Application-oriented Experiment Design for Industrial Model Predictive Control

CHRISTIAN A. LARSSON

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

TRITA-EE 2014:052
ISSN 1653-5146
ISBN 978-91-7595-304-5

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i reglerteknik fredagen den 31 oktober 2014, klockan 14.00 i sal F3, Kungliga Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

Abstract

Advanced process control and its prevalent enabling technology, model predictive control (MPC), can today be regarded as the industry best practice for optimizing production. The strength of MPC comes from the ability to predict the impact of disturbances and counteract their effects with control actions, and from the ability to account for constraints. These capabilities come from the use of models of the controlled process. However, relying on a model is also a weakness of MPC. The model used by the controller needs to be kept up to date with changing process conditions for good MPC performance. In this thesis, the problem of closed-loop system identification of models intended to be used in MPC is considered.

The design of the identification experiment influences the quality and properties of the estimated model. In the thesis, an application-oriented framework for designing the identification experiment is used. The specifics of experiment design for identification of models for MPC are discussed. In particular, including constraints in the controller results in a nonlinear control law, which complicates the experiment design.

The application-oriented experiment design problem with time-domain constraints is formulated as an optimal control problem, which in general is difficult to solve. Using Markov decision theory, the experiment design problem is formulated for finite state and action spaces and solved using an extension of existing linear programming techniques for constrained Markov decision processes. The method applies to general noise and disturbance structures but is computationally intensive. Two extensions of MPC with dual control properties which implement the application-oriented experiment design idea are developed. These controllers are limited to output error systems but require less computations. Furthermore, since the controllers are based on a common MPC technique, they can be used as extensions of already available MPC implementations. One of the developed controllers is tested in an extensive experimental validation campaign, which is the first time that MPC with dual properties is applied to a full scale industrial process during regular operation of the plant.

Existing experiment design procedures are most often formulated in the frequency domain and the spectrum of the input is used as the design variable. Therefore, a realization of the signal with the right spectrum has to be generated. This is not straightforward for systems operating under constraints. In the thesis, a framework for generating signals, with prespecified spectral properties, that respect system constraints is developed. The framework uses ideas from stochastic MPC and scenario optimization. Convergence to the desired autocorrelation is proved for a special case and the merits of the algorithm are illustrated in a series of simulation examples.

Sammanfattning

Modern processreglering är ofta baserad på modeller av processen som ska regleras. Modellbaserad prediktionsreglering (MPC) introducerades inom petrokemisk industri som en reglerstrategi för flervariabla system med bivillkor. Utvecklingen har sedan dess gått mot att MPC implementeras inom allt fler industrier. Styrkan hos MPC kommer av att regulatorn kan, med hjälp av modellen, prediktera ett systems framtida beteende och motverka kommande störningar genom att välja lämplig styrning. Det är viktigt att den modell som MPC använder för att beräkna styrlagen överensstämmer med dynamiken hos det system som ska styras. Därför måste modellen hållas uppdaterad om det reglerade systemet ändras över tid. Systemidentifiering som ett verktyg för sådana uppdateringar av MPC-modeller behandlas här.

Systemidentifiering används för att modellera dynamiska system utifrån experimentellt uppmätta data. Hur experimentet utförs påverkar vilka systemegenskaper som kan skattas såväl som den kvalitet som dessa skattningar har. Det är därför viktigt att systemidentifieringsexperimentet utformas på ett sådan sätt att de egenskaper som är viktiga för god prediktionsreglering kan skattas väl. Hur ett systemidentifieringsexperiment ska utformas brukar benämnas experimentdesign. För system med bivillkor ger MPC en olinjär styrlag vilket försvårar experimentdesignen, framförallt om hänsyn till bivillkor ska tas även under systemidentifieringsexperimentet. Experimentdesign där modeller för MPC skattas utifrån ett applikationsorienterat perspektiv studeras i denna avhandling.

Applikationsorienterad experimentdesign kan formuleras som ett optimal reglerproblem där systemets bivillkor ingår i formuleringen. Det gör att optimeringsproblemet ofta blir mycket svårt att lösa. Genom att använda Markov-beslutsteori kan problemet formuleras för system med ändligt antal tillstånd och ändligt antal möjliga styrsignalsnivåer. En optimal styrlag som även ger signaler med goda egenskaper för systemidentifiering kan då beräknas genom semidefinit programmering. Formuleringen innefattar allmänna brus- och störningsprocesser men är ofta mycket beräkningskrävande. Två MPC-baserade regulatorer där experimentdesign ingår som ett bivillkor för regulator formuleras. Regulatorerna förutsätter en enklare brusmodell utan processtörningar men är mindre beräkningskrävande och kan därför appliceras på större system. Då dessa bygger på en välanvänd MPC-formulering bör de nya regulatorerna även vara relativt enkla att implementera som utvidgningar av befintliga MPC-implementeringar. En utförlig experimentell utvärdering av en av regulatorerna har gjorts på en industriell process under normal produktion.

Experimentdesign har studerats under lång tid inom systemidentifieringsområdet och ett antal väl utvecklade metoder finns tillgängliga. De är oftast formulerade som

optimeringsproblem i frekvensdomänen där insignalens effektspektrum används som optimeringsvariabel. För system med bivillkor är det inte rättframt att då generera en signal med de optimala spektralegenskaperna. Här introduceras en ny metod för signalgenerering under bivillkor för system med osäkerhet. Metoden bygger på resultat från stokastisk MPC och scenariooptimering. Garanterad konvergens mot den önskade autokorrelationen visas för ett specialfall och metodens egenskaper demonstreras i ett stort antal simuleringsexempel.

Acknowledgements

I want to thank my main advisor Håkan Hjalmarsson. I am grateful for the opportunity to pursue a PhD and all the guidance along the way. It has been a great experience!

I want to thank my co-advisor Cristian Rojas for all the patience with my questions and good advice over the years.

I want to thank all my collaborators, especially the partners in Autoprofit, for their valuable input and the great cooperation.

I want to thank all the past and present colleagues at the department for the fun times, great discussions, early lunches and late beers all over the world.

I want to thank the administrators Anneli, Hanna, Karin and Kristina for taking care of all the things that make the department run so smoothly.

I want to thank Mariette, Patricio and Per for their help with the proof reading.

I want to thank the Swedish Research Council (Vetenskapsrådet) and the European Commission, through the projects WIDE and Autoprofit, for the financial support that made this work possible.

I want to thank all my friends for providing meaning to life outside research.

I want to thank my parents and my sister for always believing in me, for all the support and for their unconditional love. You are awesome!

Thank you all!

Christian Larsson
Stockholm, October 2014

Contents

Acknowledgements vii

Contents viii

1 Introduction 1
- 1.1 Advanced process control in industry 1
- 1.2 A motivating example 4
- 1.3 Problem formulation 6
- 1.4 Contributions and outline 8
- 1.5 Notation 12
- 1.6 Acronyms 13

2 Background 15
- 2.1 Systems and models 15
- 2.2 Model predictive control 17
- 2.3 System identification 20
- 2.4 Input design 25
- 2.5 Dual and adaptive MPC 27
- 2.6 Reinforcement learning 31

3 Experiment design for model predictive control 33
- 3.1 Application-oriented experiment design 33
- 3.2 Optimal experiment design formulation 36
- 3.3 Measuring MPC performance 38
- 3.4 Open-loop experiment design 40

4 Markov decision process formulation of experiment design 45
- 4.1 The considered problem 46
- 4.2 System and control model 46
- 4.3 Expected average cost and constraints 50

4.4 Discounted expected cost and constraints 53
4.5 Semidefinite program 54
4.6 MDP with excitation 56
4.7 Example 60
4.8 Summary 62

5 Model predictive control formulation of experiment design 65
5.1 System and model 65
5.2 Controller 66
5.3 Model predictive control with excitation (MPC-X) 68
5.4 Model predictive control with excitation for minimum time 75
5.5 Examples 77
5.6 Summary 86
5.A MPC matrices 88

6 Experimental study of MPC-X 89
6.1 Depropanizer 90
6.2 Experimental setup 94
6.3 Simulator experiments 96
6.4 Plant experiments 103
6.5 Summary 111
6.A Residual analysis of simulator models 113
6.B Residual analysis of plant models 117

7 Signal generation for constrained systems 121
7.1 Problem formulation 122
7.2 Signal generation algorithm 124
7.3 Adaptive signal generation 130
7.4 Comparison with filtered noise 134
7.5 Examples 135
7.6 Conclusions 145

8 Conclusions 147
8.1 Experiment design for MPC 147
8.2 Signal generation 150

Bibliography 151

Chapter 1
Introduction

The use of models as a tool to better understand, quantify and visualize the world has always been central in all scientific fields. There are many types of models with different aims depending on the field of study. Regardless of the type, all models are meant to represent physical objects or processes in an objective way. All models are also simplifications or abstractions of the reality that they are intended to represent. What characterizes a good model is the ability to abstract away factors that are of little or no importance to the studied phenomenon while keeping the important factors in the model. The focus in this thesis is on constructing mathematical models of real-world physical objects or systems.

Mathematical models of systems have become an integral part of almost all engineering applications. The ability to simulate a model of a system simplifies the design, development and analysis of large, complex systems. In automatic control, the increase of computing power available for control has resulted in new control strategies that rely on models for computing the future effect of the implemented control. In this context, it is vital that the model used in the controller is able to capture the properties of the system that are relevant to achieve good control performance. It is also important not to model the properties that do not affect control performance as this would make the model unnecessarily complex. These issues are considered in this thesis from an industrial application perspective.

1.1 Advanced process control in industry

Industrial process control is typically designed according to the control hierarchy pictured in Figure 1.1. On top of the plant level, regulatory control, such as valves and proportional-derivative-integral (pid) controllers are implemented as a base control level. The setpoints of the base level controllers have traditionally been controlled manually by operators but are now increasingly often controlled by an advanced

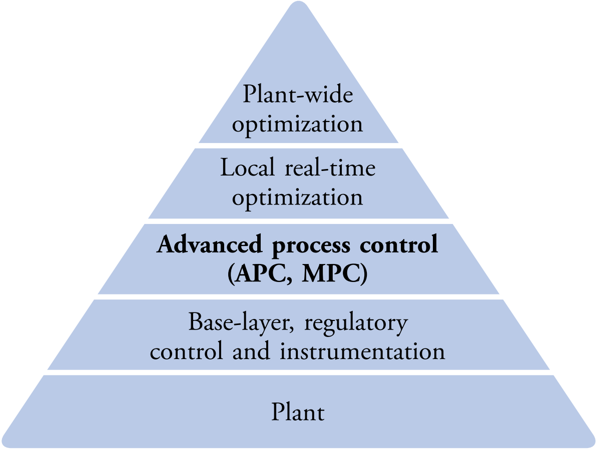

Plant-wide optimization
Local real-time optimization
Advanced process control (APC, MPC)
Base-layer, regulatory control and instrumentation
Plant

Figure 1.1 Industrial control hierarchy. Process control is often structured in a hierarchy from simpler controls to more advanced controllers. The lower level includes valves and PID controllers close to the plant. On top of that, APC is implemented to optimize production on a process level. The optimal setpoints are calculated in real-time optimization, usually on an hourly to daily basis. The top layer optimizes total plant utilization by scheduling different production units on a daily to monthly basis.

process control (APC) solution. The APC layer tries to optimize production of individual production units by driving signals to certain operating points. The time scale of the APC is in the range of minutes to hours. The real-time optimization layer calculates the economically optimal operating points for a number of units; here the time scale is in the range of hours to days. Finally, a top layer with plant-wide optimization is used to optimize plant production by scheduling production and resources according to current market trends and demands. This is done on a daily to monthly time scale. The width of the pyramid in Figure 1.1 reflects the number of individual controllers at the different levels. If the width instead reflected the computational complexity of the controllers, the pyramid would be turned upside down.

The focus in this thesis is on the APC level of process control, which since the introduction in the early 1980s has become accepted as the industry best practice. Morari and Lee (1997) estimate that all modern refineries use APC in some capacity, Canney (2003) reports that 5,000 APC applications are in use and two years later, Canney (2005) reports that the number has grown to more than 6,000. Furthermore, it is reported that *"nearly all [APC] products use Model Predictive Control as the prevailing underlying technology"*—Canney (2005). In fact, Zhu (2009) estimates that all petrochemical plants have predictive control implemented in parts of the process.

From a business point of view, Canney (2003) reports that, a typical APC implementation has a payback period of 3–9 months, each generating around a 3–5 % capacity increase. In petrochemical industries, this amounts to a profit increase of $0.05–0.3 per produced barrel. An increased profit of $0.05 per barrel may seem low, but with a current total crude oil production of 84,820,000 barrels per day,[1] it is easy to see the benefits on the economic side.

Model predictive control (MPC) is an APC strategy that was first proposed by industry to simplify optimal control for multivariate processes. The ability of MPC to handle constraints on inputs, outputs and states and the easy application to multivariate systems are the key technical properties which have lead to the success of the control strategy.

Industrial challenges

Even though MPC seems like a promising APC technology, a number of industrial challenges remain. A recurring issue is the fact that MPC performance often drops after a shorter period than the expected payback time. It is observed that initially an MPC operates at a high efficiency but due to changes and modifications of processes, operating strategies, operating conditions or a number of other factors, performance deteriorates. It is not uncommon that performance is reduced to the point that the MPC is manually turned off. Maintaining high performance is identified as a central aspect of APC technology progression by Canney (2003) and Bauer and Craig (2008). Some suggested developments to improve APC benefits include:

- Performance parameters to isolate the source of degradation.
- Monitoring and diagnosis tools.
- Testing, identification and adaptive control.
- Analysis tools that identify opportunities for performance improvement.

In particular, Canney (2003) says

> *"Automated testing and identification, and adaptive control. MPC controller performance is directly related to the accuracy of the process model. These technologies increase the accuracy by producing optimally informative process data and appropriately adjusting the model."*

[1] Production in 2011, International Energy Agency – Oil Market Report, 2012-03-19.

Still, 6 years later, Zhu (2009) reports that the most time consuming and costly part of the commissioning of an industrial MPC deployment is the modeling of the plant. Zhu (2009) also estimates that modeling efforts can take up to 90 % of the cost and time in a typical MPC commissioning. Huge benefits can be made if the modeling process is simplified. Once the MPC is up and running, maintenance of the controller mostly relies on updating the model to account for changes in the plant. This remodeling should be done in a intelligent way so that product quality does not suffer.

Commercial MPC

There are many commercially available implementations of 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 MPC technologies at the time. These are presented according to the model types employed in Figure 1.2. Linear models obtained through system identification are the most frequently used models. However, it is also reported by Qin and Badgwell (2003) that almost all commercial MPC use PRBS signals in the identification part of the commissioning. Furthermore, even for MIMO plants, inputs are most often manipulated one at a time. There is large room for improvement by tailoring the identification experiments to the plant.

1.2 A motivating example

Distillation is a central process found in many chemical and petrochemical industries, illustrated in Figure 1.3. As a simplified example, consider the binary distillation column, described by Skogestad (1997), without re-boiler and condenser. The column has two outputs, a top product and a bottom product, which are required to have a certain purity to be sold. If the average purity of either product drops below the required level, the product cannot be sold.

The system is modeled using a state space model, which has been estimated in a system identification experiment. An MPC is designed using the estimated model. To monitor the performance of the distillation column, a performance indicator is calculated online. If the performance indicator is above a given threshold, both products are within the specifications but if it drops below the threshold, money is lost due to low quality products.

The performance over time is shown in Figure 1.4. Initially, the performance is high and the system produces products with the desired purities. After a while, the

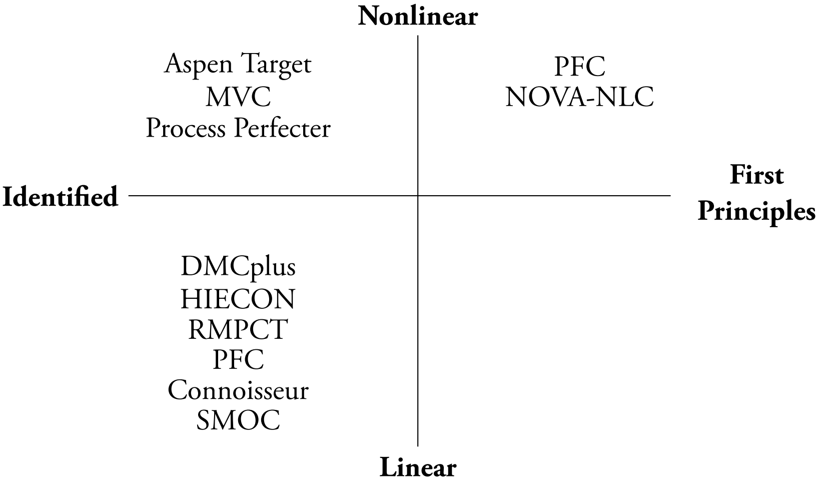


Figure 1.2 Different commercially available MPC technologies classified according to the model type used and the modeling principle. The dominant approach is to use linear models identified from experimental data (identified models).

system dynamics are changed slightly, which affects the directionality of the column. This means that the model in the MPC no longer matches the system and product quality suffers, which is reflected by the drop in the performance indicator.

To restore performance, a closed loop identification experiment is started. Extra excitation is introduced in the system to give sufficiently informative data to model the new dynamics. Since the plant is disturbed by the experiment, the performance drops even more during the identification. However, when the newly identified model is implemented, performance quickly goes back to the level before the plant change. This means that profits are back to the desired level again.

The example is implemented in a Matlab simulation environment using the technologies developed in this thesis. The details of the simulation example are given at the end of Chapter 5. The take home messages at this point are:

– Performance suffers and money is lost when the model used by the controller no longer represents the dynamics of the plant.

– Reidentifying the plant is associated with a cost.

– Performance and profits can be restored with a suitable new model.

Figure 1.3 Distillation columns in operation at an industrial oil refinery. Photo courtesy of Sasol.

1.3 Problem formulation

A model can never capture all aspects of the system it is meant to represent. Neither is it desirable that every detail is captured since this increases the model complexity. What is important is that the relevant aspects are captured; what is relevant depends on the intended model application.

In this thesis, model estimation from experimental data is considered. In particular, the problem of system identification for models used in model predictive control is studied. The quality of the estimated model can be influenced by the design of the identification experiment and the excitation properties of the applied input signal. The problem is formulated as an optimal experiment design problem which has three main goals:

1. The *performance* when the model is used should with high confidence be good.
2. The *cost* incurred due to the identification experiment should be as small as possible.
3. The *constraints* of normal plant operation should be respected.

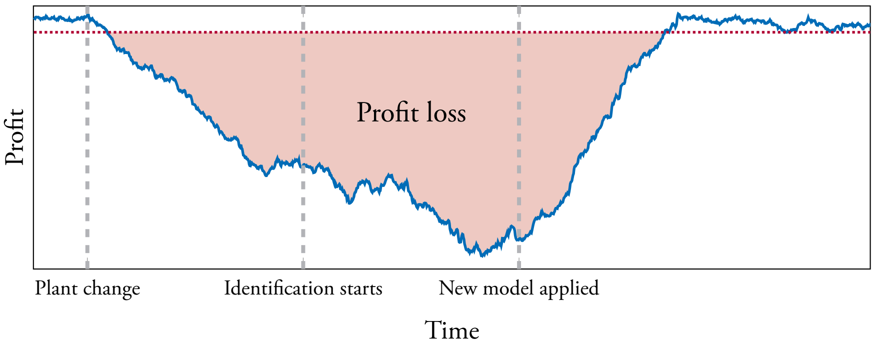


Figure 1.4 Profits from MPC. Initially the MPC results in a high profit level. When the dynamics of the plant change, performance and profits are reduced. To restore performance, a closed-loop identification experiment is performed. The experiment does not come for free and results in further reduction of profit. However, when the new model is used, profits quickly return to an acceptable level.

These goals lead to the conceptual optimization problem

$$\begin{aligned} \underset{\text{input}}{\text{minimize}} \quad & \text{Cost of experiment} \\ \text{subject to} \quad & \text{Performance specifications} \\ & \text{System constraints} \end{aligned} \tag{1.1}$$

The bulk of the thesis, Chapters 3–6, is devoted to this problem.

A second problem considered is that of signal generation under time domain constraints. The motivation is that many currently used optimal experiment design problems are formulated as optimization problems in the frequency domain. Consequently, when the optimal spectrum has been found, a signal with the right spectral properties must be generated. This is not straightforward for signals with amplitude constraints. An MPC based solution to this signal generation problem is presented in Chapter 7.

1.4 Contributions and outline

This section gives an overview of the different chapters of the thesis together with the main contributions and related publications.

Chapter 2 — Background

Theoretical background of systems, experimental modeling and model based control is summarized. First, model predictive control is outlined. Second, the main modeling tool, system identification, is presented. Experiment design and dual control for identification and adaptive control in closed loop is also outlined. Related work, in the field of adaptive and dual MPC is described, as well as reinforcement learning.

Chapter 3 — Experiment design for model predictive control

General considerations on experiment design for system identification of models used for MPC are presented here. The chapter begins with a formal formulation of the experiment design problem under industrially relevant assumptions. The formulation is done in the application-oriented input design framework. A method for MPC relevant experiment design for open loop experiment is presented. The open loop method was previously studied in

> C. A. Larsson (2011). *Toward applications oriented optimal input design with focus on model predictive control.* Licentiate thesis, KTH, Automatic Control.
>
> C. A. Larsson, M. Annergren, and H. Hjalmarsson (2011a). On Optimal Input Design for Model Predictive Control. In *Proceedings of the 50th IEEE Conference on Decision and Control.* Orlando, FL.
>
> C. A. Larsson, C. R. Rojas, and H. Hjalmarsson (2011b). MPC oriented experiment design. In *Proceedings of the 18th IFAC World Congress*. Milano, Italy.

Chapter 4 — Markov decision process formulation of experiment design

The experiment design problem is approached using Markov decision process (MDP) theory. A controller based on a constrained, finite state and action space MDP is constructed. The solution to the optimization can be found using a semidefinite program, which is an extension of the linear programming solution to constrained MDPs. The chapter is based on

C. A. Larsson, A. Ebadat, C. R. Rojas, and H. Hjalmarsson (2014a). An application-oriented approach to optimal control with excitation for closed-loop identification. *European Journal of Control*. To be submitted.

Chapter 5 — Model predictive control formulation of experiment design

Model predictive controllers extended to enable closed loop reidentification are formulated. A constraint on the information matrix related to the system identification experiment is included in the controller formulations, resulting in a controller that is called Model Predictive Control with eXcitation (MPC-X). The inclusion of this constraint gives signals suitable for identification. Simulation studies are included to show the performance of the algorithms and a comparison to existing techniques is also given. The chapter is based on

C. A. Larsson, A. Ebadat, C. R. Rojas, and H. Hjalmarsson (2014a). An application-oriented approach to optimal control with excitation for closed-loop identification. *European Journal of Control*. To be submitted.

C. A. Larsson, M. Annergren, H. Hjalmarsson, C. R. Rojas, X. Bombois, A. Mesbah, and P. E. Modén (2013a). Model predictive control with integrated experiment design for output error systems. In *Proceedings of the 2013 European Control Conference*. Zürich, Switzerland.

Some of the results and the algorithms have also been reported in

H. Hjalmarsson and C. A. Larsson (2012). Deliverable 3.1 — Novel algorithms for least-costly closed-loop re-identification. Autoprofit project.

H. Hjalmarsson, C. A. Larsson, and M. Annergren (2013a). Deliverable 3.2 — Matlab software toolbox. Autoprofit project.

H. Hjalmarsson, C. A. Larsson, P. Hägg, and A. Ebadat (2013b). Deliverable 3.3 — Novel algorithms for productivity preserving testing. Autoprofit project.

M. Annergren, D. Kauven, C. A. Larsson, M. G. Potters, Q. N. Tran, and L. Özkan (2013). On the Way to Autonomous Model Predictive Control: A Distillation Column Simulation Study. In *Proceedings of the 10th IFAC International Symposium on Dynamics and Control of Process Sytsems*. Mumbai, India.

Chapter 6 — Experimental study of MPC-X

An extensive industrial validation campaign of one of the MPC based controllers has been conducted and is reported on here. The algorithm has been implemented and tested on a depropanizer distillation column both in a full plant simulator as well as on the actual plant during normal production. Results from these experiments have previously been reported in

> C. A. Larsson, C. R. Rojas, X. Bombois, and H. Hjalmarsson (2014c). Experimental evaluation of model predictive control with excitation (MPC-X) on an industrial depropanizer. *Journal of Process Control.* Submitted.
>
> H. Guidi, C. A. Larsson, Q. N. Tran, L. Ozkan, and H. Hjalmarsson (2013). Deliverable 6.4 — results on the Sasol validation case. Autoprofit project.

Chapter 7 — Signal generation for constrained systems

It is often desirable to generate a signal with prespecified second order (autocorrelation) properties. A receding horizon algorithm for generation of signals with the desired properties under constraints is presented in this chapter. The algorithm uses ideas from MPC and scenario optimization to ensure the signal properties. Convergence to the correct autocorrelation is proven for a special case and the merits of the algorithm for more general cases are demonstrated in simulation. The chapter is based on the publications

> C. A. Larsson, P. Hägg, and H. Hjamarsson (2014b). Generation of signals with specified second order properties for constrained systems. *International journal of adaptive control and signal processing.* Submitted.
>
> C. A. Larsson, P. Hägg, and H. Hjalmarsson (2013b). Recursive generation of amplitude constrained signals with prescribed autocorrelation sequence. In *Proceedings of the 2013 American Control Conference*. Washington DC, USA.
>
> P. Hägg, C. A. Larsson, and H. Hjalmarsson (2013). Robust and adaptive excitation signal generation for input and output constrained systems. In *Proceedings of the 2013 European Control Conference*. Zürich, Switzerland.

P. Hägg, C. Larsson, A. Ebadat, B. Wahlberg, and H. Hjalmarsson (2014). Input signal generation for constrained multiple-input multiple-output systems. In *Proceedings of 19th IFAC World Congress*. Cape Town, South Africa.

An early version of the algorithm has also been published in

H. Hjalmarsson, C. A. Larsson, P. Hägg, and A. Ebadat (2013b). Deliverable 3.3 — Novel algorithms for productivity preserving testing. Autoprofit project.

Chapter 8 — Conclusions

The thesis is concluded with a summary and discussion of the results. Some future outlooks and suggestions for further research topics are also presented.

Contributions by the author

The contributions of the thesis are principally the results of the author's own work, in collaboration with the respective coauthors. The minimum time controller in Section 5.4 was intially formulated by Afrooz Ebadat. Chapter 7 is the result of close collaboration between the author and Per Hägg and the contributions are believed to be shared equally between the two collaborators.

1.5 Notation

i	Imaginary unit.
$\mathbf{N}$	Set of positive integers.
$\mathbf{R}$	Set of real numbers.
$\mathbf{R}^n$	n-dimensional vector space over $\mathbf{R}$, elements are typically denoted x, y, z and x_i denotes entry i.
$\mathbf{R}^{n\times m}$	Set of real matrices of dimensions $n \times m$, elements are typically denoted $A, B, C, \ldots$, and $A_{i,j}$ denotes entry j of column i.
$\mathbf{S}^+$	Set of positive semidefinite matrices.
$\overline{X}$	Convex hull of the set X.
$\mathcal{U}, \mathcal{X}, \mathcal{Y}$	Constraint sets for inputs, states, and outputs.
$\mathbf{U}, \mathbf{X}, \mathbf{Y}$	Finite sets of of inputs, states, and outputs.
$A^{1/2}$	Hermitian square root of the matrix A.
$\det A$	Determinant of the matrix A.
$\operatorname{diag} A$	Vector with diagonal of the matrix A.
$\operatorname{diag} x$	Matrix with vector x on the main diagonal.
$\operatorname{rank} A$	Rank of the matrix A.
$\operatorname{tr} A$	Trace of the matrix A.
$\lVert x \rVert$	$\sqrt{x^T x}$.
$\lVert x \rVert_A$	$\sqrt{x^T A x}$.
$\lambda_{\min}(A)$	Smallest eigenvalue of the matrix A.
$\lambda_{\max}(A)$	Largest eigenvalue of the matrix A.
$x \geq y$	$x_i \geq y_i,\ i = 1, 2, \ldots, n$ for $x, y \in \mathbf{R}^n$.
$A \succeq B$	$A - B \in \mathbf{S}^+$.
$\mathbb{P}$	Probability of an event.
$\mathbb{E}$	Expectation operator.
δ_{xy}	Dirac measure concentrated on x.
$\operatorname{Cov} x$	Covariance of x, $\operatorname{Cov} x \triangleq \mathbb{E}\left\{(x - \mathbb{E}\{x\})(x - \mathbb{E}\{x\})^T\right\}$.
$\mathcal{N}(\mu, \Lambda)$	Gaussian distribution with mean μ and covariance Λ.
$\chi^2(n)$	χ^2 distribution with n degrees of freedom.
$\mathcal{I}_1^N$	Information matrix for N data.
$\mathcal{I}$	Average per-sample information matrix.
θ	Parameter vector.
$\hat{\theta}_N$	Estimated parameter vector from N data samples.
$\mathcal{M}(\theta)$	Model parameterized by the vector θ.
$\mathcal{S}$	System.

