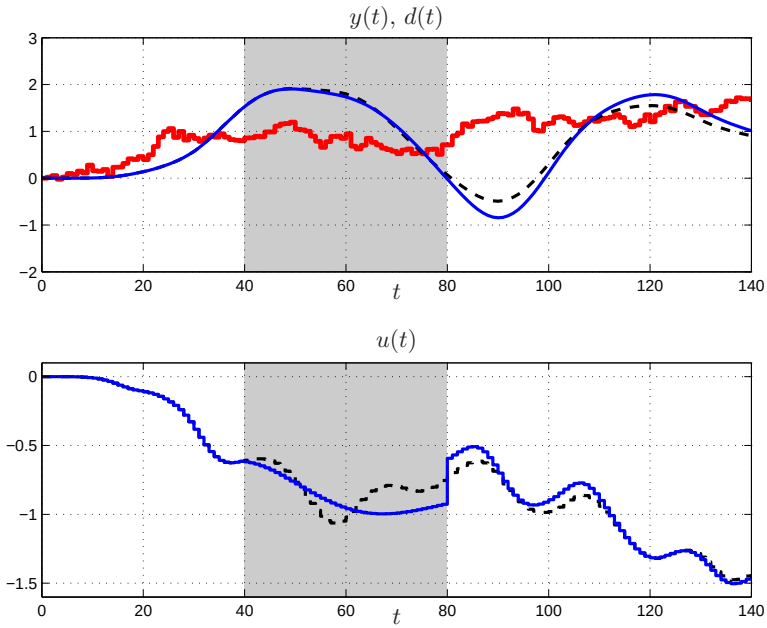
As mentioned previously in Chapter 4 the hold POC can be viewed as a crude approximation G_r of the POC G with G_r realized as in (4.2) with $A_r = K_r = C_r = 1$, see Section 4.4. The resulting simulation over the studied scenario from using this POC is shown in Figure 5.8(a).

The response is similar to the hold POC in the deterministic method scenario. We see that for the first few samples after the outage the trajectory is kept close to the loss-free one. But as before, it will after a few samples start to deviate, even though not as fast as with the zero approximation.

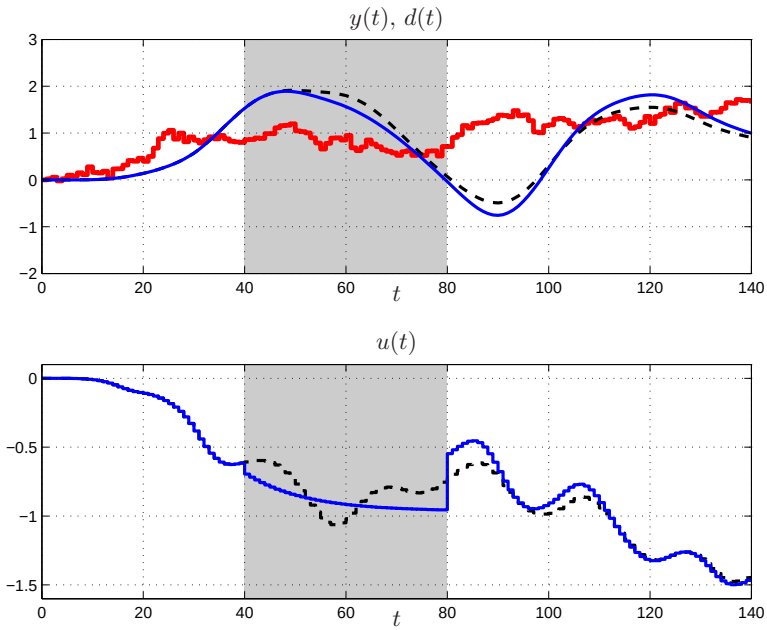
5.4.3 Zero Approximation

As the hold POC the zero POC can be seen as a crude approximation G_r of the POC G, now with G_r realized by $A_r = K_r = C_r = 0$ in (4.2), see Section 4.4. Simulating this POC over the studied scenario results in the behavior shown in Figure 5.8(b).

As for the zero POC in the deterministic method scenario, the system output rapidly starts to grow away from the desired value when the outage occurs. The prediction error during the outage causes the controller to apply a large control signal when communication is restored, in order to stabilize the system.

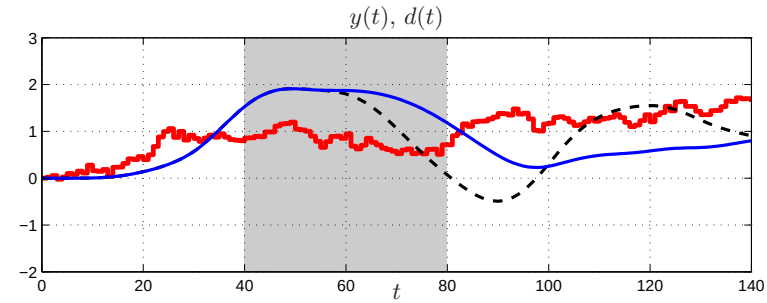


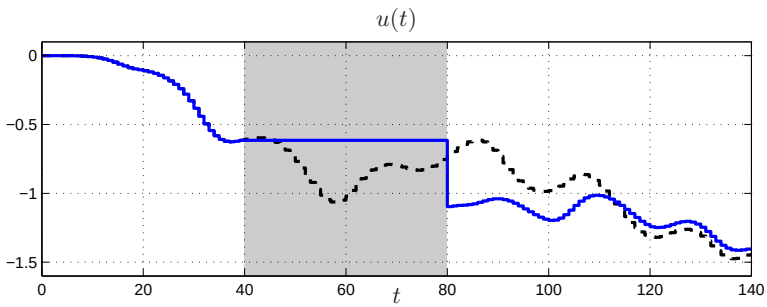
(a) reduced POC with $r = 3$



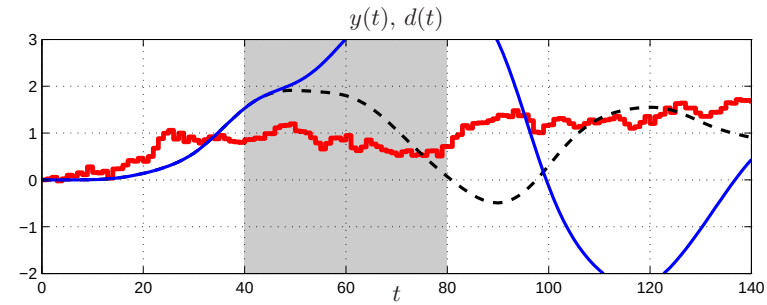
(b) reduced POC with $r = 1$

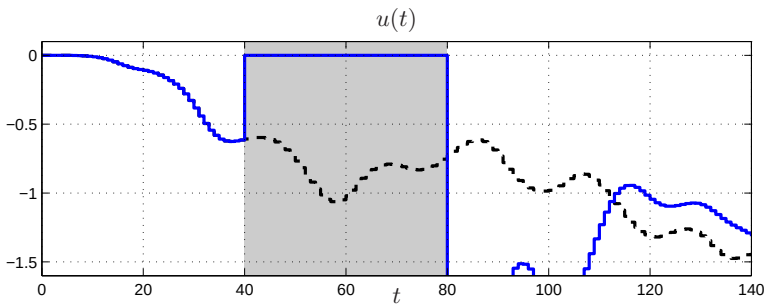
Figure 5.7: Comparison of Hankel reduced POC behavior (solid) with loss-free behavior (dashed) under disturbance d (dotted) and outage (greyed)





