Toward Applications Oriented Optimal Input Design With Focus On Model Predictive Control

CHRISTIAN A. LARSSON

Licentiate Thesis
Stockholm, Sweden 2011
Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i reglerteknik fredagen den 30 september 2011, klockan 10.15 i sal B1, Kungliga Tekniska högskolan, Brinellvägen 23, Stockholm.

© Christian A. Larsson, september 2011

Tryck: Universitetsservice US AB
Abstract

Modern control designs are, with few exceptions, in some way model based. In particular, predictive control has rapidly become a popular control strategy, implemented in a large number of industrial plants. Model predictive control (MPC) uses a model to predict the impact of future control inputs on the controlled plant. The quality of the model can have a large impact on the achievable control performance. It is widely reported that modeling is the single most time and cost consuming part of the commissioning of an industrial MPC and therefore an important research issue.

This thesis addresses the need for good modeling for MPC by introducing an optimal input design and identification method tailored to the specifics of predictive control. Parametric models are used and the influence of the individual parameters on the control performance is measured through a cost function. This leads to a set of parameters that are deemed acceptable. Optimal input design is used to ensure, with high probability, that the estimated parameters are in the set of acceptable parameters while keeping experimental cost low. It is shown that optimal input design can lead to a significant reduction of the experimental cost while still guaranteeing acceptable control performance. A toolbox for optimal input design in MATLAB is also presented.

Real world systems tend to be nonlinear and sometimes it is necessary to model them as such. Input design for two types of nonlinear systems with finite memory is considered. Similarities and differences compared to the linear case are pointed out and exploited. Convex formulations of the optimal input design problem are presented. It is shown by example that the resulting optimal design can differ greatly compared to designs for linear models.
There are many people who deserve to be recognized for their help in completing this thesis.

My main advisor Håkan Hjalmarsson for all the constructive feedback and support you have provided throughout the process leading up to the thesis. Your enthusiasm and commitment is inspiring. My co-advisor Cristian Rojas, a seemingly infinite source of insightful comments and excellent references. Bo Wahlberg and system identification group for the stimulating work environment. This thesis would not exist without you.

All colleagues at the Automatic Control Lab who make it a really great place to work. Mariette Annergren for the good collaboration and meticulous proof reading. The people who share office space with me, José Araujo, Per Hägg and André Teixeira, for all the interesting discussions and all the fun. Special thanks also to the administrators Karin Karlsson Eklund and Anneli Ström for making sure that everything runs as smoothly as it does.

My gratitude also to the The Swedish Research Council (VR) and the European Commission, through the WIDEn and AutoProfit projects, for the financial support that has made this research possible.

Finally, I want to thank all my friends and in particular my family for all their love and support over the years!

Christian Larsson
Stockholm, August 2011.
# Contents

Acknowledgements v

Contents vi

1 Introduction 1

1.1 Motivating examples .................. 3
1.2 Problem formulation .................. 6
1.3 Contributing papers .................. 7
1.4 Related work ......................... 8
1.5 Outline ............................. 9

2 Background 11

2.1 Model predictive control .............. 11
2.2 System identification .................. 13
2.3 Application requirements .............. 16
2.4 Optimal input design .................. 17
2.5 Convex optimization .................. 20
2.6 Scenario based optimization .......... 21

3 Experiment design for model predictive control 23

3.1 System description .................... 24
3.2 Identification setup ................... 27
3.3 Application cost ...................... 27
3.4 Input design .......................... 31
3.5 Identification algorithm .............. 36
3.6 Signal generation ..................... 38
3.7 Identifiability issues .................. 40
3.8 Linear models of nonlinear systems ... 42
3.9 Summary ............................ 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Experimental study: A water tank process</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>Process description</td>
<td>46</td>
</tr>
<tr>
<td>4.2</td>
<td>Simulations</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>Experiments</td>
<td>55</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>Experiment design for nonlinear systems</td>
<td>59</td>
</tr>
<tr>
<td>5.1</td>
<td>General nonlinear systems</td>
<td>60</td>
</tr>
<tr>
<td>5.2</td>
<td>Static nonlinear systems</td>
<td>62</td>
</tr>
<tr>
<td>5.3</td>
<td>Nonlinear FIR systems</td>
<td>65</td>
</tr>
<tr>
<td>5.4</td>
<td>Summary</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>Experiment design for MPC</td>
<td>73</td>
</tr>
<tr>
<td>6.2</td>
<td>Experiment design for nonlinear systems</td>
<td>75</td>
</tr>
<tr>
<td>A</td>
<td>Input design toolbox</td>
<td>77</td>
</tr>
<tr>
<td>A.1</td>
<td>Introduction</td>
<td>77</td>
</tr>
<tr>
<td>A.2</td>
<td>Tutorial</td>
<td>78</td>
</tr>
<tr>
<td>A.3</td>
<td>Moose declaration block</td>
<td>80</td>
</tr>
<tr>
<td>A.4</td>
<td>Keywords</td>
<td>80</td>
</tr>
<tr>
<td>A.5</td>
<td>Spectral factorization</td>
<td>83</td>
</tr>
<tr>
<td>A.6</td>
<td>Implementation</td>
<td>83</td>
</tr>
<tr>
<td>A.7</td>
<td>Future versions</td>
<td>84</td>
</tr>
</tbody>
</table>

Bibliography 87
Chapter 1

Introduction

“...To call in the statistician after the experiment is done may be no more than asking him to perform a post-mortem examination: he may be able to say what the experiment died of.”

Ronald Fisher

Models of reality have always been an important part of trying to understand and explain the world around us. These models range from highly personal mental models to more generally accepted models of how things work. Statistical and mathematical models are quickly becoming widely used in many areas of daily life. The local grocery store uses customer shopping pattern models to decide where different items are placed. Meteorologists use weather models to give the daily forecasts. Models are used to predict the stock market to help investors, et cetera. Even though these models often are of great help, it is clear that they are far from accurate all the time.

Automatic control is no exception and any modern control design method is in some way model based. The model can be used in simulations to test the properties of the controller or they can be used explicitly in the controller design. One such design is model predictive control (MPC), a controller structure developed by industry to easily handle multivariate, constrained control problems. The central idea is to use a model of the system to find the best possible control action. Because a model is directly used in the control, the agreement between model and plant becomes important.

A problem of many modern systems, from a modeling perspective, is their size and complexity. There is simply no way of capturing every aspect of the system in one model...
and even if it were possible, it might not be desirable. Instead the model should capture the system properties that are important for the intended use of the model. Regardless of how the model is used, it is always important to have a good understanding of what the model captures and what it does not capture.

MPC in industry

Predictive control, in the form of MPC, is becoming an increasingly popular control strategy in many areas of process industry. It was early on adopted by the petrochemical industry and MPC is now most probably used in any modern refinery (Morari and Lee, 1997). The ability to easily handle multivariate systems and to account for constraints on the system plays a big role in the success of MPC. However, these benefits come at the price of modeling.

The most time consuming and costly part of the commissioning of an industrial MPC deployment is the modeling of the plant (Zhu, 2009). It has been estimated that modeling efforts can take up to 90% of the cost and time in a typical MPC commissioning. There are huge benefits to be made if the modeling process can be simplified.
Once the \( \text{MPC} \) is up and running, maintenance of the controller mostly relies on updating the model to account for changes in the plant. This re-modeling should also be done in an intelligent way so that product quality does not suffer.

There are many commercially available flavors of \( \text{MPC} \), marketed under different names. Many of these solutions are targeted for process industry and have been developed to work as a supporting tool for the process operators rather than to replace them. Qin and Badgwell (2003) give a survey of the largest commercial \( \text{MPC} \) technologies at the time. These are presented, according to the model types employed, in Figure 1.1. Linear models obtained through system identification are the most frequently used models. However, it is also reported in Qin and Badgwell (2003) that almost all commercial \( \text{MPC} \) use PRBS signals in the identification part of the commissioning. Furthermore, even for \( \text{MIMO} \) plants, inputs are most often manipulated one at a time. There seems to be large room for improvement by tailoring the identification experiments to the plant.

### 1.1 Motivating examples

To motivate the problem considered in the thesis, a few simple but illustrative examples are presented here.

**Example 1.1 (Energy efficient buildings)**

It is estimated that buildings currently account for around 40% of the world-wide energy consumption. Besides the large energy use, the energy demand from buildings is often co-occurrent, with high peak power demands as a result. Clever automation of heating, ventilation and air conditioning (HVAC) is often suggested as a means of improving the energy efficiency of buildings. One possibility is to use \( \text{MPC} \) for this automation. By using weather and occupancy predictions in the controller, pre-heating or cooling when energy is cheap can reduce both power costs and peak load. In (Ma et al., 2011) \( \text{MPC} \) is implemented for HVAC and shown to give a 25% reduction of energy use and 28% cost savings in a building simulation environment. Guaranteeing such performance results, for example through properly identified building models, is beneficial to the wallet of the owner of the building as well as to the environment.
Introduction.

Figure 1.2 Reference tracking for a DC-motor using MPC where the output is constrained to stay within the white area. The plot shows the response of the system when the model corresponds to the system (---), when the MPC uses incorrect models (- - -) and the reference trajectory (----). Discrepancies in a model can lead to bad reference tracking and even instability.

Example 1.2 (Plant-model mismatch)
This is a modification of Example 16.1 in Glad and Ljung (2000) where reference tracking for a DC-motor is implemented using MPC. The true system is given by

\[
\begin{align*}
x(t + 1) &= Ax(t) + Bu(t), \\
y(t) &= Cx(t).
\end{align*}
\]

The model used by the controller is

\[
\begin{align*}
x(t + 1) &= (A + \Delta A)x(t) + (B + \Delta B)u(t), \\
y(t) &= (C + \Delta C)x(t).
\end{align*}
\]

The system is simulated for ten random \( \Delta A, B, C \) combinations chosen such that the model parameters can vary up to 10\% around their true values. The system is also simulated using a model that corresponds to the true system, that is \( \Delta A, B, C = 0 \). The result is presented in Figure 1.2.
1.1. Motivating examples

Example 1.3 (Static gain estimation)
Consider the 10th order finite impulse response system

\[ y(t, \theta) = \sum_{i=1}^{10} \theta_i u(t - i) + e(t), \quad (1.1) \]

where \( u(t) \) is the controlled input and \( e(t) \) is Gaussian white noise with unit variance. Let \( G_0(\theta) = \theta_1 + \cdots + \theta_{10} \) denote the static gain of (1.1). Two possible ways of estimating \( G_0(\theta) \) from \( N = 400 \) samples of data are compared:

1. A PRBS input \( u(t) = \pm 1 \) is used to estimate \( \hat{\theta}_1, \ldots, \hat{\theta}_{10} \) and \( \hat{G}_0(\theta) = \sum \hat{\theta}_i \).
2. A constant input \( u(t) = 1 \) is used to directly get \( \hat{G}_0(\theta) = \sum \theta_i \).

The resulting estimates from 100 system identification experiments on a system with randomly chosen parameters are shown in Figure 1.3. The inputs are the same in all experiments, only the noise realization was changed. The static gain estimated using the white input has a variance which is around 10 times higher than the estimate obtained using the constant input.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{static_gain_estimate.png}
\caption{The static gain estimates (o) from 100 system identification experiments with the inputs: (1) PRBS and (2) constant, respectively. The true value of the static gain is indicated by (---).}
\end{figure}

In Example 1.1 the energy use of a building was improved by using MPC in the control of the HVAC system. It will be argued that when MPC is used, the model of the controlled plant, plays a large role in the results. A model that is tailored for the intended application is one way of guaranteeing, at least with a high probability, that the performance meets an acceptable level.

Example 1.2 illustrates the importance of using a model that captures the controlled system. A system is controlled using MPC with an incorrect model. It becomes
evident that incorrect models in MPC can lead to bad reference tracking and constraint violation. The resulting trajectories also show that not all models are equally bad — some perform fairly well while others result in instability. Hence, characterizing which model variations give acceptable performance and which do not, is an important issue.

Example 1.3 shows the importance of using the right input during system identification. A static gain estimation is made using two different inputs. The variance of the resulting estimates is 10 times higher in one of the experiments compared to the other. It turns out that the input with the lower variance is in fact optimal for minimizing the variance of static gain estimates. However, the price for reducing the variance is that individual parameters of the model cannot be identified.

The conclusions from the examples are:

• Some models give better performance than others, even though neither model exactly describes the true system. The identified model should be one that gives acceptable performance when used in control design.

• The chosen input can significantly influence the quality of the estimates. The input should be such that the resulting model gives acceptable performance in the control system.

1.2 Problem formulation

Models will never fully capture the systems they are meant to describe. This discrepancy will be called the plant-model mismatch. This mismatch can have a large impact on systems controlled using MPC. In addition, the intended use of the model affects which properties of the plant are relevant to capture in the model. For important properties the mismatch should be small while for unimportant properties the mismatch can be allowed to be larger. The quality of the estimate is influenced by the chosen input, which can be exploited.

The problem considered in this thesis is how to design the input used in system identification experiments when the obtained model will be used in MPC. The principal tool is optimal input design formulated as a convex optimization problem in frequency domain. The aim of the input design is to use as little resources as possible during the identification experiment while guaranteeing that some performance specification on the control design is met. This is formulated as the optimization problem

\[
\begin{align*}
\text{minimize} & \quad \text{Experimental effort} \\
\text{subject to} & \quad \text{Performance specifications}
\end{align*}
\]
Specific problems associated with the input design problem in an MPC context are identified and discussed. A method for input design for MPC is presented. The method is illustrated on an extensive case study on a four tank process, both in simulations and on an experimental process.

The fact that real systems are nonlinear is recognized and input design for this case is considered. The crucial differences compared to input design when systems are assumed linear are pointed out. This leads up to two special cases where convex optimization formulations are obtained. These methods are illustrated on examples which further show the differences compared to the linear case.

### 1.3 Contributing papers

The material presented in this thesis has previously appeared in or been submitted to scientific publications. How these publications relate to the different chapters of the thesis is detailed below.

The MPC-oriented input design formulation presented in Chapter 3 is based on two conference publications. For the evaluation of the method in Chapter 4, the simulations are based on one conference paper while the experiments have been performed later.


C. A. Larsson, C. R. Rojas, and H. Hjalmarsson. MPC oriented experiment design. In *Proceedings of the 18th IFAC World Congress*. Milano, Italy (2011b)

The methods for input design for nonlinear systems presented in Chapter 5 are based on one conference publication.


A toolbox for optimal input design implemented in MATLAB is also presented in this thesis. The toolbox has been developed in close collaboration with Mariette Annergren. The toolbox is presented in


Throughout the thesis, examples of input design problem are presented. Whenever possible, MATLAB-code for implementing examples in the toolbox is presented.
1.4 Related work

There is a lot of ongoing research in both the fields of input design and MPC that relate to the material here. There are also other research areas where problems of similar nature are considered. Here, parts of this research and how it relates to this thesis is discussed. It should in no way be seen as a complete overview of every related research topic but instead as a discussion on contributions that the author considers closely related to what is covered in the thesis.

Input design

The optimal input design research that most closely relates to the problems in this thesis is the applications oriented input design as presented in Hjalmarsson (2009a,b) and the so called least costly input design formulation, see for example Bombois et al. (2006). The central idea is that one should not spend more effort on identification than what is necessary to ensure some control performance. In least costly identification, the experiment is designed in terms of minimizing experimental cost subject to some a priori bounds on the uncertainty of the identified model such that robust control performance is guaranteed. The applications oriented approach instead focuses on ensuring that the identified model is within a model set that ensures that the performance degradation is below a pre-specified maximum.

These formulations of the problem also relate to the plant-friendly identification ideas found mostly in chemical process control (Rivera et al., 2003). The goal is to produce identification experiments with informative data while keeping variations on inputs and outputs within some constraints. It is often desirable to keep variations in outputs low so that product quality does not suffer during identification and to keep experiments short. These plant requirements are often in contrast to the theoretical requirements, such as long data records, et cetera.