1.6 Acronyms

APC	Advanced process control
ARMAX	Autoregressive moving-average with exogenous input
ARX	Autoregressive with exogenous input
CMDP	Constrained Markov decision process
CV	Controlled variable
DV	Disturbance variable
FCC	Fluid catalytic cracker
FIR	Finite impulse response
LMI	Linear matrix inequality
LTI	Linear, time invariant
MDP	Markov decision process
MIMO	Multiple input, multiple output
MPC	Model predictive control
MPC-X	Model predictive control with excitation
MPCI	Model predictive control for identification
MV	Manipulated variable
OE	Output error
PE-MPC	Persistently exciting model predictive control
PEM	Prediction error method
PID	Proportional–derivative–integral
PRBS	Pseudo random binary sequence
SCC	Synfuels catalytic cracker
SDP	Semi-definite program
SISO	Single input, single output
SP	Setpoint

Chapter 2
Background

Model based control methods have attracted a lot of attention in the automatic control literature and are becoming increasingly popular in industry as well. Several theoretical concepts are required to formulate a practically viable predictive control system to be used in an industrial application. In particular, a model based control theory needs to be formulated and methods for obtaining the models used in the control design are necessary. The purpose of this chapter is to introduce the necessary theoretical background for the remainder of the thesis. The chapter also serves as a survey of related research areas.

2.1 Systems and models

The systems considered in the thesis are linear time invariant (LTI) multiple input, multiple output (MIMO), discrete-time systems of the form

$$\mathcal{S} : \begin{cases} x_{t+1} = Ax_t + Bu_t + v_t, \\ y_t = Cx_t + w_t, \end{cases} \tag{2.1}$$

where $x_t \in \mathbf{R}^n$ is the state, $u_t \in \mathbf{R}^m$ is the input, $y_t \in \mathbf{R}^p$ is the output and $w_t \in \mathbf{R}^p$ is measurement noise. The general setup is shown in Figure 2.1. The system is driven by the process disturbances, $v_t \in \mathbf{R}^n$, which cannot be directly controlled and the input, which is chosen by the controller. The controller bases the input choice on information about the plant from feedback measurements of the output and reference signals. Several different assumptions on the nature of the noise processes are used in different areas. Three common possibilities, used in the thesis, are given here.

Assumption 2.1 (Constant output disturbance)

$$v_t \equiv 0, \qquad\qquad w_t \equiv w.$$

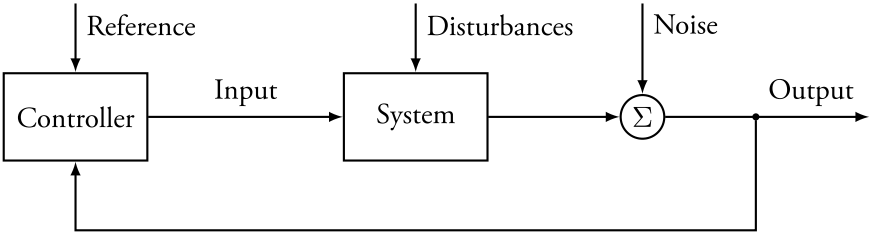


Figure 2.1 The general system setup. The system is driven by uncontrollable disturbances and controllable inputs. The input is chosen by a controller based on feedback information from the system and reference values.

Assumption 2.2 (Bounded disturbances)

$$v_t \in \mathcal{V}_t \qquad\qquad w_t \equiv 0,$$

where $\mathcal{V}_t$ is a closed and bounded set containing the origin.

Assumption 2.3 (Gaussian noise and disturbance)

$$\begin{bmatrix} v_t \\ w_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda_v & 0 \\ 0 & \Lambda_w \end{bmatrix}\right).$$

The first assumption is very common in the MPC field, and according to Maciejowski (2002) it is used in most commercial MPC products. Assumption 2.2 is typically used in the development of robust MPC and Assumption 2.3 is standard in system identification theory and many other fields of control theory.

The system is modeled by a state space model of the form

$$\mathcal{M}(\theta) : \begin{cases} x_{t+1} = A(\theta)x_t + B(\theta)u_t + K(\theta)e_t, \\ \quad\; y_t = C(\theta)x_t + e_t. \end{cases} \tag{2.2}$$

The choice of the matrices A, B, C and K, and the properties of e_t, depend on the adopted model and noise assumptions. For example, under Assumption 2.3, e_t is a zero-mean, Gaussian white noise process which means that the noise covariance Λ_e (in the SISO case, λ_e) and K are given by the Kalman filter.

The model is parameterized by $\theta \in \mathbf{R}^{n_\theta}$, which is a (often unknown) vector of parameters. It is assumed that the model $\mathcal{M}(\theta)$ can capture the true dynamics of the system. This means that there is a vector θ_o such that $\mathcal{S} = \mathcal{M}(\theta_o)$, meaning that the model and system have the same input–output properties. How an estimate of θ can be found from measured data is discussed in Section 2.3.

2.2 Model predictive control

At the core of any MPC implementation is a model of the process that is to be controlled. Typically, MPC uses a deterministic model for predicting the results of the control action and disturbances. A common choice is to use a linear, discrete time, state space model of the form

$$\begin{cases} x_{t+1} = Ax_t + Bu_t, \\ \quad y_t = Cx_t. \end{cases} \tag{2.3}$$

It is possible to use other types of models, for example transfer function or step response models. Versions of MPC with nonlinear models are also sometimes used. This thesis exclusively treats the linear, state space formulation.

The model (2.3) is used to predict future outputs and the control signals are 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 free input samples in the optimization. A quadratic cost function, C_t, is commonly used in the controller, for example

$$C_t \triangleq \sum_{k=1}^{N_y} \|\hat{y}_k - r_{t+k}\|_Q^2 + \sum_{k=0}^{N_u-1} \left(\|\Delta\hat{u}_k\|_R^2 + \|\hat{u}_k\|_S^2\right), \tag{2.4}$$

where $\hat{y}_k$ and $\hat{u}_k$ are the predicted output and input signals, r_k is the reference signal, and $\Delta\hat{u}_k = \hat{u}_k - \hat{u}_{k-1}$ The matrices Q, R and S are tunable weights. The prediction horizon is denoted N_y and the control horizon N_u. For each time instant t, an input sequence is found by solving the optimization problem

$$\begin{array}{lll} \underset{\{\hat{u}_k\}_{k=0}^{N_y-1}}{\text{minimize}} & C_t & \\ \text{subject to} & \hat{x}_{k+1} = A\hat{x}_k + B\hat{u}_k, & k = 0, 1, \ldots, N_y - 1, \\ & \hat{y}_k = C\hat{x}_k, & k = 1, 2, \ldots, N_y, \\ & \hat{x}_0 = x_t, & \\ & \hat{u}_0 = u_{t-1}, & \\ & \hat{x}_k \in \mathcal{X}, \ \hat{u}_k \in \mathcal{U}, \ \hat{y}_k \in \mathcal{Y}. & \end{array} \tag{2.5}$$

Here, u_{t-1} is the input applied to the plant at time $t - 1$, and $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{Y}$ are the constraint sets for the states, outputs and inputs respectively. In the case $N_u < N_y$, some assumption on the inputs past the control horizon has to be made. Two

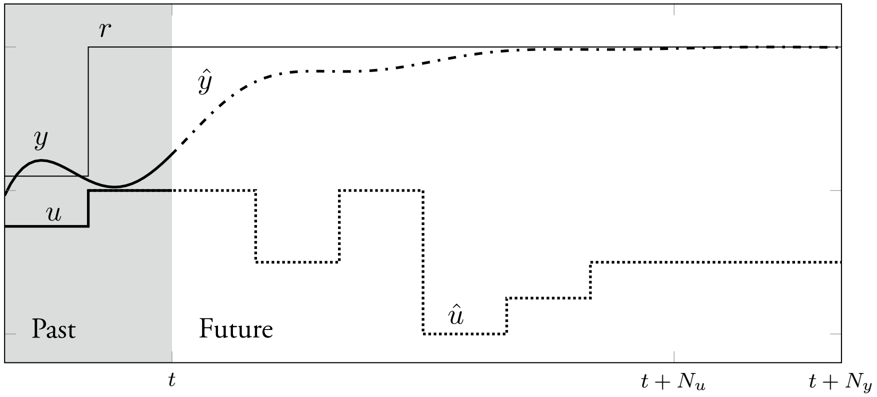


Figure 2.2 The receding horizon idea. The MPC is used to control the system to follow the reference trajectory, r, (——). At time t, the plant output, y, is predicted, $\hat{y}$, (· - · -) over the prediction horizon N_y. The predictions depend on the current control input, u, and the predicted control inputs, $\hat{u}$, (·········) over the control horizon N_u. The optimal input sequence is calculated and the first input is applied to the plant. The procedure is repeated at time $t + 1$.

possibilities are to assume that the input is constant, that is $\Delta \hat{u}_k = 0, k > N_u$, or to assume that the input is zero, that is $\hat{u}_k = 0, k > N_u$.

The solution to the optimization problem (2.5) gives a sequence of open loop optimal inputs over the control horizon. However, at time t, only the first input of the sequence is actually applied to the system, meaning that $u_t = \hat{u}_1$. The horizons are shifted one sample forward and the optimization starts over in the next time instant. This principle is illustrated in Figure 2.2.

When the horizons have been shifted one step forward, a measurement of the state is taken and used in the next optimization. This is the way that feedback enters into the receding horizon formulation. If the state, x_t, is not available for measurement, an estimate, $\hat{x}_{t|t}$, can be obtained using an observer and $\hat{x}_{t|t}$ replaces x_t in the optimization (2.5).

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, to some extent, be handled by incorporating integral action into the MPC. Modeling errors may 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 central importance in MPC applications.

Disturbances

When the state cannot be directly measured, an observer to estimate the state is needed. The way this is handled depends on which disturbance model is used. The three common cases introduced in the previous section are presented below.

Constant output disturbance

The unknown but constant disturbance case is typically handled using a deadbeat observer. This is achieved by defining the extended state space model

$$\begin{cases} \begin{bmatrix} x_{t+1} \\ w_{t+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_t \\ w_t \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ I \end{bmatrix} e_t, \\ \qquad y_t = \begin{bmatrix} C & I \end{bmatrix} \begin{bmatrix} x_t \\ w_t \end{bmatrix} + e_t, \end{cases} \tag{2.6}$$

and using this model to predict the response of the system. This strategy is also useful for offset-free tracking when the gain of the model is different from the true system gain as this introduces integration in the loop.

Gaussian noise and disturbance

In the Gaussian noise and disturbance case, the mean square optimal state estimate is given by the Kalman filter, outlined, for example, by Söderström (2002). In practice, however, the observer gain is often used as a tuning parameter in the MPC algorithm.

The predictions for the optimization can be done in a few different ways. One possibility is to replace the state measurement x_t in (2.5) by the estimate $\hat{x}_{t|t}$ and similarly replace the predictions of the noise and disturbances by their conditional expectations. Since v_t and w_t are assumed to be zero mean, this means that the remaining parts of the optimization (2.5) remain unaltered. This can be seen as a certainty equivalence formulation of MPC.

Another possibility is to keep the stochastic description of the system and reformulate the optimization (2.5) in terms of expected values. This leads to what is known as *stochastic* MPC, which is a very actively researched topic in predictive control. See, for example, the work by Hokayem et al. (2012) and Schildbach et al. (2013) and the references therein.

Bounded disturbances

The bounded disturbance case, Assumption 2.2, together with $C = I$, which means that the state is directly measurable, is common in the robust MPC formulation. The goal is to find a computationally tractable, stabilizing controller such that state and inputs satisfy constraints for all possible bounded disturbances in the set $\mathcal{V}$. For an overview of the area, see Mayne et al. (2000) or Maciejowski (2002). Using a feedback formulation decreases the conservativeness compared to using an open-loop formulation but leads to more involved optimization. Goulart et al. (2006) provide a reparameterization of the MPC problem that gives a convex formulation of the state feedback control problem.

Alternative formulations of predictive control

The MPC formulation (2.5) uses a state space description of the system, which currently seems to be the prevalent technique. However, other predictive control formulations have been proposed. The most notable difference between the formulations is in the system description. The earliest predictive control schemes, MPHC by Richalet et al. (1978) and DMC by Cutler and Ramaker (1980) use step and impulse response models of the systems. Predictive control using transfer function models can be found in the Generalized Predictive Control (GPC) scheme by Clarke et al. (1987). The GPC idea has become a popular concept, see, for example, the book by Camacho and Bordons (2004) for a thorough treatment.

2.3 System identification

The field of system identification considers the problem of making models of systems using experimental data. There are many methods for identification of the model $\mathcal{M}$. Such methods can be non-parametric or parametric and can be applied in both the time and frequency domains. Which method is used depends on the 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), while Van Overschee and De Moor (1996) covers subspace identification. This thesis only considers parametric, time-domain identification. The estimates of the model parameters θ, are found using the prediction error method (PEM) as presented by Ljung (1999). This method and the properties of the resulting estimates are described next.

Under Assumption 2.3, when a model of the form (2.2) is used, the stationary one step ahead predictor is given by

$$\begin{cases} \hat{x}_{t+1|t}(\theta) = A(\theta)\hat{x}_{t|t-1}(\theta) + B(\theta)u_t + K(\theta)(y_t - \hat{y}_{t|t-1}(\theta)), \\ \hat{y}_{t|t-1}(\theta) = C(\theta)\hat{x}_{t|t-1}(\theta). \end{cases} \tag{2.7}$$

From N samples of input–output data, $Z_N \triangleq \{y_t, u_t\}_{t=1}^{N}$, the parameter estimate is most commonly found as

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \left(y_t - \hat{y}_{t|t-1}(\theta)\right)^T \Lambda_e^{-1} \left(y_t - \hat{y}_{t|t-1}(\theta)\right). \tag{2.8}$$

When the noise covariance, Λ_e, is not known, Goodwin and Payne (1977) suggest to instead use

$$\hat{\theta}_N = \arg\min_{\theta} \det \frac{1}{N} \sum_{t=1}^{N} \left(y_t - \hat{y}_{t|t-1}(\theta)\right)^T \left(y_t - \hat{y}_{t|t-1}(\theta)\right),$$

which for Gaussian distributed noise gives the Maximum Likelihood estimate of the parameter vector.

Ljung and Caines (1979) have shown, under very general conditions, that for the choice (2.8) it holds asymptotically in N that

$$\mathcal{I}_1^N(\theta_o)^{1/2}(\hat{\theta}_N - \theta_o) \xrightarrow{d} \mathcal{N}(0, I), \tag{2.9a}$$

$$\mathcal{I}_1^N(\theta_o) \triangleq \sum_{t=1}^{N} \mathbb{E}\left\{\psi_t(\theta_o)\Lambda_e^{-1}\psi_t(\theta_o)^T\right\}, \tag{2.9b}$$

$$\psi_t^i(\theta) \triangleq \frac{\partial \hat{y}_{t|t-1}(\theta)}{\partial \theta_i}, \tag{2.9c}$$

$$\psi_t(\theta) \triangleq \begin{bmatrix}\psi_t^1(\theta) & \cdots & \psi_t^{n_\theta}(\theta)\end{bmatrix}^T, \tag{2.9d}$$

where $\xrightarrow{d}$ denotes convergence in distribution and $\mathcal{I}_1^N$ is the information matrix. Hence, the PEM estimate is consistent in probability and it achieves the Cramér-Rao lower bound. Furthermore, the distribution of the estimates implies that, asymptotically in N,

$$\hat{\theta}_N \in U(\alpha) = \left\{\theta \;\middle|\; [\theta - \theta_o]^T \mathcal{I}_1^N(\theta_o) [\theta - \theta_o] \leq \chi_\alpha^2(n_\theta)\right\}, \tag{2.10}$$

with probability α. Here $\chi_\alpha^2(n)$ is the α-percentile of the χ^2-distribution with n degrees of freedom.

The information matrix can be calculated from a state space description of the sensitivities of the predictor (2.7), shown, for example, by Ljung and Söderström

(1983). Consider the derivatives of the predictor (2.7), with respect to θ, given by

$$\begin{cases} \begin{aligned} \frac{\partial \hat{x}_{t+1|t}}{\partial \theta_i} =& \frac{\partial (A-KC)}{\partial \theta_i}\hat{x}_{t|t-1} + (A-KC)\frac{\partial \hat{x}_{t|t-1}}{\partial \theta_i} \\ &+ \frac{\partial B}{\partial \theta_i}u_t + \frac{\partial K}{\partial \theta_i}Cx_t + \frac{\partial K}{\partial \theta_i}w_t, \\ \frac{\partial \hat{y}_{t|t-1}}{\partial \theta_i} =& \frac{\partial C}{\partial \theta_i}\hat{x}_{t|t-1} + C\frac{\partial \hat{x}_{t|t-1}}{\partial \theta_i}. \end{aligned} \end{cases}$$

Introduce the matrices and extended state vector

$$\begin{aligned} \mathcal{A} &\triangleq \begin{bmatrix} A & 0 & 0 & 0 & 0 \\ KC & A-KC & 0 & 0 & 0 \\ \frac{\partial K}{\partial \theta_1}C & \frac{\partial (A-KC)}{\partial \theta_1} & A-KC & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ \frac{\partial K}{\partial \theta_{n_\theta}}C & \frac{\partial (A-KC)}{\partial \theta_{n_\theta}} & 0 & 0 & A-KC \end{bmatrix}, \\ \mathcal{B} &\triangleq \begin{bmatrix} B \\ B \\ \frac{\partial B}{\partial \theta_1} \\ \vdots \\ \frac{\partial B}{\partial \theta_{n_\theta}} \end{bmatrix}, \ \mathcal{K} \triangleq \begin{bmatrix} I & 0 \\ 0 & K \\ 0 & \frac{\partial K}{\partial \theta_1} \\ \vdots & \vdots \\ 0 & \frac{\partial K}{\partial \theta_{n_\theta}} \end{bmatrix}, \\ \mathcal{C} &\triangleq \begin{bmatrix} 0 & \frac{\partial C}{\partial \theta_1} & C & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \frac{\partial C}{\partial \theta_{n_\theta}} & 0 & 0 & C \end{bmatrix}, \\ \xi_t &\triangleq \begin{bmatrix} x_t & \hat{x}_{t|t-1} & \frac{\partial \hat{x}_{t|t-1}}{\partial \theta_1} & \cdots & \frac{\partial \hat{x}_{t|t-1}}{\partial \theta_{n_\theta}} \end{bmatrix}^T, \\ \bar{\psi}_t &\triangleq \begin{bmatrix} \psi_t^1(\theta)^T & \cdots & \psi_t^{n_\theta}(\theta)^T \end{bmatrix}^T, \end{aligned} \tag{2.11}$$

then

$$\begin{cases} \xi_{t+1} = \mathcal{A}\xi_t + \mathcal{B}u_t + \mathcal{K}\begin{bmatrix} v_t \\ w_t \end{bmatrix}, \\ \bar{\psi}_t = \mathcal{C}\xi_t \end{cases} \tag{2.12}$$

and the terms of $\mathcal{I}_1^N$ can be found from

$$\begin{aligned}&\mathbb{E}\left\{\psi_t(\theta)\Lambda_e^{-1}\psi_t(\theta)^T\right\}_{(i,j)}\\ &\qquad = \mathbb{E}\left\{\psi_t^i(\theta)^T\Lambda_e^{-1}\psi_t^j\right\} = \operatorname{tr}\left[\mathbb{E}\left\{\psi_t^j(\theta)\psi_t^i(\theta)^T\right\}\Lambda_e^{-1}\right]. \qquad (2.13)\end{aligned}$$

For a given data record, Z_N, the information matrix can be estimated by

$$I_1^N(\theta) = \sum_{t=1}^{N} \psi_t(\theta)\Lambda_e^{-1}\psi_t(\theta)^T. \qquad (2.14)$$

In fact, it holds under fairly mild conditions that

$$\lim_{N\to\infty}\frac{1}{N}I_1^N(\theta) = \lim_{N\to\infty}\frac{1}{N}\mathcal{I}_1^N(\theta), \text{ almost surely.} \qquad (2.15)$$

When the limit exists,

$$\mathcal{I}(\theta) \triangleq \lim_{N\to\infty}\frac{1}{N}\mathcal{I}_1^N(\theta) \qquad (2.15')$$

is the average per-sample information matrix.

Open- versus closed-loop identification

There is in principle no difference between open- and closed-loop identification from the data Z_N if PEM is used. The consistency results and asymptotic distribution of the estimates hold for open- and closed-loop experiments as long as the system is in the model set (there exists θ_o such that $\mathcal{S} = \mathcal{M}(\theta_o)$), the data are informative, and some technical assumptions on the feedback mechanism and the resulting closed loop. These properties may not hold for other methods, for example, nonparametric methods typically fail due to the correlation between the input and the driving noise resulting from the feedback.

Informativeness of the data is what often fails in closed-loop identification. A simple example shows what can go wrong.

Example 2.1 (Closed-loop identification with simple regulator)
The closed-loop system

$$\begin{cases} y_t = -ay_{t-1} + bu_{t-1} + e_t, \\ u_t = -ky_t, \end{cases}$$

where e_t *is white noise, is identified. Since the closed-loop system is given by*

$$y_t = -(a + bk)y_{t-1} + e_t,$$

any parameter estimates

$$\hat{a} = a - \alpha k, \qquad \hat{b} = b + \alpha, \qquad \alpha \in \mathbf{R},$$

describe the input–output behavior of the closed loop. $\Diamond$

The example shows that, even though the input signal is persistently exciting, the closed-loop data are not informative. The reason is essentially that the proportional controller is too simple. Fortunately, the situation is easily solved by choosing a more complex controller or adding a persistently exciting reference signal. The issue of informative data has been considered by Söderström et al. (1976), for example. More recently, Gevers et al. (2009) have analyzed the conditions on signal richness and controller complexity for informative experiments in closed loop.

The assumption that the model set contains the true system is more important in closed-loop identification than in open-loop identification. For experiments in open loop, a consistent estimate of the system is obtained when the noise and system models are independently parameterized. This is not the case in closed loop, where an incorrect noise model gives biased system estimates. Forssell and Ljung (1999) summarize PEM in closed-loop identification well.

When the data are informative and the true system is in the model set, there are situations when closed-loop identification is superior to open-loop identification. For example, Bombois et al. (2005) show that, under certain conditions and for independently parameterized plant and noise models, the variance of the estimated plant model is lower in a closed-loop experiment while the opposite is true for the estimates noise model. Furthermore, closed-loop identification is shown to be advantageous when the identified model is used for control design, by Hjalmarsson et al. (1996), and whenever there are constraints on the output power by Forssell and Ljung (1998). On the other hand, if there are only constraints on the input power, open-loop identification is to be preferred. These issues have also been analyzed by Agüero and Goodwin (2007), who show that open-loop designs are optimal for any type of input-constrained experiments.

2.4 Input design

Optimal design of experiments has a long tradition within the statistics literature. Many results have an important impact on the field of input design in system identification. For example, Kiefer (1959), Kiefer and Wolfowitz (1959), Karlin and Studden (1966) and Fedorov (1972) have all made important contributions in optimal experiment design for linear regressions.

Experiment design for dynamic systems essentially started to attract attention in the 1970s. The developments are well described by Mehra (1974), Goodwin and Payne (1977) and Zarrop (1979). For the experiment design problem in system identification, it is often assumed that the variance is the only source of error. This is motivated by the limiting distribution of the prediction error estimates (2.9). The focus was on open loop experiments and some scalar criterion of the average per-sample information matrix, $\mathcal{I}$, was maximized. The most common criteria are

A-optimality: $\min \operatorname{tr} \mathcal{I}^{-1}$,

D-optimality: $\max \det \mathcal{I}$,

E-optimality: $\max \lambda_{\text{min}}(\mathcal{I}^{-1})$.

The optimal experiment design problems were studied both in the time and the frequency domain. The time domain solutions are finite sequences of input data while the frequency domain formulation finds the optimal input spectrum. An important result is that the set of average information matrices for power constrained inputs is convex in the input spectrum. A second important result is that, if n parameters are estimated, any average per-sample information matrix, $\mathcal{I}$, can be obtained using a signal containing no more than $n(n+1)/2+1$ sinusoidal components. For specific systems, the number of components can be reduced further.

Ljung (1985) derived expressions for the variance of transfer function estimates, asymptotic in both data and model order. For systems of the form

$$y_t = G(q,\theta)u_t + H(q,\theta)e_t,$$

the covariance of the estimates can be approximated as

$$\operatorname{Cov}\begin{bmatrix} G(e^{i\omega},\hat{\theta}_N) \\ H(e^{i\omega},\hat{\theta}_N) \end{bmatrix} \approx \frac{n}{N}\lambda_e |H(e^{i\omega},\theta_o)|^2 \begin{bmatrix} \Phi_u(\omega) & \Phi_{ue}(\omega) \\ \Phi_{ue}(-\omega) & \lambda_e \end{bmatrix}^{-1},$$

where m is the model order, ϕ_u is the input spectrum, ϕ_{ue} is the cross spectrum between the input and the noise and λ_e is the variance of e. Based on this approximation, input design where the intended use of the model was taken into account appeared. See for example the works by Gevers and Ljung (1986), Hjalmarsson et al. (1996) and Forssell and Ljung (2000).

Along the same lines, the area of *identification for control* emerged. The question addressed was how to design the experiment and identification criterion when the model is to be used for model-based control design. Typically, some distance between nominal performance and what is achieved with the identified model was minimized. The main realization was that closed loop identification is better and the controller used during identification should match the model based controller one wants to obtain.

More recent developments in input design are mostly in the *least-costly identification* framework, introduced by Bombois et al. (2004). The framework introduces a cost for the identification which should be minimized subject to constraints on the model quality. It can be seen as the dual formulation of the identification for control idea. Rojas et al. (2008) have established equivalence between the least-costly framework and the more traditional experiment design. Bombois et al. (2006) developed the idea further in an H_∞ control framework, and Barenthin et al. (2005) and Bombois and Scorletti (2012) present some nice examples of the framework. Hjalmarsson (2005) discussed the idea that a good experiment reveals the important process properties while less important properties and attenuated. Hjalmarsson (2009) extended the least-costly experiment design paradigm to other control performance measures in the *application-oriented input design* framework.

Significant developments have been made in the application-oriented and least-costly contexts in terms of the possible experiment design criteria and constraints. Frequency-by-frequency constraints can be handled and solutions exist for open- and closed-loop design with fixed or free controllers. The identification is assumed to be performed using an input of the form

$$u_t = r_t + C(q)y_t, \tag{2.16}$$

where r_t and $C(q)$ are the design variables of the experiment design in the most general form. It is, however, easier to solve the problem in terms of the spectra Φ_u and Φ_{ue}, which have a one-to-one correspondence to Φ_r and $C(q)$.