(a) hold POC





(b) zero POC

Figure 5.8: Comparison of hold and zero POC behavior (solid) with loss-free behavior (dashed) under disturbance d (dotted) and outage (greyed)

5.4.4 Switched Balanced Truncation

Making the same stable/anti-stable decomposition of G as in Section 5.4.1 one can use an LMI solver (*e.g.*, (Sturm, 1999)) to try to find the generalized Gramians of G_s. For the given example there exists a generalized observability Gramian but the problem of finding a generalized controllability Gramian is not feasible. Hence it is not possible to perform switched balanced truncation on this example.

5.5 Summary

We have presented simulations of POCs derived using both the nominal deterministic and the optimal stochastic synthesis methods. These simulations have then been compared to simulations of the hold and zero POCs.

The simulations have shown that the performance both in the case of deterministic and stochastic disturbances is significantly improved if one uses a POC synthesized using the deterministic respectively the stochastic method, instead of using the commonly used hold or zero POC.

Further we have seen that the complexity reduction methods introduced in the thesis can be used to reduce a high order POC to a much lower order POC essentially having the same performance.

Chapter 6

Model Predictive Control based on Wireless Sensor Feedback

Here we present the design and experimental validation of a control system with both wireless sensor and actuator links. 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.

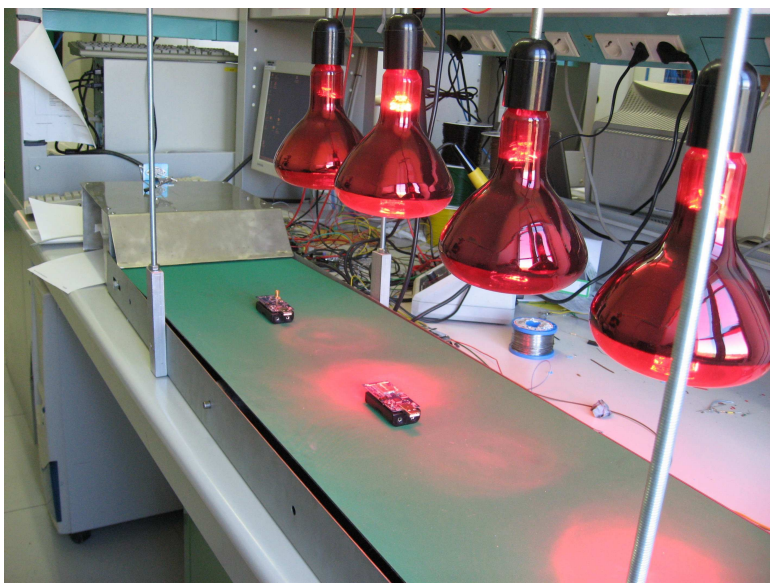


Figure 6.1: The laboratory process controlled.

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 gives interesting hybrid dynamics in the process, which we will handle using a hybrid model predictive control (MPC) controller (Bemporad and Morari, 1999). The reason for using the hybrid MPC algorithm is that it explicitly takes the hybrid nature of the system into account as well as it 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 delocalization of the MPC controller to a remote computer able to handle the computations. Both the measurements from the process to the MPC controller and the control signals from the MPC controller to the process will be transmitted over wireless links. In order for this to work 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 followed by 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 it.

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 ”part” placed on the belt. 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 ”part” is a temperature sensor equipped with a radio device able to transmit its temperature reading. The device is called a mote.

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

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

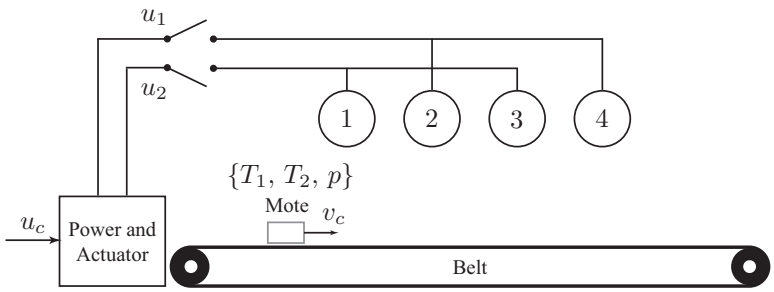


Figure 6.2: Schematics of the process.

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 $u_c \in \mathbb{R}$ and $u_1, u_2 \in \{0, 1\}$ and $T_{ss} : \mathbb{R}^3 \to \mathbb{R}$ is a static nonlinearity. 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 $\{I_1, I_2, \ldots, I_\ell\}$ and approximate f_i, $i = 1, 2$ in (6.2), by the functions

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

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

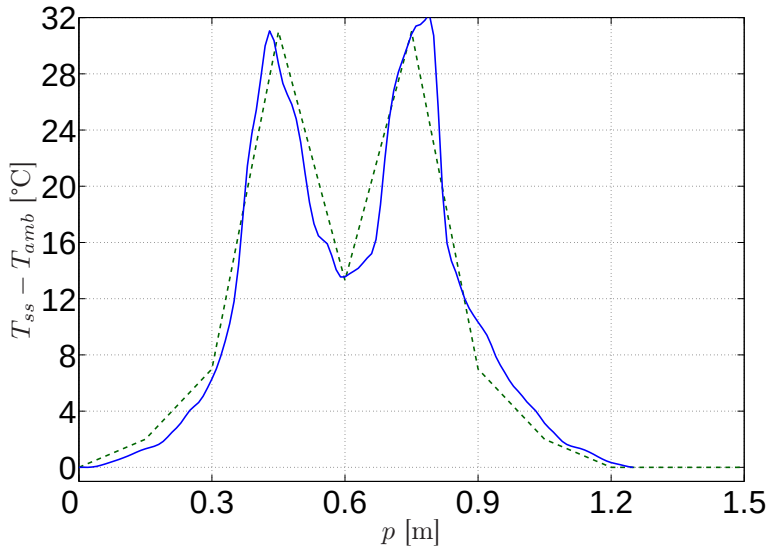


Figure 6.3: $T_{ss} - T_{amb}$ and its piecewise affine approximation.

$\chi(t)$ 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 $T_s = 250\,ms$ giving the following discrete-time system

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

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.