Identification for MPC

The use of models is inherent to the formulation of MPC and finding these models is a large part of the commissioning of an MPC installation. Experience has shown that identified models often outperform physical models when used in MPC. For example, Mantelli et al. (2005) shows in examples that gains and time constants may differ up to a factor 2–3 in physical models compared to corresponding values in identified models. As a consequence, all major MPC manufacturers use identified models almost exclusively in their products, even though they have developed proprietary physical
models for common industrial processes Zhu (2009). This identification is, however, mostly performed using PRBS signals and by manipulating one input at a time.

Jorgensen and Jorgensen (2007) presents a prediction error based identification technique specifically for MPC. They argue that the prediction error criterion should be matched to the prediction performed by the MPC. The idea is illustrated on a simulation of a distillation column.

A closed-loop identification method for MPC is proposed by MacArthur and Zhan (2007). Here, a large group of models and model orders are identified and the best model for the considered application is then searched for. The argument being that no single method will suit all conditions found in industry. The method offers an automated identification process which is a practically attractive property.

Robust MPC

Robust MPC deals with constructing predictive controllers that satisfy stability and performance requirements for uncertain models. In that sense the goal is the same as for the problem considered in the thesis: the obtained controller should give acceptable performance. The formulation is, however, the opposite. Robust MPC looks for a controller that guarantees stability or performance given an uncertainty set for the model. The problem considered here is to find a model set that guarantees performance given a controller design method.

The research on robust MPC has focused largely on robust stability. Approaches include LMI formulations of the problem (Mao, 2003; Cuzzola et al., 2002; Feng et al., 2004). There are also min-max optimization based approaches that have been suggested. One example is (Wang and Rawlings, 2004) which guarantees stability and offset free reference tracking. Morari and Lee (1997) give a good survey of the early history of MPC up to late 1990s, including an overview of ideas for robust MPC and Jalali and Nadimi (2006) treat robust MPC developments up to 2006.

1.5 Outline

The remaining chapters of the thesis are organized as follows. Chapter 2 presents the relevant parts of system identification — in particular the prediction error framework — and optimal input design. Also a brief discussion of MPC techniques is given. In Chapter 3 the particularities of input design for MPC-applications are presented. The developed ideas are illustrated on a simulation study in Chapter 4. In Chapter 5 the scope is broadened and optimal input design for certain classes of nonlinear systems is
considered. The thesis is concluded in Chapter 6 which also includes a discussion on future research directions.
Chapter 2

Background

“Science may be described as the art of systematic over-simplification — the art of discerning what we may with advantage omit.”

Karl Popper

Optimal experiment design, as it is presented in this thesis, relies on results from statistics, system identification and convex optimization. This chapter briefly summarizes the theoretical background needed for the material developed in the succeeding parts of the thesis. References to more extensive treatments of the covered topics are suggested. The thesis assumes that the intended application of the model is MPC. A general MPC formulation is therefore also presented here.

2.1 Model predictive control

Model predictive control is a flexible and generally applicable control method which rapidly is becoming more and more used in industry. In particular, petrochemical industries widely use MPC and it is estimated that most petrochemical plants now have MPC implemented in parts of the process (Zhu, 2009). Because of the increased speed of processors and the advent of explicit MPC, where an explicit control law is calculated off-line, predictive control is now finding its way into faster and faster processes.

The main advantages of MPC are the simple treatment of multivariate processes and the ability to handle constraints on state variables and signals. These constraints can
come from the physics of the plants, for example from input or output saturations. They can be design constraints that are not hard physical constraints, for example state levels that result in a deteriorated product. One may even include traditionally hard to handle constraints, such as production cost or environmental aspects.

At the core of any MPC implementation is a model of the process that is to be controlled. Typically, this is a linear, discrete time model on the form

\[
\begin{align*}
x(t + 1) &= Ax(t) + Bu(t) + v(t), \\
y(t) &= Cx(t) + w(t).
\end{align*}
\]

Here \(x(t) \in \mathbb{R}^n\) is the state vector, \(u(t) \in \mathbb{R}^m\) the controlled input signal, \(y(t) \in \mathbb{R}^p\) the measured output, and \(v(t)\) and \(w(t)\) are stationary, zero-mean, white processes commensurate with \(x(t)\) and \(y(t)\). It is possible to use other types of models, for example transfer functions. Versions of MPC with nonlinear models are also sometimes used. The linear, state space formulation will be exclusively treated in this thesis.

The model is used to predict future outputs and the control signals are then computed based on these predictions. The prediction horizon defines the number of samples of the output that are predicted and the control horizon defines the number of input samples that are used in the optimization. As an example, a common cost
function used in the controller is

\[ J(t) = \sum_{i=0}^{N_y} \| \hat{y}(t+i|t) - r(t+i) \|^2_Q + \sum_{i=0}^{N_u} \| \Delta u(t+i) \|^2_R, \]  

(2.2)

where \( \hat{y}(t+i|t) \) is the predicted output, \( r(t) \) the reference and \( \Delta u(t) \) the change in input at time \( t \). The matrices \( Q \) and \( R \) are tunable weights. The norm \( \| x \|_A \) is equal to \( \sqrt{x^T A x} \). The prediction horizon is denoted \( N_y \) and the control horizon \( N_u \). The optimal input sequence is found by solving the optimization problem

\[
\begin{align*}
\text{minimize} & \quad J(t) \\
\text{subject to} & \quad \hat{y} \in \mathcal{Y} \\
& \quad u \in \mathcal{U}
\end{align*}
\]  

(2.3)

Here \( \mathcal{Y} \) and \( \mathcal{U} \) are the constraint sets for outputs and inputs respectively.

The solution to this optimization problem gives a sequence of optimal inputs over the control horizon, that is \( U(t) = \{ u(t), \ldots, u(t+N_u) \} \). However, at time \( t \), only the first input of the sequence, \( u(t) \), is actually applied to the system. The horizons are shifted one sample forward and the optimization then starts over in the next time instant. This principle is illustrated in Figure 2.1.

The performance of the MPC depends on the quality of the model that is used in the controller. Imprecise models may result in tracking error due to wrong gain estimates. This can be handled by incorporating integral action into the MPC. Modeling error might also lead to constraint violations. In the worst case, the plant–model mismatch might even lead to instability. The conclusion is that the quality of the model is of vital importance in MPC applications.

### 2.2 System identification

System identification is the process of constructing models of dynamic systems from experimental data. The goal is to find a model \( \mathcal{M} \) that describes the relevant properties of the studied system \( S \). The experiments can be carried out in open or closed loop, the latter requires some extra considerations. In the rest of the thesis, all identification is assumed to be performed in open loop.

The systems to be identified are considered to be linear time-invariant, asymptotically stable multivariate systems on the form

\[
S: \quad \begin{align*}
  x(t+1) &= A x(t) + B u(t) + v(t), \\
  y(t) &= C x(t) + w(t),
\end{align*}
\]  

(2.4)
with known input \( u(t) \) and output \( y(t) \). The noise processes \( \nu(t) \) and \( \omega(t) \) are white, zero-mean and stationary with covariance matrices \( \Lambda_\nu \) and \( \Lambda_\omega \) respectively.

There are many methods for identification of the model \( M \). Such methods can be non-parametric or parametric and can be performed in both the time and frequency domains. Which method is used depends on intended use of the model, the considered system and, to a certain extent personal taste. Frequency domain techniques are well covered in Pintelon and Schoukens (2001). This thesis considers parametric, time domain identification only. The model structure used is

\[
\begin{align*}
M(\theta) : \quad & \dot{x}(t+1, \theta) = A(\theta)\dot{x}(t, \theta) + B(\theta)u(t) + K(\theta)e(t), \\
& y(t, \theta) = C(\theta)\dot{x}(t, \theta) + e(t),
\end{align*}
\]  

(2.5)

where \( \theta \in \mathbb{R}^n \) represents the unknown parameters to be identified and \( e(t) \) is a zero-mean, white noise process with covariance matrix \( \Lambda_e \). It is assumed that \( M(\theta) \) can capture the true system, that is there is a parameter vector \( \theta^o \) such that \( S = M(\theta^o) \). The estimates of \( \theta \) are found using the prediction error method (PEM) (Ljung, 1999). This method and the properties of the resulting estimates is described next.

**Remark 2.1** The fact that (2.5) is on innovations form is not a limitation since any state space representation (2.4) can be transformed to the innovations form by spectral factorization (Söderström, 2002).

**Prediction error method**

The prediction error method estimates parameters by minimizing the difference between measured data and what is predicted by the model. When a model of the form (2.5) is used, the one step ahead predictor is given by (Ljung, 1999)

\[
\begin{align*}
\hat{x}(t+1, \theta) & = A(\theta)\hat{x}(t, \theta) + B(\theta)u(t) + K(\theta)(y(t) - \hat{y}(t)), \\
\hat{y}(t, \theta) & = C(\theta)\hat{x}(t, \theta).
\end{align*}
\]  

(2.6)

From \( N \) samples of input–output data, the parameter estimate is found as

\[
\hat{\theta} = \arg \min_\theta \frac{1}{N} \sum_{i=1}^{N} \ell (y(t) - \hat{y}(t, \theta)).
\]  

(2.7)

The function \( \ell(\cdot) \) is scalar valued and the most common choice is the quadratic function

\[
\ell (y(t) - \hat{y}(t, \theta)) = (y(t) - \hat{y}(t, \theta))^T \Lambda_e^{-1} (y(t) - \hat{y}(t, \theta)).
\]  

(2.8)
When $\Lambda^{-1}_e$ is not known, $\ell$ can be taken as (Goodwin and Payne, 1977)\[ \text{det}\left(y(t) - \hat{y}(t, \theta)\right) \text{det}\left(y(t) - \hat{y}(t, \theta)\right)^T. \]

Under mild assumptions and $\ell$ chosen as above, the estimate $\hat{\theta}$ is consistent and the asymptotic distribution is given by (Ljung, 1999)\[
\sqrt{N}(\hat{\theta} - \theta_0) \in \mathcal{N}(0, \mathbf{P}) \text{ as } N \to \infty, \tag{2.9a}
\]
\[
\mathbf{P} = \left[ \mathbb{E}\{\psi(t, \theta_0)\Lambda^{-1}_e\psi^T(t, \theta_0)\} \right]^{-1}, \tag{2.9b}
\]
\[
\psi(t, \theta_0) = \frac{d\hat{y}(t)}{d\theta} \bigg|_{\theta=\theta_0}. \tag{2.9c}
\]

An $\alpha$-level confidence ellipsoid for the estimated parameters is given by\[
U(\alpha) = \left\{ \theta : |\theta - \theta_0|^T \mathbf{P}^{-1} |\theta - \theta_0| \leq \frac{\chi^2_\alpha(n)}{N} \right\}. \tag{2.10}
\]

The constant $\chi^2_\alpha(n)$ is the $\alpha$-percentile of the $\chi^2$ distribution with $n$ degrees of freedom. The confidence ellipsoids will prove to be useful in optimal input design.

It is useful to consider the frequency domain expression for $\mathbf{P}^{-1}$. By Parseval’s theorem, $\mathbf{P}^{-1}$ is given by the following lemma.

**Lemma 2.1** In open-loop identification, the inverse covariance matrix $\mathbf{P}^{-1}$ in (2.9b) is an affine function of the input spectrum $\Phi_u$ given by\[
\mathbf{P}^{-1} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Gamma_1(e^{j\omega})\Lambda^{-1}_e \otimes \Phi_u(\omega) \Gamma_1^H(e^{j\omega}) \, d\omega \tag{2.11a}
\]
\[
+ \frac{1}{2\pi} \int_{-\pi}^{\pi} \Gamma_2(e^{j\omega})\Lambda^{-1}_e \otimes \Lambda_e(\omega) \Gamma_2^H(e^{j\omega}) \, d\omega, \tag{2.11b}
\]
\[
\Gamma_1(e^{j\omega}) = \begin{bmatrix} \text{vec } F_u^1 \\ \vdots \\ \text{vec } F_u^n \end{bmatrix}, \quad \Gamma_2(e^{j\omega}) = \begin{bmatrix} \text{vec } F_e^1 \\ \vdots \\ \text{vec } F_e^n \end{bmatrix} \tag{2.11c}
\]
\[
F_u^i = H(q, \theta) \frac{\partial G(q, \theta)}{\partial \theta_i} \bigg|_{\theta=\theta_0}, \quad i = 1, \ldots, n, \tag{2.11d}
\]
\[
F_e^i = H(q, \theta) \frac{\partial H(q, \theta)}{\partial \theta_i} \bigg|_{\theta=\theta_0}, \quad i = 1, \ldots, n, \tag{2.11e}
\]
\[
G(q, \theta) = \mathbf{C}(\theta) (q \mathbf{I} - \mathbf{A}(\theta))^{-1} \mathbf{B}(\theta), \tag{2.11f}
\]
\[
H(q, \theta) = \mathbf{C}(\theta) (q \mathbf{I} - \mathbf{A}(\theta^0))^{-1} \mathbf{K}(\theta^0) + \mathbf{I}. \tag{2.11g}
\]
Here vec $X$ is the row vector with rows of $X$ stacked next to each other.

**Proof.** $P^{-1}$ in (2.11b) is the open-loop version of Lemma 3.1 in Barenthin Syberg (2008) with

$$G(q, \theta) = C(\theta) (qI - A(\theta))^{-1} B(\theta), \quad (2.12a)$$

$$H(q, \theta) = C(\theta) (qI - A(\theta))^{-1} K(\theta) + I. \quad (2.12b)$$

2.3 Application requirements

The performance of a controller design based on a model of a process, is directly related to the quality of the model. Plant-model mismatch can lead to a degradation of the performance. As a measure of how the plant-model mismatch influences the degradation of performance an *application cost* is introduced.

When the considered models are parameterized by $\theta$, the application cost becomes a function of the parameters. If a perfect model of the system was available, that is if $\theta^o$ was known, the performance specifications would be met perfectly. Other parameters $\theta$ will lead to reduced or equal performance. This leads to the following definition of the application cost.

**Definition 2.1 (Application cost)** If $\theta \in \mathbb{R}^n$ are parameters corresponding to the model $M(\theta)$, and $S = M(\theta^o)$, the function

$$V_{\text{app}}(\theta) : \mathbb{R}^n \to \mathbb{R}, \quad (2.13)$$

is an *application cost* if

$$V_{\text{app}}(\theta^o) = 0, \quad V_{\text{app}}'(\theta^o) = 0, \quad V_{\text{app}}''(\theta^o) \geq 0. \quad (2.14)$$

In any application, there is some limit on how much performance can be allowed to degrade. This gives the requirement that the application cost is bounded, say by

$$V_{\text{app}}(\theta) \leq \frac{1}{\gamma}. \quad (2.15)$$

Here, $\gamma$ is an application dependent, real-valued and positive constant. This bound gives a set of parameters that correspond to acceptable application performance.
Definition 2.2 (Application set) If $V_{\text{app}}(\theta)$ is an application cost and $\gamma > 0 \in \mathbb{R}$, the set

$$\Theta_{\text{app}}(\gamma) = \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\},$$

(2.16)

is an application set.

This leads to the idea that the objective of applications oriented system identification should be to deliver parameter estimates that belong to the application set. These concepts come from (Hjalmarsson, 2009b,a; Bombois et al., 2006).

2.4 Optimal input design

The idea behind optimal input design is to — by exploiting the fact that $P^{-1}$ depends on the input — shape the input in such a way that the estimates become as good as possible while minimizing limited experimental effort. Traditionally some measure of the covariance matrix has been optimized to maximize the information available in the data. Typical measures include

**D-optimality** — the determinant of $P$ is minimized,

**A-optimality** — the trace of $P$ is minimized,

**E-optimality** — the largest eigenvalue of $P$ is minimized.

These criteria were considered early in the development of input design and are still frequently found in both research and practical use.

The focus has since the last 20 years, however, shifted more toward measuring quality in terms of the application of the estimated model. This lead to the ideas of identification for control. Instead of only the covariance matrix, the distance between nominal performance and performance achieved with the model is considered. This distance is minimized in some control relevant measure, typically under some constraints on the experimental effort. See for example (Gevers and Ljung, 1986; Gevers, 1991; Hjalmarsson et al., 1996).