Research has focused on formulations of the optimization problem as semidefinite programs. In principle, two approaches exist, depending on the parameterization of the input spectrum. One possibility is a *finite-dimensional spectrum parameterization*, which restricts the spectrum to lie in a finite-dimensional subspace of the space of all spectra and therefore may result in a suboptimal solution. On the other hand, frequency-by-frequency constraints can be included in the design. The other is a *partial* *correlation parameterization*, which parameterizes the entire space of spectra in terms of a finite number of generalized moments and therefore does not result in suboptimal solutions. Jansson and Hjalmarsson (2005) present a unified framework for convex optimization formulations of input design problems based finite parameterizations. They also show how a number of quality constraints can be included in the design using a finite-dimensional spectrum parameterization of the input spectrum. The joint reference–controller input design problem, when the input is given by (2.16), is solved for finite-dimensional parameterization by Hjalmarsson and Jansson (2008) and for the partial correlation parameterization by Hildebrand et al. (2010). Hildebrand and Gevers (2013) further extend the partial correlation parameterization to MIMO systems and simplify the solution procedure by considering a central extension of the optimal generalized moments.

A restriction with all the methods based on quasi-stationary signals and spectral characterizations is their inability to take time domain constraints into consideration during the design. One possibility is to include such constraints in the generation of the signal, for example by using binary signals. Liu and Munson (1982), Boufounos (2007), Rojas et al. (2007) have proposed algorithms for this. However, de Carvalho and Clark (1983) show that the binary signals cannot have arbitrary covariance sequence and therefore not all spectra may be realizable using a binary signal. Manchester (2010) introduces a method for convex relaxations of the time domain input design problem, which will be used later in the thesis.

2.5 Dual and adaptive MPC

The concept of dual control was introduced by Feldbaum (1960–61). The central idea is that, if an unknown or uncertain system is to be controlled, the controller necessarily has two objectives. On the one hand, the controller must probe the system to learn the dynamics. On the other hand, the controller should control the system as well as possible. These two goals are in conflict: probing introduces a disturbance in the system which gives worse control performance and vice versa. A controller which generates an input suitable for both probing and control is said to have dual properties.

Adaptive controllers update the parameter estimates used for the control design online. Although dual controllers facilitate such updates, adaptive controllers do not necessarily have dual properties. In such cases, the control action is not designed to reduce the uncertainty of the estimates and any improvement in the estimates is due to the extra data available for estimation. Wittenmark (1995) and Filatov and Unbehauen (2000) survey the dual and adaptive control fields.

Adaptive MPC

Many adaptive versions of predictive controllers without dual properties have been proposed. The MHPC, DMC and GPC algorithms have all been used in an adaptive setting. There are also the Extended Horizon Adaptive Control (EHAC) by Ydstie (1984), the Extended Prediction Self-Adaptive Control by De Keyser and Van Cauwenberghe (1979) and the Multistep Multivariable Adaptive Regulator (MUSMAR) by Menga and Mosca (1980). All these approaches use a certainty equivalence formulation where recursive least-squares estimates are used as models in the predictions. No dual properties are added to the controllers. A comparative study of some of the algorithms has been done by De Keyser et al. (1988).

Dual MPC

Several algorithms which extend MPC with dual properties are based on adding a constraint which guarantees persistence of excitation in the input calculated by the controller. These algorithms are closely related to the developments in this thesis and are discussed in some detail here. To formulate the algorithms, a definition of a persistently exciting signal is needed.

Definition 2.1 (Persistence of excitation) A sequence of vectors $\{\phi_k, k = 1, 2, \dots,\}$ is *persistently exciting* if there exist real numbers $\rho_1 > 0$ and $\rho_2 > 0$ and an integer P such that

$$\rho_1 I \preceq \sum_{j=k}^{P+k-1} \phi_j \phi_j^T \preceq \rho_2 I, \quad \text{for all } k.$$

For systems of the form

$$\begin{aligned} y_t &= \sum_{k=1}^{n_b} b_k u_{t-k} + e_t = \theta^T \phi_t + e_t, \\ \theta &= \begin{bmatrix} b_1 & \cdots & b_{n_b} \end{bmatrix}^T, \\ \phi_t &= \begin{bmatrix} u_{t-1} & \cdots & u_{t-n_b} \end{bmatrix}, \end{aligned} \tag{2.17}$$

persistence of excitation guarantees consistent estimation of θ, for example noted by Bitmead (1984). For systems where past outputs y_t are included in the regressors ϕ_t, sufficiently rich inputs can often guarantee persistence of excitation. This issue is investigated by, for instance, Green and Moore (1986). Finally, Lai and Wei (1982) have derived conditions for consistent estimation of the parameters θ in the case of stochastic regressors (2.17).

Based on Definition 2.1, different constraints on the input can be included in MPC. The upper bound is guaranteed by the amplitude constraints of the input while the lower bound needs to be considered by the MPC. The exact formulation of the constraint differs in the different approaches. The choice of the lower bound ρ_1 becomes an application dependent design choice (like the choice of γ in the application-oriented schemes) and is often a difficult one.

MPCI

Genceli and Nikolaou (1996) have proposed model predictive control with simultaneous model identification (MPCI). The MPC is modified to include the constraint

$$\sum_{k=t}^{t+N_u} \phi_k \phi_k^T \succeq (\rho_1 - \mu) I, \tag{2.18}$$

which forces the input to be persistently exciting over the control horizon N_u. Beyond the control horizon, MPCI constrains the input sequence to be periodic. The extra tuning parameters are ρ_1, which sets the level of the excitation, and μ, which is used for constraint softening with the ideal value $\mu = 0$. The resulting optimization problem is nonconvex, but numerical schemes for finding local solutions are proposed.

Multiobjective MPC with identification

Aggelogiannaki and Sarimveis (2006) have formulated a multiobjective MPC where the excitation is included as the top priority objective. The constraint (2.18) from MPCI is used but, in a first step, an optimal μ is found as

$$\mu^\star = \arg\min_{u,\mu} \mu \tag{2.19}$$

subject to the regular MPC constraints and the constraint (2.18). The other MPC objectives are then optimized sequentially with $\mu^\star$ imposed as a constraint. The idea is to remove the possibility of an infeasible MPC and at the same time get the highest possible excitation. The problem formulation is nonconvex and is solved using an evolutionary algorithm.

PE-MPC

Marafioti (2010) introduced persistently exciting model predictive control (PE-MPC), which in a way reverses the formulation of MPCI. Instead of ensuring persistence of excitation over the future control horizon, PE-MPC uses a *backward* window of P samples. More formally, the constraint

$$\sum_{k=0}^{P-1} \phi_{t-k+1}\phi_{t-k+1}^T \succeq \rho_1 I, \tag{2.20}$$

is added to the MPC formulation at time instant t. From (2.17) it follows that

$$\sum_{k=0}^{P-1} \phi_{t-k+1}\phi_{t-k+1}^T = \sum_{k=1}^{P-1} \begin{bmatrix} u_{t-k} \\ u_{t-1-k} \\ \vdots \\ u_{t-n_b-k} \end{bmatrix} \begin{bmatrix} u_{t-k} \\ u_{t-1-k} \\ \vdots \\ u_{t-n_b-k} \end{bmatrix}^T + \begin{bmatrix} u_t \\ u_{t-1} \\ \vdots \\ u_{t-n_b} \end{bmatrix} \begin{bmatrix} u_t \\ u_{t-1} \\ \vdots \\ u_{t-n_b} \end{bmatrix}^T,$$

and hence u_t is the only decision variable in the constraint (2.20), which therefore applies only to the next input sample. PE-MPC results in a nonconvex optimization problem but can, for certain systems, be solved using two convex quadratic programs. The extra tuning parameters of PE-MPC are ρ_1 and the horizon length P. Marafioti (2010) shows that PE-MPC, if feasible, admits a (possibly sub-optimal) P-periodic solution.

Dual Control by Information Maximization

Rathouský and Havlena (2011) propose adding the constraint

$$\sum_{k=t}^{t+M} \phi_k\phi_k^T \succeq \rho_1 I, \quad n_b \leq M \ll N_u, \tag{2.21}$$

and consider several numerical schemes for finding approximate solutions to the resulting optimization. Choosing ρ_1 and M is the main concern. An alternative solution, where the maximum ρ_1 is found subject to constraints on the controller performance loss, is also proposed. Rathouský and Havlena (2013) further develop the latter idea and propose an approximation of the original nonconvex optimization problem. The approximation finds an upper bound on the smallest eigenvalue of the information matrix by considering a set of quadratic forms, which is computationally simpler than directly computing the eigenvalue and can approximate the original problem to arbitrary precision. The authors also show the benefit of multi-step predictions (over single-step predictions) in the constraint (2.21) since this allows for uniform excitation of the parameter space.

Other dual MPC formulations

A number of other dual MPC formulations have also been proposed. Three of these MPC formulations with dual properties are summarized here.

González et al. (2013) propose an MPC formulation suitable for closed-loop reidentification. Focus is on guaranteeing stability of the closed loop while having a persistently exciting input. This is achieved by constructing a target invariant set where excitation is possible.

Žáčeková et al. (2013) consider MPC with persistently exciting inputs specifically for controllers with zone control, often found in, for example, temperature control. Informative data is achieved by maximizing the smallest eigenvalue of the information matrix in the MPC formulation.

Heirung et al. (2013) present yet another MPC with dual effect. The optimization problem is extended with a constraint relating to the covariance of least-squares estimates of the model parameters. The dual effect comes from including a term related to the predicted future parameter error covariance in the optimization objective.

Persistence of excitation versus application-oriented experiments

The dual MPC formulations presented in this section are all algorithms that generate input signals suitable for estimation of the model *parameters*. Inputs that are persistently exciting guarantee consistent parameter estimates. The methods developed in this thesis are aimed at generating input signals suitable for estimation of a *model* for control design. This is in line with the least-costly identification paradigm; the intended use of the model is explicitly taken into account when designing the experiment. By doing so, the system is not excited unnecessarily to reveal properties that are not important for good control performance and thereby the cost of the experiment can be kept low and the modeling task is simplified.

2.6 Reinforcement learning

The field of reinforcement learning is an area in machine learning that is related to the field of dual and adaptive control in the area of automatic control. Both fields deal with the problem of taking the right action while, at the same time, learning the results of the actions. In reinforcement learning, an agent balances *exploration*, where the agent tries to learn about its environment, and *exploitation*, where the obtained knowledge is used to achieve the desired result. This is the same balance that a controller with dual properties aims at achieving between probing and control. The purpose of this section is not to give a detailed treatment of reinforcement learning but rather to point at the similarities and some differences with dual control. The survey by Kaelbling et al. (1996) is a good introduction to the field.

Reinforcement learning is traditionally formulated for finite state and action spaces and the process is modeled as a Markov decision process. The agent takes an action based on the current state and receives an instantaneous reward. Based on the previously received rewards, the agent must decide on which action to take next. The goal of the reinforcement learning algorithm is to find a policy that maximizes some long-term measure of the rewards. As an illustration, consider the k-armed bandit problem, where an agent in a number of turns has the option of k actions. Each action gives a reward according to some unknown probability. How should the agent act? On the one hand, the agent wants to find a good action, that is one with high probability of giving a reward. On the other hand, trying the different actions, to find the best one, means that some turns are wasted on bad actions. The strategy of how the exploration of new arms should be performed can be seen as the experiment design of the reinforcement learning algorithm. For a deeper and general treatment of bandit problems, the book by Berry and Fristedt (1985) is suggested.

A notable difference to adaptive control is the use of reinforcement leaning techniques that do not require a model of the process to find the optimal reward. For example, the adaptive heuristic critic of Barto et al. (1983) and the Q-learning algorithm by Watkins (1989) learn a control policy without explicitly learning a model of the process being controlled. Recently, Lewis et al. (2012) have used reinforcement learning techniques to formulate adaptive controllers that converge to optimal control solutions. They also show how Q-learning can be used to learn an optimal control solution for discrete-time systems.

Chapter 3
Experiment design for model predictive control

Considering the current situation of industrial MPC, there is a need for technologies enabling adaptation to changing operating conditions and plant changes. Implementations of MPC show promise but benefits are held back by the fact that control performance degrades before payback on the investment can be made. This leads to distrust in MPC solutions, both from process operators who experience the worsened control first-hand, and from managers, who do not see a clear return on investment.

The recent research interest in the area of dual MPC shows that the academic community has recognized these issues, and algorithms aimed at maintaining accurate models have started to appear. Although the algorithms that have appeared so far enable closed-loop reidentification, the intended use of the model is not explicitly taken into account. This chapter considers closed-loop reidentification for MPC from an application-oriented standpoint. This means that the intended use of the model enters explicitly in the design of the reidentification experiment. The benefit is that the cost of the experiments can be kept as low as possible while at the same time performance guarantees on the resulting closed-loop performance can be given.

3.1 Application-oriented experiment design

In this section, the experiment design problem formulated in (1.1) in Section 1.3 is discussed in a general setting. The experiment design was formulated as

$$\begin{aligned} \underset{\text{input}}{\text{minimize}} \quad & \text{Cost of experiment}, \\ \text{subject to} \quad & \text{Performance specifications}, \\ & \text{System constraints}. \end{aligned} \tag{3.1}$$

Cost of the experiment

In an industrial application it is often desirable to have identification experiments that disturb normal operation as little as possible. What constitutes a disturbance depends on the situation at hand. Bombois et al. (2006) give three general situations together with relevant measures of the experimental cost. Somewhat modified, these are:

Situation 1 The resources consumed by the experiment and/or the resulting degradation of the product are the important factors.

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

Situation 3 Both resources and time are important factors.

Characterizing good models

The quality of a model can be measured in many ways. When the intended use of the model is control, one way is to consider the model quality in terms of the resulting performance when the model is used in the control design. Let the performance degradation be measured by an *application cost*, $V_{\text{app}} : \mathbf{R}^{n_\theta} \to \mathbf{R}$, defined as a function of the model parameters θ. If the true system parameters were known, the control design should result in the best possible control performance and no performance degradation. For other values of the parameters, the control performance may degrade. Taking these observations into account, the application cost should have the properties

$$V_{\text{app}}(\theta_o) = 0, \qquad V_{\text{app}}(\theta) \geq 0, \text{for all } \theta, \tag{3.2}$$

which imply, if $V_{\text{app}}(\theta)$ is twice differentiable in a neighborhood of θ_o, that

$$V'_{\text{app}}(\theta_o) = 0, \qquad V''_{\text{app}}(\theta_o) \succeq 0, \tag{3.3}$$

where prime denotes the gradient and double prime denotes the Hessian matrix.

A model is considered good if the performance degradation is sufficiently small. This gives a set of acceptable models or parameters

$$\Theta_{\text{app}}(\gamma) \triangleq \left\{ \theta \,\middle|\, V_{\text{app}}(\theta) \leq \frac{1}{\gamma} \right\},$$

where γ is an application specific constant which determines the required accuracy of the model.

Guaranteed performance for estimated models

The set Θ_{app} contains the model parameters that give acceptable performance and any $\hat{\theta}$ that lies in Θ_{app} is an acceptable model from the application perspective. Hence, the goal is to find estimated model parameters that belong to this set. Since the estimates from the system identification are random variables with (asymptotically in the data-size) Gaussian distribution (2.9), this can only be guaranteed with some probability. Therefore, the experiment design should be such that

$$\mathbb{P}\left\{\hat{\theta} \in \Theta_{\text{app}}\right\} \geq \alpha, \tag{3.4}$$

for some lower bound α close to 1. Calculating the probability in (3.4) is in general very involved but this chance constraint can be approximately reformulated as follows.

Using the asymptotic Gaussian distribution of the estimates in (2.9), a standard confidence ellipsoid around the true parameters θ_o can be constructed as

$$U(\alpha) \triangleq \left\{\theta \;\middle|\; [\theta - \theta_o]^T \mathcal{I}_1^N(\theta_o)\, [\theta - \theta_o] \leq \chi^2_\alpha(n_\theta)\right\},$$

where $\mathcal{I}_1^N(\theta_o)$ is the information matrix and $\chi^2_\alpha(n_\theta)$ is the α-percentile of the χ^2 distribution with n_θ degrees of freedom. The ellipsoid $U(\alpha)$ contains the estimated parameters $\hat{\theta}_N$ with probability α which means that (3.4) can (for sufficiently large number of data) be approximately enforced by requiring that

$$U(\alpha) \in \Theta_{\text{app}}.$$

For general application costs, the set Θ_{app} may be nonconvex, however, it can be approximated by an ellipsoidal set using a second order Taylor expansion and the properties (3.3), which results in the ellipsoid

$$\mathcal{E}_{\text{app}}(\gamma) \triangleq \left\{\theta \;\middle|\; \frac{1}{2}[\theta - \theta_o]^T V''_{\text{app}}(\theta_o)[\theta - \theta_o] \leq \frac{1}{\gamma}\right\} \approx \Theta_{\text{app}}.$$

Hence, the chance constraint (3.4) can be further approximated by

$$U(\alpha) \in \mathcal{E}_{\text{app}}. \tag{3.5}$$

The ellipsoidal inclusion (3.5) can be simplified to

$$\mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\theta_o), \tag{3.6}$$

as shown by Hjalmarsson (2009). The constraint (3.6), which is an LMI in the elements of $\mathcal{I}_1^N$, will be referred to as the *experiment design constraint*.

3.2 Optimal experiment design formulation

Based on the considerations in the previous section, an optimal controller for the application-oriented experiment design problem for systems with constraints can be formalized. Two principal situations will be considered. The first is the situation where the cost of the experiment relates to the degradation of product quality. Therefore, it is desired that the control cost is kept small during the identification. The second is the situation where the time of the experiment is the crucial cost. It is acceptable to sacrifice some product quality, as long as this results in a shorter experiment.

The experiment designs for the two situations can be formulated as:

1. With the least possible effect on the control cost C, during an experiment of length N, excite the system $\mathcal{S}$ enough to accurately model the important system dynamics needed for acceptable control performance.

2. During the shortest possible experiment time N, excite the system $\mathcal{S}$ enough to accurately model the important system dynamics needed for acceptable control performance.

Consider the system $\mathcal{S}$ in (2.1) modeled by $\mathcal{M}(\theta)$ in (2.2) and operating under the constraints

$$x \in \mathcal{X}, \qquad u \in \mathcal{U}, \qquad y \in \mathcal{Y}. \tag{3.7}$$

The application-oriented experiment design problem can be formulated as the finite-horizon optimal control problem

$$\min_{u} \left\{ J \;\middle|\; \begin{aligned} x_{t+1} &= A(\theta_o)x_t + B(\theta_o)u_t + K(\theta_o)e_t, \\ y_t &= C(\theta_o)x_t + e_t, \\ & x_t \in \mathcal{X}, y_t \in \mathcal{Y}, u_t \in \mathcal{U}, \\ & \mathcal{I}_1^N(\theta_o) \succeq \frac{\gamma\chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\theta_o) \end{aligned} \right\}. \tag{3.8}$$

The objective J is the control cost C in situation 1 and the experiment length N in situation 2. The horizon considered by the controller (3.8) is the experiment length. Hence, the controller should find an input of length N that satisfies the experiment design constraint at the lowest cost.

Alternatively, an infinite-horizon controller can be formulated for the application-oriented experiment design problem. This means that the control cost is calculated for a control horizon that is infinite. The experiment length N, on the other hand, remains finite. In this case, the average per-sample information matrix, scaled by the experiment lenght, is used to formulate the experiment design constraint. This means that

$$\mathcal{I}(\theta_o) \succeq \frac{\gamma\chi^2_\alpha(n_\theta)}{2N} V''_{\text{app}}(\theta_o)$$

replaces the last constraint in (3.8).

Dual properties

Explicitly including the experiment design constraint in the controller gives the controller dual properties. The minimization of the cost, ensures good control performance while the experiment design constraint ensures that the data contains enough information for good estimation. The formulation in (3.8) relates to the MPC with excitation and dual MPC formulations presented in Section 2.5. The significant difference is that these new formulations *explicitly* take the intended application into account when the excitation is designed. The algorithms presented in Section 2.5, on the other hand, are primarily concerned with persistence of excitation.

Challenges

The input design problem in (3.8) is in general a difficult optimization problem. There are a number of challenges that need to be addressed. The solution may be closed-loop control and therefore the correlation between the input and any noise sources present must be accounted for. This is further complicated by the possibility that the controller is nonlinear or gives non-stationary signal distributions. When the system is subject to time-domain constraints on the signals, it is often observed that the resulting signal distributions are not Gaussian even when the system is driven by Gaussian sources.

A notable exception is the quasi-stationary case with a linear output feedback controller such that the input is given by (2.16). When no time-domain constraints are present, the two experiment design situations have essentially been solved for a number of frequency-domain constraints, see, for example Hjalmarsson and Jansson (2008), Hildebrand et al. (2010), or Hildebrand and Gevers (2013).

Finally, quantifying performance is often not straightforward. This is further elaborated on in the next section.

3.3 Measuring MPC performance

The central idea in application-oriented experiment design is that control performance should be guaranteed. Performance can be measured in many ways, even for one application, and it is often not a clear cut decision which measure to use.

A common approach is to use the performance index by Harris (1989), which compares the control performance to a minimum variance controller. Huang et al. (2006) have extended the idea to multivariate controllers. These type of measures put a focus on the control performance, in terms of how well the signals are controlled, and less focus on the overall production performance. Another approach is to measure MPC performance in economic terms. One such measure, specifically for performance monitoring, is introduced by Modén and Lundh (2013). Bauer and Craig (2008) have also studied the general problem of economic assessment of APC implementations.

Performance measures are often used for performance monitoring and diagnosis. In application-oriented experiment design, the performance measure is the application cost. The specifics for the application cost in the MPC setting are discussed in the following sections.

MPC specifics

A significant factor for the success of MPC is the explicit handling of constraints in the controller. Maciejowski (2002) explains it in the following way, illustrated in Figure 3.1. If the process is affected by random disturbances, reducing the variance of the outputs as much as possible means that the process can operate as close as possible to the constraints without violating them, which is often near the optimal performance. When a linear controller is used, the variance may be reduced but the shape of the distribution remains the same. However, an MPC that is aware of constraints can react differently depending on if a disturbance moves the output toward or away from a constraint. This reshapes the distribution of the output which may allow the process to operate closer to the constraints than what is possible with a linear controller. This effect is central to MPC and the ability to operate closer to constraints should be reflected by the performance measure.

Control cost

The quadratic control cost, C_t, is the main tuning knob of the MPC in (2.5) and tuning is done through the choice of the weighting matrices Q, R and S. The importance of the different outputs is reflected in the choice of the matrix Q and the cost of using the different inputs is reflected by the matrices R and S. Hence, many different

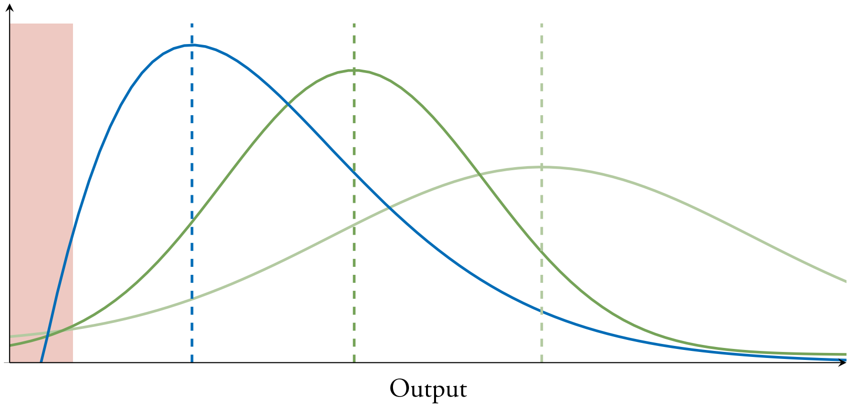


Figure 3.1 Setpoint optimization using MPC. The uncontrolled system output distribution (—) has a large variance. Using a controller unaware of the constraints (▭), the variance can be reduced but the shape of the distribution (—) remains the same. Since MPC is aware of constraints, the controller skews the distribution (—), allowing the setpoint to be closer to the constraint. The setpoints for the three cases are the dashed lines of the corresponding colors.

considerations are weighted together into a single value. This can make assessing control performance based on the cost C_t difficult. Several methods have been proposed for relating some performance measure to the choice of C_t. One such approach, which uses the sample variance as performance measure, is the practical auto-tuning method by Annergren et al. (2013). Since, in the thesis, the controller is considered fixed and tuning is not the main focus, this issue is not discussed further here.

Application cost

The application-oriented experiment design idea requires an application cost which measures performance degradation related to model plant mismatch. In the MPC context, a measure that relates the output of the system when the wrong model is used to the ideal output, is a reasonable choice for the application cost. This is adopted here and the application cost

$$V_{\text{app}}(\theta) = \frac{1}{T} \sum_{t=1}^{T} \|y_t(\theta) - y_t(\theta_o)\|^2 \tag{3.9}$$

is used. The application cost compares the output from the system when the controller uses arbitrary model parameters, $y_t(\theta)$ to the output when a perfect description of the system is used, $y_t(\theta_o)$ over a time window T. The cost (3.9) satisfies the requirements (3.2) and can be approximated by the Taylor expansion

$$V_{\text{app}}(\theta) \approx (\theta - \theta_o)^T V''_{\text{app}}(\theta_o)(\theta - \theta_o).$$

The application cost in (3.9) is based on knowledge of the true parameters θ_o, which are not available; were they available, there would be no need for system identification. Therefore, the construction (3.9) is mostly of theoretical interest and not practically applicable. Instead, in practice, the true parameter vector must be replaced by the best available estimate of θ_o. By doing so, a simulation based algorithm for calculating the Hessian of V_{app} was presented by Ebadat et al. (2014).

3.4 Open-loop experiment design

In this section, the open-loop experiment design problem for MPC, as presented by Larsson (2011), is briefly outlined. The problem formulation is for the infinite-horizon case, assuming stationary signals in open-loop operation of the process with no active constraints during the identification experiment. This means that the frequency-domain techniques, for example by Jansson and Hjalmarsson (2005), can be applied. The outcome of the experiment design is an optimal input spectrum and any time-domain constraints need to be taken care of in the signal generation.

The input is designed to minimize the input power, which can be expressed as

$$\operatorname{tr} \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega)\, \mathrm{d}\omega.$$

The goal of the system identification is to estimate a model with accaptable performance, as specified by (3.6), after experiment of length N samples. This means that the the problem can be formally stated as

$$\begin{aligned} \underset{\Phi_u(\omega)}{\text{minimize}} \quad & \operatorname{tr} \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_u(\omega)\, \mathrm{d}\omega \\ \text{subject to} \quad & \mathcal{I}(\theta_o) \succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2N} V''_{\text{app}}(\theta_o) \\ & \Phi_u(\omega) \succeq 0 \quad \text{for all } \omega. \end{aligned} \tag{3.10}$$

The optimization is done over the trigonometric, matrix polynomial Φ_u representing the input spectrum. The last constraint in the optimization ensures that the polynomial is a spectrum. In the chosen formulation, the cost of the experiment is taken as the input power. However, the same principal formulation can be used for the case of output power, a combination of the two, or for the experiment length.

The last constraint in (3.10) is infinite dimensional but can be treated by using a finite-dimensional spectrum parameterization. For example, the input can be restricted to be generated by an Mth order FIR-filter, which gives the input spectral density

$$\Phi_u(\omega) = \sum_{k=-(M-1)}^{M-1} c_k e^{i\omega k}, \tag{3.11}$$

where $c_k \in \mathbf{R}^{m\times m}$. Positivity of the spectrum defined by (3.11) can be enforced by the Kalman-Yakubovich-Popov lemma.

Lemma 3.1 (Kalman-Yakubovich-Popov)
Let $\{A, B, C, D\}$ be a controllable state-space form of $\sum_{k=0}^{M-1} c_k e^{i\omega k}$. Then there exists a matrix $Q = Q^T \succeq 0$ such that

$$\begin{bmatrix} Q - A^TQA & -A^TQB \\ -B^TQA & -B^TQB \end{bmatrix} + \begin{bmatrix} 0 & C^T \\ C & D + D^T \end{bmatrix} \succeq 0, \tag{3.12}$$

if and only if

$$\Phi_u(\omega) = \sum_{k=(M-1)}^{M-1} c_k e^{i\omega k} \succeq 0 \quad \textit{for all } \omega. \tag{3.13}$$

Proof. The lemma follows from the positive-real lemma, see for example the work of Yakubovich (1962). □

Furthermore, by Parseval's theorem, the average per-sample information matrix is an affine function of the input spectrum. This means that the optimization problem (3.10) can be approximately reformulated as a finite dimensional SDP in the variables c_k and Q. The approximation comes from the restriction to FIR spectra, which limits the solution to a subspace of all possible spectra.