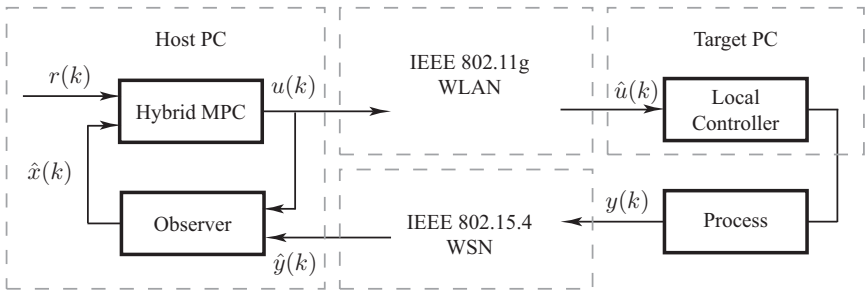


Figure 6.4: Wireless control architecture.

6.2 Control System Architecture

To be able to place the MPC controller 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 algorithm 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 delocalization of the remote MPC controller from the process site. Measurements are sent from the process to the observer over a wireless sensor network (WSN) implemented on the network standard IEEE 802.15.4, while the commands from the MPC controller 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" this means that

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

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

where $u(k)$ and $y(k)$ are the outputs from the MPC controller 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 controller to the local controller is lost the local controller applies a hold mechanism giving $\hat{u}(k) = \hat{u}(k-1)$, this is in fact a hold POC as described in Example 3.1 in Chapter 3. 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 2.

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 Algorithm

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 constrain the acceleration of the belt. 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 $\big(v_c(k) - x_u(k)\big)/T_s$. By this constraints on the acceleration can be expressed as state constraints. The extended system model becomes

$$\begin{aligned} x(k+1) &= \begin{pmatrix} a_{11} & 0 & 0 & 1 & 0 \\ a_{21} & a_{22} & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x(k) + \begin{pmatrix} b_{11}\chi(k) \\ b_{11}\chi(k) \\ b_{32}v_c(k) \\ 0 \\ v_c(k) \end{pmatrix}, \\ y(k) &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} x(k), \end{aligned} \tag{6.5}$$

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

In order to apply the hybrid model predictive control algorithm the system model in (6.5) 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 (6.5) one get a MLD system as

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k), \\ y(k) &= Cx(k), \\ E_2\delta(k) + E_3z(k) &\leqslant E_1u(k) + E_4x(k) + E_5, \end{aligned} \tag{6.6}$$

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 algorithm based on the following optimization problem, solved at each time step k,

$$\begin{aligned} &\min\ J(\{u(n), \delta(n|k), z(n|k)\}_0^{N-1}, x(k)) \triangleq \\ &\quad q_\rho\rho^2 + \sum_{n=0}^{N-1} \left(q_{v_c}v_c(n)^2 + q_z \left(\frac{1}{T_s}\right)^2 \big(v_c(n) - x_u(n|k)\big)^2 + \|Q_y(y(n|k) - y_r)\|_2 \right) \\ &\text{subject to (6.6) and} \\ &\quad \begin{pmatrix} 20 \\ 20 \\ 0 \end{pmatrix} \leqslant \begin{pmatrix} T_1(n|k) \\ T_2(n|k) \\ p(n|k) \end{pmatrix} \leqslant \begin{pmatrix} 50 \\ 50 \\ 1.2 \end{pmatrix}, \quad n = 1, \ldots, N \\ &\qquad -0.1 \leqslant v_c(n|k) \leqslant 0.1, \qquad n = 0, \ldots, N-1 \\ &\qquad u_1(n|k),\ u_2(n|k) \in \{0,1\}, \qquad n = 0, \ldots, N-1, \end{aligned} \tag{6.7}$$

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

The tuning parameters of the hybrid MPC problem (6.7) 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 algorithm 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 (6.7) is solved with $x(0|k) = \hat{x}(k)$;
3. The first optimal input $u^*(0)$ 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 since it in this case is a simple solution. It is given by

$$\begin{aligned} \hat{x}(k+1|k+1) &= \Phi\hat{x}(k|k) + \xi(k) + K\big[\hat{y}(k+1) - C\big(\Phi\hat{x}(k|k) + \xi(k)\big)\big], \\ \xi(k) &= \begin{pmatrix} b_{11}\chi(k|k) \\ b_{21}\chi(k|k) \\ T_s v_c(k) \end{pmatrix}, \quad K = \begin{pmatrix} k_{11} & k_{12} \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \end{aligned} \tag{6.8}$$

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

If the packet at time k is lost the estimation evolves in open loop so that (6.8) 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 2 which also updates its internal states if the measurement is received and evolves in open loop if it is lost.

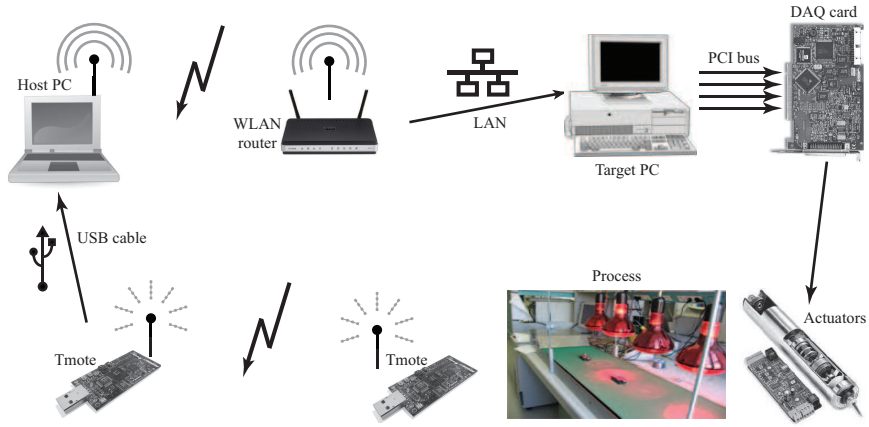


Figure 6.5: Hardware architecture.

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 controller and the observer runs in the *Host PC*, which is a $1.2\,GHz$ Pentium-M™ 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 National Instruments® DAQ-board is used. The process belt is moved using a belt roller with an encapsulated motor controlled using a motor servo. 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 through the DAQ board. The ”parts” or 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 temperature and communicates it to another mote connected to the USB port of the Host PC.

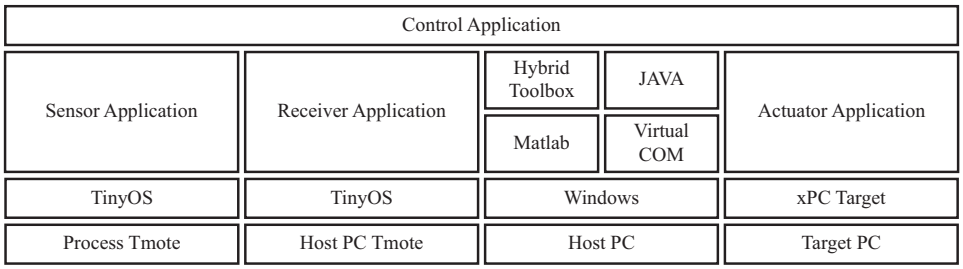


Figure 6.6: Software architecture

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 algorithm. The underlying optimization software used in the execution of the MPC is CPLEX™ 9.0 (ILOG, Inc., 2004). Concurrently to the MPC algorithm, 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 toolbox provides a transparent way to use a standard PC, in our case the Target PC, as a microcontroller. It also provides a hardware abstraction for TCP/IP communication with the *Host* as well as an abstraction towards the DAQ card. The full software abstraction of the Target PC from the WLAN router to the DAQ 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 algorithm is implemented on the *Host* within the Hybrid Toolbox for MATLAB. The system model (6.5) is written in `HYSDEL` (Torrisi and Bemporad, 2004) and automatically converted by the associated compiler into the MLD system

(6.6). The optimal control problem (6.7) 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, respectively 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 the hold POC, 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 controller 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.