The least costly identification approach turned the problem around and instead considers minimizing the experimental effort under constraints on plant-model mismatch. The ideas in the thesis are very close to the least costly identification formulation and will be called applications oriented experiment design. The general formulation of the
applications oriented experiment design problem is

\[
\begin{aligned}
\text{minimize} & \quad \text{Experimental effort} \\
\text{subject to} & \quad \theta \in \Theta_{\text{app}}(\gamma)
\end{aligned}
\]

where \(\Theta_{\text{app}}\) is an application set according to Definition 2.2.

The estimated parameters are random variables, typically with unbounded support, and consequently it is impossible to guarantee that estimates fall within \(\Theta_{\text{app}}(\gamma)\). Instead, this requirement has to be relaxed to a certain level of probability. One possibility is to use the confidence ellipsoid (2.10) and impose the condition that

\[
\theta \in \Theta_{\text{app}}(\gamma) \quad \text{for all} \quad \theta \in U(\alpha)
\]

or equivalently that

\[
U(\alpha) \subseteq \Theta_{\text{app}}(\gamma).
\]

According to Lemma 2.1, \(P^{-1}\) is an affine function of the input spectrum \(\Phi_u\) in open loop identification. As a consequence, by a linear parameterization of the input spectrum, input spectrum constraints can be formulated as LMIs. Other types of constraints are also possible to formulate as Linear Matrix Inequalities (LMI). For further details on this, Jansson and Hjalmarsson (2005) is a good reference.

**Parameterizations of signal spectra**

The spectral density of a stationary signal \(u(t)\) can be written as

\[
\Phi_u(\omega) = \sum_{k=-\infty}^{\infty} c_k B_k(e^{j\omega}),
\]

where the scalar basis functions \(\{B_k(e^{j\omega})\}_{k=0}^{\infty}\) are proper, stable, and rational such that \(B_{-k}(e^{j\omega}) = B_k(e^{-j\omega})\) and \(c_{-k} = c_k\) are real coefficients. One choice of basis functions is \(B_k(e^{j\omega}) = e^{-jk}\). This means that the coefficients correspond to the auto-correlation sequence of the input signal

\[
c_k = \mathbb{E}\left\{u(t)u^T(t-k)\right\},
\]

and hence, (2.19) corresponds to the usual spectrum definition, see for example Söderström (2002).
With (2.19) in (2.11b), $P^{-1}$ is a linear function of the parameters $\{c_k\}$. However, it is not computationally feasible to work with infinitely many coefficients $c_k$. Therefore a finite dimensional parameterization is required. The spectral density is a positive function and thus the coefficients also have to be chosen such that

$$\Phi_u(\omega) \succeq 0, \quad \text{for all } \omega. \quad (2.21)$$

Here the notation $X \succeq 0$ means that the matrix $X$ is positive semi-definite. This is an infinite dimensional constraint and therefore computationally hard to work with. Two suggested parameterization with finite number of coefficients solve these problems. The \textit{Partial correlation parameterization} (Jansson and Hjalmarsson, 2005) considers the parameterization to be a truncation of the full spectrum and requires that sequence of coefficients can be extended to define a spectrum. This approach is exact but the shape of the resulting $\Phi$ depends on the structure of $P^{-1}$. The \textit{Finite dimensional parameterization} requires that the parameterization with finite number of coefficients define a spectrum. This is the parameterization of choice for the remainder of the thesis and further elaborated on next. The parameterization may restrict the set of possible spectra compared to the partial correlation parameterization. However, if frequency by frequency constraints must be imposed, the finite dimensional parameterization is often required.

\textbf{Finite dimensional parameterization}

The sum (2.19) is truncated and the positive real part

$$\Phi_u(\omega) = \Phi_+(\omega) + \Phi_u^+(\omega), \quad (2.22)$$

$$\Phi_+(\omega) = \sum_{k=0}^{m-1} c_k B_k(e^{j\omega}), \quad (2.23)$$

is used to parameterize the spectrum. The constraint (2.21) can be handled by the \textit{KYP-lemma}.

\textbf{Lemma 2.2 (Kalman-Yakubovich-Popov)} Let $\{A, B, C, D\}$ be a controllable state-space form of $\Phi_+(\omega)$. Then there exists a matrix $Q = Q^T \succeq 0$ such that

$$K(Q, \{A, B, C, D\}) \triangleq \begin{bmatrix} Q - A^TQA & -A^TQB \\ -B^TQA & -B^TQB \end{bmatrix} + \begin{bmatrix} 0 & C^T \\ C & D + D^T \end{bmatrix} \succeq 0 \quad (2.24)$$

if and only if

$$\Phi_u(\omega) = \Phi_+(\omega) + \Phi_u^+(\omega) \succeq 0. \quad (2.25)$$
An alternative parameterization is to consider the complete trigonometric polynomial $\Phi_n(\omega)$ instead of only the positive real part. Dumitrescu (2007) presents a trace parameterization of such polynomials which can be used to ensure positivity. For univariate polynomials, with scalar coefficients $\{c_k\}$ this parameterization gives conditions equivalent to the kyp-lemma. The computational complexity of the trace parameterization is lower, since it requires $O(n)$ variables instead of $O(n^2)$ as for the kyp-lemma.

The trace parameterization can also be used for multivariate polynomials or matrix coefficients $\{c_k\}$ but then the polynomials will be restricted to be sum-of-squares. The kyp-lemma has as of now only been generalized to polynomials in two variables (Yang et al., 2008).

2.5 Convex optimization

A convex optimization problem is of the form

$$\begin{align*}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & g_i \leq 0, \quad i = 1, \ldots, m
\end{align*}$$

where the functions $f$ and $g$ are convex. A particular case is the so called Semi Definite Program (SDP) where the objective function $f$ is linear and the constraints $g_i \leq 0$ are given by LMI s. In this context the notion of positive (semi-) definite matrices is important. The generalized inequalities

$$A \succeq 0, \quad A \succ 0,$$

are used to indicate that $A$ is positive definite and semi-definite respectively and

$$A \succeq B \iff A - B \succeq 0.$$

Convex problems have the important property that any local optimal point is globally optimal. There are also efficient solvers for convex optimization problems. This makes these problems attractive both from a theoretical and computational point of view. Boyd and Vandenberghe (2003) provide a very good and thorough treatment of convex optimization.

Optimization software

Many solvers for convex optimization problems exist and are available freely, in particular for SDPs and cone programs. The choice for the problems presented here
has been to use SDPT3 (Toh et al., 1999) simply because it seems to work best for these problems. No in-depth analysis of solvers and their capabilities has been made though.

There are parsers available that simplify setting up the optimization problems in MATLAB. For this purpose, CVX (Grant and Boyd, 2011) was used.

### 2.6 Scenario based optimization

The "scenario approach" is a method for relaxing a semi-infinite convex optimization problem. Such problems have a finite number of decision variables but an infinite number of constraints. This makes these problems computationally hard to handle. Robust optimization is one example where problems are semi-infinite.

Consider the following general robust problem

\[
\begin{align*}
\text{minimize} & \quad c^T \eta \\
\text{subject to} & \quad f(\eta, \delta) \leq 0, \quad \delta \in \Delta.
\end{align*}
\]  

(2.26)

If \( f(\eta, \delta) : \mathbb{R}^n \times \Delta \to \mathbb{R} \) is convex for every \( \delta \in \Delta \), then (2.26) is a robust convex problem.

In this robust optimization formulation, the parameter \( \delta \) is uncertain and the set \( \Delta \) represents this uncertainty. The admissible values of \( \delta \) lie in this set. The interpretation of (2.26) thus becomes that the optimal solution should minimize the objective function while guaranteeing that a set of constraints is satisfied in all possible situations, that is, for all possible values of \( \delta \).

This description of (2.26) involves the satisfaction of an infinite number of constraints, one per each value of \( \delta \in \Delta \). Even though this corresponds to a convex optimization problem, the infinite number of constraints means that it is in general computationally intractable (Ben-Tal and Nemirovski, 1998).

A relaxation of (2.26) is provided by the scenario approach presented in Calafiore and Campi (2006). The basic idea is to select a finite number of the original constraints to include in the optimization problem. To do this, a probabilistic description of the uncertainty is presumed. In other words, a probability distribution \( P_r \) over \( \Delta \). Instances of the uncertain parameter \( \delta \), so called scenarios, are randomly selected according to the probability \( P_r \). The corresponding constraints are then used in the optimization problem. A formal description of the scenario based relaxation follows next.
Extract $N_k$ independent identically distributed samples $\delta^{(1)}, \ldots, \delta^{(N_k)} \in \Delta$ according to the probability $P_r$ and solve the scenario convex program

$$\begin{align*}
\text{minimize} & \quad \epsilon^T \eta \\
\text{subject to} & \quad f(\eta, \delta^{(i)}) \leq 0, \quad i = 1, \ldots, N_k.
\end{align*}$$

(2.27)

The resulting optimization problem (2.27) is a standard finite dimensional convex optimization problem with a finite number of constraints. The computational cost of (2.27) can be quite reasonable, provided that $N_k$ is not too large. The solution to the relaxed problem (2.27) relates to the solution of the original problem (2.26) in the following sense (Calafiore and Campi, 2006):

The scenario-based optimization program (2.27) provides a solution $\eta^*$ which, with probability $1 - \beta$, satisfies all the constraints in $\Delta$, except for a fraction with a small probability $\epsilon$ (with respect to the probability measure $P_r$).

Here $\beta$ can be regarded as a confidence parameter and $\epsilon$ a violation parameter. These variables are user choices which determine the minimum number of scenarios $N$ to be randomly selected.

The minimum number of scenarios $N_k$ required for (2.27) to give a reasonable approximation of the solution to (2.26) in the sense stated above has been studied in several publications, see for example Alamo et al. (2007, 2008); Calafiore and Campi (2006); Campi and Garatti (2007). To date, the tightest bound on the minimum $N_k$ has been established in Campi and Garatti (2007). The bound states that the number of scenarios $N_k$ has to satisfy

$$\sum_{i=0}^{n-1} \binom{N_k}{i} \epsilon^i (1 - \epsilon)^{N_k-i} \leq \beta.$$  

(2.28)
Chapter 3

Experiment design for model predictive control

“Prediction is very difficult, especially about the future.”
Niels Bohr

Using a model of a system for controller design puts high demand on model quality. A general framework for obtaining application specific models from experimental data was outlined in Chapter 2. In this chapter, the framework is applied and extended to the particulars of MPC. The input design scheme is combined with the prediction error method into an identification method for MPC models. The method can be used as a one shot identification or as an iterative scheme with the possibility of improving the input design in each iteration. Challenges related to input design for MPC are identified and discussed.

A major advantage of MPC is the ability to handle signal and state constraints on the process in the controller. This, however, leads to that there is no explicit solution to the optimization problem in the controller (Maciejowski, 2002). We will see that this is a limiting factor in the experiment design and requires numerical calculations.

There are two major challenges with the practical implementation of the method. The first is the fact that the optimal input design relies on knowledge of the true system parameters. These are, of course, not known at the time of input design. The two proposed ways around this are to design inputs that are robust to parameter variations...
Experiment design for model predictive control

For example, Rojas (2008), or to use an initial parameter estimate instead of the true parameters in the optimal input design. The latter approach will be considered here.

The second challenge relates to the use of time domain constraints in the MPC. There is, as of yet, no good way of including such constraints in the input design formulation that is considered here. The solution here is to include them in the calculation of the application cost but not to consider them in the identification part of the method. It may be possible to enforce some time domain constraints when the optimal signal is generated. Another idea is proposed in where the input design is first optimized under power constraints and then iteratively optimized under amplitude constraints. One could also design the input signal directly in the time domain as is done in Manchester (2010).

3.1 System description

The considered systems are linear, time-invariant and stable systems given by the state space description

\[ S : \begin{align*} x(t+1) &= Ax(t) + Bu(t) + v(t), \\ y(t) &=Cx(t) + w(t). \end{align*} \tag{3.1} \]

Here \( x(t) \in \mathbb{R}^n \) is the state vector, \( u(t) \in \mathbb{R}^m \) the controlled input signal, \( y(t) \in \mathbb{R}^p \) the measured output, and \( v(t) \) and \( w(t) \) are stationary, zero-mean, white processes commensurate with \( x(t) \) and \( y(t) \). Their covariance matrices are \( \Lambda_v \) and \( \Lambda_w \) respectively.

The process is assumed to be controlled using MPC with a finite horizon quadratic cost function of the form

\[ J(t) = \sum_{i=0}^{N_r} \| \hat{y}(t+i|t) - r(t+i) \|_Q^2 + \sum_{i=0}^{N_c} \| \Delta u(t+i+1|t) \|_R^2. \tag{3.2} \]

Here, \( \hat{y}(t+i|t) \) is the \( i \)-step prediction of the output and \( \Delta u(t+i+1|t) \) the input update. The reference trajectory is denoted \( r(t) \) and assumed to be known over the prediction horizon.

The system can be constrained in terms of inputs and outputs (or states). If the prediction horizon is longer than the control horizon, the input is set to be constant for the time instants after the control horizon. At time \( t \) the input is found from the
3.1 System description

The optimization problem

\[
\begin{align*}
\text{minimize} & \quad J(t) \\
\text{subject to} & \quad \hat{y}(t + i) < y_{\text{lim}}(t + i), \quad i = 1, \ldots, N_y \\
& \quad u(t + i) < u_{\text{lim}}, \quad i = 1, \ldots, N_u \\
& \quad u(t + i) = u(t + N_u), \quad i = N_u + 1, \ldots, N_y
\end{align*}
\] (3.3)

The cost function is minimized with respect to the whole input sequence \( U(t) = \{u(t), \ldots, u(t + N_u)\} \) but only the input \( u(t) \) is actually applied to the system. The optimization is performed in each time step in accordance with the receding horizon control philosophy.

The response of the system is predicted using the model of the system in the MPC. These predictions are calculated as

\[
\hat{y}(t + k|t) = CA^k x(t) + \sum_{i=0}^{k-1} A^{k-i-1} Bu(t + i).
\] (3.4)

To perform these calculations, the state of the system is needed. If these are not available for measurement or if only part of the state is available, a state estimator must be used. It is assumed that the state estimates come from a Kalman filter with the same plant model as is used by the MPC. The complete plant and controller setup is illustrated in Figure 3.1.

To find the output predictions that are used in the optimization — possibly also for state estimation — a model of the process is needed. When an accurate model is used,
Experiment design for model predictive control

The performance of the MPC can be expected to be good. A model with low accuracy, on the other hand, can be expected to give worse performance.

Example 3.1 (System and MPC)
Reference tracking for the system
\[
\begin{align*}
  x(t + 1) &= \theta_2^0 x(t) + u(t), \\
  y(t) &= \theta_1^0 x(t) + e(t),
\end{align*}
\]

is implemented using the MPC (3.3) with
\[
J(t) = \sum_{i=0}^{N_f} \| \hat{y}(t + i) - r(t + i) \|^2
\]
and \( N_f = N_y = 5 \). The parameters are \( \theta^0 = [0.6 \ 0.9]^T \). To simplify comparison, the noise variance is set to zero, that is \( \lambda_e \), so that the MPC has full state information. The reference trajectory is a series of unit steps over 10 samples. The resulting inputs and outputs for the two cases \( u_{\text{lim}} = \infty \) and \( u_{\text{lim}} = 1 \) are shown in Figure 3.2.

Figure 3.2 The resulting outputs and inputs for the MPC setup in Example 3.1. Unconstrained inputs (---), constrained inputs (--- - -) and reference trajectory (-----). The system is considerably slower with constraints than without.
3.2 Identification setup

The system is modeled by

\[
\mathcal{M}(\theta) : \begin{align*}
\dot{x}(t+1, \theta) &= A(\theta)x(t, \theta) + B(\theta)u(t) + K(\theta)e(t), \\
y(t, \theta) &= C(\theta)x(t, \theta) + e(t).
\end{align*}
\] (3.5)

It is assumed that there exists a parameter vector \( \theta^o \) such that the model (3.5) describes the true system, denoted \( S \). All identification experiments are performed on plants operating in open loop. The parameter estimates are obtained using \( \text{pem} \) with quadratic cost, hence the estimates have the asymptotic distribution in (2.9).