Identification algorithm

The complete open-loop application-oriented identification method is presented here. The method requires an initial model to base the input design on. If no such model is available, one can be obtained in a first step using, for example, a white noise input.

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 an 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.

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 θ_o such that $\mathcal{S} = \mathcal{M}(\theta_o)$, the estimates will converge to their true values. Therefore, one can expect the input design to converge to what would be obtained had θ_o been known. Gerencsér et al. (2009) discuss this and give a formal proof for this for the case of ARX systems in an adaptive input design setting.

Signal generation

The solution to the input design problem (3.10) 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)

$$\Phi_u(\omega) = H(\omega)\Lambda H^{\mathrm{H}}(\omega). \tag{3.14}$$

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 time-domain constraints on the inputs. If time domain signal constraints need to be considered in the signal generation, the framework in Chapter 7 may be useful.

Experimental evaluation

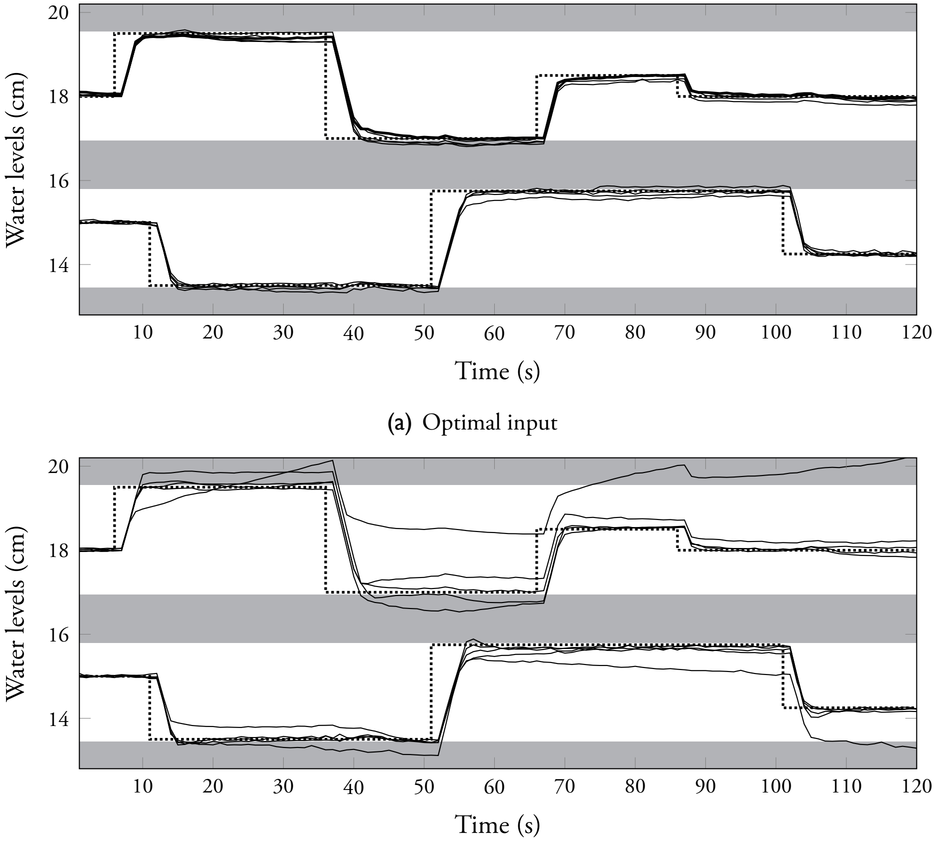
The method for open loop input design is evaluated on a laboratory four tank system. The process is described in detail in Example 3 of Chapter 7. The experimental setup consists of four interconnected tanks and two pumps. The system runs at a sampling frequency of 1 Hz and the input design, system identification and MPC are all implemented in MATLAB. The application cost is chosen according to (6.2) and evaluated using an initial estimate obtained in a white noise experiment.

The procedure for the experiment was:

1. Apply input voltages corresponding to the desired operating point and allow the water levels in the tanks to settle.
2. Run a 2 minute (120 samples) initial identification experiment. The input used was Gaussian white noise with variance 0.1.
3. Based on the estimate from 2, calculate $V''_{\text{app}}(\hat{\theta}_N)$ and find the optimal input spectrum $\Phi_u^\star(\omega)$.
4. Run a 10 minute (600 samples) identification experiment using an input with spectrum $\Phi_u^\star(\omega)$.
5. Identify a model of the system using all 12 minutes (720 samples) of data.
6. Construct an MPC with the model obtained in step 5 and run a reference tracking scenario.

To illustrate the advantage of using the optimal input, the same procedure is performed with the optimal input replaced by a Gaussian white noise input with the same variance as the optimal input. The reference tracking capabilities of the resulting MPCs are shown in Figure 3.2. 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 when white noise identified models are used are much more spread out around the desired reference trajectories.

The water levels violate constraints for both cases presented in Figure 3.2. 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 from white noise excitation experiments. To further improve on reducing constraint violations, a penalty for this in the application cost, might be useful.



(a) Optimal input

(b) White noise input

Figure 3.2 The trajectories (——) and references (·········) of the four tank process controlled by MPC with models based on estimates from 5 identification experiments. In plot (a) the identification was made with optimal inputs whereas in plot (b) white inputs were used.

Chapter 4
Markov decision process formulation of experiment design

The application-oriented experiment design problem formulated in Chapter 3 is in general a difficult optimization problem. The nonlinear nature of the controller and the presence of constraints complicate the calculation of the resulting signal distributions. When the system is driven by disturbance processes and the controller is used to counteract the effect of the disturbances, there is always correlation between the input and the disturbance. Markov decision processes (MDP) offer a framework for making sequential decisions based on current knowledge when the result of the decision is uncertain, which is the case for the systems considered here. A good introductory reference to MDPs is the book by Puterman (1994).

In this chapter, constrained Markov decision processes (CMDPs) are used to formulate a controller that implements the application-oriented experiment design problem in (3.8). The problem is formulated using theory for discrete states and controls (or actions), which often will be an approximation since (3.8) is formulated for more general cases. The benefit of the approach is that very general noise and disturbance situations may be considered. The downside is that the approach results in a new type of controller that is not found in implementations. Furthermore, finding the controller is computationally expensive.

The chapter starts by recalling the considered problem and an introduction to the system and control model used in the MDP. Then the optimal control problem is discussed in the expected average and discounted cost cases. The discounted case does not correspond to the original problem but results in a smaller optimization problem and may have a simpler solution. The expected average cost case matches the original problem but it may result in more complicated controllers. A semidefinite program (SDP) can be used to find the optimal control in both cases. The application-oriented experiment design problem is implemented as a CMDP in a simulation example.

4.1 The considered problem

The infinite horizon application-oriented experiment design problem

$$\min_{u} \left\{ C(u) \;\middle|\; \begin{aligned} x_{t+1} &= Ax_t + Bu_t + v_t, \\ y_t &= Cx_t + w_t, \\ x_t &\in \mathcal{X}, y_t \in \mathcal{Y}, u_t \in \mathcal{U}, \\ \mathcal{I}(\theta_o) &\succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2N} V''_{\text{app}}(\theta_o) \end{aligned} \right\}, \tag{4.1}$$

where $C(u)$ a control objective, is considered. Since the infinite horizon case is considered, the average per-sample information matrix is used in (4.1). The last constraint in (4.1) ensures that, after an identification experiment of length N, the resulting estimates fulfill the application requirements with probability α. As it stands, the optimization problem in (4.1) is typically intractable. The idea here is to formulated a finite CMDP corresponding to the original optimization problem. Since the state and control spaces in (4.1) are more general than the finite, discrete spaces that are considered for the CMDP, a discretization of the original spaces is typically needed. The resulting CMDP can then be solved using an SDP. Before reformulating the problem (4.1), a general framework for CMDPs with matrix constraints is defined. The first step in the CMDP modeling is to formalize the system and control models used, which is done in the following section.

4.2 System and control model

The system is observed at discrete times $t = 1, 2, \ldots$, over an infinite horizon. The evolution of the system from one time instant to the other depends on the state of the system, the action taken by the controller and the process noise. Depending on the chosen action, the system transitions randomly from one state to another with different probabilities. It is assumed that the system has the Markov property, which means that the probability of transitioning from one state to another depends only on the current state and not on how the current state was attained.

To each state and action there is an associated cost, and a number of associated rewards. The goal of the controller is to find a control policy that minimize the cost while ensuring that the rewards are above some given constraints. To implement the controller in (4.1), the definition of CMDPs from Altman (1999) is extended to matrix valued rewards in Definition 4.1.

Definition 4.1 (Matrix constrained Markov decision process) A finite, matrix constrained Markov decision process is the tuple $\{\mathbf{X}, \mathbf{U}, \mathcal{P}, c, d\}$ where:

- $\mathbf{X}$ is a discrete state space containing a finite number of states. Generic notation for states is x, y.
- $\mathbf{U}$ is a discrete action space containing a finite number of actions. An individual action is denoted u and $\mathbf{U}(x) \subseteq \mathbf{U}$ are the actions available in state x.
- $\mathcal{P} = \{p_{xy}(u)\}$ are the transition probabilities from state x to state y when action u is taken, that is $p_{xy}(u) = \mathbb{P}\{x_{t+1} = y \,|x_t = x, u_t = u\}$.
- $c\colon \mathbf{X} \times \mathbf{U} \to \mathbf{R}$ is an immediate cost for state–action pair (x, u) related to the control objective.
- $d = \left\{d^k\colon \mathbf{X} \times \mathbf{U} \to \mathbf{R}^{n_k \times n_k}, k = 1, 2, \ldots, m\right\}$ are m symmetric, matrix valued immediate rewards for state–action pair (x, u) related to the constraints.

Remark 4.1 If $n_k = 1$ for all k in Definition 4.1, d can be taken as the k-dimensional vector of immediate costs in the standard definition of CMDPs.

A sequence of rules for how an action is chosen at a specific time is called a policy and is denoted $\pi = (\pi_1, \pi_2, \ldots)$. Policies can be categorized in the following classes:

Markov policies, Π_M: $\pi \in \Pi_M$ if for any time t, the rule π_t depends only on the state x_t and not on how the state was attained. Hence, the rule π_t is a mapping $\pi_t\colon \mathbf{X} \times \mathbf{U}(x) \to [0, 1]$, which gives the conditional probability of taking control action u in state x at time t, that is $\pi_t(x, u) = \mathbb{P}\{u_t = u \,|x_t = x\}$.

Stationary policies, Π_S: $\pi \in \Pi_S$ if $\pi_1 = \pi_2 = \cdots$, that is, the rule does not depend on time. The stationary policies are a subset of Π_M.

Stationary deterministic policies, Π_D: $\pi \in \Pi_D$ if the action u_t is a function of the state x_t. Hence, the rule π_t is a mapping $\pi_t\colon \mathbf{X} \to \mathbf{U}(x)$. The stationary deterministic policies are a subset of Π_S.

The initial distribution of the states is denoted β, that is, $\beta(x) = \mathbb{P}\{x_0 = x\}$. A given policy π, and an initial distribution β determine a unique probability measure $\mathbb{P}^\pi_\beta$ for the state and action trajectories. The corresponding expectation operator is denoted $\mathbb{E}^\pi_\beta$.

Controller with matrix constraints

The objective for the controller is to minimize a cost, $C_\beta(\pi)$, based on the immediate costs c, while maintaining rewards, $D_\beta^k(\pi)$, based on the immediate rewards d^k, above a given level. For the expected average case, given an initial distribution β and policy π, the cost and rewards are defined as

$$C_\beta(\pi) \triangleq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\beta^\pi\{c(x_t, u_t)\}, \tag{4.2}$$

$$D_\beta^k(\pi) \triangleq \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\beta^\pi\Big\{d^k(x_t, u_t)\Big\}. \tag{4.3}$$

For the discounted case, the cost and rewards are defined as

$$C_\beta^\alpha(\pi) \triangleq \limsup_{T\to\infty} (1-\alpha) \sum_{t=1}^{T} \alpha^{t-1}\, \mathbb{E}_\beta^\pi\{c(x_t, u_t)\}, \tag{4.4}$$

$$D_\beta^{k,\alpha}(\pi) \triangleq \liminf_{T\to\infty} (1-\alpha) \sum_{t=1}^{T} \alpha^{t-1}\, \mathbb{E}_\beta^\pi\Big\{d^k(x_t, u_t)\Big\}, \tag{4.5}$$

for a fixed discount factor $\alpha, 0 < \alpha < 1$. Using the cost and rewards, the matrix constrained Markov decision process (matrix CMDP) can be formulated. The controller should find a feasible policy $\pi^\star$ that is optimal in the sense that

$$\pi^\star = \arg\min_\pi \Big\{C_\beta(\pi) \;\Big|\; D_\beta^k(\pi) \succeq B^k,\ k = 1, 2, \ldots, m\Big\}. \tag{4.6}$$

Here, B_k are matrices, of dimensions commensurate with $D_\beta^k(\pi)$, defining the constraints on the rewards. Comparing (4.1) and (4.6), it is seen that the experiment design problem has similarities with in the matrix CMDP. Firstly, the state and control constraints $\mathcal{X}$ and $\mathcal{U}$ relate to the state and action spaces $\mathbf{X}$ and $\mathbf{U}$. Secondly, the dynamics given by the state space model in (4.1) together with the distribution of the disturbance process v_t is an alternative description of the transition probabilities $p_{xy}(u)$. Thirdly, the excitation constraint is a matrix valued reward subject to a lower constraint. The formulation of the experiment design problem as a matrix CMDP is formalized later in this chapter. First, a small example is presented to illustrate the CMDP modeling and how the constraints can be used to give inputs suitable for reidentification.

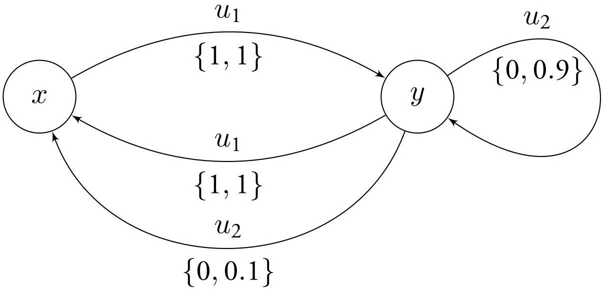


Figure 4.1 The Markov chain considered in Example 4.1. The circles represent the states and the arcs are the possible transitions. The symbol above an arc indicates the chosen action. The first value in the bracket below the arc is the cost related to the action and the second value is the probability of the transition given the action.

Example 4.1 (Matrix constrained Markov chain)

Consider first the two-state unconstrained *Markov chain illustrated in Figure 4.1. The chain is defined by:*

States: $\mathbf{X} = \{x, y\}$.
Actions: $\mathbf{U} = \{u_1, u_2\}$ *with* $\mathbf{U}(x) = \{u_1\}$ *and* $\mathbf{U}(y) = \{u_1, u_2\}$.
Transition probabilities:

$$\begin{aligned} &p_{xx}(u_1) = 0, && p_{yx}(u_1) = 1, && p_{yx}(u_2) = 0.1 \\ &p_{xy}(u_1) = 1, && p_{yy}(u_1) = 0, && p_{yy}(u_2) = 0.9. \end{aligned}$$

Costs:

$$c(x, u_1) = 1, \qquad c(y, u_1) = 1, \qquad c(y, u_2) = 0.$$

The initial state is x or y with equal probability and hence $\beta = [0.5\ 0.5]^T$. The control criterion is to minimize the expected average cost (4.2) *(without any constraints), which can be done using the policy*

$$\pi^\star_{uc} : u = \begin{cases} u_1 & \textit{in state } x, \\ u_2 & \textit{in state } y. \end{cases}$$

The policy $\pi^\star_{uc}$ is not good in a situation where new information on the effect of action u_1 needed. Since, under $\pi^\star_{uc}$, the action u_1 is never taken in state y, it is not possible to *completely model the effect of u_1 using this policy. To model that effect, a constraint can be added to ensure that, on average, the action u_1 is taken in state y at least in, for example, 10 % of the total control actions. To achieve this, introduce the rewards*

$$d(x, u_1) = 0, \qquad d(y, u_1) = 1, \qquad d(y, u_2) = 0,$$

which give a reward for using action u_1 in state y. The controller is changed such that the expected average cost is minimized subject to that the expected average reward is above 0.1, which can be done using the policy

$$\pi_c^\star : u = \begin{cases} u_1 & \textit{in state } x, \\ u_1 & \textit{with probability } 0.12 \textit{ in state } y, \\ u_2 & \textit{with probability } 0.88 \textit{ in state } y. \end{cases}$$

In Monte Carlo simulations of the Markov chain, under the policy $\pi_c^\star$, the state–action pair (y, u_1) occurs in 10 % of the total state–action pairs, while under $\pi_{uc}^\star$, the pair (y, u_1) (of course) never occurs. ◇

Example 4.1 shows two things. First, the optimal policy in the unconstrained case does not result in a controller which can be used to fully model the system, since the effect of action u_1 in state y is never explored. Second, suitably chosen constraints gives the controller dual properties.

To solve the problem (4.6), an SDP can be used. The idea is to introduce a feasible set for the SDP such that the decision variables correspond to probability measures of pairs of states and actions in the controlled Markov chain. In the following two sections, these feasible sets are constructed for the expected average cost and the discounted cost cases.

4.3 Expected average cost and constraints

In this section, the case with expected average cost and constraints is considered for general Markov chains. The expected average cost for initial distribution β can be written as

$$\begin{aligned} C_\beta(\pi) &= \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\beta^\pi\{c(x_t, u_t)\} \\ &= \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{x_0 \in \mathbf{X}} \sum_{\substack{x\in\mathbf{X} \\ u\in\mathbf{U}(x)}} \beta(x_0)\, \mathbb{P}_{x_0}^\pi\{x_t = x, u_t = u\}\, c(x, u), \end{aligned}$$

where $\mathbb{P}^{\pi}_{x_0}\{x_t = x, u_t = u\}$ is the probability of the state–action pair (x, u) given initial state x_0 and policy π. The expected average rewards can be written as

$$\begin{aligned} D^k_\beta(\pi) &= \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}^{\pi}_{\beta}\Big\{ d^k(x_t, u_t)\Big\} \\ &= \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \sum_{x_0\in\mathbf{X}} \sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} \beta(x_0)\,\mathbb{P}^{\pi}_{x_0}\{x_t = x, u_t = u\}\, d^k(x, u). \end{aligned}$$

Introduce the finite-horizon expected state–action frequencies

$$z^T_{xu}(\pi) \triangleq \frac{1}{T}\sum_{t=1}^{T}\sum_{x_0\in\mathbf{X}} \beta(x_0)\,\mathbb{P}^{\pi}_{x_0}\{x_t = x, u_t = u\}\,.$$

The sets $\{z^T_{xu}(\pi)\}_{(x\in\mathbf{X}, u\in\mathbf{U}(x))}$ can be seen as probability measures (sometimes denoted occupation measures), which give the probability $z^T_{xu}(\pi)$ to the state–action pair (x, u).

Let Z_π denote the limit points, $z(\pi)$, of the vectors $z^T(\pi) = \{z^T_{xu}(\pi),\ T = 1, 2, \ldots\}$. The elements of Z_π are the infinite-horizon probability measures assigning probabilities for the state–action pairs. In general, for a given initial distribution and policy, there may be an infinite set of limit points. Therefore, the notion of a convergent policy is useful.

Definition 4.2 (Convergent policy) A policy π is *convergent* if Z_π consists of a single element. The class of convergent policies is denoted Π_C.

The set Z_π depends on which class the policy belongs to. Introduce the sets L, L_M, L_C, L_S, and L_D defined by

$$\begin{aligned} L &\triangleq \{z(\pi) \in Z_\pi \mid \pi \text{ is an arbitrary policy.}\}\,, \\ L_M &\triangleq \{z(\pi) \in Z_\pi \mid \pi \in \Pi_M\}\,, \\ L_C &\triangleq \{z(\pi) \in Z_\pi \mid \pi \in \Pi_C\}\,, \\ L_S &\triangleq \{z(\pi) \in Z_\pi \mid \pi \in \Pi_S\}\,, \\ L_D &\triangleq \{z(\pi) \in Z_\pi \mid \pi \in \Pi_D\}\,. \end{aligned}$$

Theorem 4.1
$L = L(M) = L(C) = \overline{L(S)} = \overline{L(D)}$, *where* $\overline{X}$ *is the convex hull of* X.

Proof. See, for example, Hordijk and Kallenberg (1984). □

Theorem 4.1 shows that, since the limit points for arbitrary policies have a corresponding convergent policy, there is no restriction in considering only convergent policies. Furthermore, the theorem shows that, in general, the optimal policy for the infinite-horizon cost situation may be non-stationary.

For convergent policies, the expected average cost and rewards can be expressed in terms of the state–action frequencies as

$$C_\beta(\pi) = \sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} c(x,u)z_{xu}(\pi), \qquad D_\beta^k(\pi) = \sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} d^k(x,u)z_{xu}(\pi).$$

Introduce the vector set

$$\mathbf{Q}_\beta = \left\{ z \;\middle|\; \begin{array}{l} \displaystyle\sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} (\delta_{xy} - p_{xy}(u))z_{xu} = 0,\ y \in \mathbf{X} \\ \displaystyle\sum_{u} z_{yu} + \sum_{(x,u)} (\delta_{xy} - p_{xy}(u))\tilde{z}_{xu} = \beta(y)s,\ y \in \mathbf{X} \\ z_{xu}, \tilde{z}_{xu} \geq 0,\ (x,u) \in \mathbf{X}\times\mathbf{U}(x) \end{array} \right\}.$$

The set $\mathbf{Q}_\beta$ is a polytope and can be related to the set L by the following theorem.

Theorem 4.2

$L = \mathbf{Q}_\beta$.

Proof. See, for example, Hordijk and Kallenberg (1984). □

The theorem shows that the polytope $\mathbf{Q}_\beta$ corresponds to the set of probability measures of the state–action pairs for infinite horizon. Hence, it is possible to optimize over the state–action frequencies by considering vectors $z \in \mathbf{Q}_\beta$. Before formulating the SDP to solve (4.6), the discounted cost situation is studied.

4.4 Discounted expected cost and constraints

The discounted expected cost can be written as

$$\begin{aligned} C_\beta^\alpha(\pi) &= \liminf_{T\to\infty}\,(1-\alpha)\sum_{t=1}^{T}\alpha^{t-1}\,\mathbb{E}_\beta^\pi\{c(x_t,u_t)\} \\ &= (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}\sum_{x_0\in\mathbf{X}}\sum_{\substack{x\in\mathbf{X}\\u\in\mathbf{U}(x)}}\beta(x_0)\,\mathbb{P}_{x_0}^\pi\{x_t=x,u_t=u\}\,c(x,u), \end{aligned}$$

where the limit exists because of the finite state and action spaces. The discounted rewards $D_\beta^{k,\alpha}(\pi)$ can be written analogously as

$$\begin{aligned} D_\beta^{k,\alpha}(\pi) &= \limsup_{T\to\infty}\,(1-\alpha)\sum_{t=1}^{T}\alpha^{t-1}\,\mathbb{E}_\beta^\pi\Big\{d^k(x_t,u_t)\Big\} \\ &= (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}\sum_{x_0\in\mathbf{X}}\sum_{\substack{x\in\mathbf{X}\\u\in\mathbf{U}(x)}}\beta(x_0)\,\mathbb{P}_{x_0}^\pi\{x_t=x,u_t=u\}\,d^k(x,u). \end{aligned}$$

Introduce the discounted state–action frequencies

$$z_{xu}^\alpha(\pi) \triangleq (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}\sum_{x_0\in\mathbf{X}}\beta(x_0)\,\mathbb{P}_{x_0}^\pi\{x_t=x,u_t=u\}\,.$$

The set $\{z_{xu}^\alpha(\pi)\}_{(x\in\mathbf{X},u\in\mathbf{U}(x))}$ is again a probability measure that gives the probability $z_{xu}^\alpha(\pi)$ to the state–action pair (x,u). Contrary to the infinite-horizon situation, for a given policy and initial distribution, there is only one such probability measure. Therefore, the discounted cost and rewards can be expressed in terms of the discounted state–action frequencies as

$$C_\beta^\alpha(\pi) = \sum_{\substack{x\in\mathbf{X}\\u\in\mathbf{U}(x)}} c(x,u)z_{xu}^\alpha(\pi), \qquad D_\beta^{k,\alpha}(\pi) = \sum_{\substack{x\in\mathbf{X}\\u\in\mathbf{U}(x)}} d^k(x,u)z_{xu}^\alpha(\pi).$$

Introduce the sets of vectors $z^\alpha(\pi) = \{z_{xu}^\alpha(\pi)\}_{(x\in\mathbf{X},u\in\mathbf{U}(x))}$ defined by

$$\begin{aligned} K &\triangleq \{z^\alpha(\pi)\mid \pi \text{ is an arbitrary policy.}\}\,, \\ K_M &\triangleq \{z^\alpha(\pi)\mid \pi\in\Pi_M\}\,, \\ K_S &\triangleq \{z^\alpha(\pi)\mid \pi\in\Pi_S\}\,, \\ K_D &\triangleq \{z^\alpha(\pi)\mid \pi\in\Pi_D\}\,. \end{aligned}$$

Further introduce the set of vectors

$$\mathbf{Q}_\beta^\alpha \triangleq \left\{ z \,\middle|\, \begin{array}{r} \displaystyle\sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} (\delta_{xy} - \alpha p_{xy}) z_{xu} = (1-\alpha)\beta_y, \quad y \in \mathbf{X} \\ z_{xu} \geq 0 \quad (x,u) \in \mathbf{X} \times \mathbf{U}(x) \end{array} \right\}$$

The set $\mathbf{Q}_\beta^\alpha$ can be related to the sets of probability measures by the following theorem.

Theorem 4.3

$K = K_M = K_S = \overline{K_D} = \mathbf{Q}_\beta^\alpha$, where $\overline{X}$ is the convex hull of X.

Proof. See, for example, Altman (1999, Chapter 3). □

The theorem shows that, for the discounted case, it is no restriction to consider stationary policies, and that optimizing over the vectors $z \in \mathbf{Q}_\beta^\alpha$ corresponds to optimizing over the discounted state–action frequencies. In the next section, the SDP formulation that can be used to solve (4.6) is introduced.

4.5 Semidefinite program

To solve the constrained control problem (4.6), consider the optimization problem

$$\min_z \left\{ \sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} c(x,u) z_{xu} \,\middle|\, \begin{array}{c} z \in \mathbf{Q} \\ \displaystyle\sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} d^k(x,u) z_{xu} \succeq B^k, \; k = 1,\dots,m \end{array} \right\}, \tag{4.7}$$

where $\mathbf{Q} = \mathbf{Q}_\beta^\alpha$ for the discounted problem and $\mathbf{Q} = \mathbf{Q}_\beta$ for the expected average problem. Comparing $\mathbf{Q}_\beta^\alpha$ and $\mathbf{Q}_\beta$, it is seen that the vector $\tilde{z}$ is not required in the specification of $\mathbf{Q}_\beta^\alpha$. This means that the optimization problem (4.7) has half the number of decision variables in the discounted cost case compared to the expected average case.