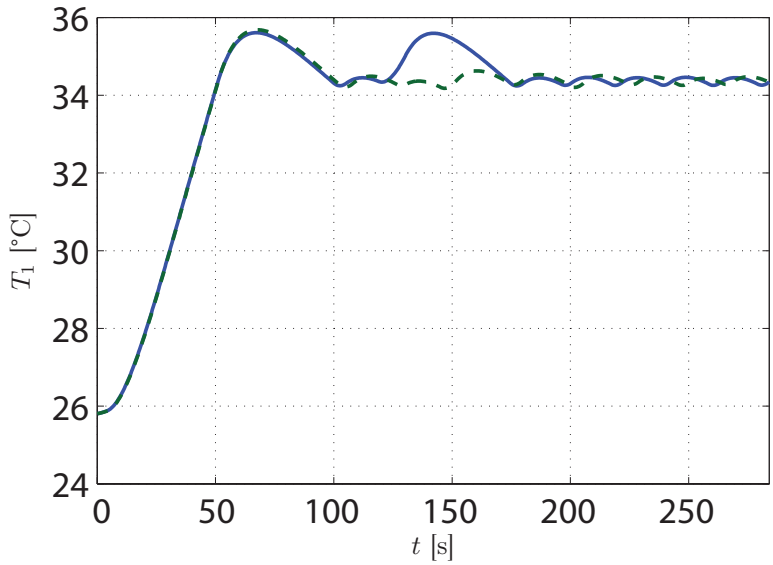
First we analyze the behavior with respect to data losses in the communication link between the MPC controller and the local controller, denoted the forward channel, where the input commands are sent over the WLAN network. Then we look at data losses in the sensor to MPC link, denoted the backward or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

6.5.1 Losses in the MPC Controller to Local Controller Communication Link

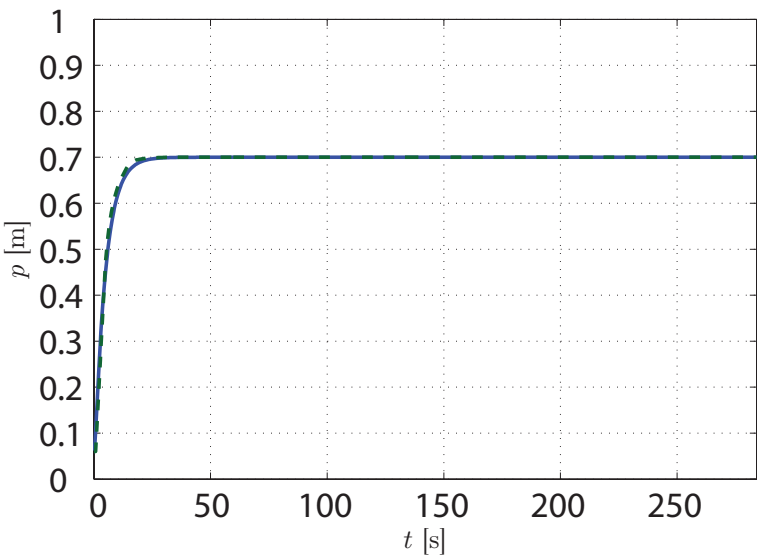
For the forward link case we use a constant reference $y_r = [35,\ 0.7]^T$ for the temperature and position respectively. Since the WLAN network used is very reliable, data losses are introduced on purpose by discarding packets according to a data-loss profile obtained from a sensor network. In this 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 $t = 150\,s$ 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.

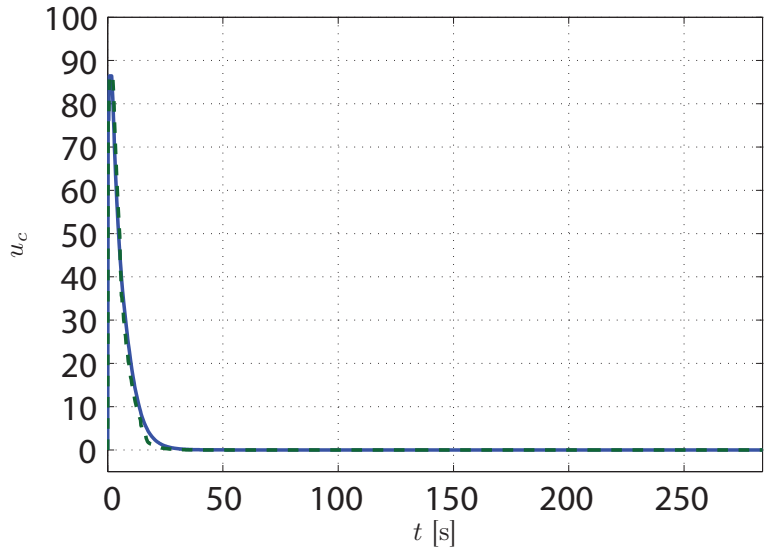


(a) Temperature.

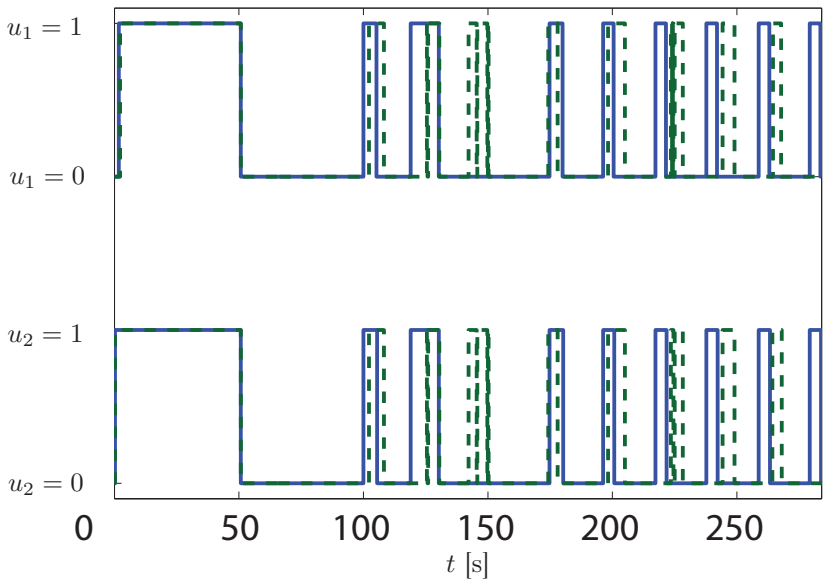


(b) Position.