3.3 Application cost

The purpose of the application cost is twofold. Firstly, it relates the model accuracy to the performance of the plant; any parameter vector can be directly related to the expected performance of the control system. Secondly, and more importantly, it shows in which directions in the parameters space the performance is sensitive to parameter variations. These directions are important to model well for good performance and it becomes the objective of the input design to find inputs that excite the plant in a suitable way.

An application cost should compare the performance that is achieved using the estimated model to what would have been obtained if the true system was known. However, since \( \text{mpc} \) with constraints is not a linear time-invariant controller, this option is not feasible. Another possibility is to compare the resulting output trajectory of the system with a perfect model to the output trajectory with the estimated model. This idea will be used for the \( \text{mpc} \) situation.

A simple, albeit reasonable, choice for the application cost is to use

\[
V_{\text{app}}(\theta) = \frac{1}{N} \sum_{t=1}^{N} \| y(t, \theta^o) - y(t, \theta) \|, \quad (3.6)
\]

where \( y(t, \theta^o) \) is the output using parameters that corresponds to the true system and \( y(t, \theta) \) is the output when other parameters are used. This is an application cost according to Definition 2.1. This choice makes sense if there is some nominal trajectory that one wants the system to follow.

Remark 3.1 Even though any function that fulfills Definition 2.1 can be used as an application cost, it is important that the choice reflects the intended application. If this
is not the case, the input design based on the used application cost might give model properties that are not useful.

An acceptable performance degradation together with the application cost defines the application set. This upper limit is highly application dependent. A possible choice when (3.6) is used is

$$\gamma = \lambda \sum_{t=0}^{N-1} \| y(t, \theta^0) \|^2, \quad \lambda > 0 \in \mathbb{R}. \quad (3.7)$$

This relates the degradation to what is possible to achieve with a perfect model and is used to set the level. For example, $\lambda = 100$ gives a $1\%$ performance loss.

**Approximation of the application set**

The application set given by $V_{app}$ and $\gamma$ typically contains an infinite number of parameters. There is also no guarantee that the set is convex. Two approximate descriptions of the application set are presented here. One based on a convex, ellipsoidal approximation, the other on the scenario approach.

**Ellipsoidal approximation**

The application cost can be approximated around the true parameters $\theta^0$ using the Taylor series expansion

$$V_{app}(\theta) \approx V_{app}(\theta^0) + [\theta - \theta^0]^T V'_{app}(\theta^0) + \frac{1}{2} [\theta - \theta^0]^T V''_{app}(\theta^0) [\theta - \theta^0] \quad (3.8)$$

$$= \frac{1}{2} [\theta - \theta^0]^T V''_{app}(\theta^0) [\theta - \theta^0]. \quad (3.9)$$

The last equality holds because of the properties of the application cost, see Definition 2.1. Consequently, the inequality that defines the application set can be approximated by

$$[\theta - \theta^0]^T V''_{app}(\theta^0) [\theta - \theta^0] \leq \frac{2}{\gamma}, \quad (3.10)$$

which in turn means that the application set can be approximated by the ellipsoid

$$\Theta_{app}(\gamma) \approx \mathcal{E}_{app} = \left\{ \theta : [\theta - \theta^0]^T V''_{app}(\theta^0) [\theta - \theta^0] \leq \frac{2}{\gamma} \right\}. \quad (3.11)$$

The approximate set $\mathcal{E}_{app}$ will be called the *application ellipsoid*. The approximation error depends both on the application cost and the parameter $\gamma$. For large values of $\gamma$ the approximation will be better than for small values.
3.3. Application cost

\[ \theta_1 \]

\[ \theta_2 \]

\[ \theta_1 \]

\[ \theta_2 \]

(a) Unconstrained input, \( u_{\text{lim}} = \infty \)

(b) Constrained input, \( u_{\text{lim}} = 1 \)

Figure 3.3 Level curves (-----) for the application cost of the system in Example 3.2. The innermost level curve corresponds to the required accuracy \( \gamma \). The ellipsoidal approximation (-----) is much better in the unconstrained than in the constrained case. Scenarios for both cases are also shown (\( \gamma \)).

Remark 3.2 For low performance demands, that is for low values of \( \gamma \), the second order approximation may not be very good. An alternative could be to consider higher order expansions. This approach is investigated in Hjalmarsson and Egebrand (2011) where it is shown that this leads to problems that are polynomial in the coefficients of the spectrum parameterization. It is also shown by example that the difference between the ellipsoidal approximation and higher order approximations can be quite large for low \( \gamma \).

Scenario based approximation

Larsson et al. (2011a) introduced the scenario based approach to the applications oriented optimal input design formulation. The central idea to randomly select parameters, so called scenarios, from \( \Theta_{\text{app}}(\gamma) \) to represent the set. If enough scenarios are used, a good enough approximation of the original set is obtained. The application set is simply replaced by a set of samples

\[ \left\{ \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(N_k)} \right\}, \]

(3.12)

taken from a uniform distribution over \( \Theta_{\text{app}}(\gamma) \).
Example 3.2 (Application set)
Consider the system from Example 3.1 and introduce the application cost and set

\[ V_{\text{app}}(\theta) = \frac{1}{10} \sum_{t=1}^{10} \| y(t, \theta) - y(t, \theta^*) \|^2, \]  \hspace{1cm} (3.13)

\[ \Theta_{\text{app}}(\gamma) = \left\{ \theta : V_{\text{app}}(\theta) \leq \frac{1}{100} \right\}. \] \hspace{1cm} (3.14)

That is, the required accuracy is \( \gamma = 100 \). The level curves together with the ellipsoidal approximation of the application set and a number of scenarios, uniformly distributed over \( \Theta_{\text{app}}(\gamma) \) are shown in Figure 3.3 for the two different constraints \( u_{\text{lim}} = \infty \) and \( u_{\text{lim}} = 1 \). It is seen that imposing constraints on the input enlarges the application set. However, the contours of \( V_{\text{app}}(\theta) \) are closer to ellipses in the unconstrained case.

Remark 3.3 Example 3.2 shows that the application set grows when the inputs are constrained. Input constrains make the system slower, which is related to lowering the bandwidth of the system. This can be expected to reduce the requirements on the model.

Remark 3.4 In practice, the shape and size of \( \Theta_{\text{app}}(\gamma) \) may not be known in advance, hence sampling from the set can be difficult. In this case the Hessian of \( V_{\text{app}}(\theta) \) could be used as guidance when choosing the uniform distribution.

Approximate application cost
The constrained optimization problem in the MPC lacks a closed form solution. Consequently, there is no closed form solution to the expression for the application cost (3.6), and numerical evaluation becomes unavoidable.

In an application, it is unlikely that one can evaluate (3.6) using outputs from the real process, as this would require controlling the process based on models with more or less arbitrary parameter values. Instead an approximation of \( V_{\text{app}} \) where the true system is replaced with the linear model using estimated parameter values is introduced. This gives

\[ \tilde{V}_{\text{app}}(\theta, \hat{\theta}) = \frac{1}{N} \sum_{t=1}^{N} \| y(t, \hat{\theta}, \hat{\theta}) - y(t, \theta, \hat{\theta}) \|, \] \hspace{1cm} (3.15)
where the second argument of $\gamma$ is the parameter used by the $\text{MPC}$ and the third argument the parameter playing the role of the true parameter vector in a simulations-based experiment.

To obtain the corresponding approximations when the approximate application cost is used, $V_{\text{app}}(\theta)$ can simply be replaced by $\tilde{V}_{\text{app}}(\theta, \hat{\theta})$ in the relevant expressions.

### 3.4 Input design

The optimal input design problem considered for the $\text{MPC}$ case is to minimize some experimental cost while guaranteeing that the identified model will give the required performance. The motivation for this formulation is that in a practical situation it is often desirable to have identification experiments that disturb normal operation as little as possible. In Bombois et al. (2006) three situations are described together with relevant measures of the experimental cost. These are

**Situation 1** The input power consumed by the experiment and/or the resulting output power are the important factors.

**Situation 2** The length of the experiment is the important factor.

**Situation 3** Both power and time, that is the input and/or output energy, are important factors.

All three situations have relevance to the $\text{MPC}$ case that is considered here. However, from (2.11b) it can be seen that time, in terms of number of samples $N$, and the input spectrum $\Phi_n(\omega)$ enter in the same way in the expression for the inverse covariance matrix. Consequently, if the input design formulation for, say situation 1, is found, the same formulation can be used in situation 2 *mutatis mutandis*. There is in principle no difference in the problem setup when considering the input or output power or both; the solutions will, however, be different.

The input design problem considered here is situation 1 with only input power as the experimental cost. The problem can formally be stated as

$$
\begin{align*}
\text{minimize} & \quad \Phi_n(\omega) \text{ trace } \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_n(\omega) d\omega \\
\text{subject to} & \quad U(\alpha) \subseteq \Theta_{\text{app}} \\
& \quad \Phi_n(\omega) \geq 0 \quad \forall \omega.
\end{align*}
$$

(3.16)
The input is assumed to be generated by an \( \text{FIR-filter} \). The input spectral density can hence be parameterized — for an \( M \)th order filter — as
\[
\Phi_u(\omega) = \sum_{k=-(M-1)}^{M-1} c_k e^{j\omega k},
\] (3.17)
where \( c_k \in \mathbb{R}_{m \times m} \). The positive real part of (3.17) has a controllable state space realization given by
\[
A = \begin{bmatrix} 0_{m \times M} & 0_{m \times m} \\ I_{M \times m} & 0_{m \times m} \end{bmatrix}, \quad B = \begin{bmatrix} I_{m \times m} & 0_{m \times M} \end{bmatrix}, \\
C = \begin{bmatrix} c_1 & c_2 & \cdots & c_{M-1} \end{bmatrix}, \quad D = \frac{1}{2} c_0.
\]

Based on the two alternative approximations of the application set and the \( \text{kYP-lemma} \), two different relaxations of (3.16) can be formulated.

**Ellipsoidal relaxation**

Using expression (2.10) for the confidence ellipsoid and the ellipsoidal approximation (3.11) of the application cost, (3.16) can be relaxed to
\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega) \, d\omega \\
\text{subject to} & \quad \Phi_u(\omega) = \sum_{k=-(M-1)}^{M-1} c_k e^{j\omega k} \\
& \quad K \left( Q, \{ A, B, C, D \} \right) \succeq 0 \\
& \quad \frac{N}{\kappa} P^{-1} \succeq \gamma V_{\text{app}}(\theta^0).
\end{align*}
\] (3.18)
Table 3.1 The required input power to give accuracy $\gamma = 100$ for the problem in Example 3.3 for the optimal designs and a white input.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Ellipsoid</th>
<th>Scenario</th>
<th>White</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconstrained system</td>
<td>0.27</td>
<td>0.31</td>
<td>2.5</td>
</tr>
<tr>
<td>Constrained system</td>
<td>0.4</td>
<td>0.5</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Scenario based relaxation

The optimization problem (3.16) can be seen as a robust convex optimization problem with infinitely many constraints which can be relaxed using the scenario approach. Using expression (2.10) for the confidence ellipsoid and $N_k$ scenarios drawn uniformly from $\Theta_{app}(\gamma)$ gives the relaxation

$$\min_{\{c_k\}_{k=0}} \quad \text{trace} \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega) d\omega \right\}$$

subject to

$$\Phi_u(\omega) = \sum_{k=-(M-1)}^{M-1} c_k e^{i\omega k}$$

$$K (Q, \{A, B, C, D\}) \succeq 0$$

$$[\theta^{(l)} - \theta^0] \gamma \frac{N}{\kappa} P^{-1} [\theta^{(l)} - \theta^0] \geq \gamma V_{app}(\theta^{(l)}),$$

$$l = 1, \ldots, N_k.$$

Example 3.3 (Input design)

Using the applications sets from Example 3.2 input designs are generated. The design setup is $M = 40$, $N = 100$, $\gamma = 100$, $\alpha = 0.95$ and $\lambda_\nu = 1$. The resulting application ellipsoids for the optimal input designs are shown in Figure 3.4 and the optimal input spectra are shown in Figure 3.5. For comparison the white input spectrum that gives the same accuracy is calculated. Also the D-optimal design with the same input variance as the optimal design is found. The latter designs have high input power in frequency bands where the optimal spectra are low. This power is in principle wasted on modeling parts of the system that are not important for the application. The required input power for the different designs is presented in Table 3.1.

The input design problem can be implemented in the MATLAB toolbox Moose as

```
A = 0.9; Aparam = 1;
B = 1; Bparam = 0;
```
C = 0.6; Cparam = 1;
D = 0; Dparam = 0;
le = 1; Ts = 1;

mooseBegin
objective minimize(inputPower)
model A(Aparam) B(Bparam) C(Cparam) K(Kparam) le Ts
identification constraints
    spectrum FIR(40)
    probability \(0.95\)
    numSamples 100
application constraints
    ellipsoid(VappBiss,100)
mooseEnd

To use the scenario approach, ellipsoid should be changed to scenarios.
3.4. Input design

Figure 3.4 The level curves of the application cost (---), the inner most curve corresponds to the desired accuracy. The resulting confidence ellipsoids when the ellipsoidal relaxation (----) and when the scenario approach relaxation (-----) is used are shown. The scenario approach requires more energy and therefore comes closer to lying inside the application set.

Figure 3.5 The optimal input spectra for the different relaxations, ellipsoidal relaxation (---) and the scenario approach relaxation (-----). Both relaxations give similar results. For comparison the white spectrum (-----) that gives equal accuracy and the D-optimal spectrum (-----) with equal input variance are shown. The frequency response of the system is also shown (-----).
3.5 Identification algorithm

The complete application oriented identification method is presented here. The algorithm can be used once or can be iterated to improve the input design. The method is also presented in Figure 3.6.

Algorithm

**Step 0** Set the initial input spectrum to be flat, with equal power at all frequencies, to obtain a white noise input sequence in the first identification experiment.

**Step 1** Generate an input corresponding to the desired input spectrum.

**Step 2** Estimate system parameters in identification experiment.

**Step 3** Find the application cost based on simulations of the model with the parameter estimates.

**Step 4** Design the optimal input signal based on the application cost and parameter estimates.

**Step 5** Find a new estimate of the model parameters using the optimal input signal in the system identification experiment.

Remark 3.5 If a good initial guess of the parameters is available, for example through physical insight of the process, this guess can replace the initial estimation in Step 0.

The algorithm can be iterated so that the estimate from Step 5 is used in Steps 1 and 2 to calculate a new input design. As more and more data are used in the identification step and if there exist parameters $\theta^o$ such that $S = M(\theta^o)$, the estimates will converge to their true values. Therefore, one can expect the input design to converge to what would be obtained had $\theta^o$ been known. A discussion on this and a formal proof for the case with ARX systems are found in Gerencsér and Hjalmarsson (2009).
3.5. Identification algorithm

Figure 3.6 The method for system identification with optimal input design for MPC. The initial identification is performed with an input signal spectrum $\Phi^o_u$. The system $S$ is approximated by the estimated model $M(\hat{\theta})$ to calculate the application cost in simulations (grey). The resulting input design is optimal for $M(\hat{\theta})$ but may not be optimal for $S$. 

\[ \Phi^o_u \rightarrow \text{Signal generation} \]

\[ \Phi^o_u \rightarrow \text{MPC}(\hat{\theta}) \]

\[ \Phi^o_u \rightarrow \text{Input design} \]
Example 3.4 (Full identification)
The identification algorithm is implemented on the system from Example 3.1. An initial estimate of the system is obtained in an identification with white noise with variance 0.01 as input. Based on the initial model, an optimal input design is found. This design is used to estimate 100 models in Monte Carlo simulations. The resulting estimates and the system identification confidence ellipse are presented in Figure 3.7 for the unconstrained system and in Figure 3.8 for the constrained system.