Proposition 4.1

The optimization problem (4.7) *is an SDP.*

Proof. The constraints given by $d^k(x,u)$ and B^k are LMIs in the decision variables z and the set $\mathbf{Q}$ is defined by linear constraints on z. Furthermore, the objective is linear. Therefore the problem is an SDP, which proves the proposition. □

Since the problem (4.7) is an SDP, it can be solved by standard techniques, see, for example, the book by Boyd and Vandenberghe (2003). Given a solution z to the SDP (4.7), define the stationary policy π^∞ by

$$\pi^\infty_{xu} = \begin{cases} z_{xu}/z_x, & \text{if } z_x > 0, \\ \tilde{z}_{xu}/\tilde{z}_x, & \text{if } z_x = 0, \tilde{z}_x > 0, \\ \text{arbitrary}, & \text{otherwise}, \end{cases} \tag{4.8}$$

$$z_x = \sum_{u \in \mathbf{U}(x)} z_{xu}, \quad \tilde{z}_x = \sum_{u \in \mathbf{U}(x)} \tilde{z}_{xu}.$$

The relationship between the matrix CMDP (4.6) and the SDP (4.7) is formalized in the following theorem.

Theorem 4.4

The following hold

(i) The semidefinite program (4.7) *is feasible if and only if* (4.6) *is feasible.*

(ii) The optimal values of (4.7) *and* (4.6) *are equal.*

(iii) If π is an optimal policy for (4.6)*, then $z(\pi)$ is optimal for* (4.7).

For the discounted case, $\mathbf{Q} = \mathbf{Q}^\alpha_\beta$, it further holds that

(iv) If z is an optimal solution of (4.7)*, then the policy π^∞ defined in* (4.8) *is a stationary optimal policy for* (4.6).

Proof. The proof is analogous to the linear programming case in Altman (1999, Chapter 3) for the discounted case and Hordijk and Kallenberg (1984) for the expected average case.
(i)-(iii): For the discounted case, by Theorem 4.3, $K_S = \mathbf{Q}^\alpha_\beta$ and the statements follow. In the expected average case, by Theorems 4.1 and 4.2, $L_C = \mathbf{Q}_\beta$. Furthermore, for any convergent policy,

$$C_\beta(\pi) = \sum_{\substack{x \in \mathbf{X} \\ u \in \mathbf{U}(x)}} c(x,u) z_{xu}, \qquad D^k_\beta(\pi) = \sum_{\substack{x \in \mathbf{X} \\ u \in \mathbf{U}(x)}} d^k(x,u) z_{xu}$$

and the statements follow.
(iv): See, for example, Altman (1999, Chapter 3). □

The stationary policy π^{∞} is optimal for the discounted problem but may not be optimal for the expected average problem. However, optimality can be checked by the following lemma.

Lemma 4.1
If $z_{xu}/z_x = \tilde{z}_{xu}/\tilde{z}_x$ for all u and $x \in \{x|z_x > 0, \tilde{z}_x > 0\}$, then the stationary policy π^{∞} is an optimal policy for the expected average version of problem (4.6).

Proof. See, for example, Hordijk and Kallenberg (1984). □

Remark 4.2 It is possible to prove that if $(z, \tilde{z})$ is an optimal solution of (4.7), a corresponding optimal Markov (but not necessarily stationary) policy always exists. However, in practice, stationary policies are simpler to work with and therefore only such policies are considered also in the expected average case.

4.6 MDP with excitation

A matrix CMDP that approximately solves the experiment design problem can now be formulated. This means that the tuple $\{\mathbf{X}, \mathbf{U}, \mathcal{P}, c, d\}$ corresponding to the problem (4.1) needs to be specified. The first step is to find a suitable state space description for the dynamics of the Markov chain. The superscript c is used to distinguish states defined on a continuous state space from the states on the discrete state space used by the CMDP. Definition 4.1 requires that the immediate costs and rewards are functions of the current state. Since the excitation constraint uses the information matrix, the state space description

$$\xi_{t+1}^{c} = \mathcal{A}\xi_t^{c} + \mathcal{B}u_t + \mathcal{K}\begin{bmatrix} v_t \\ w_t \end{bmatrix}, \tag{4.9}$$

from (2.12) is used. The terms in the information matrix can be calculated from the state ξ_t^c using the relation (2.13).

To simplify notation for the constraints on the information matrix, the single output case is considered. However, extension to multiple outputs is possible. The components of the matrix CMDP are described in the following sections.

State and action spaces

The problems (4.6) and (4.7) are formulated for *finite* state and action spaces while the experiment design problem in (4.1) is defined for more general spaces. Therefore, an approximation of the state and action spaces by some suitable discretization may be needed. This is a well studied problem where a lot of attention has been paid to the curse of dimensionality. Here, a simple griding technique is employed.

Partition the sets $\mathcal{X}$, $\mathcal{U}$ into discrete regions, $\mathcal{X}_x$ and $\mathcal{U}_u$, indexed by $\mathbf{X} = \{1, 2, \ldots, n_x\}$ and $\mathbf{U} = \{1, 2, \ldots, n_u\}$, such that $\mathcal{X} = \bigcup_{x \in \mathbf{X}} \mathcal{X}_x$ and $\mathcal{U} = \bigcup_{u \in \mathbf{U}} \mathcal{U}_u$. A continuous state ξ^c is categorized as belonging to a certain discrete state x, for example, by

$$x = \arg\min_{x \in \mathbf{X}} \|\xi^c - \xi_x\|, \tag{4.10}$$

where ξ_x are the centers of the discretization regions. The continuous input u^c is categorized in the same way. This allows the system to be approximated by a finite Markov chain on the discrete state space $\mathbf{X}$ and the discrete action space $\mathbf{U}$, see, for example, the work by Lunze (1998) for a discussion on the properties of the approximation.

Remark 4.3 In modern industrial applications, control is most often implemented digitally which means that the action space is already discrete.

Transition probabilities

The transition probabilities for the discretized state and action spaces need to be calculated. This can be done in different ways, resulting in different approximations of the true state transitions. One possibility is to consider the conditional probability

$$p_{xy}(u) = \mathbb{P}\left\{\xi^c_{t+1} \in \mathcal{X}_y \,|\xi^c_t = \xi_x, u^c_t = u_u\right\}, \tag{4.11}$$

which means that the starting point for the transitions is considered to always be at the center of a discretization region. Assuming that the noise sources v_t and w_t have the joint Gaussian distribution

$$\begin{bmatrix} v_t \\ w_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda_v & 0 \\ 0 & \Lambda_w \end{bmatrix}\right),$$

the state ξ^c_{t+1} is, conditioned on $\xi^c_t = \xi_x, u^c_t = u_u$, distributed as

$$\xi^c_{t+1} \sim \mathcal{N}\left(\mathcal{A}\xi_x + \mathcal{B}u_u, \mathcal{K}\begin{bmatrix} \Lambda_v & 0 \\ 0 & \Lambda_w \end{bmatrix}\mathcal{K}^T\right),$$

which can be used to calculate the needed transition probabilities (4.11) by evaluating the resulting multivariate Gaussian integrals. This can, for example, be done numerically using the Monte Carlo techniques by Hammersley and Handscomb (1964). The

Monte Carlo techniques are also useful for other distributions of the noise processes. This and other schemes for discretization of MDPs and calculating the transition probabilities are discussed by Bertsekas (1975), Chow and Tsitsiklis (1991) and Munos and Moore (2002), for example.

Control cost

A quadratic immediate cost, similar to the quadratic cost (2.4) used in the MPC, is introduced according to

$$\begin{aligned} c(x,u) &= \mathbb{E}\left\{ x_{t+1}^T Q x_{t+1} + u_t^T S u_t + \Delta u_t^T R \Delta u_t \middle| \, x_t = x, u_t = u \right\} \\ &= \sum_{y \in \mathbf{X}} p_{xy}(u) y^T Q y + u^T S u + \Delta u^T R \Delta u, \end{aligned} \tag{4.12}$$

where Q, R and S are positive, semidefinite matrix weights. The expected average cost (4.2) or the discounted cost (4.4) can be formulated using the immediate costs $c(x,u)$.

Remark 4.4 The cost (4.12) includes Δu terms which require memory of the past input u_{t-1}. This can always be included in the MDP by adding an extra state, $\tilde{x}$, which evolves as $\tilde{x}_{t+1} = u_t$.

Excitation constraint

The excitation constraint can be formulated using an appropriate immediate reward. In the single output case, consider

$$\begin{aligned} I(x,u) &= \frac{1}{\lambda_v} \mathbb{E}\left\{ \mathcal{C} x_{t+1} x_{t+1}^T \mathcal{C}^T \middle| \, x_t = x, u_t = u \right\} \\ &= \frac{1}{\lambda_v} \sum_{y \in \mathbf{X}} p_{xy}(u) \mathcal{C} y y^T \mathcal{C}^T, \end{aligned} \tag{4.13}$$

where $\mathcal{C}$ comes from the extended state space system (2.12) used to calculate the information matrix. The $I(x,u)$ terms can be interpreted as matrix valued, immediate rewards for the matrix CMDP, corresponding to the $d^k(x,u)$ terms in Definition 4.1.

The average per-sample information matrix can, for convergent policies, be written in terms of the immediate rewards (4.13) as

$$\mathcal{I}(\theta_o) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_\pi^\beta \{ I(x_t, u_t) \} = \sum_{\substack{x \in \mathbf{X} \\ u \in \mathbf{U}(x)}} z_{xu} I(x,u). \tag{4.14}$$

The expression for $\mathcal{I}(\theta_o)$ in (4.14) corresponds to one of the expected average rewards D^k_β in the general matrix CMDP. In the discounted case, introduce

$$\mathcal{I}^\alpha(\theta_o) = \sum_{\substack{x\in\mathbf{X}\\ u\in\mathbf{U}(x)}} z^\alpha_{xu} I(x,u), \tag{4.15}$$

which corresponds to a discounted reward. This is not the actual average per-sample information matrix but can be used as an approximation thereof as the following lemma shows.

Lemma 4.2

For any convergent policy $\pi \in \Pi_C$

$$\mathcal{I}(\pi) = \lim_{\alpha\to 1^-} \mathcal{I}^\alpha(\pi).$$

Proof. By Hordijk and Kallenberg (1984, Theorem 3), $z_{xy}(u) = \lim_{\alpha\to 1^-} z^\alpha_{xy}(u)$ for any convergent policy and the lemma follows. □

The right hand side of the excitation constraint, which is given by

$$\frac{\gamma\chi^2_\alpha(n_\theta)}{2N} V''_{app}(\theta_o),$$

corresponds to the B^k term in the general matrix CMDP.

Optimal policy

The controller for the application-oriented experiment design problem (4.1) is given by the optimal policy $\pi^\star$ such that

$$C_\beta(\pi^\star) = \min_\pi \left\{ C_\beta(\pi) \;\middle|\; \mathcal{I}(\theta_o) \succeq \frac{\gamma\chi^2_\alpha(n_\theta)}{2N} V''_{\text{app}}(\theta_o) \right\}, \tag{4.16}$$

for the matrix CMDP defined on the discrete state and action spaces discussed previously in this section. The problem (4.16) can be solved using the SDP (4.7). Given a solution to the SDP, a stationary policy π_X can be found from (4.8).

Algorithm

The MDP with excitation is summarized in Algorithm 4.1.

Algorithm 4.1 MDP with excitation

```
Require: $V''_{app}(\theta_o), \gamma, \alpha, N$
solve (4.16) to obtain $\pi_X$.
while $t < N$ do
    measure $y_t$
    update $\hat{\xi}_t^c$ using (2.12)
    find discrete state $x$ using (4.10)
    find $u_t$ from $\pi_X$
    apply $u_t$ to the system
    $t \leftarrow t + 1$
end while
```

Remark 4.5 In Algorithm 4.1, the state is estimated using the regular Kalman filter and then a discrete state is found for the estimate. This is in the certainty equivalence spirit of MPC and simple to use. However, it is certainly not the optimal estimate in many cases. An alternative could be to introduce states for the distribution of the state estimate and, based on these, calculate a distribution over the discrete states.

4.7 Example

This example serves as a simple proof of concept for the presented matrix CMDP approach to solving the application-oriented experiment design problem. It is also shown how including the noise model in the design affects the expression for the information matrix and the resulting estimates. Consider the ARX system

$$\begin{cases} x_{t+1} = -\theta_1 x_t + \theta_2 u_t - \theta_1 v_t, \\ \quad y_t = x_t + v_t, \end{cases}$$

where v_t is Gaussian white noise with variance $\lambda_v = 1 \times 10^{-3}$, and the parameter vector $\theta = [\theta_1, \theta_2]$ is unknown and needs to be estimated. The true system is given by $\theta_0 = [0.5, 0.5]^T$, and the stationary optimal one step ahead predictor is

$$\hat{y}_{t+1|t} = -\theta_1 y_t + \theta_2 u_t. \tag{4.17}$$

The state space is extended with states for the predictor sensitivities according to (2.12), which gives

$$\xi_{t+1} = \begin{bmatrix} \theta_1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \xi_t + \begin{bmatrix} \theta_2 \\ 0 \\ 1 \end{bmatrix} u_t + \begin{bmatrix} \theta_1 \\ -1 \\ 0 \end{bmatrix} v_t,$$

$$\xi_t = \begin{bmatrix} x_t & \frac{\partial \hat{y}_t}{\partial \theta_1} & \frac{\partial \hat{y}_t}{\partial \theta_2} \end{bmatrix}^T.$$

The terms (4.14) needed for the excitation constraint are given by

$$\begin{aligned} I(x,u) &= \frac{1}{\lambda_v} \mathbb{E} \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xi_{t+1} \xi_{t+1}^T \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \middle| x_t = x, u_t = u \right\} \\ &= \frac{1}{\lambda_v} \mathbb{E} \left\{ \begin{bmatrix} (-x_t - v_t)^2 & (-x_t - v_t)u_t \\ (-x_t - v_t)u_t & u_t^2 \end{bmatrix} \middle| x_t = x, u_t = u \right\} \\ &= \frac{1}{\lambda_v} \begin{bmatrix} x^2 + \lambda_v & -xu \\ -xu & u^2 \end{bmatrix}. \end{aligned}$$

Therefore, no additional states are actually needed to include the excitation constraint.

The state space is discretized into 51 regions, uniformly spaced on the interval $[-1, 1]$ and the input into 21 regions, uniformly spaced on the interval $[-2, 2]$. The transition probabilities are calculated numerically by evaluating the transitions for each possible pair (x, u) in simulations. A quadratic control cost based on the immediate costs (4.12) is used with $Q = 2$, $R = 1$ and $S = 0$.

The identification objective is setup such that, after an experiment length of $N = 500$ samples, the average per-sample information matrix should satisfy

$$\mathcal{I}(\theta_o) \succeq \begin{bmatrix} 400 & -300 \\ -300 & 800 \end{bmatrix}. \tag{4.18}$$

The resulting SDP (4.7) is solved and a stationary policy π_X is found through (4.8). For comparison, a controller for experiment design (formulated in the next chapter) assuming that the system is of OE type, is also used. ARX models are estimated using the predictor (4.17) for the two methods in a Monte Carlo study with 100 trials to evaluate the performance of the methods. The 99 % confidence ellipses corresponding to the information matrices calculated from the two methods and the estimates are plotted in Figure 4.2. Both methods indicate that the performance specifications are met since the calculated confidence ellipses are inside the performance specification ellipsoid. However, with the OE model, the calculated ellipse is not the true one. This

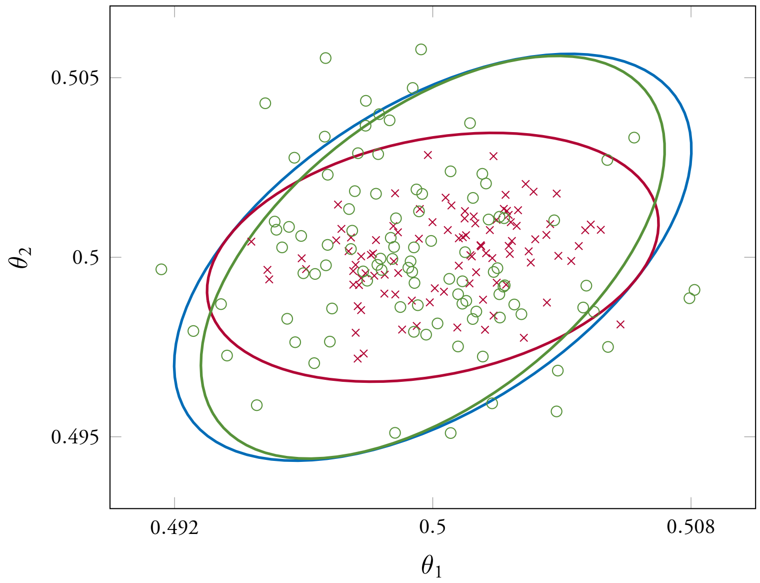


Figure 4.2 The confidence ellipses and estimates using an ARX model and an OE model for experiment design for the system in the example in Section 4.7. The performance specification is given by (——). Using an ARX model results in (——) and the estimates (×), while an OE model gives the confidence ellipse (——) and results in the estimates (○) In both cases, ARX models corresponding to the true system structure are estimated from the data. The ellipses are scaled to contain 99 % of the estimated parameters.

results in 15 of the estimates not fulfilling the requirements when the output error assumption is used in the experiment design, while only 1 of the estimates from the matrix CMDP approach does not satisfy the requirements.

4.8 Summary

In this chapter, a solution to the infinite-horizon application-oriented experiment design problem was formulated using a finite, matrix constrained Markov decision process formulation. Since the original problem is formulated for more general state and action spaces, the finite spaces used in the matrix CMDP formulation typically results in an approximate solution to the initial problem.

The formulation suffers from the curse of dimensionality if continuous state and action spaces need to be discretized and is therefore often impractical in applications. Furthermore, the formulation requires the implementation of a completely new type of controller. This is not feasible in many industrial situations. Nevertheless, the formulation may be of theoretical interest as it allows to study the problem for very general noise and disturbance processes. The resulting controller may also be used as a benchmark solution for other approximate schemes. Finally, for systems that inherently have discrete states and actions, the approach can be viable.

Chapter 5
Model predictive control formulation of experiment design

MODEL PREDICTIVE CONTROLLERS routinely require maintenance of the prediction models to give acceptable control performance as plant dynamics change over time. In the previous chapter, an optimal controller with dual properties that enable such model maintenance was developed. However, in applications where an MPC is already used, it is desirable to include the dual properties in the formulation of the MPC. Two controllers along these lines are presented in this chapter. The controllers are based on the application-oriented experiment framework. The major benefit of these formulations is that they build on existing MPC controllers. However, the algorithms are limited to OE models, unlike the MDP solution in Chapter 4, which works for very general noise structures.

The chapter starts by defining the assumed system and control structure. Then, the MPC from Chapter 2 is reformulated to facilitate the development of the MPC with excitation. Controllers for the minimum control cost and minimum time problems are designed and evaluated in simulations.

5.1 System and model

The considered systems are of output error (OE) type. Hence, the system is described by a state-space model of the form

$$\begin{cases} x_{t+1} = Ax_t + Bu_t, \\ y_t = Cx_t + w_t, \end{cases} \tag{5.1}$$

where $x_t \in \mathcal{X} \subseteq \mathbf{R}^n$ is the state, $u_t \in \mathcal{U} \subseteq \mathbf{R}^m$ is the input, $y_t \in \mathcal{Y} \subseteq \mathbf{R}^p$ is the output and $w_t \in \mathbf{R}^p$ is a zero mean, white, Gaussian noise process with covariance matrix Λ_v. The constraint sets $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{Y}$ are all convex sets. The system is modeled using

$$\begin{cases} \hat{x}_{t+1|t} = A(\theta)\hat{x}_{t|t-1} + B(\theta)u_t, \\ \quad y_t = C(\theta)\hat{x}_{t|t-1} + e_t, \end{cases} \tag{5.2}$$

where the parameter vector θ is estimated using PEM system identification. The estimate is based on N samples of input–output data. It is assumed that there exists a θ_o such that the model captures the true system and the asymptotic properties (2.9) hold for the estimate $\hat{\theta}_N$.

5.2 Controller

The controller is based on the general MPC formulation in (2.5). The cost is chosen as the quadratic cost

$$C_t = \sum_{k=1}^{N_y} \|\hat{y}_k - r_{t+k}\|_Q^2 + \sum_{k=0}^{N_u-1} \left(\|\Delta\hat{u}_k\|_R^2 + \|\hat{u}_k\|_S^2\right), \tag{5.3}$$

and the optimization problem solved at sample t is

$$\begin{aligned}
\underset{\{u_k\}_{k=0}^{N_y-1}}{\text{minimize}} \quad & C_t \\
\text{subject to} \quad & \hat{x}_{k+1} = A(\theta)\hat{x}_k + B(\theta)\hat{u}_k, && k = 0, 1, \ldots, N_y - 1, \\
& \hat{y}_k = C(\theta)\hat{x}_k, && k = 1, 2, \ldots, N_y, \\
& \hat{x}_0 = x_t, \\
& \Delta u_0 = \hat{u}_t - u_{t-1}, \\
& \hat{u}_{k+1} = \hat{u}_k, && k = N_u, N_u + 1, \ldots, N_y - 1, \\
& \hat{u}_k \in \mathcal{U}, && k = 0, 1, \ldots, N_y - 1, \\
& \hat{x}_k \in \mathcal{X}, && k = 1, 2, \ldots, N_y, \\
& \hat{y}_k \in \mathcal{Y}, && k = 1, 2, \ldots, N_y.
\end{aligned} \tag{5.4}$$

The choice to constrain the input to be constant after the control horizon is purely for simplicity and imposes no limitation in the formulation. The theory of MPC has become fairly well established and formal results on stability and robustness exists, see for example the survey by Mayne et al. (2000). Here, stability is handled by the following assumption.

Assumption 5.1 The MPC (5.4) is feasible for each time instant, the closed loop is stable for all initial states x_0 in the set $\mathcal{X}$.

Remark 5.1 The recursive feasibility and stability of Assumption 5.1 can be enforced using terminal constraints on the states, as discussed by Mayne et al. (2000).

Reformulation of the cost

It is convenient to reformulate the cost function (5.3) by including the dynamics of the predictions in (5.2). Introduce the notation

$$\bar{x} \triangleq \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{N_y} \end{bmatrix} \qquad \bar{y} \triangleq \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{N_y} \end{bmatrix}, \qquad \bar{r}_t \triangleq \begin{bmatrix} r_{t+1} \\ r_{t+2} \\ \vdots \\ r_{t+N_y} \end{bmatrix},$$

$$\bar{u} \triangleq \begin{bmatrix} \hat{u}_0 \\ \hat{u}_1 \\ \vdots \\ \hat{u}_{N_y} \end{bmatrix}, \qquad \bar{u}_t^{\star} \triangleq \begin{bmatrix} u_{t-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad \Delta\bar{u}_t \triangleq \begin{bmatrix} \hat{u}_0 - u_{t-1} \\ \hat{u}_1 - \hat{u}_0 \\ \vdots \\ \hat{u}_{N_u} - \hat{u}_{N_y-1} \end{bmatrix},$$

for the signals over the prediction horizon. The states and the output can be expressed as

$$\begin{aligned} \bar{x} &= \Upsilon(\theta)\bar{u} + \Psi(\theta)x_0, \\ \bar{y} &= \Sigma(\theta)\bar{x} = \Sigma(\theta)\left(\Upsilon(\theta)\bar{u} + \Psi(\theta)x_0\right), \end{aligned}$$

where the matrices $\Psi(\theta)$, $\Upsilon(\theta)$ and $\Sigma(\theta)$ are calculated using the prediction model (5.2). Further, introduce the matrix Γ such that

$$\Delta\bar{u}_t = \Gamma\bar{u} - \bar{u}_t^{\star}.$$

By defining

$$\begin{aligned} \mathcal{H}(\theta) &\triangleq \Upsilon(\theta)^T\Sigma(\theta)^T\left(I_{N_y} \otimes Q\right)\Sigma(\theta)\Upsilon(\theta) + \Gamma^T\left(I_{N_u} \otimes R\right)\Gamma + \left(I_{N_u} \otimes S\right), \\ \mathcal{G}_t(\theta) &\triangleq 2\left[\Sigma(\theta)\Psi(\theta)x_0 - \bar{r}_t\right]^T\left(I_{N_y} \otimes Q\right)\Sigma(\theta)\Upsilon(\theta) - 2\bar{u}_t^{\star T}\left(I_{N_u} \otimes R\right)\Gamma, \end{aligned}$$

where I_n is an identity matrix of dimensions $n \times n$ and $\otimes$ denotes the Kronecker product, the cost function (5.3) can be written as

$$\begin{aligned} C_t &= \bar{u}^T \mathcal{H}(\theta)\bar{u} + \mathcal{G}_t(\theta)\bar{u} + \text{constant} \\ &= \operatorname{tr} \mathcal{H}(\theta)\bar{u}\bar{u}^T + \mathcal{G}_t(\theta)\bar{u} + \text{constant}, \end{aligned} \tag{5.5}$$

where the cyclic property of the trace has been used and the constant term does not influence the optimization. The expressions for the matrices $\Psi(\theta)$, $\Upsilon(\theta)$, $\Sigma(\theta)$ and Γ can be found in Appendix 5.A.

Using the expression (5.5) and the remaining constraints, the MPC problem (5.4) can be written as the optimization problem

$$\begin{aligned} \underset{\bar{u}}{\text{minimize}} \quad & \operatorname{tr} \mathcal{H}(\theta)\bar{u}\bar{u}^T + \mathcal{G}_t(\theta)\bar{u} \\ \text{subject to} \quad & \hat{x}_0 = x_t, \\ & \bar{u} \in \widehat{\mathcal{U}}, \\ & \Upsilon(\theta)\bar{u} + \Psi(\theta)\hat{x}_0 \in \widehat{\mathcal{X}}, \\ & \Sigma(\theta)\left(\Upsilon(\theta)\bar{u} + \Psi(\theta)\hat{x}_0\right) \in \widehat{\mathcal{Y}}, \end{aligned} \tag{5.6}$$

where $\widehat{\mathcal{U}}$, $\widehat{\mathcal{X}}$ and $\widehat{\mathcal{Y}}$ are the constraint sets for the states and the signals over their respective horizons. This optimization program serves as the basis for the MPC formulations with dual properties.

5.3 Model predictive control with excitation (MPC-X)

In this section, an extended controller aimed at solving the the finite horizon minimum cost experiment design problem

$$\min_{\{u_t\}_{t=1}^N} \left\{ C \;\middle|\; \begin{aligned} x_{t+1} &= Ax_t + Bu_t, \\ y_t &= Cx_t + w_t, \\ u_t &\in \mathcal{U}, x_t \in \mathcal{X}, y_t \in \mathcal{Y}, \\ \mathcal{I}_1^N(\theta_o) &\succeq \frac{\gamma \chi_\alpha^2(n_\theta)}{2} V''_{\text{app}}(\theta_o) \end{aligned} \right\}, \tag{5.7}$$

is considered. In this formulation, the control cost C is calculated over a finite horizon of length N. The corresponding excitation constraint has been added to give a sufficiently large information matrix after an experiment of the same length N. As discussed in Chapter 3, this optimization problem can be very involved. Therefore, a receding horizon approach is taken using the controller in (5.6) as a starting point.

Augmented controller

The excitation constraint in (5.7) is included in the MPC (5.6). The receding horizon principle is used also for the formulation of the excitation constraint in the augmented MPC. The idea is to consider the information available from past data plus the predicted contribution over a suitable horizon, denoted N_I. Hence, at time t, the input is found by solving the optimization problem

$$\begin{aligned}
\underset{\bar{u}}{\text{minimize}} \quad & \operatorname{tr} \mathcal{H}(\theta_o)\bar{u}\bar{u}^T + \mathcal{G}_t(\theta_o)\bar{u} \\
\text{subject to} \quad & \hat{x}_0 = x_t, \\
& \bar{u} \in \widehat{\mathcal{U}}, \\
& \Upsilon(\theta_o)\bar{u} + \Psi(\theta_o)\hat{x}_0 \in \widehat{\mathcal{X}}, \\
& \Sigma(\theta_o)\left(\Upsilon(\theta_o)\bar{u} + \Psi(\theta_o)\hat{x}_0\right) \in \widehat{\mathcal{Y}}, \\
& \mathcal{I}_1^{t+N_I}(\theta_o) \succeq \kappa_t \frac{\gamma \chi_\alpha^2(n)}{2} V''_{\text{app}}(\theta_o).
\end{aligned} \tag{5.8}$$

The controller (5.8) will be called Model Predictive Control with eXcitation, or MPC-X.