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

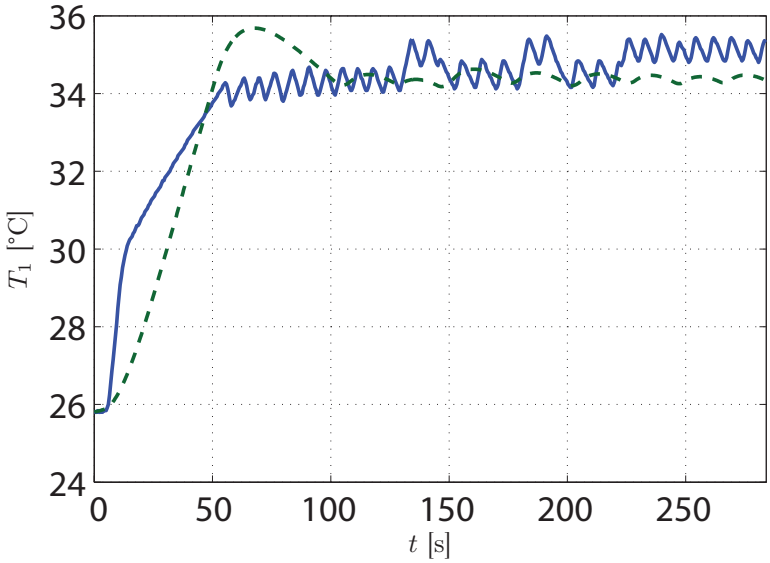


(a) Belt motor command.

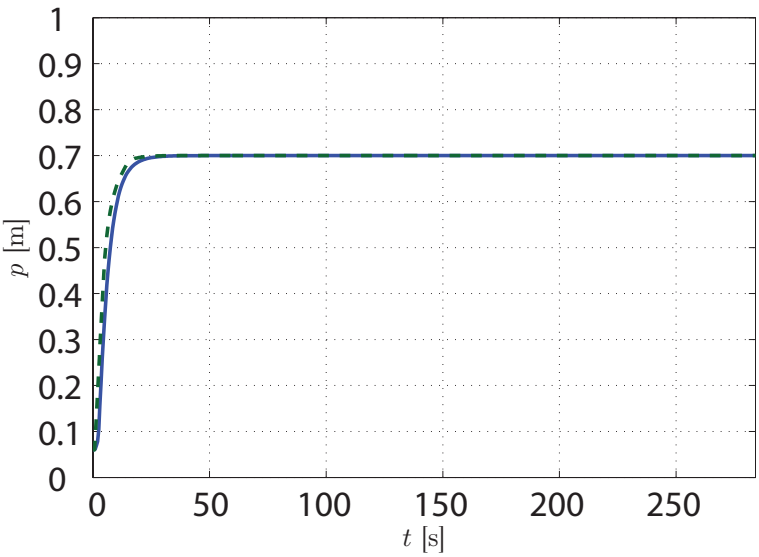


(b) Lamp commands.

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

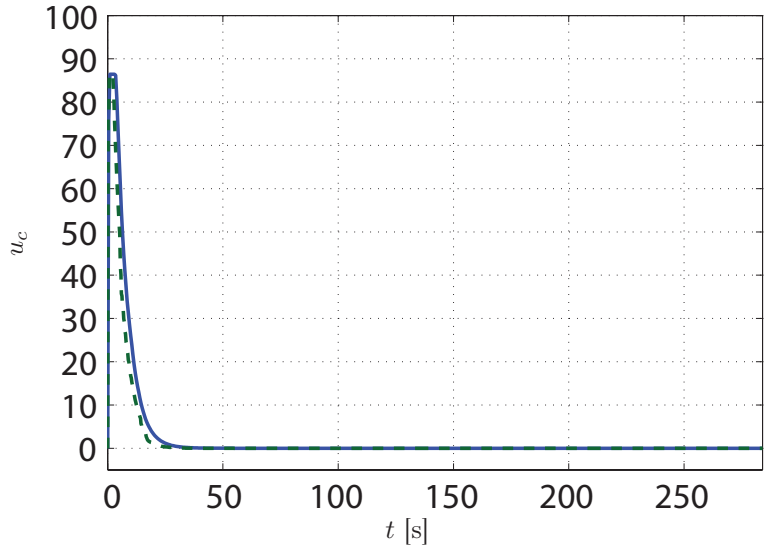


(a) Temperature

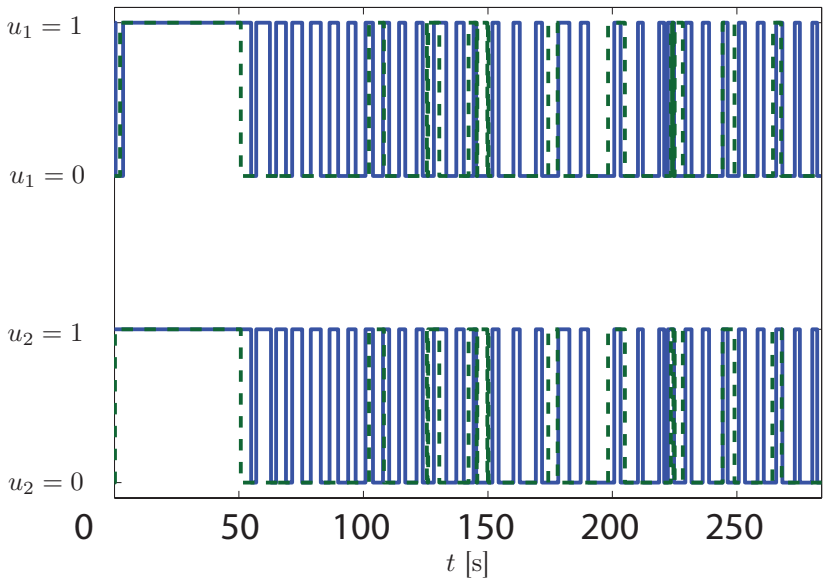


(b) Position

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

6.5.2 Losses in the Sensor to MPC Controller Communication Link

Now let us consider the case where losses occur in the feedback channel between the sensor and the MPC controller. In these experiments, the temperature reference is a square wave with maximum 42 $^{\circ}C$, minimum 38 $^{\circ}C$ 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.