3.6 Signal generation

The solution to the input design problem (3.16) is an optimal input spectrum. To be useful in practice, a time domain realization corresponding to the spectrum is required. Typically this is done by filtering Gaussian white noise through a spectral factor of the spectrum. Assuming that the spectrum is rational and nonsingular, by spectral factorization the optimal input spectrum can be factorized as (Söderström, 2002)

\[ \Phi_s(\omega) = H(\omega) \Lambda H^H(\omega). \]  

(3.20)

If the input is realized as \( u(t) = H(q) \Lambda^{1/2} e(t) \) where \( e(t) \) is white noise with unit variance, \( u(t) \) will have the right spectral characteristics. However, this solution cannot impose constraints on the inputs.

Input constraints

If there are physical constraints on the plant or constraints that cannot be disregarded for other reasons, such as safety, the input that is applied to the plant, also during identification, must obey these constraints.

Let the input be constrained to the interval \([-u_{\text{lim}}, u_{\text{lim}}]\). One possibility is to use a binary input

\[ u(t) = \pm u_{\text{lim}}. \]  

(3.21)

There are many proposed methods to realize such a signal, see for example (Liu and Munson, 1982; Boufounos, 2007). A receding horizon approach is presented in Rojas et al. (2007). The algorithm is shown to converge for pseudo white noise sequences. Another possibility is to use the result for the auto-correlation function of a "clipped" Gaussian process found in Hannan (1970). The result is presented here in Theorem 3.1.
3.6. Signal generation

Figure 3.7 Resulting input design for the unconstrained system in Example 3.1 using the ellipsoidal relaxation. The application ellipsoid from the initial estimate (——) resembles the true application ellipse (⋯⋯). The optimal confidence ellipsoid (⋯⋯⋯⋯) is almost contained within the true application ellipse.

Figure 3.8 Resulting input design for the constrained system in Example 3.1 using the ellipsoidal relaxation. The application ellipsoid from the initial estimate (——) does not match the true application ellipse (⋯⋯⋯⋯). However, the important semi-axis has the right size. The optimal confidence ellipsoid (⋯⋯⋯⋯⋯⋯) is almost contained within the true application ellipse.
Theorem 3.1 Let \( \{x_n\} \) be a strictly stationary Gaussian process with \( E\{x_n\} = 0 \), \( E\{x_n^2\} = 1 \) and auto-correlation \( E\{x_{n+\tau}x_n\} \). Then the auto-correlation of the process \( u_n = \text{sgn} \ x_n \) is given by

\[
E\{u_{n+\tau}u_n\} = \frac{2}{\pi} \arcsin E\{x_{n+\tau}x_n\}.
\]

(3.22)

It is however known that the covariance sequence of a binary signal is limited (de Carvalho and Clark, 1983) and hence the obtained signal spectrum can be suboptimal.

Output constraints

When output constraints are present, it may be necessary to take such constraints into account during the identification process. One possibility would be a sort of back-of-mechanism that monitors outputs and reduces input power if constraints are close to being violated.

3.7 Identifiability issues

In this chapter it was assumed that the MPC uses a discrete time model of the controlled plant. When there is some physical insight into the properties of the plant, this knowledge can be used to reduce the number of free parameters in the model. Physical modeling, however, typically leads to continuous time models which then have to be discretized to be used in the MPC. This can give identifiability problems and it may not be straightforward to investigate the identifiability of the chosen model. This is discussed in Ljung (1999) and will not be further elaborated on here. Instead, possible consequences of loss of identifiability for the input design problems are investigated.

If the model is such that the inverse covariance matrix does not have full rank, it may be impossible to find a design that satisfies the original input design formulation. A singular inverse covariance matrix has at least one eigenvalue that is zero, say \( \lambda_0 \) with eigenvector \( v_0 \). Consequently, the corresponding confidence ellipsoid degenerates and is infinitely extended in the direction of the eigenvector \( v_0 \). This ellipsoid can never lie inside an application set that is not infinitely extended in the same direction. This is illustrated in Example 3.5.
3.7. Identifiability issues

Figure 3.9 Level curves for the application cost (solid) and the confidence ellipsoids of the parameter estimates (dash dotted) for two different application cost functions. In (a) it is not possible to satisfy the constraint $U(\alpha) \in \Theta_{app}(\gamma)$ with any input, while in (b) it is.

Example 3.5 (Identifiability)

This example shows how infeasibility in the input design can arise from identifiability problems in the chosen model structure. Consider a system on the form

$$x(t + 1) = \theta_1 u(t),$$
$$y(t) = \theta_2 x(t) + e(t).$$

This can be seen as a simplified version of a system where the states depend on the input through some conversion, such as voltage to torque in a DC-motor, and the output is measured with an instrument that also has some conversion, such as angle to voltage.

The inverse covariance matrix is given by

$$P^{-1} = \begin{bmatrix} (\theta_2^\alpha)^2 & \theta_2^\alpha \theta_1^\alpha \\ \theta_1^\alpha \theta_2^\alpha & (\theta_1^\alpha)^2 \end{bmatrix}, \quad \text{rank } P^{-1} = 1.$$

Now consider two different application requirements, the first one where it is desirable to keep the physical interpretation of $\theta_1$ and $\theta_2$ and the second one where only the input–output relationship is important. Possible application costs for the two cases are

$$V_1(\theta) = ||\theta^\alpha - \theta||^2,$$
$$V_2(\theta) = ||\theta_1^\alpha \theta_2^\alpha - \theta_1 \theta_2||^2,$$
$$V_1''(\theta^\alpha) = 2I,$$
$$V_2''(\theta^\alpha) = \begin{bmatrix} (\theta_2^\alpha)^2 & \theta_2^\alpha \theta_1^\alpha \\ \theta_1^\alpha \theta_2^\alpha & (\theta_1^\alpha)^2 \end{bmatrix}.$$
where $V_1$ corresponds to case one and $V_2$ to case two. Level curves of ellipsoidal approximations of the two application costs together with the confidence ellipsoids for the estimated parameters are shown in Figure 3.9. It is clear that in case two it is possible to find an input design while in case one it is not.

Remark 3.6 Trying to identify the two parameters in Example 3.5 is a bad choice since the two parameters are obviously not individually identifiable. However, in large state-space models these types of products between parameters can appear in the predictor even if it is not obvious from the state space formulations. Furthermore, if one wants to retain a physical interpretation of the model parameters, it may not be possible to avoid such products.

Remark 3.7 When the ellipsoidal approximation of the application set is used, the situation described in case 1 in Example 3.5 leads to a relaxed optimization problem that is infeasible. If on the other hand the scenario approach is used, the relaxed problem can still be feasible, the resulting solution is however not sensible. Therefore, one should be careful when using the scenario approach in such cases.

3.8 Linear models of nonlinear systems

The presented framework assumes the true system to be linear and contained in the model class, which consists only of linear systems. Whenever the true system is nonlinear, this assumption does not hold. It may however still be useful to find approximate linear models of nonlinear systems. The properties of such models are studied in detail in, for example Enqvist (2005). Some aspects of input design for identification of linear models of nonlinear systems are discussed here.

Consider an optimal input design $\Phi_u^*(\omega)$ for a certain $N_1$, $\gamma$ and $\kappa$. From the optimization problem (3.16) it is seen that for a fixed $N$, changing $\gamma$, say to $10\gamma$, results in an optimal design $10\Phi_u^*(\omega)$. If on the other hand $\Phi_u^*(\omega)$ is fixed, $10N$ samples are needed to increase the accuracy. For linear systems, changing one or the other does not matter. For nonlinear systems this need not be true. This is illustrated by Example 3.6.
Example 3.6 (Linear model of nonlinear system)
Consider the static nonlinear system
\[ y(t) = \theta^o \arctan u(t) + e(t), \]  
(3.23)
and the linear model
\[ y(t) = \theta u(t) + e(t). \]  
(3.24)
Here \( u(t) \) is a known input and \( e(t) \) is Gaussian white noise with variance \( \lambda_e = 1 \). Introduce the application set
\[ \Theta_{app} = \left\{ \theta : |\theta - \theta^o|^2 < \frac{1}{\gamma} \right\}. \]  
(3.25)
The solution to (3.16) a constant input with amplitude
\[ A^* = \sqrt{\frac{2\gamma \kappa}{N}}. \]  
(3.26)
The accuracy \( \gamma \) is increased by increasing \( N \) and by increasing \( A \) in different experiments. The percentages of identified models that satisfy the application requirements from the two experiments are shown in Figure 3.10.
3.9 Summary

Model predictive control has rapidly become a very popular control strategy widely used in industry. Therefore, tools for finding good process models to be used in MPC are now an important issue. This chapter presented a strategy for experiment design for identification when the goal is to use the model in MPC applications. The ideas from the applications oriented system identification framework in Chapter 2 have been tailored to the specifics of the MPC case.

The lack of explicit solutions to the MPC criterion leads to an application cost without explicit solutions. Hence, numerical calculations are required for evaluation. To avoid using the true system in such evaluations, an approximate application cost was introduced.

An optimization problem in terms of the input spectral density for the input design formulation was presented. For efficient solutions, two convex relaxations of the initial problem were given. The first is based on ellipsoidal approximations of the application set. The second is an application of the scenario approach.

The method has two shortcomings. The first is the fact that the input and output constraints are not directly included in the input design. However, as pointed out, there are methods to include them later in the signal generation phase. The second is the memory requirements for the optimization problem. The size of the problem grows quite fast with the dimension of the process model. Exploiting the structure of the model, such as exploiting sparsity in matrices, would be a way to reduce this problem.
Chapter 4

Experimental study: A water tank process

“A good simulation, be it a religious myth or scientific theory, gives us a sense of mastery over experience. To represent something symbolically, as we do when we speak or write, is somehow to capture it, thus making it one’s own. But with this appropriation comes the realization that we have denied the immediacy of reality and that in creating a substitute we have but spun another thread in the web of our grand illusion.”

Heinz R. Pagels

The method developed for optimal input design and system identification is evaluated in this chapter on a process of four interconnected water tanks. This process is well understood and easily modeled from first order principles. Furthermore, the process is nonlinear but well modeled as a linear process close to an operating point. This makes it suitable for experimental validation of the developed method.

Evaluation has been made both in simulations and in experiments on a real process. The simulations provide a situation closer to theory as it allows for a linear model of the water tanks to represent the system. The experiments on the real system provide a situation closer to reality as effects of nonlinearities and other practical considerations have to be taken into account.
Table 4.1 The physical constraints of the four tank process.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_i,\text{max}$</td>
<td>25 cm</td>
<td>maximum water level of tank $i$</td>
</tr>
<tr>
<td>$h_i,\text{min}$</td>
<td>0 cm</td>
<td>minimum water level of tank $i$</td>
</tr>
<tr>
<td>$u_j,\text{max}$</td>
<td>5 V</td>
<td>maximum voltage of pump $j$</td>
</tr>
<tr>
<td>$u_j,\text{min}$</td>
<td>0 V</td>
<td>minimum voltage of pump $j$</td>
</tr>
</tbody>
</table>

All identification in this chapter has been performed using the grey box capabilities of the System Identification toolbox (Ljung, 2010) in MATLAB. The input design problems were not solved using Moose (see Appendix A) because of lack of grey box support in version 1.0.

4.1 Process description

The considered process is shown in Figure 4.1. The main components of the considered process are two lower tanks (1 and 2), two upper tanks (3 and 4) and two pumps. The pumps are connected so that pump 1 delivers water into tanks 1 and 4 while pump 2 delivers water to tanks 2 and 3. The fraction of the water flow that is delivered to the upper and lower tank respectively can be changed and determines whether the process is minimum phase or not. This is represented by the two valves.

There are pressure sensors at the bottom of each tank. The signals from the sensors in the two lower tanks, tanks 1 and 2, are the output signals of the process. The input signals are the voltages applied to the two pumps. There are a number of physical constraints on the process, such as the maximum input voltages to the pumps and the possible water levels in the tanks. These constraints are presented in Table 4.1.

Nonlinear model

The four tank process can be accurately described using a system of nonlinear differential equations. Some simplifying assumptions are made:

- Water is used and therefore all flows are incompressible.
- All tanks have the same cross sectional areas.
- The dynamics of the pumps are much faster than those of the tanks and are neglected.
4.1. Process description

Figure 4.1 The four tank process. Water is pumped from the reservoir into the four tanks. The flow from pump 1 fills tanks 1 and 3 while the flow from pump 2 fills tanks 2 and 4. The flow is divided between the tanks according to the settings of the two valves, \( \gamma_1 \) and \( \gamma_2 \). The water levels of the tanks are \( x_i, \ i = 1, 2, 3, 4 \). The inputs to the process are \( u_1 \) and \( u_2 \).

The change in volume \( V \) of water in a tank per unit time is the difference in volumetric flow rate into, \( q_{\text{in}} \), and out of, \( q_{\text{out}} \), the tank, that is

\[
\frac{dV}{dt} = q_{\text{in}} - q_{\text{out}}. \tag{4.1}
\]

The flow out of a pump is proportional to the applied voltage, that is \( q = ku \), and this flow is divided between two tanks according to

\[
q_1 = \gamma ku, \quad q_2 = (1 - \gamma) ku, \quad \gamma \in [0, 1]. \tag{4.2}
\]

The flow out of a tank can be found using Torricelli’s principle and the cross sectional area of the outlet hole of the tank, \( a \). This gives

\[
q_{\text{out}} = a \sqrt{2gx},
\]
Experimental study: A water tank process

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>Cross sectional area of outlet of tank $i$</td>
</tr>
<tr>
<td>$\gamma_i$</td>
<td>Fraction of flow from pump $i$ pumped to lower tank</td>
</tr>
<tr>
<td>$k_i$</td>
<td>Voltage to volumetric flow rate proportionality constant of pump $i$</td>
</tr>
<tr>
<td>$A$</td>
<td>Cross sectional area of the four tanks</td>
</tr>
<tr>
<td>$l_i$</td>
<td>Water level to voltage proportionality constant of sensor $i$</td>
</tr>
</tbody>
</table>

where $g$ is the gravitational acceleration and $x$ the height of the water level in the tank. The water levels in the two lower tanks, $x_1$ and $x_2$, are measured by pressure sensors. The sensor readings are proportional to the water levels and noisy. The conversion factor for the pressure sensors is known to be $l_1 = l_2 = 5 \text{ cm/V}$. The measurement noise is modeled as zero mean Gaussian noise that is uncorrelated between the two measurement channels. All together this gives the model

\[
\frac{dx_1}{dt} = -\frac{a_1}{A} \sqrt{2gx_1(t)} + \frac{a_3}{A} \sqrt{2gx_3(t)} + \frac{\gamma_1 k_1}{A} u_1(t), \\
\frac{dx_2}{dt} = -\frac{a_2}{A} \sqrt{2gx_2(t)} + \frac{a_4}{A} \sqrt{2gx_4(t)} + \frac{\gamma_2 k_2}{A} u_2(t), \\
\frac{dx_3}{dt} = -\frac{a_3}{A} \sqrt{2gx_3(t)} + \frac{(1 - \gamma_2) k_2}{A} u_2(t), \\
\frac{dx_4}{dt} = -\frac{a_4}{A} \sqrt{2gx_4(t)} + \frac{(1 - \gamma_1) k_1}{A} u_1(t), \\
y_1(t) = l_1 x_1(t) + e_1(t), \\
y_2(t) = l_2 x_2(t) + e_2(t). 
\]

The physical parameters used in (4.3a) are explained in Table 4.2.

Linear model

The MPC uses a discrete time, linear model of the process of the form

\[
x(t + 1) = Ax(t) + Bu(t), \\
y(t) = Cx(t) + e(t). 
\]
Therefore (4.3a) is linearized around a working point $x^0$ and $u^0$ and discretized. Linearization using first order Taylor expansions gives

$$A_c = \begin{bmatrix} -\tau_1 & 0 & \tau_3 & 0 \\ 0 & -\tau_2 & 0 & \tau_4 \\ 0 & 0 & -\tau_3 & 0 \\ 0 & 0 & 0 & -\tau_4 \end{bmatrix}, \quad B_c = \begin{bmatrix} \frac{k_1 \gamma_1}{A} & 0 \\ 0 & \frac{k_2 \gamma_2}{A} \\ 0 & 0 \\ \frac{k_3 (1-\gamma_1)}{A} & 0 \end{bmatrix},$$

$$C_c = \begin{bmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \end{bmatrix},$$

$$\tau_i = \frac{a_i}{A} \sqrt{\frac{g}{2x_i^0}}.$$

Discretization, using zero-order hold sampling at a sampling rate of 1 Hz, gives

$$A = e^{A_c}, \quad B = \int_0^1 e^{A_c (1-t)} B_c \, dt, \quad C = C_c.$$

**Reference model**

A reference model and parameter vector is useful when evaluating the different input designs and quality of obtained models. Since the constants of the true system are unknown, these have to be estimated. One possibility is to use a long identification experiment to get high quality estimates based on the linearized model (4.4), using PEM. It is desirable that the process does not operate too far away from the linearization point during this experiment. Therefore a low variance Gaussian white input was used to collect 2 hours of data, corresponding to 7200 samples at 1 Hz.