A scaling factor κ_t is introduced to be able to control the excitation level at a given instant. This is needed since it is not reasonable to require that the excitation constraint is completely satisfied over the shorter horizon considered in MPC-X. For $\kappa_t = 1$, the information in the past data plus the contribution predicted by the MPC fulfills the original excitation constraint. One possible choice for the scaling is

$$\kappa_t = \frac{t + N_I}{N},$$

which should give an even excitation during the experiment. Further aspects of the choice of κ_t are discussed later in this section. An optimization based choice for the scaling, aimed at finding the highest possible excitation in each time instant, is introduced in Section 5.4.

In practice, the MPC-X controller in (5.8) is not possible to implement since it relies on knowledge of θ_o. Therefore, one has to rely on the best available model instead of the true description of the system. This means that in (5.8), the parameters θ_o would be replaced by an estimate, which will be denoted $\hat{\theta}$.

Predicted information matrix

The dependence on the input in the predicted information matrix of the excitation constraint in MPC-X needs to be made explicit for the formulation of the optimization problem. Using the fact that the system is OE and the best available estimate of the parameters, $\hat{\theta}$, the information matrix can be approximated by

$$\mathcal{I}_1^{t+N_I}(\theta_o) \approx I_1^{t+N_I}(\hat{\theta}) = I_1^t(\hat{\theta}) + I_{t+1}^{t+N_I}(\hat{\theta}), \tag{5.9}$$

where also the dependence of past and future data has been made explicit.

The evolution of the information matrix over the horizon can be described by defining the matrices Σ_ξ, Ψ_ξ and Υ_ξ in the same manner for the system (2.12) as the matrices Σ, Ψ and Υ are defined for the model (5.2), see Appendix 5.A. Then $\bar{\psi}_t$ in (2.12) is given by

$$\begin{bmatrix} \bar{\psi}_{t+N_I}^T & \bar{\psi}_{t+N_I-1}^T & \cdots & \bar{\psi}_{t+1}^T \end{bmatrix}^T = \Sigma_\xi \left(\Psi_\xi \xi_t + \Upsilon_\xi \bar{u} \right). \tag{5.10}$$

Using (5.10), the matrix

$$\begin{aligned} \Sigma_\xi \left(\Psi_\xi \xi_t + \Upsilon_\xi \bar{u} \right) \left(\Psi_\xi \xi_t + \Upsilon_\xi \bar{u} \right)^T \Sigma_\xi^T &= \Sigma_\xi \Psi_\xi \xi_t \xi_t^T \Psi_\xi^T \Sigma_\xi^T + \\ & \Sigma_\xi \Psi_\xi \xi_t \bar{u}^T \Upsilon_\xi^T \Sigma_\xi + \Sigma_\xi \Upsilon_\xi \bar{u} \xi_t^T \Psi_\xi^T \Sigma_\xi^T + \Sigma_\xi \Upsilon_\xi \bar{u} \bar{u}^T \Upsilon_\xi^T \Sigma_\xi^T, \end{aligned} \tag{5.11}$$

can be formed. The diagonal blocks of (5.11) are the terms $\mathcal{C}\xi_t\xi_t^T\mathcal{C}^T$, with $\mathcal{C}$ defined in (2.11), and the elements of $I_{t+1}^{t+N_I}(\hat{\theta})$ are sums of the elements of these blocks according to (2.13) and (2.14). Note that the terms $\mathcal{C}\xi_t\xi_t^T\mathcal{C}^T$ are quadratic in the decision variable $\bar{u}$ due to the last term of the right hand side of (5.11). However, by introducing a lifting variable, U, such that

$$U = \bar{u}\bar{u}^T, \tag{5.12}$$

and using (5.12) in (5.11), it is seen that the expression is linear in U and $\bar{u}$. As a result, the experiment design constraint is an LMI in the decision variables U and $\bar{u}$. It should be pointed out that constraint (5.12) is a nonlinear constraint in the new decision variables U and $\bar{u}$, which can be rewritten as

$$\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \qquad \operatorname{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1.$$

Lifted optimization problem

The optimization problem in (5.8) can be expressed using the new decision variables U and $\bar{u}$, and the best available estimate $\hat{\theta}$, as the lifted problem

$$
\begin{aligned}
\underset{U,\bar{u}}{\text{minimize}} \quad & \operatorname{tr} \mathcal{H}(\hat{\theta})U + G_t(\hat{\theta})\bar{u} \\
\text{subject to} \quad & \hat{x}_0 = x_t, \\
& \bar{u} \in \widehat{\mathcal{U}}, \\
& \Upsilon(\hat{\theta})\bar{u} + \Psi(\hat{\theta})\hat{x}_0 \in \widehat{\mathcal{X}}, \\
& \Sigma(\hat{\theta}) \left(\Upsilon(\hat{\theta})\bar{u} + \Psi(\hat{\theta})\hat{x}_0\right) \in \widehat{\mathcal{Y}}, \\
& I_1^t(\hat{\theta}) + I_{t+1}^{t+N_I}(\hat{\theta}) \succeq \kappa_t \frac{\gamma\chi_\alpha^2(n_\theta)}{2} V''_{\text{app}}(\hat{\theta}), \\
& \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \qquad \operatorname{rank} \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1.
\end{aligned}
\tag{5.13}
$$

The formulation of the lifted optimization problem in MPC-X is inspired by the input design problem considered by Manchester (2010). In the lifted problem (5.13), the constraints should be rewritten in terms of the new decision variables U and $\bar{u}$. How this is done depends on the nature of the constraints. A very common situation in practice is that the input and output (or states) are subject to box constraints. How such constraints can be handled is shown in the following example.

Example 5.1 (Box constraints)

Consider the constraints

$$
\begin{aligned}
\bar{u} \in \widehat{\mathcal{U}} &= \{u \,|u_{min} \leq u \leq u_{max}\}, \\
\Sigma\left(\Upsilon\bar{u} + \Psi\hat{x}_0\right) \in \widehat{\mathcal{Y}} &= \{y \,|y_{min} \leq y \leq y_{max}\},
\end{aligned}
$$

where u_{min},u_{max}, y_{min} and y_{max} are scalars. In the new decision variables U and $\bar{u}$, the above constraints can be written as

$$
\begin{aligned}
\operatorname{diag} U - \bar{u}u_{max} - \bar{u}u_{min} &\leq -u_{min}u_{max}, \\
\operatorname{diag} \Sigma\Upsilon U \Upsilon^T \Sigma^T - \tilde{y}_{min}\Sigma\Upsilon\bar{u} - \tilde{y}_{max}\Sigma\Upsilon\bar{u} &\leq -\tilde{y}_{min}\tilde{y}_{max}, \\
\tilde{y}_{min} &= y_{min} - \Sigma\Psi\hat{x}_0, \\
\tilde{y}_{max} &= y_{max} - \Sigma\Psi\hat{x}_0.
\end{aligned}
$$

◊

Proposition 5.1

The optimization problem in (5.13) *is nonconvex.*

Proof. The constraint

$$\operatorname{rank}\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1,$$

is nonconvex, since a convex combination of rank 1 matrices need not have rank 1. Therefore, the problem (5.13) is nonconvex. □

Remark 5.2 The nonconvexity of the feasible set in (5.13) is concentrated in the rank constraint. All other constraints are convex in the decision variables.

Convex relaxation

Although (5.13) is nonconvex, a convex relaxation of the problem can be found by simply dropping the rank constraint. This corresponds to the relaxation introduced by Shor (1987). The solution to this relaxed problem is the matrix U and the vector $\bar{u}$. An input can be found by drawing a sample of the random variable

$$\tilde{u} = \bar{u} + \beta D^T \zeta, \qquad DD^T = U - \bar{u}\bar{u}^T, \qquad \zeta \sim \mathcal{N}(0, I), \tag{5.14}$$

as suggested by Manchester (2010) and Luo et al. (2010). Here, β is a scaling parameter chosen as large as possible such that the constraints are satisfied. Finally, the input applied to the process can be extracted from the m first elements of $\tilde{u}$.

Analysis

A complication due to the receding horizon principle used in MPC-X is that the addition to the information matrix is only a predicted contribution. Since only the first input from the optimal solution of (5.13) is implemented on the system, only the corresponding part is added to the information matrix. Therefore, it is possible that a badly tuned MPC-X controller postpones part of the needed excitation indefinitely. However, under certain conditions, the experiment design constraint is guaranteed to be fulfilled. First, an assumption on the recursive feasibility of the optimization is needed.

Assumption 5.2 The optimization problem (5.13) is feasible at each time instant.

Remark 5.3 The optimization problem (5.13) may be infeasible due to the conflicting nature of the constraints. In such a case κ_t can always be adjusted to give a feasible problem.

Observation 5.1 If κ_t is chosen such that (5.13) is feasible in each time instant, MPC-X renders the closed loop system stable, assuming that the MPC (5.4) is used for $t > N$. This follows from the fact that recursive feasibility of MPC-X ensures that $x_N \in \mathcal{X}$ and stability follows from Assumption 5.1.

Proposition 5.2

A necessary condition for feasibility of (5.13) *for every time instant* t *is that* $N_I \geq \operatorname{rank} V''_{app}(\hat{\theta})$.

Proof. Feasibility of (5.13) requires that

$$I_{t+1}^{t+N_I}(\hat{\theta}) \succeq \kappa_t \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta}) - I_1^t(\hat{\theta}),$$

where the right hand side has up to $\operatorname{rank} V''_{\text{app}}(\hat{\theta})$ positive eigenvalues. Since $I_{t+1}^{t+N_I}(\hat{\theta})$ is the sum of N_I rank 1 matrices, it has at most N_I positive eigenvalues. Hence, $N_I \geq \operatorname{rank} V''_{\text{app}}(\hat{\theta})$ is required for feasibility for every time instant. □

Theorem 5.1

Suppose that Assumption 5.2 holds. If $V''_{app}(\hat{\theta})$ *has full rank,* κ_t *is deterministic and increasing such that* $\kappa_t \to \infty$*, and the input is amplitude bounded, then, for a fixed* $\hat{\theta}$*, there exists an* N *such that*

$$I_1^N(\hat{\theta}) \succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta}).$$

If Assumption 5.2 is strengthened so that (5.13) *is feasible for all possible* x *at each time instant, then also*

$$\mathcal{I}_1^N(\hat{\theta}) \succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta}).$$

Proof. Let $\tau = \frac{\gamma \chi^2_\alpha(n_\theta)}{2}$. Assumption 5.2 gives

$$I_1^{t-1}(\hat{\theta}) + I_t^t(\hat{\theta}) \succeq \kappa_t \tau V''_{\text{app}}(\hat{\theta}) - I_{t+1}^{t+N_I}(\hat{\theta}), \tag{5.15}$$

where $I_t^t(\hat{\theta}_N)$ is known to be added to the information matrix since the system is OE. Take the eigenvector z corresponding to the smallest eigenvalue of the right hand side of (5.15) such that $\|z\| = 1$, then

$$\begin{aligned}
\lambda_{\min}\left(I_1^{t-1}(\hat{\theta}) + I_t^t(\hat{\theta})\right) &\geq z^T \left[\kappa_t \tau V''_{\text{app}}(\hat{\theta}) - I_{t+1}^{t+N_I}(\hat{\theta})\right] z \\
&\geq \kappa_t \min_{\|z\|=1} z^T \tau V''_{\text{app}}(\hat{\theta}) z - \max_{\|z\|=1} z^T I_{t+1}^{t+N_I}(\hat{\theta}) z \\
&= \kappa_t \lambda_{\min}\left(\tau V''_{\text{app}}(\hat{\theta})\right) - C,
\end{aligned}$$

where $C < \infty$ is an upper bound on the largest eigenvalue of $I_{t+1}^{t+N_I}(\hat{\theta}_N)$, which is bounded since u_t and x_t are bounded. Hence, for N such that

$$\kappa_N \geq \frac{\lambda_{\max}\left(\tau V''_{\text{app}}(\hat{\theta})\right) + C}{\lambda_{\min}\left(\tau V''_{\text{app}}(\hat{\theta})\right)},$$

which is bounded since V''_{app} is invertible, $\lambda_{\min}\left(I_1^N(\hat{\theta})\right) \geq \lambda_{\max}\left(\tau V''_{app}(\hat{\theta})\right)$, which implies that $I_1^N(\hat{\theta}) \succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta})$.

Under the assumptions, N is deterministic. Furthermore, under the stronger assumption that (5.13) is feasible for all possible x at each time instant, $I_1^N(\hat{\theta}) \succeq \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta})$ holds for every possible trajectory. Hence

$$\mathcal{I}_1^N(\hat{\theta}) = \mathbb{E}\left\{I_1^N(\hat{\theta})\right\} \succeq \mathbb{E}\left\{\frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta})\right\} = \frac{\gamma \chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta}).$$

□

Remark 5.4 The only stochastic part of the right-hand side of the excitation constraint is the parameter estimate. For a fixed parameter vector, the right-hand side is completely deterministic.

Remark 5.5 The bound on κ_t and the resulting N in Theorem 5.1 are very loose because of the unnecessarily strong requirements on the eigenvalues of the information matrix. In practice, the constraint is typically fulfilled for much lower values of N.

Remark 5.6 The assumption that (5.13) is feasible for all possible x in every time instant is a fairly strong requirement on the scaling κ_t. The possible choices of κ_t depend on the assumptions made on the disturbances.

MPC-X algorithm

The complete MPC-X algorithm is given in Algorithm 5.1.

Algorithm 5.1 MPC with excitation (MPC-X)

```
Require: $\hat{\theta}, V''_{app}(\hat{\theta}), \gamma, \alpha, \kappa_t$.
t = 0, ξ_0 = 0.  ($t = 0, \xi_0 = 0.$)
while $I_1^t \prec \frac{\gamma\chi^2_\alpha(n_\theta)}{2} V''_{app}$ do
    Measure $y_t$ and estimate $x_t$.
    Solve (5.13) without the rank constraint.
    Obtain feasible input sequence $\tilde{u}$ from (5.14).
    Apply $\tilde{u}_t$ to the system.
    $t \leftarrow t + 1$.
end while
```

5.4 Model predictive control with excitation for minimum time

An extension of the controller (5.4) for the minimum time experiment design problem

$$\min_{\{u_t\}_{t=1}^N} \left\{ N \;\middle|\; \begin{array}{r} x_{t+1} = Ax_t + Bu_t, \\ y_t = Cx_t + w_t, \\ u_t \in \mathcal{U}, x_t \in \mathcal{X}, y_t \in \mathcal{Y}, \\ \mathcal{I}_1^N(\theta_o) \succeq \dfrac{\gamma\chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\theta_o) \end{array} \right\},$$

is considered in this section. The idea is to find the maximal allowed excitation in each time instant and thereby get the shortest experiment time. The idea is inspired by the work of Žáčeková et al. (2012).

Augmented controller

To get the excitation required to fulfill the excitation constraint in the shortest time, the controller is formulated to maximize the scaling κ_t in the augmented controller (5.8) in each time instant. Hence, at time t, the input is found from the optimization problem

$$\begin{aligned}
\underset{\bar{u},\kappa_t}{\text{maximize}} \quad & \kappa_t \\
\text{subject to} \quad & \hat{x}_0 = x_t, \\
& \bar{u} \in \widehat{\mathcal{U}}, \\
& \Upsilon(\theta_o)\bar{u} + \Psi(\theta_o)\hat{x}_0 \in \widehat{\mathcal{X}}, \\
& \Sigma(\theta_o)\left(\Upsilon(\theta_o)\bar{u} + \Psi(\theta_o)\hat{x}_0\right) \in \widehat{\mathcal{Y}}, \\
& \mathcal{I}_1^{t+N_I}(\theta_o) \succeq \kappa_t \frac{\gamma\chi^2_\alpha(n)}{2} V''_{\text{app}}(\theta_o), \\
& \operatorname{tr} \mathcal{H}(\theta_o)\bar{u}\bar{u}^T + \mathcal{G}_t(\theta_o)\bar{u} \leq C_t^\star + \Delta C.
\end{aligned} \tag{5.16}$$

The controller (5.16) will be called Minimum time MPC-X.

The last constraint in (5.16) is added to limit the control performance degradation due to the excitation. The optimal value from the solution of the regular MPC problem (5.4) at time t, denoted $C_t^\star$ in (5.16), is used as a reference value for the control performance. If maximal excitation is wanted, $\Delta C = \infty$ can be chosen, or equivalently the last constraint can be removed. Note that constraints on either u, x, y or the control cost are needed for a well-defined problem.

Just as in the case for MPC-X, the minimum time MPC-X is not implementable, as it is formulated in (5.16), since it relies on knowledge of θ_o. Therefore, one again has to rely on the best available estimate $\hat{\theta}$, and the approximation (5.9) for the information matrix. To solve the optimization problem (5.16), an upper bound for the scaling κ_t is found in the following section.

Adapting κ_t.

The value of κ_t can be maximized by considering a suitable upper bound. Introduce $\tilde{V} = \frac{\gamma\chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta})$, which can be written as $\tilde{V} = \tilde{V}^{1/2}\tilde{V}^{1/2}$ since the information matrix is symmetric and positive semidefinite. Then the excitation constraint is equivalent to

$$\begin{aligned}
& I_1^t(\hat{\theta}) + I_{t+1}^{t+N_I}(\hat{\theta}) \succeq \kappa_t \frac{\gamma\chi^2_\alpha(n_\theta)}{2} V''_{\text{app}}(\hat{\theta}) \\
\Leftrightarrow\; & \tilde{V}^{-1/2}\left[I_1^t(\hat{\theta}) + I_{t+1}^{t+N_I}(\hat{\theta})\right]\tilde{V}^{-1/2} \succeq \kappa_t I \\
\Leftrightarrow\; & \lambda_{\min}\left\{\tilde{V}^{-1/2}\left[I_1^t(\hat{\theta}) + I_{t+1}^{t+N_I}(\hat{\theta})\right]\tilde{V}^{-1/2}\right\} \geq \kappa_t.
\end{aligned} \tag{5.17}$$

Equation (5.17) gives an upper bound on the value of κ_t which can be maximized by solving the lifted optimization problem

$$\begin{aligned} \underset{U,\bar{u}}{\text{maximize}} \quad & \lambda_{\min}\left\{\tilde{V}^{-1/2}\left[I_1^t(\hat{\theta}) + I_{t+1}^{t+N_u}(\hat{\theta})\right]\tilde{V}^{-1/2}\right\} \\ \text{subject to} \quad & \hat{x}_0 = x_t, \\ & \bar{u} \in \widehat{\mathcal{U}}, \\ & \Upsilon(\hat{\theta})\bar{u} + \Psi(\hat{\theta})\hat{x}_0 \in \widehat{\mathcal{X}}, \\ & \Sigma(\hat{\theta})\left(\Upsilon(\hat{\theta})\bar{u} + \Psi(\hat{\theta})\hat{x}_0\right) \in \widehat{\mathcal{Y}}, \\ & \operatorname{tr}\mathcal{H}(\hat{\theta})U + \mathcal{G}_t(\hat{\theta})\bar{u} \leq C_t^\star + \Delta C, \\ & \begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} \succeq 0, \qquad \operatorname{rank}\begin{bmatrix} U & \bar{u} \\ \bar{u}^T & 1 \end{bmatrix} = 1, \end{aligned} \tag{5.18}$$

where new decision variables have been introduced in the same way as was done in (5.13). The optimization problem (5.18) is nonconvex for the same reasons as for (5.13), see Proposition 5.1. A convex relaxation can be found using the same relaxation techniques as for (5.13).

Remark 5.7 An alternative interpretation is that (5.18) works as a back-off mechanism. In case there is some unforeseen disturbance in the process, the algorithm finds the maximal allowed excitation. It thus adapts the level of excitation to the current operating conditions.

Observation 5.2 Under Assumption 5.1, minimum time MPC-X is feasible in each time instant and the resulting closed loop is stable. This follows from the fact that, under the same assumption, MPC is feasible and any feasible solution to (5.4) is also feasible for (5.18). Furthermore, feasibility of (5.18) ensures $x \in \mathcal{X}$ which gives stability by Assumption 5.1.

Minimum time MPC-X algorithm

The complete MPC-X algorithm is given in Algorithm 5.2.

5.5 Examples

Here the performance of the two MPC controllers with excitation developed in this chapter are evaluated in simulation studies. First, the MPC-X algorithms are compared

Algorithm 5.2 Minimum time MPC-X

```
Require: $\hat{\theta}, V''_{app}(\hat{\theta}), \gamma, \alpha$.
$t = 0, \xi_0 = 0$.
while $I_1^t \prec \frac{\gamma\chi^2_\alpha(n_\theta)}{2} V''_{app}$ do
    Measure $y_t$ and estimate $x_t$.
    Solve (5.4) to obtain $C_t^\star$.
    Solve (5.18) without the rank constraint.
    Obtain feasible input sequence $\tilde{u}$ from (5.14).
    Apply $\tilde{u}_t$ to the system.
    $t \leftarrow t + 1$.
end while
```

to one of the persistently exciting MPC formulations presented in Section 2.5. Second, the motivating example in Section 1.2 is revisited and studied in detail.

Example 1 — Comparison with PE-MPC

This example compares the properties of the MPC-X algorithm (5.13) and the minimum time MPC-X (5.18) to the properties of the PE-MPC by Marafioti (2010), outlined in Section 2.5. The optimization in the two MPC-X algorithms is implemented in Matlab using CVX by Grant and Boyd (2011) and the resulting SDP is solved using SDPT3 by Toh et al. (1999). PE-MPC is implemented as it is presented by Marafioti (2010). The two resulting quadratic programs are solved using the quadprog function from the Matlab optimization toolbox.

The example is a simple model of two interconnected tanks. An upper tank is connected to a pump with input u_t. The tank has a hole in the bottom with free flow into a lower tank, which has a hole with free flow out of the tank. The level in the lower tank is the output, y_t, which can be measured. The system is modeled using the discrete-time, output error model

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_3 & \theta_4 \\ 1 & 0 \end{bmatrix} x_t + \begin{bmatrix} 4.5 \\ 0 \end{bmatrix} u_t, \\ \quad y_t = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix} x_t + e_t. \end{cases} \tag{5.19}$$

The true system is given by the parameter values $\theta_o = \begin{bmatrix} 0.12 & 0.059 & 0.74 & -0.14 \end{bmatrix}^T$ and the noise variance is $\lambda_e = 0.01$. The goal is to control the level in the lower tank using MPC with the following settings: $N_y = N_u = 5, Q = 10, R = 1$ and $S = 0$. The considered scenario is such that the identification is started at steady state conditions of the plant. The input is constrained to be between -1 and 1. No constraints are imposed on the outputs.

The true parameter values are used for the initial model. In a practical application, one would instead have to use estimated parameters, for example from a commissioning identification experiment. The application cost is chosen as

$$V_{\text{app}}(\theta) = \sum_{t=1}^{T} \|y_t(\theta_o) - y_t(\theta)\|_2^2, \tag{5.20}$$

over a step response of the system with MPC running. Hence, it is desired that, when the identified model is used in MPC, the step response is close to what it had been if the true system parameters were available. The desired accuracy is set to $\gamma = 100$ and should be achieved with 99 % probability. The experiments are set up so that they run until the achieved information matrix fulfills the experiment design constraint. Two different comparisons are made.

First, the benefit of the experiment design constraint over the persistence of excitation constraint is investigated in experiments of the same length. MPC-X is configured with experiment length $N = 200$ and linear scaling, $\kappa_t = \frac{t}{N}$. The PE-MPC is configured with $\rho = 0.5$ and backward horizon $P = 5$, which gives a similar evolution of the smallest eigenvalue of the information matrix as with MPC-X and therefore similar experiment length.

Second, the benefit of the experiment design constraint in a minimum time experiment is investigated. The PE-MPC is configured with $\rho = 1$, which is the highest value that gives a feasible solution to the optimization problem. The minimum time MPC-X is configured with $\Delta C = 5$, which gives a similar evolution of the information matrix. For comparison, regular MPC is also used and the resulting information matrix is calculated.

The results of the simulations are summarized in Table 5.1. First, it is seen that PE-MPC with $\rho = 0.5$ satisfies the experiment design constraint after 211 samples which is approximately the same as the 200 samples required for MPC-X. The input variances are comparable but the resulting output variance using PE-MPC is 0.12 compared to 0.074 for MPC-X, which is also reflected in the higher control cost for PE-MPC, 287, compared to 180 for MPC-X. Second, PE-MPC with $\rho = 1$ requires 162 samples to fulfill the experiment design constraint compared to 82 samples for the minimum time MPC-X. This is despite the fact that the PE-MPC results in higher signal variances and control cost (per sample and total). For the chosen application, MPC-X results in cheaper experiments in both scenarios. On the other hand, a clear benefit of PE-MPC

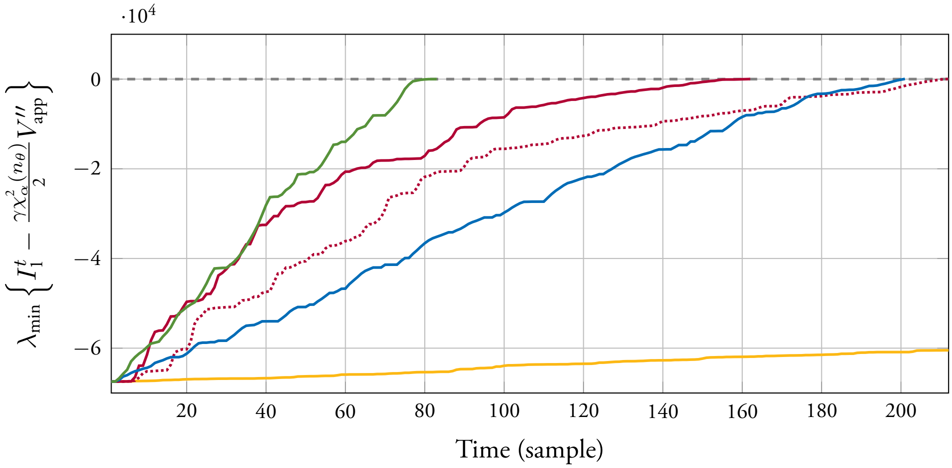


Figure 5.1 Evolution of the smallest eigenvalue of the difference between the information matrix and the scaled application cost Hessian from simulation where regular MPC (——), MPC-X (——), the PE-MPC algorithm with $\rho = 1$ (——) and $\rho = 0.5$ (⋯⋯), and minimum time MPC-X (——) are used on the system in Example 1. The algorithms terminate when the eigenvalue becomes positive.

Table 5.1 Results for the algorithms used in Example 1. The value c_t is the average control cost per sample and $\sum c_t$ is the total cost over the whole experiment length.

Algorithm	var u	var y	N	c_t	$\sum c_t$
MPC	0.036	0.016	—	0.22	—
MPC-X	0.164	0.074	200	0.90	180
PE-MPC, $\rho = 0.5$	0.175	0.120	211	1.36	287
MPC-X, minimum time	0.203	0.146	82	1.66	138
PE-MPC, $\rho = 1$	0.246	0.184	162	2.07	335

is a much lower computational demand. The two convex quadratic programs in PE-MPC typically require much less time and computations to solve than the SDPs in the two MPC-X formulations. Figure 5.1 shows the evolution of the smallest eigenvalue of the experiment design constraint, which should be positive for the constraint to hold. Clearly, all the dual MPC formulations increase the information in the data and regular MPC is not useful for identification. Typical signals from the algorithms are shown in Figures 5.2 and 5.3.