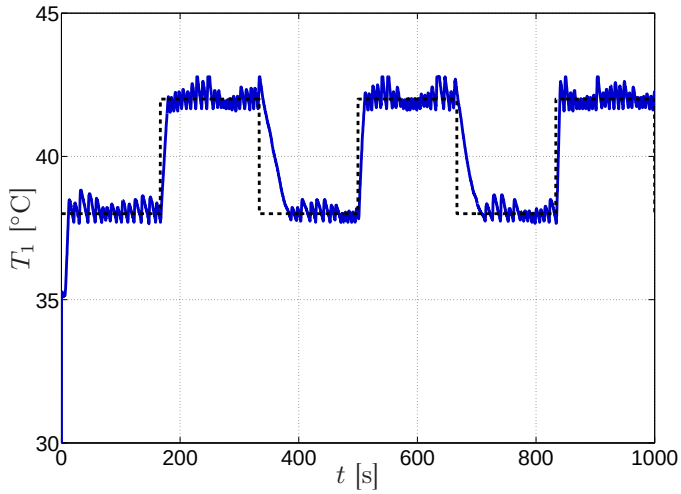
To induced packet loss an extra sensor node is used to disturb the communication of the sensor on the belt by sending large amounts of data into the network. Even though the base station is able to discard the data sent by this extra sensor, the extra traffic and processing required will cause packet 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.

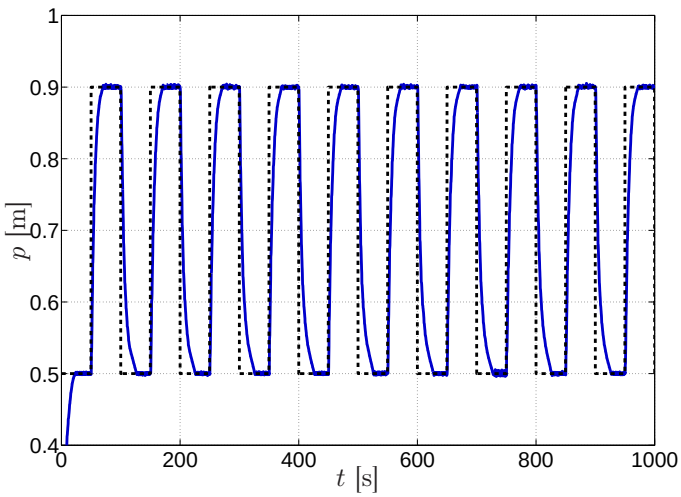
Low Loss Scenario

We first study the low loss scenario. Figure 6.11 shows the output as perceived by the MPC controller. 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 controller 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 setpoint. These are due to the discrete nature of the lamp switching. The tracking of the position reference is even higher. This is because of 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.

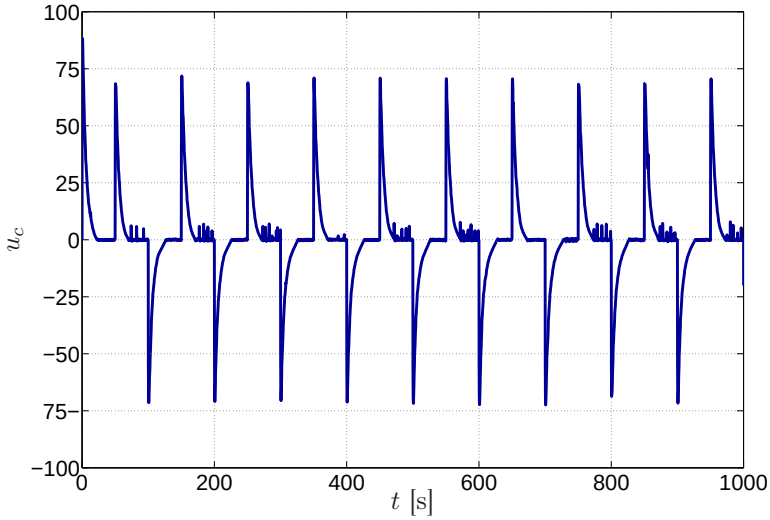


(a) Temperature: In case the measurement is lost the predicted value is shown.

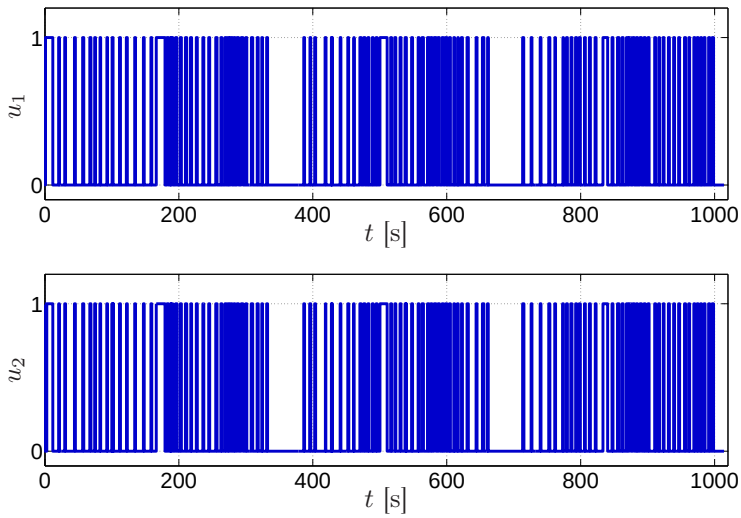


(b) Position

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 controller.

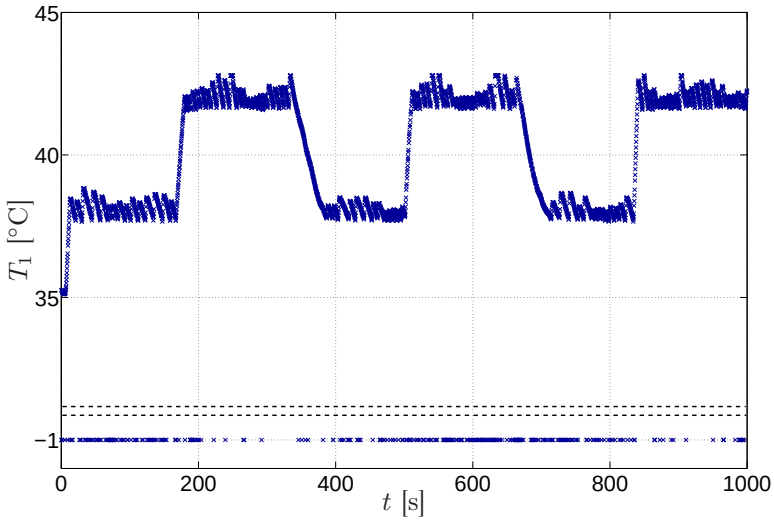


(a) Belt motor command.

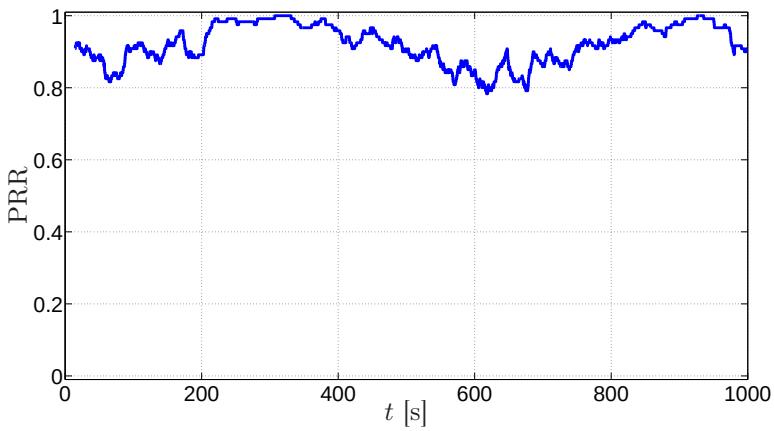


(b) Lamp commands.