The resulting estimates along with their estimated variances are presented in Table 4.3. These estimated values are also used as the true parameter values in simulations whenever the true system is considered to be linear.

**Control strategy**

The objective for the controller is to perform reference tracking of the water levels in the two lower tanks. This is implemented using MPC with the quadratic cost function

$$J(t) = \sum_{i=0}^N \| y(t + i) - r(t + i) \|^2.$$ (4.5)
Table 4.3 The parameter of the four tank process. The nominal values are obtained from a 2 h (7200 samples) experiment on the real process. The estimated variances are also presented.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Nominal value</th>
<th>Variance ($\times 10^{-3}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>0.148 cm$^2$</td>
<td>0.001</td>
</tr>
<tr>
<td>$a_2$</td>
<td>0.167 cm$^2$</td>
<td>0.0005</td>
</tr>
<tr>
<td>$a_3$</td>
<td>0.0673 cm$^2$</td>
<td>0.002</td>
</tr>
<tr>
<td>$a_4$</td>
<td>0.0619 cm$^2$</td>
<td>0.0003</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.620</td>
<td>0.005</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>0.621</td>
<td>0.012</td>
</tr>
<tr>
<td>$k_1$</td>
<td>2.51 cm$^3$/s</td>
<td>0.088</td>
</tr>
<tr>
<td>$k_2$</td>
<td>2.25 cm$^3$/s</td>
<td>0.141</td>
</tr>
</tbody>
</table>

There is no cost associated with the input and all effort is focused on the output following the reference, $r(t)$. The control horizon is chosen to be the same as the prediction horizon, that is $N_u = N_p = N = 10$.

The system outputs are constrained to be in a narrow band around the reference trajectory. The inputs are constrained to never go below 0.5 V, this ensures that there is water in all tubes all the time; the reason for imposing this constraint is that if a tube is drained, an additional delay is introduced into the system, which is not captured by the chosen model. The optimization problem at time $t$ becomes

$$\text{minimize} \sum_{i=0}^{N-1} \| \hat{y}(t+i) - r(t+i) \|^2$$

subject to

$$\hat{y}(t+i) = C [A \hat{x}(t+i) + Bu(t+i)], \quad i = 0, \ldots, N-1$$

$$y^o_1 - 0.21 \leq \hat{y}_1(t+i) \leq y^o_1 + 0.31, \quad i = 0, \ldots, N-1$$

$$y^o_2 - 0.31 \leq \hat{y}_2(t+i) \leq y^o_2 + 0.16, \quad i = 0, \ldots, N-1$$

$$0.5 - u^o \leq u(t+i) \leq 3 - u^o, \quad i = 0, \ldots, N-1$$

The desired reference trajectory and the nominal response (when model perfectly describes the system) is shown in Figure 4.2 along with the allowed operating regions for the outputs. The reference trajectory is kept the same in all simulations and experiments to simplify comparison.
Figure 4.2: The nominal response of the linear four tank system, that is when $\mathcal{S} = \mathcal{M}(\theta)$. The reference trajectory (---) should be tracked with inputs and outputs outside of the constraint regions ( ). The plot shows the signals for tanks 1 ( ) and 2 ( ).

4.2 Simulations

The reference model is used as the true system and controlled with the MPC (4.1). This setup is chosen as it gives a system that can be captured by the model class. That is, there exists a parameter vector $\theta^0$ that corresponds to the true system. The situation is hence the one assumed in Chapter 3. The chosen application cost is (3.6).

Input designs

The optimal design obtained when using $\theta^0$ and designs based on initial estimates of the parameter values are compared. These estimates were obtained using a zero mean
white Gaussian excitation signal with variance 0.01. The optimal FIR input spectra with $M = 40$ parameters for 1% performance degradation, that is $\gamma$ given by (3.7), are shown in Figure 4.3. The number of samples in the identification experiment is $N = 600$. The optimization problem is implemented in CVX and solved using SDPT3.

The optimal spectrum is clearly temporally colored but almost spatially white. The spectrum has high energy at low frequencies, indicating that the static gain of the system is important. This is expected since the application cost relates to reference tracking and therefore emphasizes the static gain.

The designs obtained based on the initial estimates resemble the true optimal spectrum. The main, lowpass characteristics are the same. The optimal input spectrum gives an input with covariance matrix

$$
\begin{bmatrix}
0.0019 & 0.0001 \\
0.0001 & 0.0040 \\
\end{bmatrix},
$$

If a white input is used the covariance matrix would have to be

$$
\begin{bmatrix}
0.0453 & 0 \\
0 & 0.0945 \\
\end{bmatrix},
$$

to guarantee the same performance. That is, the input power would have to be almost 24 times higher for the white input than for the optimal input.

The Hessian of the application cost indicates which directions in the parameter space give a high performance degradation. The two largest eigenvalues of the Hessian indicate that it is important to estimate combinations of the parameters $\{a_2, a_3, a_4, \gamma_2\}$ and $\{a_1, a_2, a_3, \gamma_1\}$ respectively. The smallest eigenvalues correspond to the parameters $k_1$ and $k_2$.

**Control performance comparison**

To motivate why optimal input design should be performed, the water tank process is identified using an optimal input with minimum power and a white noise input with the same power. The resulting performance of the MPC controllers based on these estimates are then compared. The simulation is performed with the controller (4.1). The input design is setup to guarantee a performance degradation of at most 1%, with 95% probability.

A Monte Carlo simulation with 100 independent trials is performed. The resulting output trajectories can be seen in Figure 4.4.

This Monte Carlo trial shows that the optimal input signal outperforms white noise in terms of satisfying the specification on control performance. Of the models
4.2. Simulations

Figure 4.3 The input spectra obtained using optimal input design based on \( \theta^0 \) (---) and an initial estimate of the parameters (---). \( \Phi_{ij}(\omega) \) is the cross spectrum between \( u_i \) and \( u_j \).
Figure 4.4 The trajectories of the process controlled by MPC with models based on estimates from 100 identification experiments. In plot (a) the optimal input has been used in the identification whereas in plot (b) a white input has been used. 96% of the trajectories in plot (a) satisfy the application requirements while only 37% of the trajectories in plot (b) satisfy the requirements.

estimated using the optimal input, 96% satisfy the performance requirement. This should be compared to only 37% of the models estimated using white noise.

Performance degradation in nonlinear systems

In this simulation, the true system is represented by the nonlinear model (4.3a). Hereby, the effects of decreasing the upper limit on the performance degradation, when a linear model is used for a nonlinear process, can be investigated. To decrease the application cost, the estimates should have lower variance, at least in the sensitive directions of $V_{app}$. Had the process been linear, increasing input power, var ($u$), or the
experiment length, $N$, would both reduce the variances of the estimated parameters; input power can always be traded for experiment time or vice versa. However, when the process is nonlinear, increasing input power might drive the process too far from the linearization point for the model to be accurate. Therefore, increasing experiment time may be the only way of reducing parameter variance.

Two experiments are performed to investigate the effect of the nonlinearities in the four tank process. In the first experiment $N = 100$ samples are used for identification. This gives an optimal input design with a high input power solution. In the second experiment, $N = 10,000$ samples are used. The resulting input design has a low power solution. The total input energy that is used is, however, the same in both experiments. The allowed performance degradation is 0.01% in both experiments, that is $\gamma = 10,000/V_{\text{app}}(\theta^p)$.

Out of 100 estimated models, all from the long, low power experiments and only 1 from the short, high power experiments fulfills $V_{\text{app}} \leq 1/\gamma$. The results show that increasing the power can indeed degrade the quality of the estimates when the identified plant is nonlinear. If instead experiment time is allowed to increase, higher quality estimates are obtained.

### 4.3 Experiments

The method for applications oriented identification is evaluated on a laboratory setup of the presented four tank system. The process is described in detail in Section 4.1. The experimental setup consists of the four tank process, two Universal Power Modules for power supply and signal amplifications, a National Instruments DAQ-card to sample measurements and a Windows PC running MATLAB with the identification and MPC implementations. The system runs at a sampling frequency of 1 Hz which gives plenty of time for online optimization in the MPC.

#### Optimal input design and identification

The full identification method with input design presented in Section 3.5 was performed on the four tank process. The procedure was:

1. Apply input voltages corresponding to the desired operating point and allow the water levels in the tanks to settle.

2. A 2 minute (120 samples) initial identification experiment. The input used was Gaussian white noise with covariance matrix $\Lambda_r = 0.1I$. 
3. Based in the estimate from 2, calculate $\hat{\nu}_{app}(\theta, \hat{\theta})$ and find the optimal input spectrum $\Phi^u_{\ast}(\omega)$.

4. A 10 minute (600 samples) identification experiment using an input with spectrum $\Phi^u_{\ast}(\omega)$.

5. Construct an MPC with the model obtained in 5 and try reference tracking.

To illustrate the advantage of using the optimal input, the same procedure is performed but the optimal input is replaced by a Gaussian white input with the same variance as the optimal input. The reference tracking capabilities of the resulting MPCs are shown in Figure 4.5. It is evident that using the optimal input during the identification experiment results in models that are suitable for the intended application. The system responses from when white noise identified models are used are more spread out around the desired reference trajectories.

The water levels violate constraints for both cases presented in Figure 4.5. However, since the responses are closer together for models originating from optimal input designs, it is in this case possible to operate at references closer to the output constraints. The references would have to be far from the constraints to guarantee no constraint violations for the models obtained in white noise identifications. To further improve on reducing constraint violations, a penalty for this in the application cost, might be useful.

### 4.4 Summary

The applications oriented input design formulation and the identification procedure for MPC applications from Chapter 3 has been evaluated on a four tank process.

Simulations were performed to illustrate the theory, where a linear system that is within the model class is assumed. The results show that the applications oriented input design can guarantee, with high probability, that the resulting model gives acceptable control performance. The reduction in experimental effort compared to if a white noise input is used can also be significant.

A simulation where the true system is represented by a nonlinear model of the tank process was also performed. It shows that when linear models of nonlinear systems are identified, high variance or amplitude inputs can degrade the model quality. To increase model quality, longer experiment may be necessary. This is in contrast to identification of linear systems, where experiment length can be traded for input power.

Finally, the real tank process was identified using the procedure in Section 3.5. The experiment was repeated five times and the resulting MPC controllers were evaluated.
The process was also identified using white noise with the same power as the optimal input. The controllers from the optimal experiments give significantly better reference tracking than what is achieved with models from white noise identification.
Many real systems exhibit nonlinear behavior, particularly when operating far from chosen operating points. The previously presented water tank process is just one example of this. Oftentimes it suffices to use linear models for control design. However, some systems might require nonlinear models to achieve acceptable performance. Therefore it is of interest to have good tools also for identification of systems that are not linear.

This chapter presents input design for system identification of nonlinear FIR-type systems in the prediction error framework. The starting point is the asymptotic properties of PEM estimates, as was done for linear systems. First the general situation is presented. Some similarities to the linear case are found and explored. To proceed further, two specific system structures are studied and presented together with numerical examples. The first is static nonlinear systems and the second is nonlinear systems with finite memory. Input sequences are designed in terms of their statistical properties and not directly in time domain. The fact that the inverse covariance matrix of the estimated parameters is linear in the input probability density function is exploited to formulate...
convex optimization problems. The main issue considered is the parameterization of the Probability Density Function (PDF) of the input. The PDF which should be flexible enough but also keep the number of free variables in the optimization problem small. Generating signals with the obtained PDF is also discussed to some extent.

### 5.1 General nonlinear systems

In contrast to the class of linear systems, the class of nonlinear systems attributes no specific properties to the considered system. In fact, one might just as well consider the class of all systems. By specifying a very general model structure the difference between this very general class and linear systems — in an input design context — can be discussed.

Consider the general model given by

\[
\begin{align*}
y(t) &= g(U(t), Y(t-1), \theta) + e(t), \\
U(t) &= [u(t), u(t-1), \ldots], \\
Y(t) &= [y(t), y(t-1), \ldots],
\end{align*}
\]  

(5.1)

where \(u(t)\) is a known exogeneous input and \(y(t)\) is the observed output. The function \(g(\cdot)\) is a nonlinear mapping from past inputs and outputs to the system output. The mapping is parameterized by the unknown vector \(\theta\). The stochastic sequence \(e(t)\) is an independent, identically distributed Gaussian random processes with variance \(\lambda_e\).

Assuming that there exists a parameter vector \(\theta\) such that the model describes the true system and PEM is used to estimate the parameters, these estimates will have the covariance matrix

\[
\sqrt{\mathcal{N}(\hat{\theta}_N - \theta_0)} \in \mathcal{N}(0, P),
\]

(5.2a)

\[
P = \lambda_e E \left\{ \psi(t, \theta_0)\psi^T(t, \theta_0) \right\}^{-1},
\]

(5.2b)

\[
\psi(t, \theta_0) = \left. \frac{d\hat{y}(t)}{d\theta} \right|_{\theta = \theta_0}.
\]

(5.2c)

In the linear case, \(\psi(t)\) contains delayed samples of the input \(u(t)\) only and the resulting matrix contains second order moments of the input. It was seen in Chapter 2 that the inverse covariance matrix can then be expressed as a linear function of the input spectrum in the frequency domain. For nonlinear systems this is no longer true as \(\psi(t)\) in general contains higher order moments of the input or other, arbitrary functions of the input.
Despite this difference, it would be nice to be able to use the well-developed tools from linear systems even for input design for nonlinear systems. The key result for linear systems was the fact that $P^{-1}$ is linear in the input spectrum. A similar result can be found in the nonlinear case. The expectation operator in the expression for $P$ is the multivariate integral over the input PDF,

$$P^{-1} = \lambda \int \cdots \int_{U(t)} \psi(t, \theta_0) \psi^T(t, \theta_0) p(U(t)) \, dU(t).$$  \hspace{1cm} (5.3)

From expression (5.3) it is seen that the inverse covariance matrix is linear in the input PDF. In fact the PDF has taken exactly the role of $\Phi_\omega$ in (2.11b). This has been known since the 1970s and is greatly discussed in the statistics literature; for an early example see Fedorov (1972). Concerning the input design problem — if one chooses to work with the input PDF in place of the input spectral density — the situation should be similar to the linear case. However, the crucial difference is that while the spectral density is a univariate function of frequency, the PDF will often be multivariate.

**Properties of the PDF**

The joint PDF of a stochastic process has some specific properties.

1. The total probability must be one, that is

$$\int_{U(t)} p(U(t)) \, dU(t) = 1.$$  \hspace{1cm} (5.4)

2. The PDF is always non-negative, that is

$$p(U(t)) \geq 0, \text{ for all } U(t).$$  \hspace{1cm} (5.5)

When a stationary distribution is desired, some extra properties in terms of the marginal distributions need to be imposed. These properties will be addressed later in the specific cases.

**Input design formulation**

Much inspired by the linear case, the idea behind the convex formulation of the input design problem for nonlinear systems is to:

1. Find a suitable linear parameterization of the input PDF.
2. Find a convex representation of the optimization problem in the PDF parameters.

3. Generate an input signal with the desired PDF.

The main focus will be on finding good PDF parameterizations that give computationally tractable optimization problems. Often times this means that the parameterization should be linear and finite. For the signal generation, readily available methods will be used.