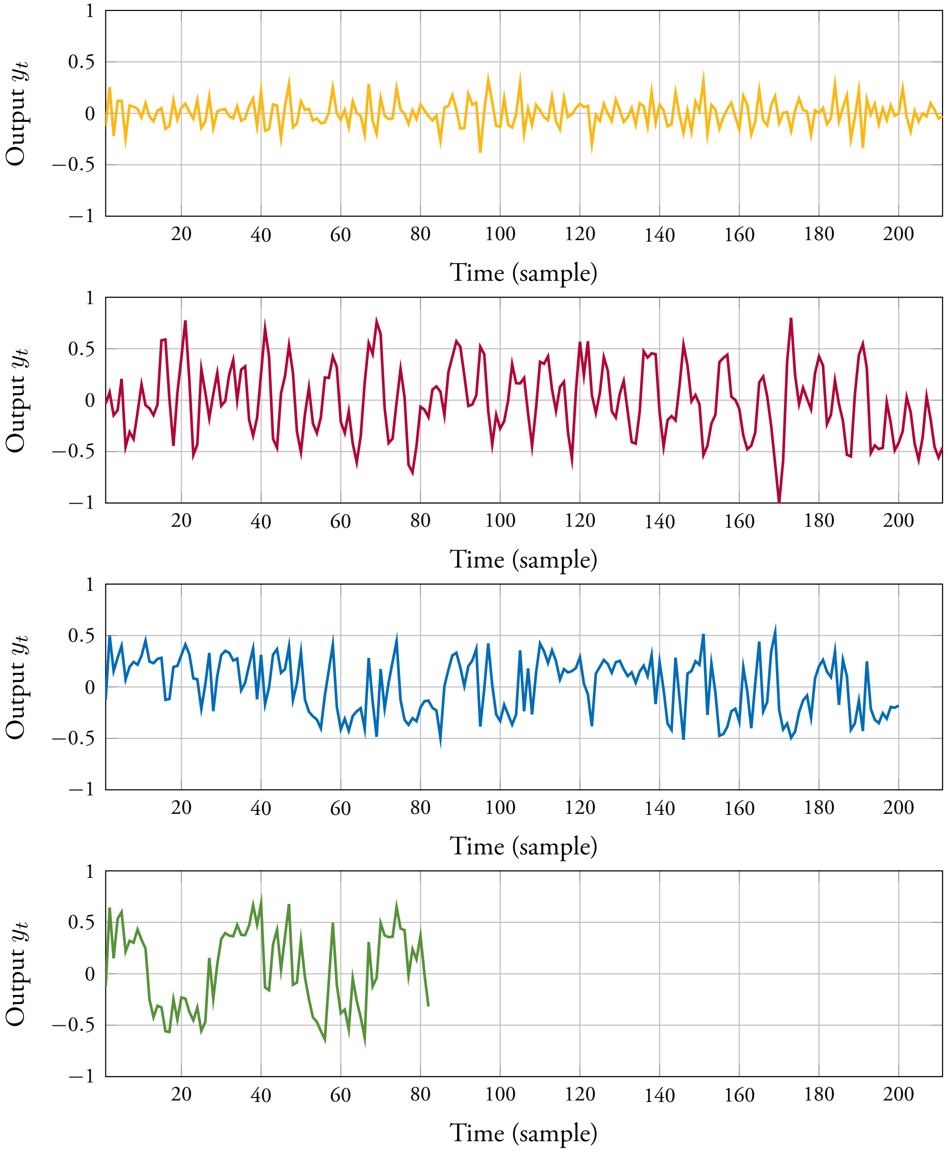


Figure 5.2 Typical output signals from the system in Example 1, when the input is generated by regular MPC (——), PE-MPC with $\rho = 1$ (——), MPC-X (——), and minimum time MPC-X (——).

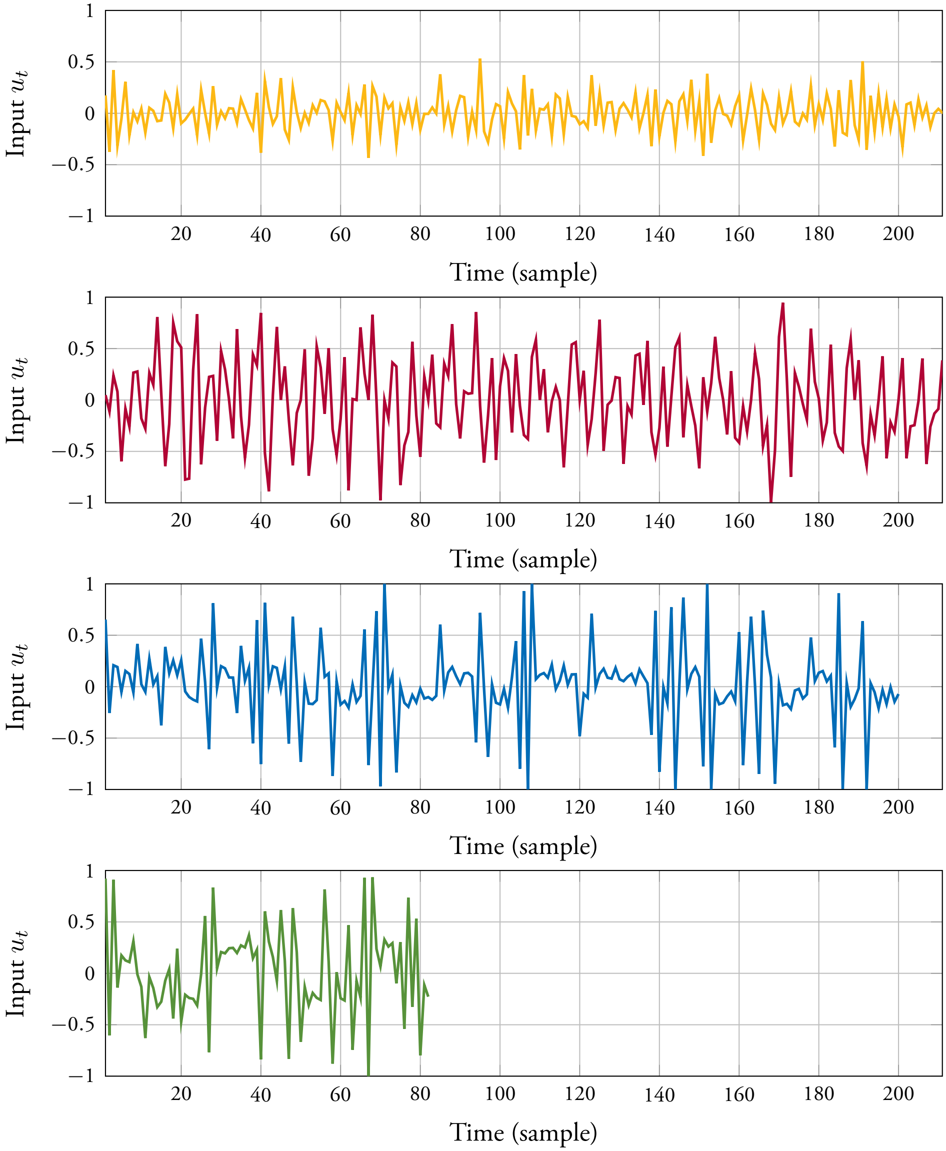


Figure 5.3 Typical input signals generated by regular MPC (—), PE-MPC (—) with $\rho = 1$, MPC-X (—), and minimum time MPC-X (—) for the system in Example 1.

Example 2 — Distillation column simulation

The example from Section 1.2 is revisited and studied in more detail. The MPC-X algorithm is applied to a nonlinear model of the binary distillation column presented by Skogestad (1997). The column is configured with 110 trays. The feed is located at tray 39 and the nominal feed rate is 219 kmol/min. The relative volatility of the two substances is $\alpha = 1.35$ and the liquid holdup, M, is constant at 30 kmol. The feed composition, z_F, is set to 0.65 mole fraction. The column is controlled in LV-configuration, using the liquid flow, L, and vapor flow, V, as control inputs while the feed rate and composition are considered unmeasured disturbances. The outputs are the top and bottom compositions, denoted y^{t} and y^{b}, respectively. The setpoints of the outputs are $r^{\mathrm{t}} = 0.95$ and $r^{\mathrm{b}} = 0.05$. For the example, only the distillation column is modeled, while auxiliary systems such as reboiler and condenser are not included in the model. Disturbances are introduced in the system by adding low-pass filtered noise to the nominal values of the feed rate and composition, such that $\mathrm{var}(F) = 64$ and $\mathrm{var}(z_F) = 2.5 \times 10^{-3}$. White measurement noise with variance 0.1 is added to the outputs.

Model and controller

A linear state space model with two states, two inputs and two outputs is used to model the process for control purposes. Each element of the state space matrices A, B and K corresponds to one of the parameters in the parameter vector θ and the measurement matrix C is considered to be known. The linear model can be written as

$$\begin{cases} x_{t+1} = \begin{bmatrix} \theta_1 & \theta_2 \\ \theta_3 & \theta_4 \end{bmatrix} x_t + \begin{bmatrix} \theta_5 & \theta_6 \\ \theta_7 & \theta_8 \end{bmatrix} u_t + \begin{bmatrix} \theta_9 & \theta_{10} \\ \theta_{11} & \theta_{12} \end{bmatrix} e_t, \\ y_t = Cx_t + e_t, \end{cases} \tag{5.21}$$

where C is known and θ needs to be estimated. Since MPC-X assumes that the system is of OE type, the input design is made assuming $\theta_9, \dots, \theta_{12} = 0$ in (5.21). However, in the identification, the parameters in K are estimated. The initial model is found by system identification of the nonlinear model.

The MPC-X is configured with prediction and control horizons $N_y = N_u = 8$. A practical auto-tuning method proposed by Annergren et al. (2013) is used to find the tuning weights Q and R. Since the MPC is used for reference tracking the matrix S is set to zero. The tuning finds a trade-off between robustness to modeling errors, and closed-loop bandwidth.

The performance of the controller is measured according to the ideas in Section 3.3 and the performance measure is defined as

$$p(\theta) = \sum_{t=1}^{N} \left\{ \frac{\mathbb{I}\left(y_t^{\mathrm{t}} < y_{\min}^{\mathrm{t}}\right) + \mathbb{I}\left(y_t^{\mathrm{b}} > y_{\max}^{\mathrm{b}}\right)}{N} + |y_t^{\mathrm{t}} - r^{\mathrm{t}}| + |y_t^{\mathrm{b}} - r^{\mathrm{b}}| \right\}. \tag{5.22}$$

In the first two terms of the sum, $\mathbb{I}$ is the indicator function and used to estimate the probability of constraint violations, where $y_{\min}^{\mathrm{t}} = 0.92$ and $y_{\max}^{\mathrm{b}} = 0.08$ are the constraints on the top and bottom products of the column. The two last terms of the sum penalize deviations from the setpoint values. Both components are considered equally important and weighted equally. The performance measure is calculated over a window of $N = 400$ samples.

The Hessian, V''_{app}, needed for the specification of the experiment design constraint is calculated numerically based on the performance measure (5.22).

Simulation scenario

The simulation is setup to show the benefit of MPC-X when closed-loop reidentification is necessary to restore controller performance. The simulation is divided into four phases, each lasting for 400 samples. The phases are:

1. Normal operation using initial model.

2. Operation under plant change.

3. Reidentification phase.

4. Operation using MPC based on the new model from the third phase.

The performance measure (5.22) is calculated for each phase. A Monte Carlo study with 310 trials is made to evaluate the identification capabilities. The benefit of the extra excitation added in MPC-X is evaluated by running simulations using the same noise sequences but without the excitation constraint active in the MPC.

The plant change is made by making a small rotation in the input gain on the process. This choice is motivated by the well-known fact that the directionality is a nuisance of distillation columns. The effect is similar to the situation with uncertainty presented by Skogestad et al. (1988). The rotation is achieved by multiplying the input signals calculated by the controller by the matrix

$$\begin{bmatrix} \cos\left(-\frac{\pi}{5}\right) & \sin\left(-\frac{\pi}{5}\right) \\ -\sin\left(-\frac{\pi}{5}\right) & \cos\left(-\frac{\pi}{5}\right) \end{bmatrix},$$

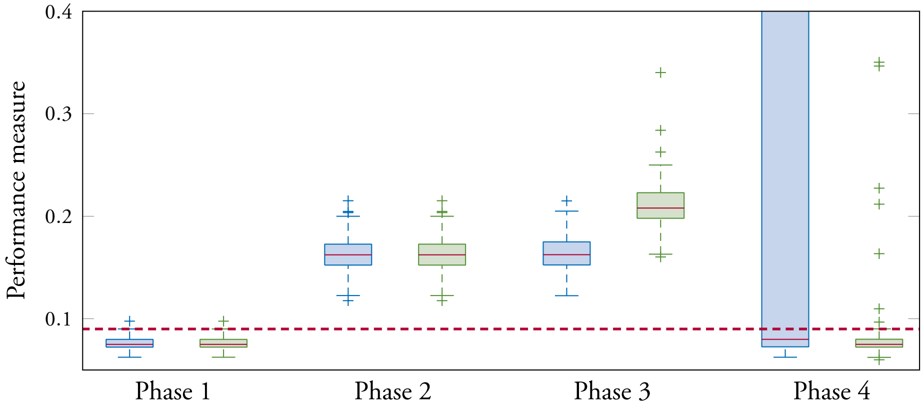


Figure 5.4 Box plots of the performance measures of the four phases of the 310 Monte Carlo trials when regular MPC (🟦) and MPC-X (🟩) is used. During the excitation phase 3, the cost of using MPC-X is higher than for regular MPC where no extra excitation is added. However, when the new models are used in the controllers in phase 4 the number of models with acceptable performance is significantly higher for the MPC-X case. The acceptable level is below the dashed line.

An explanation of the plant change and resulting performance drop is given in more detail by Tran et al. (2012).

Results

The performance index for the four different phases of the two simulation setups are illustrated in Figure 5.4. It is clear that adding the constraint results in degradation of performance during the reidentification phase. However, the new model results in acceptable performance in 94 % of the cases. Note that the spread of the resulting performance measure is small. This should be compared to the case without the excitation constraint, where the reidentification does not degrade performance, but the resulting models are able to restore performance in only 62 % of the cases. Furthermore, while the median performance measure is reasonable, the spread is very large. Another observation is that when the model has been identified from normal closed loop data, 13 % of the cases result in a closed loop that is unstable, while this happened for only 2 % of the models identified data generated by MPC-X. One typical simulation is shown in Figure 5.5.

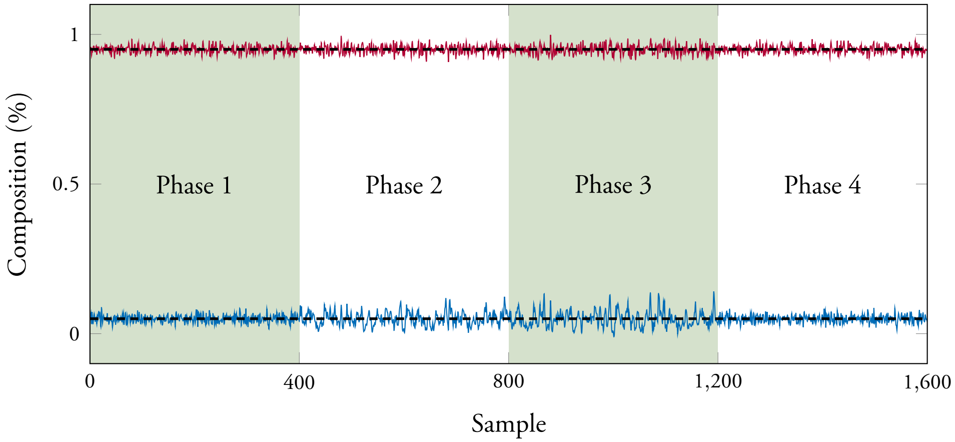


Figure 5.5 The top (——) and bottom (——) compositions for the plant change scenario when MPC-X with extra excitation during phase 3 is used. The performance measures for the different phases are 0.08, 0.15, 0.20, 0.07, respectively. The setpoints for the top and bottom compositions are shown as (----).

Discussion

The goal of 99 % reidentified models giving acceptable performance is not reached. There are a number of reasons that contribute to the discrepancy. Firstly, the model in the controller cannot fully capture the plant, which is nonlinear. Secondly, MPC-X calculates the excitation based on an OE model, while the plant is not OE. It is seen in Figure 4.2 that not using the true noise model can lead to too little excitation and a higher number of models which give unsatisfactory performance. However, the results are still clearly better when the extra excitation is added using MPC-X compared to when no excitation is added.

5.6 Summary

The application-oriented experiment design problem has been added to existing formulations for MPC to formulate two new controllers: MPC-X and Minimum time MPC-X. The inclusion of the experiment design constraint ensures that closed loop data can be used for reidentification of the process model. The MPC-X controllers have dual properties and the intended use of the model is explicitly taken into account in the generation of the excitation. The controllers are limited to OE models for the calculation of the needed excitation, which is the major restriction of these methods at present.

It was shown, that under stability assumptions on the closed loop, the generated excitation is sufficient to reidentify a model which satisfies the application performance demands. The merits of the two controllers in comparison to PE-MPC, where the use of the model is not explicitly considered, was shown in simulation examples. The two proposed MPC-X controllers generate data that result in models with acceptable performance in cheaper experiments than when only persistently exciting inputs are generated by the controller. This is not surprising since PE-MPC is not designed to take the experiment design constraint into account. The price to be paid for the cheaper experiment is higher computational complexity: MPC-X requires the solution of an SDP while PE-MPC is solved by two convex quadratic programs.

5.A MPC matrices

Using the model (5.2), the output at time t can be predicted by

$$\hat{y}_t = C(\theta)A(\theta)^t x_0 + \sum_{k=1}^{t} C(\theta)A(\theta)^{t-k}B(\theta)u_{k-1}.$$

Introducing the matrices $\Psi \in \mathbf{R}^{nN_y \times n}$, $\Sigma \in \mathbf{R}^{pN_y \times nN_y}$, and $\Upsilon \in \mathbf{R}^{pN_y \times mN_u}$ defined by

$$\Sigma(\theta) \triangleq I_{N_y} \otimes C(\theta),$$

$$\Psi(\theta) \triangleq \begin{bmatrix} A(\theta) \\ A(\theta)^2 \\ \vdots \\ A(\theta)^{N_y} \end{bmatrix},$$

$$\Upsilon(\theta) \triangleq \begin{bmatrix} B(\theta) & 0 & \cdots & 0 \\ A(\theta)B(\theta) & B(\theta) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A(\theta)^{N_y-N_u}B(\theta) & A(\theta)^{N_y-N_u-1}B(\theta) & \cdots & B(\theta) \\ A(\theta)^{N_y-N_u-1}B(\theta) & A(\theta)^{N_y-N_u-2}B(\theta) & \cdots & A(\theta)B(\theta)+B(\theta) \\ \vdots & \vdots & \vdots & \vdots \\ A(\theta)^{N_y-1}B(\theta) & A(\theta)^{N_y-2}B(\theta) & \cdots & \sum_{k=N_u}^{N_y} A(\theta)^{N_y-k}B(\theta) \end{bmatrix},$$

where $u_{t+N_u} = \cdots = u_{t+N_y}$ is used to construct the last columns of $\Upsilon(\theta)$, gives

$$\begin{aligned} \bar{x} &= \Psi(\theta)x_0 + \Upsilon(\theta)\bar{u} \\ \bar{y} &= \Sigma(\theta)x_0 = \Sigma(\theta)\left(\Psi(\theta)x_0 + \Upsilon(\theta)\bar{u}\right). \end{aligned}$$

The matrix

$$\Gamma \triangleq \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 1 \end{bmatrix},$$

gives

$$\Delta\bar{u}_t = \Gamma\bar{u} - \bar{u}_t^{\star}.$$

Chapter 6
Experimental study of MPC-X on an industrial distillation column

The MPC-X algorithm developed in Chapter 5 has been implemented and evaluated in experiments on an industrial process at the Sasol Synthetic Fuels refinery in Secunda, South Africa. The experiments were conducted during two separate, four weeks long site visits in April and October 2013. The considered test process is a distillation column, a depropanizer, where propane and lighter hydrocarbons are separated from butane and heavier components. The depropanizer is part of the cold side of the Synfuels Catalytic Cracker (SCC) unit where high-boiling, high-molecular weight hydrocarbon fractions of synthetic crude oil are converted to gasoline, alkene gases and other products similar to an FCC unit in a conventional refinery. The SCC unit and the depropanizer are shown in Figure 6.1.

The goal of the experiments was to demonstrate the capabilities of MPC-X for generating signals useful for closed-loop reidentification in a typical industrial application. The experiments include initial process identification, an MPC commissioning, and controller tuning. Thereafter, a performance drop is induced by creating a plant–model mismatch. Control performance is restored by updating the model in the MPC by means of closed-loop reidentification.

The experiments were first conducted on a full-plant simulator built for operator training, which is implemented in the Honeywell UniSim Operations Suite. The simulator is believed to represent the dynamics of the actual plant well. Experiments were also performed on the actual depropanizer. These experiments were conducted during regular production. This means that the conditions represent the intended application of MPC-X well. It also means that it was vital that any experiments did not disrupt the normal operations of the plant or degrade product quality. The experiments show that by using MPC-X, it is possible to improve performance by reidentification after a performance drop has been observed.

Figure 6.1 The SCC unit of the Sasol Synfuels plant in Secunda, South Africa. The depropanizer is the middle column in the background. Photo courtesy of Sasol.

6.1 Depropanizer

The depropanizer is a total reflux distillation column used to separate three-carbon hydrocarbons (C_3) and lighter components, such as propane and propene, from four-carbon hydrocarbons (C_4) and heavier components, such as butane and butene. The distillation is possible using a 56 tray column because of the significant difference in molecular weight of the two components. A schematic overview of the depropanizer is shown in Figure 6.2. There are four upstream units which supply the depropanizer feed and a feed surge drum is used to even out the feed to the column. To allow for further control of the feed to the depropanizer, part of the flow from the drum can be sent to another downstream unit, referred to as CatPoly. The feed enters the depropanizer on tray 34 and product is taken from the depropanizer at a side draw on tray 39. Water is removed from the product in the liquid dryer stage before being sent further downstream to the C_3 splitter. The C_4 rich bottom product of the depropanizer is cooled and sent to CatPoly. On tray 19, vapor for pressurizing the feed surge drum is drawn from the depropanizer. The description of the depropanizer is based on

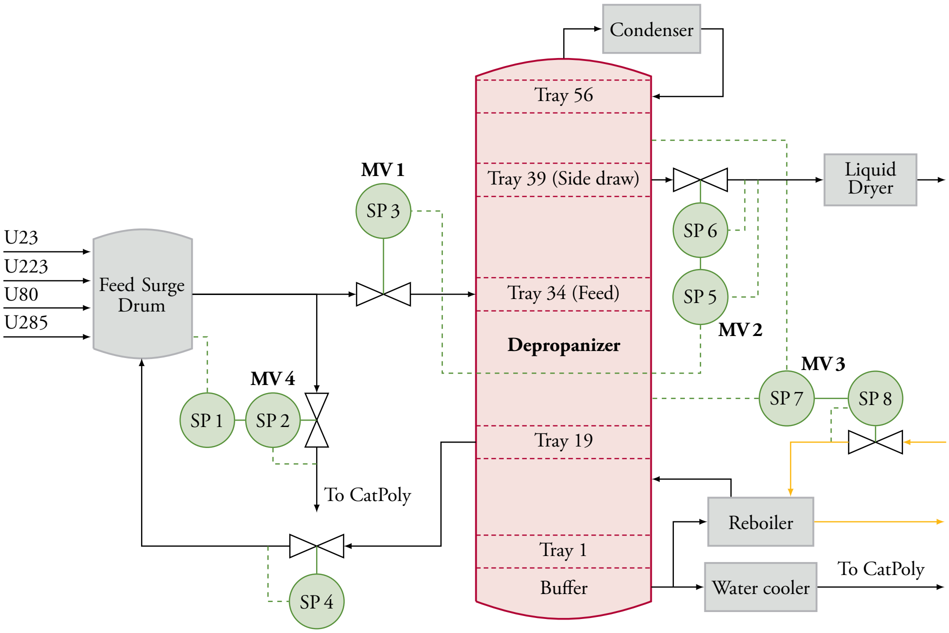


Figure 6.2 Schematic overview of the depropanizer system. Green indicates control points, where circles are low level controllers, typically PID, solid lines are controller outputs and dashed lines are feedback measurements. The set-points that are controlled by MPC are indicated as MV 1–4. The inputs to the feed surge drum come from four upstream units.

information from Kotze et al. (2011) and Pieterse et al. (2011).

Base layer control

Several PID loops are used to achieve the base level control for the depropanizer.

Feed surge drum level is controlled using a set of cascaded PID loops and maintained by regulating the flow from the drum to CatPoly. The setpoint of the PID control is given by operators by changing SP 1 or by MPC by directly changing SP 2. When the level is set by the operator, SP 2 is a measurable disturbance in the MPC.

Feed flow to the depropanizer is controlled by a PID loop. The setpoint of the PID is SP 3 which can be controller by MPC or by the operator, in which case it becomes a measurable disturbance in the MPC.

Feed surge drum pressure is controlled by a PID loop where the setpoint SP 4 is set by the operator.

Side draw flow is controlled by two cascaded PID loops and MPC controls the setpoint of the outer loop, SP 5.

Differential pressure between the top and bottom of the depropanizer is controlled by two cascaded PID loops which control the flow to the reboiler. The setpoint of the outer PID loop, SP 7, is set by MPC.

MPC objectives

The control objective of the MPC for the depropanizer is to maximize the side draw flow, which maximizes the C_3 production rate, while keeping product within specifications. In this case, the specifications are reflected in the amount of remaining C_4 components in the side draw. Secondary objectives include prevention of column flooding, excess flaring and sudden C_4 breakthrough by maintaining variables within their limits. The MPC uses 11 controlled variables (CVs) and 4 manipulated variables (MVs). The variables and their constraints are summarized in Tables 6.1 and 6.2.

The MPC is setup to move CV 1 to a setpoint value to maintain product specifications, while all other controlled variables are maintained within their limits. Therefore, level control of the feed drum (CV 2) is maintained by lower level controllers and MV 4 (which is mostly used to control the feed drum level) is not used for control but instead considered to be a measurable disturbance. It is desirable to have a steady flow through the process and therefore MV 1 should be used only to prevent flooding of the column. Consequently, MV 2 and MV 3 are left to control the distillation process. For technical reasons, CV 7 is not available to the MPC, however, an increase in MV 2 increases the product yield. Therefore, a setpoint of 69.5 % is introduced for MV 2. These considerations are reflected in the choice of the tuning matrices of the MPC, which are set to

$$\begin{aligned} Q &= \operatorname{diag}\begin{bmatrix} q_y & 0 & \cdots & 0 \end{bmatrix}, \\ R &= \operatorname{diag}\begin{bmatrix} 25 & 1 & 1 \end{bmatrix}, \\ S &= \operatorname{diag}\begin{bmatrix} 5 & 2 \times 10^{-5} & 0 \end{bmatrix}. \end{aligned}$$

The value of q_y is used to tune the bandwidth of the controller based on the practical auto tuning method presented by Annergren et al. (2013). The goal is to find a balance between high bandwidth and good robustness properties.

Table 6.1 Manipulated and controlled variables of the depropanizer and their limits.

	Variable	Low	High	Unit
MV 1	Column feed flow	45	65	×[c]
MV 2	Side draw to feed ratio	40	70	%
MV 3	Column differential pressure	30	39	kPa
MV 4	Feed to C_3 header	1	70	×[c]
CV 1	C_4 in side draw	—	2 000	ppm
CV 2	Feed drum level	40	70	%
CV 3	Feed drum level ROC[a]	—	—	%
CV 4	Column pressure	1 170	1 400	kPa
CV 5	Bottom temperature	—	90	°C
CV 6	Feed flow OP[b]	1	99	%
CV 7	Side draw flow OP[b]	1	99	%
CV 8	Reboil	0	14	×[c]
CV 9	Reboil flow OP[b]	1	99	%
CV 10	Reflux flow OP[b]	1	99	%
CV 11	Feed to C_3 header OP[b]	1	99	%

[a]Rate of change, [b]Valve opening percentage, [c]Not shown for confidentiality reasons

Performance measures on the depropanizer

Two performance measures are used to evaluate the performance of the MPC on the depropanizer. The key process variable for the depropanizer is CV 1, which is also the only output that is controlled to a setpoint value. Therefore, both measures use only the signal CV 1.