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



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



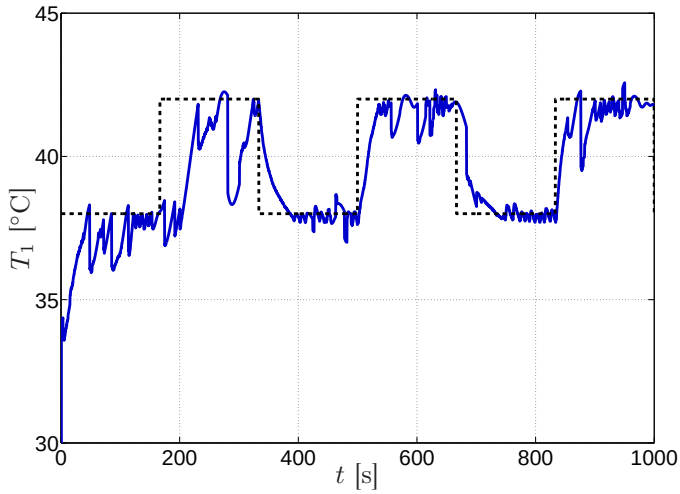
(b) Packet reception rate PRR(t) computed over a moving centered window of 30s.

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

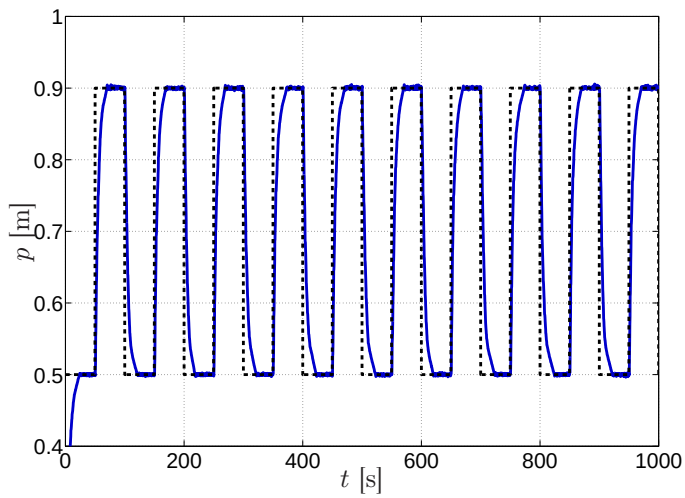
High Loss Scenario

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 controller. The control signal computed by the MPC controller 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.

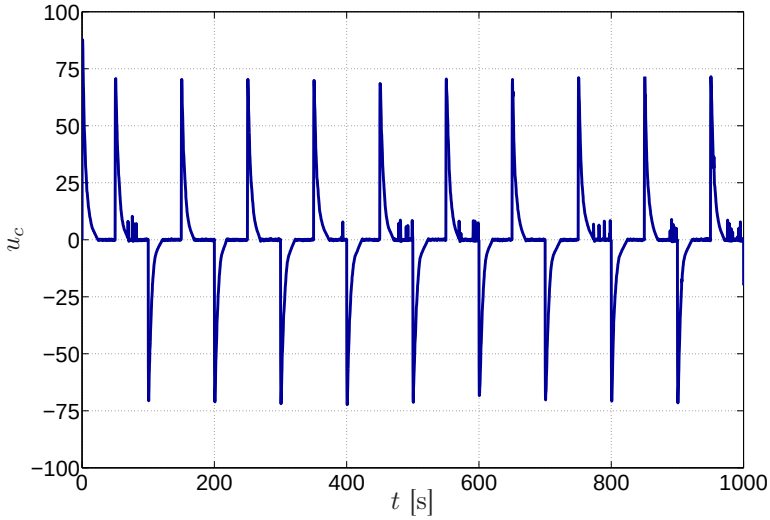


(a) Temperature: In case the measurement is lost the predicted value is shown.

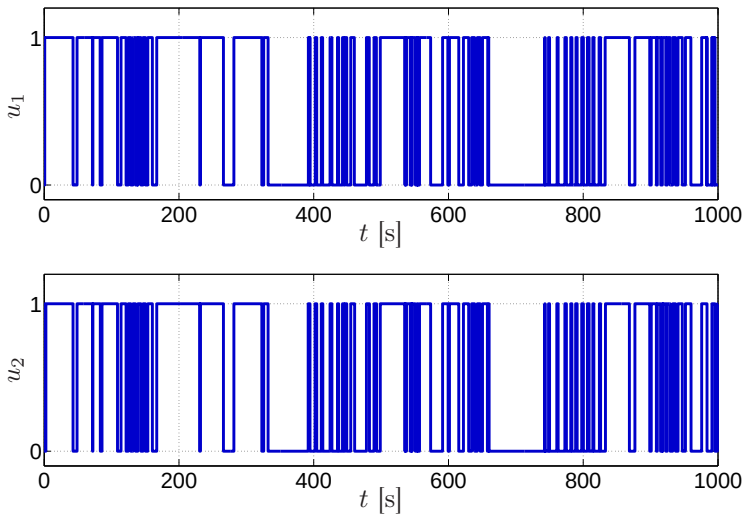


(b) Position.

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 controller.

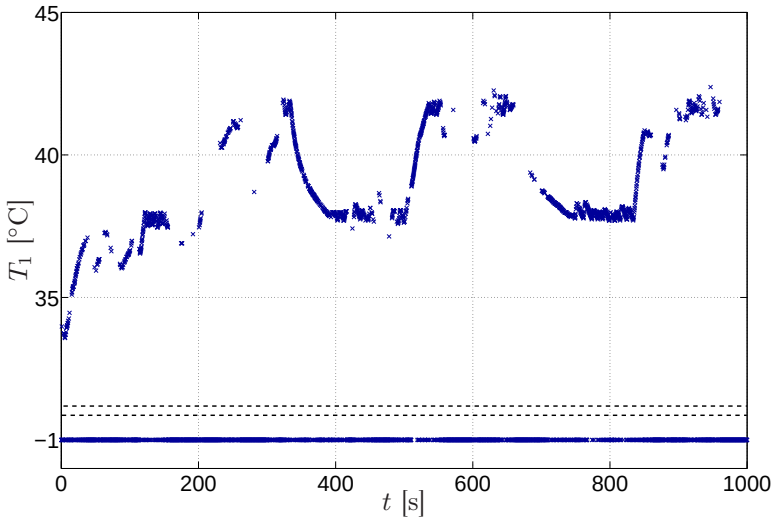


(a) Belt motor command.