To proceed further it is necessary to restrict the considered system class to certain types of nonlinear systems. Common such classes are Hammerstein and Wiener systems, nonlinear FIR-systems and Neural Networks, to name a few. In the following two sections, two special classes, where convex formulations can be found, are considered.

5.2 Static nonlinear systems

The first special case considered is the class of static nonlinear systems. Such systems can be represented by models of the type

\[ y(t) = g(u(t), \theta) + e(t), \]

where \( g \) is a nonlinear mapping between the input \( u(t) \) and the output \( y(t) \). The signal \( e(t) \) is Gaussian, white noise. The mapping is parameterized by \( \theta \), which is unknown and needs to be identified. The predictor is in this case given by

\[ \hat{y}(t) = g(u(t), \theta). \] (5.6)

This class, although simple, contains some important and common nonlinearities such as saturations, dead zones, ideal relays, to name a few.

For this class of systems the inverse covariance matrix is given by

\[ P^{-1} = \lambda \int \psi(t, \theta_0) \psi^T(t, \theta_0) p(u(t)) \, du(t), \]

\[ \psi(t, \theta_0) = \frac{dg(u(t))}{d\theta} \bigg|_{\theta=\theta_0}. \] (5.7)

The expression (5.7) is a univariate function of \( u(t) \) and the PDF has taken exactly the place of the input spectral density in Lemma 2.1. Therefore, many of the techniques from linear systems carry over to static nonlinear systems.
5.2. Static nonlinear systems

Parameterization of the PDF

The first step is to find a suitable way of parameterizing the input PDF. A flexible choice is the finite, linear parameterization

\[ p(u(t)) = \sum_{k=-M}^{M} c_k B_k(u(t)). \]  

(5.8)

where \( B_k \) are some basis functions and the \( c_k \) are the parameters that will be the free variables in the optimization.

There are many possible choices for the bases \( B_k \). One could for example choose a set of pre-specified distribution functions. Another example is to use the complex exponentials as bases and hence a trigonometric polynomial representation of the PDF. This also has the interpretation of a truncated Fourier series expansion of the PDF, which is known to approximate a large class of functions well (Vretblad, 2003). Another benefit of using the exponential basis is the available results from trigonometric polynomials which can be used to enforce positivity.

Input generation

There are many algorithms available that can be used to generate independent samples from a given distribution. One can, for example, use the acceptance-rejection or inverse transform algorithms. Both methods are well understood and easily implemented (Law and Kelton, 2000).

Numerical example

Consider the nonlinear static system shown in Figure 5.1. The mapping \( g \) is a polynomial function of the input and the system is given by

\[ y(t) = \theta_1 u(t) + \theta_2 u^3(t) + \theta_3 u^5(t) + e(t) = \theta^T \varphi(u(t)) + e(t), \]

\[ \theta = [\theta_1 \ \theta_2 \ \theta_3]^T, \quad \varphi(u(t)) = [u(t) \ u^3(t) \ u^5(t)]^T. \]  

(5.9)
Here the input $u(t)$ takes values on the interval $[-u_{\text{max}}, u_{\text{max}}]$ and $e(t)$ is Gaussian, zero mean white noise with variance $\lambda_e = 1$. For this system, the optimal output predictor is given by

$$\hat{y}(t) = \theta^T \varphi(u(t)).$$  \hfill (5.10)

Suppose that it is of interest to have a good estimate of the sum of the parameters $\theta$, that is

$$\theta_1 + \theta_2 + \theta_3 = L^T \theta, \quad L = \begin{bmatrix} 1 & 1 \end{bmatrix}^T.$$ \hfill (5.11)

The variance of the sum of the parameters is given by

$$E \left\{ L^T \hat{\theta} \hat{\theta}^T L \right\} = L^T E \left\{ \hat{\theta} \hat{\theta}^T \right\} L = L^T P L,$$ \hfill (5.12)

and a relevant input design problem would then be to design the signal $u(t)$ such that this variance is minimized. This corresponds to the optimization problem

$$\begin{align*}
\text{minimize} & \quad \gamma \\
\text{subject to} & \quad L^T P L \leq \gamma \\
& \quad p(u(t)) \geq 0, \quad \text{for all } u(t). \\
& \quad \int p(u(t)) \, du(t) = 1
\end{align*}$$ \hfill (5.13)

The input PDF $p(u(t))$ is parameterized as the trigonometric polynomial

$$p(u(t)) = \sum_{k=0}^{M} \alpha_k e^{j u_k} + \sum_{k=1}^{M} \alpha_k e^{-j u_k}, \quad u_k = \frac{\pi k u(t)}{u_{\text{max}}},$$

which gives a function $p(u(t))$ that is periodic with period $2u_{\text{max}}$. Since the input is known to lie in the interval $[-u_{\text{max}}, u_{\text{max}}]$, $p(u(t)) = 0$ for $u(t)$ outside this interval. This is achieved if

$$\int_{-u_{\text{max}}}^{u_{\text{max}}} p(u(t)) \, du(t) = 1 \iff \alpha_0 = \frac{1}{2u_{\text{max}}},$$ \hfill (5.14)

Let $\{A, B, C, D\}$ be a controllable state space realization of $p^+(u(t))$, the KYP lemma can be used to impose non-negativity of $p(u(t))$. 
The convex formulation of the optimization problem becomes

\[
\text{minimize } \gamma \\
\text{subject to } \begin{bmatrix} \gamma & L^T \\ L & P^{-1} \end{bmatrix} \succeq 0 \\
K(Q, \{A, B, C, D\}) \succeq 0
\] (5.15)
\[\alpha_0 = \frac{1}{2u_{\text{max}}} \]

The variance constraint \( L^T P L \leq \gamma \) has been transformed to a convex constraint in \( P^{-1} \) using the Schur complement.

Numerical results

The optimization problem (5.15) is implemented in MATLAB using CVX (Grant and Boyd, 2011) and solved with SeDuMi (Sturm, 1999). The number of coefficients in the parameterization is chosen as \( M = 30 \). Two values for the maximum input amplitude are used, (1) \( u_{\text{max}} = 3 \) and (2) \( u_{\text{max}} = 6 \) and the optimal input PDF for both cases is calculated. The results are shown in Figure 5.2. It is interesting to note that the optimal PDFs for the two cases have very different characteristics.

5.3 Nonlinear FIR systems

The second case considered is when (5.1) is a nonlinear FIR system, i.e.,

\[
y(t) = g(U(t), \theta) + \epsilon(t), \\
U(t) = \{u(t), u(t-1), \ldots, u(t-k)\}
\] (5.16)

where \( k \) is the number of time lags in the dynamics of the system. By choosing to work with this type of model, the amount of time lag in the moments of \( P^{-1} \) is also limited. This means that the inverse covariance matrix becomes a linear function of the joint PDF \( p(U(t)) \) given by

\[
P^{-1} = \lambda, \int \psi(t, \theta_0) \psi^T(t, \theta_0) p(U(t)) \, dU(t),
\]

\[
\psi(t, \theta_0) = \frac{dg(t)}{d\theta} \bigg|_{\theta = \theta_0}.
\] (5.17)

The expression (5.17) looks similar to (5.7). There is, however, a crucial difference: the inverse covariance matrix is no longer a function of one variable. Instead, the expression is a function of a multivariate joint distribution.
Parameterization of the PDF

Considering the usefulness of the parameterization (5.8) it might be tempting to use the same ideas for the nonlinear FIR case. However, there are two problems that require other solutions.

Firstly, the number of coefficients needed for the parameterization quickly grows very large. This means that the number of free variables in the optimization problem becomes too large to handle. As an example, consider the case with just two time lags in (5.16), that is \( k = 2 \), and \( M = 50 \) coefficients each, as was used in the static case, for \( u(t) \), \( u(t - 1) \) and \( u(t - 2) \). This would give 125,000 free variables in the parameterization of the PDF alone. This is not a practical solution.

Secondly, in the linear and nonlinear static case, the parameterizations lead to
5.3. Nonlinear FIR systems

univariate trigonometric polynomials. Such polynomials are easily constrained to be positive on the unit circle. For multivariate polynomials this is in general more difficult. Even though the trace parameterization is applicable to multivariate polynomials, it is only possible to constrain the polynomial to be sum-of-squares, a smaller set than the set of positive polynomials. The kyp-lemma has been extended to handle two frequency dimensions (Yang et al., 2008) but not to higher dimensions.

Discrete inputs

Because of the aforementioned reasons, other parameterizations of the input that give numerically tractable problems are needed. A tradeoff between flexibility in the parameterization and computational complexity has to be made. One natural restriction to the inputs considered is to use discrete levels. This is not an uncommon situation as more and more systems are quantized.

Consider the case where the input is restricted to take one of \( m \) possible levels, that is, the input at time \( t \) is \( u(t) \in \{u_1 \ldots u_m\} \). Let \( I = [i_1, \ldots, i_k] \) be a vector of indices corresponding to the signal values of a particular input sequence and let \( U_I(t) = \{u_{i_1}, \ldots, u_{i_k}\} \) be that particular sequence. Then the joint pdf of the input sequence \( U(t) \) can be parameterized as

\[
p(U(t)) = \sum_I p_I \delta(U(t) = U_I(t)),
\]

where \( p_I \) are the probability weights of the sequence given by the indices \( I \) and \( \delta \) is the Kronecker delta. This parameterization is linear in the parameters \( \alpha \) and gives a linear parametrization of the inverse covariance matrix,

\[
P^{-1} = \lambda_c \sum_I \psi(t, \theta_0)\psi^T(t, \theta_0) p(U(t)).
\]

To ensure that the discrete distribution corresponds to a stationary process, the constraints

\[
\sum_{i=1}^m p(u_i(t), u(t-1), \ldots, u(t-k)) = \sum_{i=1}^m p(u(t-1), \ldots, u(t-k), u_i(t)) \quad (5.18)
\]

must hold for all possible combinations of \( \{u(t-1), \ldots, u(t-k)\} \) (Grillenberger and Krengel, 1976).

This parameterization comes with two choices that have to be made: the number of levels, \( m \), and the number of time lags considered, \( k \). For FIR systems, the natural choice for \( k \) is the length of the memory in the system.
The number of levels will also impact the solution. When the system operates using discrete input levels, these levels are the natural choice. For systems with amplitude continuous inputs, restricting the number of levels can give suboptimal solutions but tractable problems. Since there are \( m \) possible inputs at any given time and inputs up to \( k \) time lags in (5.16) there are \( m^k \) possible input combinations for the sequence \( U(t) \) and hence \( m^k \) free variables in \( P^{-1} \). As a comparison, if a ternary input signal is used, \( m = 3 \) and \( 3^k = 125,000 \) gives \( k \approx 11 \). This shows that this approach can give computationally tractable formulations for systems with longer memory in the dynamics. One should, however, be aware that there are conditions which a quantized input impose on the possible moments of the distribution. In de Carvalho and Clark (1983) the characteristics of autocorrelations of binary sequences are discussed.

### Input generation

Sampling from a multivariate stationary distribution can be done through a number of Markov Chain Monte Carlo (MCMC) methods, one such example is Gibbs sampling (Casella and George, 1992). The common idea is to construct a Markov chain with the desired distribution as its equilibrium. When the stationary distribution of the Markov chain has been reached, samples from the desired distribution can be taken.

A recent and interesting MCMC method is the so called Coupling from the past. The method delivers, at least in principle, perfect samples from the stationary distribution of the Markov chain. The idea behind coupling from the past is as follows. If a Markov chain has been running for all time, then the state at time 0 has the stationary distribution. Find a time in the past such that all possible paths coalesce to one value by time 0. This value will be a sample of the stationary distribution (Propp and Wilson, 1997; Murdoch and Rosenthal, 2000).

### Numerical example

Consider the nonlinear FIR system shown in Figure 5.3 where

\[
\begin{align*}
y(t) &= G_1(q, \theta)u(t) + G_2(q, \theta)u^2(t) + e(t), \\
G_1(q, \theta) &= \theta_1 + \theta_2q^{-1}, \\
G_2(q, \theta) &= \theta_3 + \theta_4q^{-1},
\end{align*}
\]

where \( e(t) \) is Gaussian white noise with variance \( \lambda_e = 1 \). This model is commonly used in communications as it describes, for example, nonlinear distortions in satellite links and ADSL communication (Bai, 2005). The optimal predictor for this system is

\[
\hat{y}(t) = G_1(q, \theta)u(t) + G_2(q, \theta)u^2(t).
\]
5.3. Nonlinear FIR systems

For the identification experiment, a ternary input signal with levels \{-1, 0, 1\} is used. The input design problem is to find the D-optimal distribution of the input. This is the solution to the optimization problem

\[
\begin{align*}
\text{maximize} & \quad \det \mathbf{P}^{-1}(\theta) \\
\text{subject to} & \quad p(U(t)) \geq 0 \\
& \quad \int p(U(t)) \, dU(t) = 1
\end{align*}
\] (5.21)

The naïve approach would be to disregard the fact that the system is nonlinear and simply use the input signal that would be optimal if the system had been linear. It is a well known result that a white input is a D-optimal input distribution for a FIR system (Levin, 1960). However, it is seen below that the D-optimal ternary signal for the system (5.19) is in fact not white.

The system (5.19) has a memory of \(k = 2\) and \(m = 3\) input levels are used. Therefore the parameterization of the PDF becomes

\[
p(u(t), u(t-1)) = \sum_{i=1}^{3} \sum_{j=1}^{3} p_{ij} \delta(u(t) = u_i, u(t-1) = u_j),
\]

which has 9 free variables. This gives the inverse covariance matrix

\[
\mathbf{P}^{-1} = \lambda \sum_{u(t), u(t-1)} \psi(t, \theta_0) \psi^\top(t, \theta_0) p(u(t), u(t-1)),
\]

which is linear in the parameters \(\alpha_{ij}\).

The positivity constraint for the PDF (5.5) is simply

\[
p_{ij} \geq 0, \quad i, j = 1, 2, 3,
\]
the constraints for stationarity (5.18) becomes
\[ p_{12} + p_{13} = p_{21} + p_{31}, \]
\[ p_{21} + p_{23} = p_{12} + 2p_{32}, \]
\[ p_{31} + p_{32} = p_{13} + p_{23}, \]
and finally the constraint for the total mass (5.4) becomes
\[ \sum_{i=1}^{3} \sum_{j=1}^{3} p_{ij} = 1. \] (5.22)

All in all, this gives the convex formulation of (5.21)

\[
\begin{align*}
\text{maximize} & \quad \det \mathbf{P}^{-1}(\theta) \\
\text{subject to} & \quad p_{ij} \geq 0, \quad i, j = 1, 2, 3 \\
& \quad \sum_{i=1}^{3} \sum_{j=1}^{3} p_{ij} = 1 \\
& \quad p_{12} + p_{13} = p_{21} + p_{31} \\
& \quad p_{21} + p_{23} = p_{12} + 2p_{32} \\
& \quad p_{31} + p_{32} = p_{13} + p_{23}
\end{align*}
\] (5.23)

The optimization problem (5.23) is implemented in MATLAB using CVX and solved with SDPT3. The resulting distribution is presented in Figure 5.4 along with the distribution for a white ternary input. It is clear that the optimal input is not white. In fact, in the optimal input, a zero in one time step is never followed by a zero in the next time step.

The resulting parameter variances and the determinant of the inverse covariance matrix are presented in Table 5.1. For comparison, the values obtained when using a white input where all points in the distribution have equal weights are also presented. We see that the D-optimal input reduces the variance of \( \theta_1 \) and \( \theta_2 \) while it increases for \( \theta_3 \) and \( \theta_4 \).

5.4 Summary

This chapter raised the problem of input design for nonlinear systems. Inspired by the successful linear case, the starting point is the statistical properties of prediction.
Figure 5.4 The optimal input PDF (a) for minimizing the total variance of the estimated parameters in the nonlinear FIR-systems (5.19). The white distribution (b), which is optimal for linear FIR-systems, is presented for comparison. The size of a circle is proportional to the corresponding $p_{ij}$.

Table 5.1 The resulting determinant of the parameter covariance matrix using D-optimal and white inputs and the parameter variances. Some parameter estimates are slightly better with the white input, but the determinant is reduced 40% with the D-optimal input.

<table>
<thead>
<tr>
<th>Input</th>
<th>det $P$</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
<th>$\theta_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-optimal</td>
<td>5.7</td>
<td>1.28</td>
<td>1.28</td>
<td>2.73</td>
<td>2.73</td>
</tr>
<tr>
<td>White</td>
<td>9.1</td>
<td>1.50</td>
<td>1.50</td>
<td>2.70</td>
<td>2.70</td>
</tr>
</tbody>
</table>

error estimates. The similarities between input design for identification of nonlinear models and the much studied linear case are illustrated. A key point is that linear parameterizations of the probability density function of the input lead to convex optimization problems and for certain classes, tractable problems can be formulated. Two such cases were studied in detail.

A convex formulation of the input design problem for static nonlinear systems was presented. Much of the techniques from the linear case can be applied with minor changes. It is worth noticing that amplitude constraints on the input affect the resulting optimal input to a large extent.
Nonlinear FIR systems were considered. Because of the rapidly growing number of optimization variables, the input has in this case been restricted to be ternary. The ideas can however be extended to also cover inputs with more levels. One should, however, be aware that the computational complexity grows quickly as the number of decision variables increases.

An important observation is that results from optimal input design of linear systems are not guaranteed to give optimal results for the nonlinear case. This is an indication that optimal input design for nonlinear systems should be studied further to be better understood.
Chapter 6

Conclusions

“We have the duty of formulating, of summarizing, and of communicating our conclusions, in intelligible form, in recognition of the right of other free minds to utilize them in making their own decisions.”

Ronald Fisher

The thesis discusses optimal input design formulated as convex optimization problems in two areas. The first is the applications oriented input design formulation presented and evaluated in Chapters 3–4. The second is optimal input design for two classes of nonlinear systems. The contributions of the thesis are summarized and some directions for future work are suggested.

6.1 Experiment design for MPC

A framework for applications oriented input design and system identification in an model predictive control context was presented. The idea is to minimize the required experimental effort in a system identification experiment while guaranteeing that when the obtained model is used, the system performance is acceptable. This has been considered earlier but the application to MPC presents some new problems. The motivation for these applications is the growing popularity of the MPC strategy, both in industry and academia.
The performance degradation due to plant-model mismatch is measured by the introduction of an application cost. Due to lack of closed form solutions to the constrained MPC problem the application cost needs to be evaluated numerically. To avoid the need for using the real process for these calculations, an approximate application cost, where the system is replaced by an estimated linear model, is used. This strategy is used even for nonlinear processes.

The original problem formulation is non-convex and infinite dimensional. Two relaxations that give finite dimensional, convex problems are presented. The optimal input design method requires knowledge of the true parameter values. These are obviously not available. The proposed solution is to use an initial estimate instead. These can be obtained in an identification experiment or through knowledge of the system.

The input design and identification strategy is illustrated on a four tank process where reference tracking is implemented using MPC. Optimal inputs for the identification experiments are designed and show that one can expect the required input power in the optimal designs to be lower than for a white input, if the same accuracy is desired. The method is evaluated both in simulations and on real process data.

Examples also show that the trade-off between input signal power and number of samples used in identification experiment plays a larger role for identification of nonlinear systems. This is because a linear model is used in the input design. A high variance of the input signal may drive the process state far from its linearization point and thus the model will no longer be accurate.

Future work

There are many interesting open questions that should be investigated for the MPC oriented input design. These can be divided into theoretically interesting problems and more practically important problems.

It is clear that the constraints in the MPC can have an impact on the application cost and hence on the application set. It should be investigated if it is possible to characterize systems where the impact will be significant and systems where there is little or no impact of constraints.

Since an estimated model is used in the design, the obtained input spectrum is often based on parameter values that are correct. Investigating when such designs are close to the true optimal design is an important topic.

It is possible to run the identification and input design method iteratively. If more and more samples are used in the identification, the parameters estimates will converge to their true values and the resulting input design should eventually be the optimal
design. To show this behavior in the MPC case, as has been done for ARX systems, is a challenging but good contribution.

More practically, the method has two shortcomings. The first is the fact that the input and output constraints are not directly included in the input design. However, as pointed out, there are methods to include them later in the signal generation phase. The second is the memory requirements for the optimization problem. The size of the problem grows quite fast with the size of the process model. Exploiting the structure of the model, in this case the matrices are quite sparse, should be a way to reduce this problem. Also using alternatives to the KYP-lemma for spectrum positivity will reduce the computational complexity. Another interesting idea is to use scenario based optimization also for the constraints on the input spectrum.

An important issue in maintenance of industrial MPC systems is the ability to perform re-identification during the life-time of a plant. It is often preferable to perform this identification without shutting down normal production. This means that the identification method and input design must work in closed-loop. Modifying the presented method to work in such cases should therefore be a priority.

### 6.2 Experiment design for nonlinear systems

This chapter raised the problem of input design for nonlinear systems. Inspired by the successful linear case, the starting point is the statistical properties of prediction error estimates. The similarities between input design for identification of nonlinear models and the much studied linear case are illustrated. A key point is that linear parameterizations of the probability density function of the input lead to convex optimization problems and for certain classes tractable problems can be formulated. Two such cases were studied in detail.

A convex formulation of the input design problem for static nonlinear systems was presented. Much of the techniques from the linear case can be applied with minor changes. It is worth noticing that amplitude constraints on the input affect the resulting optimal input to a large extent.

Nonlinear FIR-systems were considered. Because of the rapidly growing number of optimization variables, the input was this case been restricted to be ternary. The ideas can be extended to also cover inputs with more levels. However, one should be aware that the computational complexity grows quickly as the number of decision variables increases.

An important observation is that results from optimal input design of linear systems is not guaranteed to give optimal results for the nonlinear case. This is an
indication that optimal input design for nonlinear systems should be studied further
to be better understood.

Future work

This work is a first investigation of the possibilities of using statistical properties to
design the input for identification of nonlinear systems. The intention is in no way
to present a complete solution. Other parameterizations of the probability density
functions can be considered. A family of predefined PDFs could be used for example.

The KYP-lemma was used in the static case to ensure positivity of the PDF but not
in the nonlinear Fir-system. The alternative trace parameterization could be used for
higher dimensional PDFs or a scenario based approach.

This work has only considered two classes of nonlinear systems. Finding other
classes where convex optimization problems can be formulated is probably possible.
MOOSE: Model based optimal input signal design toolbox for Matlab

“Give us the tools, and we will finish the job.”

Winston Churchill

Moose is a model based optimal input design toolbox developed for MATLAB. The goal of the toolbox is to simplify the implementation of optimal input design problems. It provides an extra layer between the user and a convex optimization environment. The interface to the user is very much inspired by cvx Grant and Boyd (2011).

The toolbox has been developed in close collaboration with Mariette Annergren.

A.1 Introduction

Moose is designed to handle input design problem of the form

\[
\text{minimize} \quad \Phi_u \quad \text{subject to} \quad U(\alpha) \subseteq \Theta_{app}(\gamma) \\
\alpha(\omega) \leq \Phi_u(\omega) \leq \beta(\omega), \quad \forall \omega.
\]

The optimization problem is solved through the ellipsoidal relaxation or the scenario optimization approach. It is set up in MATLAB through a series of keywords in a Moose
A.2 Tutorial

This tutorial presents the process of declaring and solving an optimal input design problem in Moose.

Installation

If Moose is used for the first time, it needs to be installed in the \texttt{\textasciitilde\textbackslash moose\textbackslash path}. This is done by going to the Moose directory and executing the command \texttt{mooseSetup}. Moose is based on the package \texttt{cvx} which also needs to be installed to run Moose.

Setting up the problem

Consider input design for the system

\[
G(q, \theta) = \theta_1 u(t - 1) + \theta_2 u(t - 2) + \epsilon(t)
\]  

(A.1)

with true parameter values \( \theta^0 = [10 \ -9 \ 1] \) and noise variance \( \text{var} \ e = 1 \). The goal is to solve the optimization problem

\[
\begin{align*}
\text{minimize} & \quad \text{var} \ u \\
\text{subject to} & \quad U(0.95) \subseteq \Theta_{app}(100) \\
& \quad \Phi_u(\omega) \geq 0, \quad \forall \omega
\end{align*}
\]

using an \texttt{fir}-input spectrum with 20 coefficients, 100 samples of data and the ellipsoidal relaxation (3.18). A \texttt{MATLAB} implementation of the problem is presented below.

```matlab
% Setup system and model
theta = [10 -9]';
G = tf([0;theta],1,1,'variable','z^-1');
H = 1;
Re = 1;
% Moose declaration block
beginMoose
```
A.2. Tutorial

```plaintext
objective minimize(inputPower)
model G H Re
identification constraints
    spectrum phiU = FIR(20)
    probability 0.95
    numSamples 100
application constraints
    ellipsoid(@Vapp,100)
endMoose
optimalFilter = mooseProblem.spectralFactor;
```

The first three lines setup the models for the nominal system and noise transfer functions and the noise variance.

The Moose declaration block begins with the `mooseBegin` command. This creates the `mooseProblem` variable in the workspace where the input design is stored.

The eight basic keywords are then used to setup the problem:

- **objective** sets the optimization goal function, in the example the input power is minimized.
- **model** sets the nominal system and noise model.
- **identification constraints** has no formal action but can be used to make the declaration clearer.
- **spectrum** sets the spectrum type. The non-negativity constraint is handled automatically.
- **probability** is the level of the confidence ellipsoid.
- **numSamples** sets the number of samples used in the identification experiment.
- **application constraints** has no formal action but can be used to make the declaration clearer.
- **ellipsoid** adds the ellipsoid defined by the Hessian of $V_{\text{app}}$ to the set $\Theta_{\text{app}}$.

The Moose declaration block ends with the `mooseEnd` command. In addition to closing the block, it also calls an optimization problem solver to find the optimal design.

The last line gets the optimal spectral factor which can be used to realize an input signal.

In the example some keywords have been assigned variable names. This makes the keyword available inside to Moose declaration block so that it can be used in, for example, the objective function. When the Moose declaration block is closed, named variables remain available in the MATLAB workspace.
A.3 Moose declaration block

An input design problem is defined in Moose in a *Moose declaration block*.

The **mooseBegin** command

The `mooseBegin` command declares the beginning of a Moose declaration block. When the command is called, two things happen. First the other keywords needed in the problem declaration are activated. Second, a variable `mooseProblem` is created in the workspace to handle the input design declaration.

**Remark A.1** The Moose keywords are not available outside of a Moose declaration block. This is done to minimize conflict with other MATLAB functions.

The **mooseEnd** command

The `mooseEnd` declares the end of a Moose declaration block. When the command is called the input design problem is passed to `cvx` and solved. The solver print out and optimal value are shown in the command window.

A.4 Keywords

The declaration of an optimal input design problem in Moose is done using a set of eight keywords. They correspond to the important parameters of the convex formulation of the optimization problem.

**objective keyword**

Declares the objective function of the optimization. The syntax is

```
objective fun
```

`fun` can be any function that can be evaluated by MATLAB using the variables available in the workspace and that results in a convex optimization problem.

**model-keyword**

Declares the nominal model used. Moose supports two types of models, transfer functions and state space models on innovations form. The same keyword is used for both model types, only the arguments change.

To declare a transfer function model the syntax
A.4. Keywords

model G H Re

is used. G and H are discrete time transfer function models in the variable $z^{-1}$ and Re is the noise covariance matrix.

To declare the state space model

$$
x(t + Ts) = Ax(t) + Bu(t) + Ke(t), \quad y(t) = Cx(t) + e(t). \quad (A.2)
$$

the syntax

model A B K C Re Ts

is used. A, B, C and K are the state space matrices, Re is the noise covariance matrix and Ts is the sampling interval.

It is possible to declare which model parameters are known and which are estimated. This is done by adding an index argument in the model declaration. As an example, a transfer function can be declared with

$$
G = tf([10 -9],[10 0],1);
H = tf([1],[1],1);
Gindex = '{{[1 2]},{0 0}}';
Hindex = '{{0},{0}}';
model G(Gindex) H(Hindex) Re
$$

The known parameters all have index 0 and the estimated parameters are indexed 1, 2, .... The parameter index should appear in the same position as its nominal value in the transfer function declaration. Here parameter 1 has nominal value 10 and appears as the first coefficient in the numerator of $G$, for instance. The indexing is the same for state space models, but uses matrices commensurate with A, B, C and K instead.

If no indexing is specified, the parameter indexing is the order in which they appear in the standard MATLAB declaration for transfer functions and column-wise for the state space matrices.

**Remark A.2** Using the indexing it is also possible to declare that a parameter appears in more than one place in the model by entering the corresponding parameter index more than once.
identification constraints keyword
This keyword does nothing, it is provided to make the declaration more readable. The spectrum, probability and numSamples keywords belong to the identification constraints.

spectrum keyword
Declares the spectrum type and number of coefficients with the syntax

\( \text{spectrum type}(M) \)

This declaration gives a type-spectrum with \( M \) coefficients in the parameterization. Currently the only supported type is \( \text{FIR} \) which uses the parameterization in (2.22).

probability keyword
Declares the level of the parameter confidence ellipsoid, that is the parameter \( \alpha \). The syntax is

\( \text{probability \ alpha} \)

numSamples keyword
Declares the number of samples used in the identification experiment, that is the parameter \( N \). The syntax is

\( \text{numSamples \ N} \)

application constraints keywords
This keyword does nothing, it is provided to make the declaration more readable. Currently ellipsoid and scenarios are available to define the application constraints.

ellipsoid
Adds an ellipsoid to the application constraints. The syntax is

\( \text{ellipsoid}(\text{fun}, \gamma) \)

The argument \( \text{fun} \) can be a function handle to an application cost or the Hessian at the true parameter values. \( \gamma \) is the accuracy which sets the value for the level curve that gives the ellipsoid, \( \frac{1}{\gamma} \) is the level curve.
A.5. Spectral factorization

scenarios

Adds scenario constraints to the application constraints. The syntax is

scenarios(mat, gamma)

The argument mat is a wide matrix where the first rows contain the scenario and the last row the application cost value at that scenario. gamma is the accuracy which sets the maximum application cost value, 1/\gamma.

A.5 Spectral factorization

Once a Moose problem has been solved, it is possible to obtain the stable, minimum phase spectral factor of the optimal spectrum. The syntax for this is

```
optH = mooseProblem.spectralFactor;
```

A.6 Implementation

The implementation of Moose uses the object-oriented programming capabilities of MATLAB. The structure of the implementation is presented in Figure A.1. The design is built around a predefined set of interfaces for the necessary component of the optimal input design problem.

All user interaction with Moose is done through the user interface and in some cases through the help functions. The user interface is based on a set of keywords that define an optimal input design problem.

The central object is the class oidProblem where the input design problem is stored. Every instance of oidProblem contains a Model- and a Spectrum-instance and one or more Constraint-instances.

Abstract classes are used for the model, spectrum and constraint classes to define interfaces. This allows for easy implementation of new models, spectra and constraint classes.

There is no optimization implemented in Moose. Instead the toolbox relies on external SDP-solvers for solving the defined optimization problem. An abstract parser class defines the interface of the parsers used to construct the SDP from an oidProblem. Currently the only implemented parser is to CVX.

The specifications for the implemented interfaces are detailed in Larsson and Annergren (2011).
A.7 Future versions

At the time of writing, the Moose project is in its initial stages and the first version of the toolbox is just released. The project is however very active with many planned extensions in future versions of the toolbox. The features with highest priority for implementation are presented here. If any other features are requested, contacting the creators is most appreciated. Any other comments are also welcome.

A complete API for the toolbox is planned to be released in the near future. Users are encouraged to implement their own classes as they are needed. Well implemented contributions may be included in future versions of the toolbox.
Planned features

The most highly prioritized features for coming version of Moose are:

- Spectrum types, in particular discrete spectra, from signals that are sums of sinusoids, and spectra that use the partial correlation parameterization.
- Physically parameterized state space models.
- Optimization objectives, in particular output variance, joint input and output variance.
- Multiple application costs.
Bibliography


"I'm finished!"

Daniel Plainview