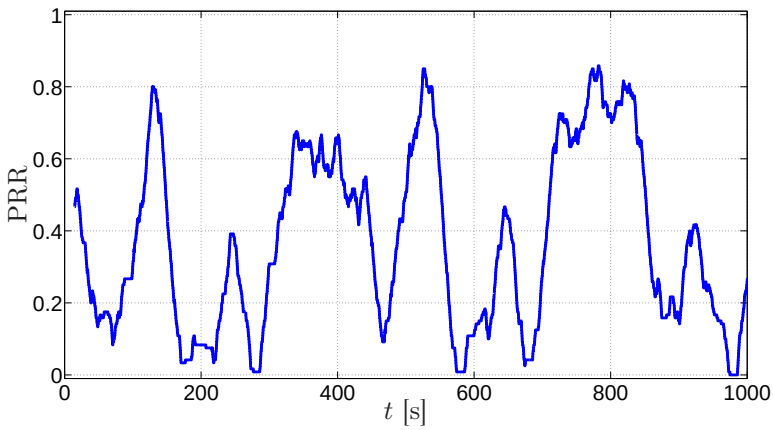


(b) Lamp commands.

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



(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

We conclude the thesis with a brief summary of the main results followed by interesting problems for future research efforts within the area. We start by summarizing the results and then discuss ideas for future work.

7.1 Summary

We have presented a new methodology for compensating for communication losses in networked control systems. The proposed Predictive Outage Compensator (POC) has been shown to give significantly improved performance compared to previously used compensation schemes.

In particular, we derived two methods to synthesize a POC, one for SISO systems affected by deterministic disturbances and one for MIMO systems affected by stochastic disturbances and noise. For both methods prediction error bounds are presented. Methods have also been developed to reduce the complexity of a POC by means of Hankel norm approximation and switched balanced truncation. A priori approximation error bounds for these reduction methods are presented. Finally, different POCs where demonstrated and compared on a simulated tank system.

Further, the implementation of a hybrid MPC design for control over wireless networks was presented. We have shown that setup was easy to tune and that the use of hybrid MPC has several advantages. The most obvious 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

We note that the Kalman filter in the optimal stochastically synthesized POC estimates the full state of the entire 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 strategies for networked systems.

It is also interesting to characterize for which POCs it is possible to perform switched balanced truncation, *i.e.*, for which POCs there exists a generalized controllability Gramian and a generalized observability Gramian. Another problem of interest is to further investigate the problem of Hankel norm approximation where a decaying weight, as a function of outage length, is added to the performance measure as discussed in Remark 4.2.1. This problem gives insight to how the POC complexity relates to the maximum tolerable outage length. For the proposed hybrid MPC architecture it is interesting to implement a more advanced POC on the actuator side of the system, to see how that could enhance the overall system performance.

There are several generalizations and extensions of the predictive outage compensation framework that are of interest to explore. The first is to extend the analysis to handle the case when POCs are simultaneously used within the same loop, to compensate for losses of both control commands and sensor measurements. A natural further extension is to study the case when multiple control loops share the same network, and for this case investigate how the interactions between the controllers and POCs affect the overall control performance. In the case of MIMO systems it is interesting to extend the POC theory to, instead of handling losses of the entire vector of measurements y or control signals u, handle losses in the individual channels y_i and u_i.

To avoid introducing computations at the actuator side of the network an alternative POC placement is of interest. One possibility is to place the POC at the controller and use it to, given past and present control commands, compute a predicted future trajectory of the control signal. This trajectory can then be sent to the actuator, which then only needs to play out values from a buffer if no new command is received. This receding-horizon approach has been investigated for the case of MPC in (Bemporad, 1998) where the entire predicted open-loop trajectory is sent to the actuator to be used as a play out buffer.

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.

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

P. Antsaklis and J. Baillieul. Special Issue on Networked Control Systems. *IEEE Transaction on Automatic Control*, 49(9), September 2004. Guest Editors.

P. Antsaklis and J. Baillieul. Special Issue on Technology of Networked Control Systems. *Proceedings of the IEEE*, 95(1), January 2007. Guest Editors.

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*, 2-3:261–279, 2007. Invited Paper.

A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. In *Proceedings IEEE Conference on Decision and Control*, pages 2133–2138, Tampa, FL, December 1998.

A. Bemporad. *Hybrid Toolbox – User's Guide*, December 2003. `http://www.dii.unisi.it/hybrid/toolbox`.

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*, AC-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 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*, 2009. To Appear.

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

P.P. Caffier, U. Erdmann, and P. Ullsperger. Experimental evaluation of eye-blink parameters as a drowsiness measure. *European Journal of Applied Physiology*, 89 (3–4):319–325, May 2003.

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, December 1999.

E. N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, 1960.

K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L_∞-error bounds. *International Journal of Control*, 39:1115–1193, 1984.

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

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

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 IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.

HART Communication Foundation. WirelessHART, 2007. `http://www.hartcomm2.org`.

E. Henriksson. Hybrid Model Predictive Control based on Wireless Sensor Feedback. Master's thesis, School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden, 2007.

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

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

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, January 2007.

ILOG, Inc. *CPLEX 9.0 User Manual*. Gentilly Cedex, France, 2004.

International Society of Automation. ISA100 Family of Standards, 2009. `http://www.isa100.org`. Ongoing.

V. Kawadia and P. R. Kumar. A cautionary perspective on cross layer design. *IEEE Wireless Communication Magazine*, 12:3–11, 2005.

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

S. Lall and C. Beck. Error-bounds for balanced model-reduction of linear time-varying systems. *IEEE Transactions on Automatic Control*, 48(6):946–956, June 2003.

D. Liberzon. *Switching in Systems and Control*. Birkhäuser, 2003.

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

X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2823–2829, Atlantis, Paradise Island, Bahamas, 2004.

L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. The MIT Press, 1983.

B. C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26 (1):17–32, February 1981.

Moteiv Corporation. *Tmote sky*, March 2007. `http://www.moteiv.com/products/tmotesky.php`.

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.

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

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, May 2008.

J-P. Richard. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39(10):1667–1694, October 2003.

W. J. Rugh. *Linear System Theory*. Prentice-Hall, second edition, 1996.

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

H. Sandberg and A. Rantzer. Balanced truncation of linear time-varying systems. *IEEE Transactions on Automatic Control*, 49(2):217–229, February 2004.

L. Schenato. To zero or to hold control inputs with lossy links? *IEEE Transactions on Automatic Control*, 54(5):1093–1099, May 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. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S S Sastry. Distributed control applications within sensor networks. *Proceedings of the IEEE*, 91:1235–1246, August 2003.

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

SOCRADES, Integrated Project, EU Sixth Framework Programme. `http://www.socrades.eu`.

B. Stenlund and A. Medvedev. Level control of cascade coupled flotation tanks. *Control Engineering Practice*, 10:443–448, 2002.

J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

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, August 1993.

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, March 2004.

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

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

K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, New Jersey, 1996.