The first measure is a variance based measure, which will be the main performance measure for the depropanizer. The chosen measure is

$$p(\theta) = \frac{1}{T}\sum_{t=1}^{T} y_t(\theta)^2 = \frac{1}{T}\|\tilde{y}(\theta)\|^2, \tag{6.1}$$

where $\tilde{y} = \begin{bmatrix} y_1(\theta) & y_2(\theta) & \cdots & y_T(\theta) \end{bmatrix}^T$ and $y_t(\theta)$ is the deviation of CV 1 around the steady state value when the depropanizer is controlled using MPC with a model based on the parameters θ.

The second measure is an application cost used for the specification of the excitation constraint in MPC-X. The application cost measures the degradation resulting

Table 6.2 Allowed move sized of the manipulated variables.

	Variable	Up	Down
MV 1	Column feed flow	0.3	0.3
MV 2	Side draw to feed ratio	0.5	0.25
MV 3	Column differential pressure	0.2	0.2
MV 4	Feed to C_3 header	5	5

from using a controller based on model parameters $\theta \neq \theta_o$. For the depropanizer, the cost is chosen to be

$$V_{\text{app}}(\theta) = \frac{1}{T}\sum_{t=1}^{T}(y_t(\theta) - y_t(\theta_o))^2 = \frac{1}{T}\|\tilde{y}(\theta) - \tilde{y}(\theta_o)\|^2. \tag{6.2}$$

The application cost can be seen as one of the tuning knobs of MPC-X, which controls the excitation added to the system. It is a means for achieving the ultimate goal of good performance, as measured by $p(\theta)$. The application cost, $V_{\text{app}}(\theta)$, and the performance measure, $p(\theta)$, can be related through

$$\begin{aligned}\sqrt{p(\theta)} &= \frac{1}{\sqrt{N}}\|\tilde{y}(\theta) - \tilde{y}(\theta_o) + \tilde{y}(\theta_o)\| \\ &\leq \frac{1}{\sqrt{N}}\|\tilde{y}(\theta) - \tilde{y}(\theta_o)\| + \frac{1}{\sqrt{N}}\|\tilde{y}(\theta_o)\| \\ &\leq \sqrt{\frac{1}{\gamma}} + \sqrt{p(\theta_o)},\end{aligned}$$

which implies that

$$p(\theta) \leq \left(\sqrt{\frac{1}{\gamma}} + \sqrt{p(\theta_o)}\right)^2. \tag{6.3}$$

Hence, the choice of the application cost and the related upper bound γ result in a corresponding upper bound for the performance measure. Since, in practice, θ_o is not available, it is replaced by the best current estimate, $\hat{\theta}$, see Section 3.3. For the depropanizer this estimate is taken from the initial commissioning identification.

6.2 Experimental setup

The goal of the experiments is to demonstrate the capabilities of MPC-X on a full scale, typical, distillation column found in petrochemical refineries. This is done by setting up an MPC-X controller for the plant, inducing a performance drop and restoring performance to acceptable levels by reidentifying a new model from closed-loop data where MPC-X has excited the system. The experiments are set up to be as similar as possible on the simulator as on the plant; any significant differences between the tests are explained when necessary. The main steps of the experiment are now outlined.

1. Initial identification

The initial model needed for setting up the MPC-X controller is identified on the open-loop system in various identification experiments, both using step inputs and using random, binary signals. On the plant, due to limited experiment time, only tests using random, binary signals are conducted.

2. Controller design and tuning

Based on the model from the initial identification, an MPC-X controller is setup and tuned for the optimal closed-loop bandwidth according to the control objectives in Section 6.1.

3. Induced drop in performance

On the simulator, it would be possible to change the dynamics of the plant to emulate the system degrading over time or a sudden change in the plant characteristics. However, on the real plant this is not possible since the system is used in production. Therefore, a change in system dynamics is made by modifying the dynamics of the model used by the MPC-X controller. In other words, a model-plant mismatch is caused by keeping the plant dynamics unchanged while altering the prediction model of the controller.

The chosen strategy is to scale the state transition matrix by a factor α, which moves the poles of the model. That is, the two models are related through

$$A_\alpha = \alpha A, \tag{6.4}$$

where A_α is the state transition matrix of the degraded model.

4. Closed-loop reidentification

The excitation mode of MPC-X is turned on to excite the system to give signals rich enough for closed-loop identification. The data is used to identify new models to be used in the controller.

5. Performance assessment

The control performance using the new model from the closed-loop identification is assessed by running the MPC-X in regular control mode without excitation and evaluating the resulting output variances and application costs.

Results

The results on the simulator and plant are presented in the following two sections. The focus in this chapter lies on the results of the reidentification experiments where MPC-X is the enabling technology. For an in depth discussion on the tuning, performance measure and diagnosis results, see Guidi et al. (2013) or Annergren et al. (2013).

6.3 Simulator experiments

Summary

Two experiments with excitation from MPC-X and closed-loop reidentification were performed. The change in the controller model resulted in an increase in the performance measure, $p(\theta)$, from the nominal value 0.0082 to 0.043 in the degraded case, which is considered unacceptable performance. In both cases, MPC-X restores performance to an acceptable level when the new model is used. The scaling factor, κ_t in MPC-X is changed in a different manner in the two experiments, resulting in different excitation signals and final bounds on the achievable performance. In Table 6.4 it is seen that $p(\theta)$ is reduced to 0.012 and 0.018 in the two experiments. Both results are below the bound (6.3) and the resulting application costs are below $1/\gamma$. The experiments are described in more detail in the following sections.

Operating point

One operating point is chosen for all the experiments on the simulator. The corresponding manipulated variables values and the resulting steady state values of the controlled variables are presented in Table 6.3.

Initial model

The model structure chosen for the simulator is a state space model with 11 states, 3 inputs (MV 1 – MV 3) and 1 measured disturbance (MV 4). Due to the physics of the depropanizer, the model is fitted to the logarithm of the first output, CV 1. This is a

Table 6.3 Operating point of the manipulated variables and corresponding steady state values of the controlled variables for the simulator experiments.

	Variable	Value	Unit
MV 1	Feed flow	55	×[c]
MV 2	Side draw to feed ratio	47	%
MV 3	Column differential pressure	32.78	kPa
CV 1	C_4 in side draw	5.44	log ppm
CV 2	Feed drum level	50	%
CV 4	Column pressure	1 302.4	kPa
CV 5	Bottom temperature	81.96	°C
CV 6	Feed flow OP[b]	58.0	%
CV 8	Reboil	10.48	×[c]
CV 9	Reboil flow OP[b]	89.6	%
CV 10	Reflux flow OP[b]	63	%
CV 11	Feed to C_3 header OP[b]	3.19	%

[b] Valve opening percentage, [c] Not shown for confidentiality reasons

common technique in modeling of distillation columns. Therefore, the model is

$$\begin{cases} x_{t+1} = A(\theta)x_t + B(\theta)u_t + B_d(\theta)d_t, \\ y_t = C(\theta)x_t + e_t, \\ z_t = \begin{bmatrix}\exp y_{1,t} & y_{2,t} & \cdots & y_{9,t}\end{bmatrix}^T, \end{cases}$$

where z_t is the output, u_t the input and d_t the measured disturbance.

In the open-loop identification experiments, the simulator is excited using a random, binary signal. A white Gaussian signal is filtered through an eight-order, non-causal Butterworth low-pass filter and the sign of the resulting signal, appropriately scaled, is used. The amplitudes used are ±2000 for MV 1, ±2.25 for MV 2 and ±1.25 for MV 3.The values of the parameters are obtained using PEM in the System identification toolbox of Matlab. The resulting fit is shown in Figure 6.3 and the step responses from the MVs to CV 1 are shown in Figure 6.4.

Controller setup

Based on the initial model, the sampling time of the controller is chosen to be 0.5 Hz. The prediction and control horizons are chosen as $N_y = 50$, such that the predictions

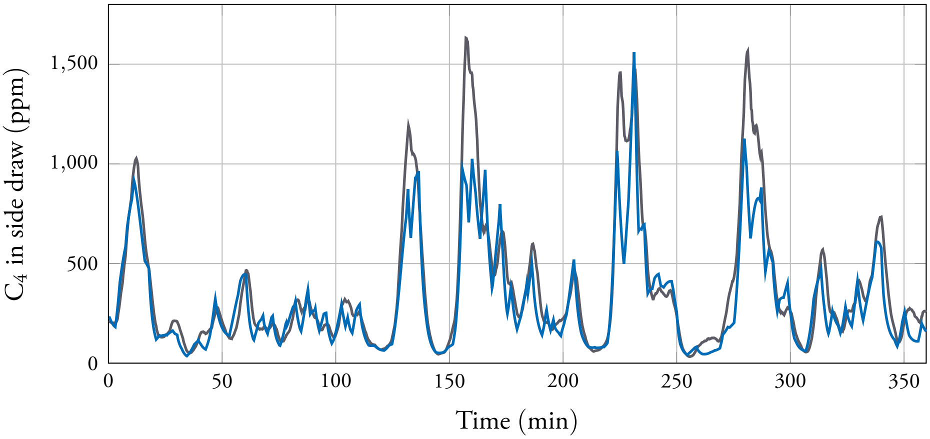


Figure 6.3 The simulator output (——) and the data from simulation of the model obtained from open-loop identification (——). The achieved fit is 53.4 %.

cover the main dynamics of the process, and $N_u = 10$, which is a trade-off between, on the one hand, sufficiently long horizon for good control and fulfillment of the excitation constraint and, on the other hand, computational tractability. The controller is tuned to give good disturbance rejection, which is a balance between bandwidth and robustness to modeling errors. In Figure 6.5, where the spectrum of CV 1 is shown, it can be seen that the two main low frequency oscillations in the output are removed by the controller. However, it is also seen that some new, high frequency oscillations have appeared in the closed loop.

Induced performance drop

The model of the MPC is modified according to (6.4) with $\alpha = 0.98$. This increases the variance of CV 1 from 0.0082 to 0.043 as shown in Table 6.4. This is an increase of 420 % in the main performance measure $p(\theta)$ and considered to be a significant drop in performance. The resulting application cost is 0.062.

MPC-X reidentification

Two closed-loop reidentification experiments are performed using excitation from MPC-X. The value of the scaling factor κ_t, which controls the excitation level, is adjusted manually in the experiments and an a priori performance level, $^1/_\gamma$, can therefore not be guaranteed. Instead the experiments run for a pre-determined number of samples and the resulting application cost bound is calculated at the end of the experiments. The excitation signals from both experiments are shown in Figure 6.6.

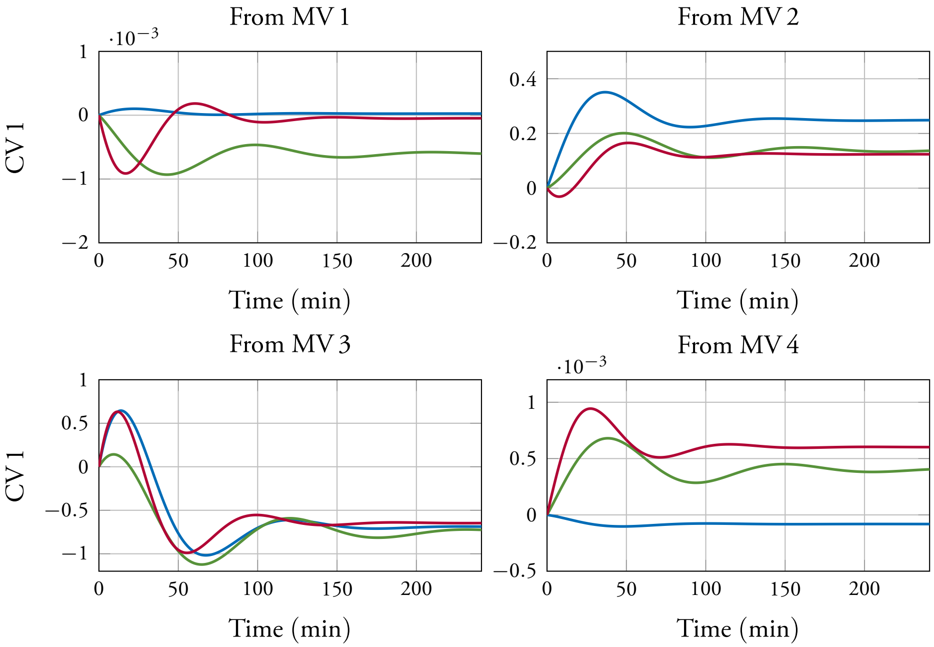


Figure 6.4 The step responses to CV 1 of the model fitted to the logarithmically scaled open-loop data (——) and MPC-X data from experiment 1 (——) and experiment 2 (——).

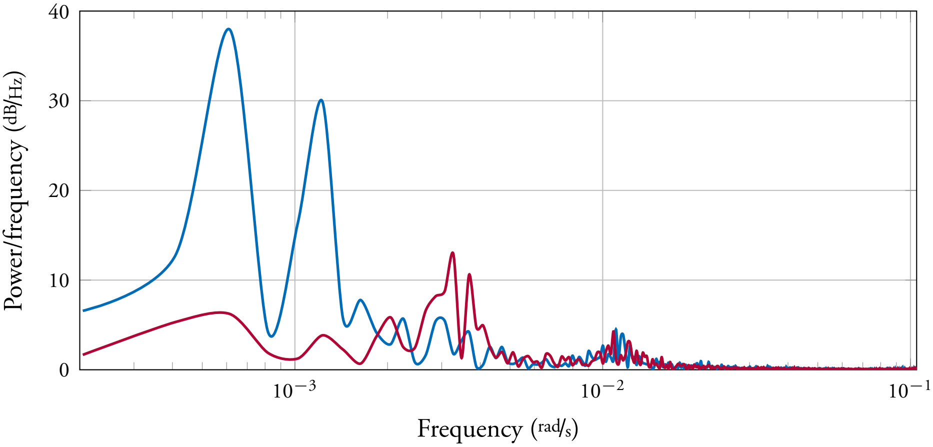


Figure 6.5 The one sided power spectrum estimate of the simulator CV 1 in open loop (——) and in closed loop (——) with the initial MPC. The controller rejects the two low frequency, periodic disturbances well. An oscillation around 0.003 rad/s seems to be induced by the controller.

First experiment

The first closed-loop identification experiment ran for 1 530 samples, which is approximately 12 hours of experiment time. The factor κ_t was increased in a quite aggressive way. The value is pushed until infeasibility and then reduced just enough to give a feasible problem. The value was then kept for a number of samples. Consequently, the level of required excitation in each sample is reduced. The procedure was repeated when the excitation level became too low.

Second experiment

The second experiment ran for 1 430 samples, which again correspond to 12 hours of experiment time. The factor κ_t was increased less aggressively compared to what was done in experiment 1. The value was changed such that a clear "disturbance" was seen in the signals, this value was kept until the disturbance was no longer visible and the procedure repeated.

Comparison

The step responses from the manipulated variables to CV 1 of the models resulting from the two MPC-X experiments are shown in Figure 6.4. MV 2 and MV 3 are the important signals for the control and the models for these channels are hence most important. This is reflected in the excitation produced by MPC-X which can be seen in Figure 6.6. MV 2 and 3 are excited a lot, while MV 1 remains mostly unexcited.

In both experiments, the excitation has large peaks which result in peaks also in the output. The reason for these peaks is not completely understood. The less aggressive tuning of κ_t in Experiment 2 may be the reason for the smaller peaks. However, there is a random element in the excitation which complicates the comparison.

From the step responses in Figure 6.4, it can be seen that the general dynamics are the same for all the identified models but time constants and gains differ. In particular, the gain from MV 2 to CV 1 is lower for both models obtained from closed-loop data than for the model obtained from open-loop data. Note that the channel from MV 1 is expected to not be very accurately modeled since it is considered less important for control and hence not excited by MPC-X. Furthermore, since MV 4 is a measured disturbance, it is not excited by MPC-X at all and the model quality in that channel can therefore not be influenced.

Residual analysis is also used to validate the obtained models, shown in further detail in Appendix 6.A. Notice that, because MV 3 is important for the chosen control setup, it is considered important to model the channel from MV 3 to CV 1. The correlation plots of the residuals in Figures 6.12–6.14 show that the models from the MPC-X experiments have captured this channel better than the model identified from open-loop data. In Figure 6.12 the cross-correlations between MV 3 and the residuals are outside the confidence bounds, indicating that there is information in the data that is not captured by the model from the open-loop experiments. In Figures 6.13 and 6.14, the cross-correlations indicate that the effects of MV 2 are well captured by the model from the MPC-X experiments.

Performance analysis

The two models obtained using the MPC-X data are tested for controlling the process using the same tuning as was used with the model from open-loop data. In Table 6.4 it is seen that the new models improve performance compared to when the degraded model is used. In Experiment 1, the variance $p(\theta)$ is reduced from 0.043 to 0.012 and in Experiment 2 it is reduced to 0.018. The resulting variances in both experiments are below their respective bounds (6.3) given by γ and $p(\theta_0)$. Looking at the application cost, it is reduced to 0.020 in Experiment 1 and 0.026 in Experiment 2, compared to 0.062 when the degraded model is used. In both cases, the cost is below the $1/\gamma$ bound. Analyzing the excitation shows that there is less information in the data from Experiment 2, which is reflected by the lower γ value in that experiment. The highest γ value such that the excitation constraint is still fulfilled is 32 in Experiment 1, and 16 in Experiment 2.

Conclusions

The two experiments were successful in generating excitation that leads to identified models that restore performance to acceptable levels. The major issue in setting up the experiments is the choice of the scaling κ_t. This scaling influences the aggressiveness of the excitation and the choice has consequences for the achievable performance when the identified model is used in the controller. The generated excitation, shown in Figure 6.6, also indicates that too large increases in κ_t may give very large excursions from the operating point.

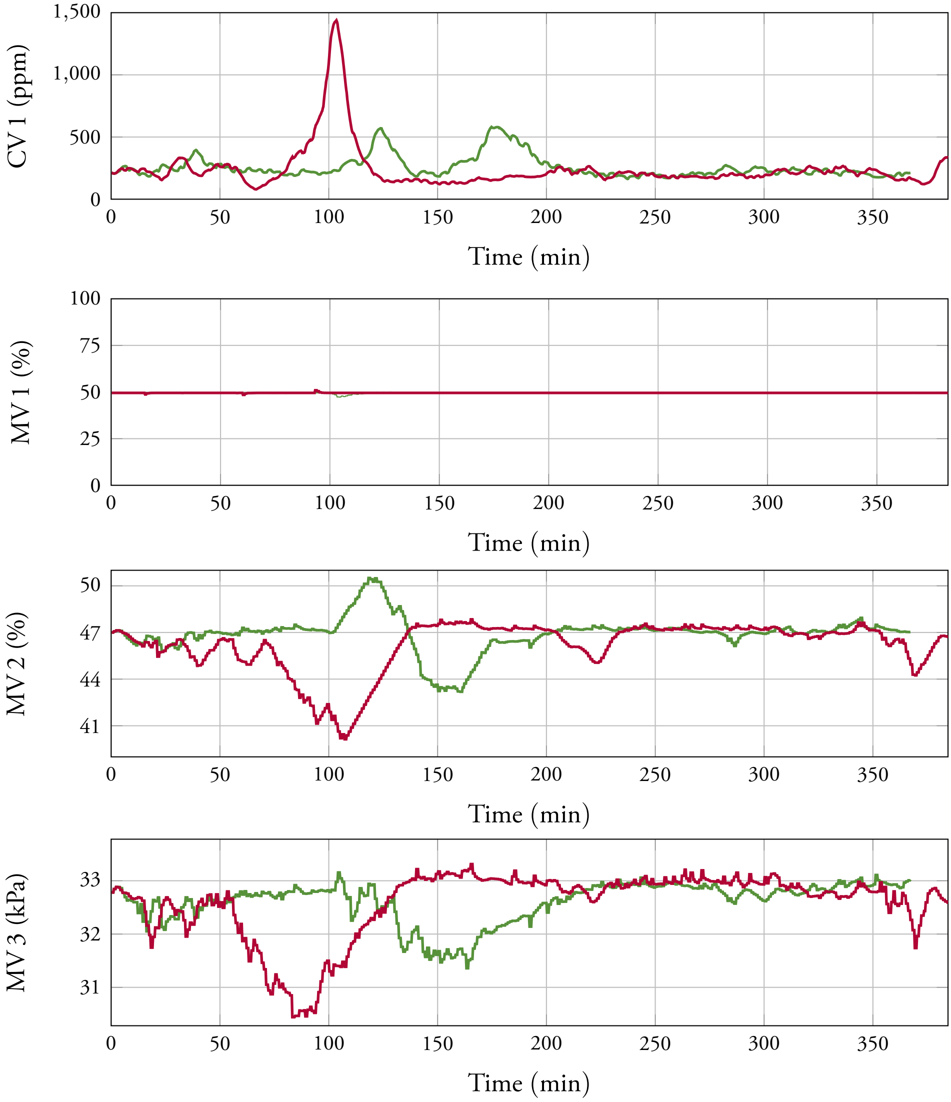


Figure 6.6 The excitation signals generated by MPC-X in experiment 1 (🟥) and experiment 2 (🟩) on the simulator. The two signals important for the control, MV 2 and 3, are excited while MV 1, which is typically not used by the controller, is not excited. The scale for MV 1 covers the full variable range in the MPC.

Table 6.4 The resulting performance using the different models for predictions in the MPC. The models obtained from the MPC-X experiments result in acceptable performance according to the defined specification bound.

Model	p_θ	V_{app}	$1/\gamma$	Bound (6.3)
Open-loop identification	0.0082	—	—	—
Open-loop identification, degraded	0.043	0.062	—	—
Closed-loop MPC-X, Experiment 1	0.012	0.020	0.031	0.11
Closed-loop MPC-X, Experiment 2	0.018	0.026	0.063	0.17

6.4 Plant experiments

Summary

One experiment with excitation from MPC-X and closed-loop reidentification was performed. The degraded controller model resulted in a significant drop in performance. After the MPC-X reidentification, the performance is improved. Changes in the operating conditions between the experiments — most significantly an increased disturbance level — complicates the comparison. The performance measure is modified to account for this. However, a clear connection to the application cost is lost and therefore the γ value is not used. The need for correct disturbance modeling is highlighted by the experiments. The details of the experiment are given in the following sections.

Effects not on simulator

The setup of the plant differs in some ways from the simulator, which gives some different considerations in the modeling and controller setup. Firstly, on the plant the C_4 in the side draw (CV 1) cannot be measured directly, instead a soft sensor is used. The bias of the sensor is corrected by an analyzer measurement every 20 minutes. This soft sensor can give negative values even though this is not physically possible, therefore the logarithm of CV 1 was not used in the model of the plant.

Secondly, C_4 content in the feed to the depropanizer varies which affects the distillation. This does not happen on the simulator where the C_4 content in the feed is constant. On the plant, measurements of the feed compositions are available and can be included as a measured disturbance in the model and used for feed forward. This could improve the control performance if included in the MPC.

Thirdly, the plant has noise and disturbances, which are not implemented on the

Table 6.5 Operating point and corresponding steady state values for the plant experiments.

	Variable	Value	Unit
MV 2	Side draw to feed ratio	56.9	%
MV 3	Column differential pressure	34.0	kPa
CV 1	C_4 in side draw	250	ppm
CV 2	Feed drum level	70	%
CV 4	Column pressure	1 219	kPa
CV 5	Bottom temperature	62.8	°C
CV 6	Feed flow OP[b]	95	%
CV 8	Reboil	12.05	×[c]
CV 9	Reboil flow OP[b]	75.3	%
CV 10	Reflux flow OP[b]	73.6	%
CV 11	Feed to C_3 header OP[b]	13.8	%

[b]Valve opening percentage, [c]Not shown for confidentiality reasons

simulator. One such disturbance is the feed variation mentioned earlier. Other sources are, for example, down and up stream changes in demand or production, impact of outside the temperature and weather, and measurement noises in all sensors, to name a few. Therefore, a model which includes a noise model was used in the plant experiments.

Finally, the experiments are performed during normal operations of the plant and care has to be taken to make sure that the added excitation does not disturb production in downstream units which rely on the output of the depropanizer. Such considerations could be included in the constraints of the MPC during the identification experiment. However, only the constraints from regular operation are included in this case.

Operating point

The values of the manipulated variables and the resulting values of the controlled variables at the operating point at the beginning of the plant experiments are presented in Table 6.5.

Initial model

The model of the plant is

$$\begin{cases} x_{t+1} = A(\theta)x_t + B(\theta)u_t + B_d(\theta)d_t + Ke_t, \\ \quad y_t = C(\theta)x_t + e_t, \end{cases}$$

where y_t is the output, u_t the input, d_t the measured disturbance and e_t unmeasured disturbances. The benefit of including the feed variation as a measured disturbance is investigated by using two different models, one with MV 4 as the only measured disturbance and one model with two measured disturbances, MV 4 and C_4 in feed (henceforth MV 5).

For the open-loop identification experiments, low-pass filtered, random, binary excitation is added to the manipulated variables. A Gaussian white noise signal is filtered through an eighth-order, non-causal Butterworth filter and the sign of the resulting signal, appropriately scaled, is used. The amplitudes are ±2000 for MV 1, ±2.25 for MV 2 and ±1.25 for MV 3. No excitation is added to the two signals considered to be measured disturbances. The values of the parameter vector are obtained using PEM in the System identification toolbox of Matlab. The resulting fit is shown in Figure 6.7 and the step responses from the manipulated variables to CV 1 are shown in Figure 6.8. The benefit of including the second measured disturbance is not visible in these data. However, residual analysis in Appendix 6.B shows that including the extra disturbance whitens the residuals of the output. The figures also include the currently used model for reference. This model follows the slower trends well but does not capture the faster dynamics in the process.

Control setup

Based on the initial model, the sampling time of the controller is chosen to be 0.5 Hz. The prediction and control horizons are chosen as $N_y = 60$ and $N_u = 10$. The controller is tuned to give good disturbance rejection, which is a balance between bandwidth and robustness to modeling errors.

MPC-X reidentification

The application cost (6.2) is used also for the performance specifications on the actual depropanizer. The level κ_t is again manually tuned such that the optimization problem is feasible. The strategy for updating the scaling is similar to the one used in Experiment 2 on the simulator. The excitation is increased so that it is visible but intentionally kept small not to upset down stream production units. The signals during the experiment are shown in Figure 6.9. Note that, even though the model structure includes a K matrix and therefore is not of output error type, the experiment design part of MPC-X uses an output error model where K is removed. Implications of this are further discussed in the conclusions.

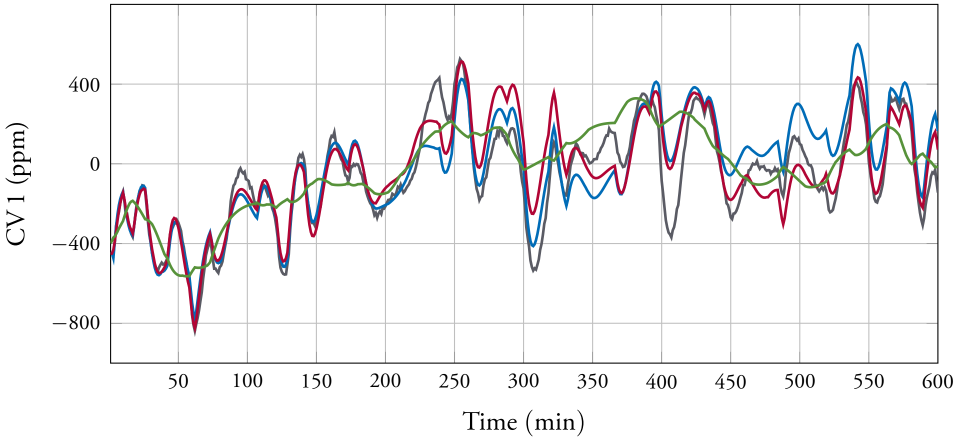


Figure 6.7 Measured data and simulated responses. The plant output (——) and data from simulation of the model with feed forward (——), the model without feed forward (——) and the model used by the current MPC (——).

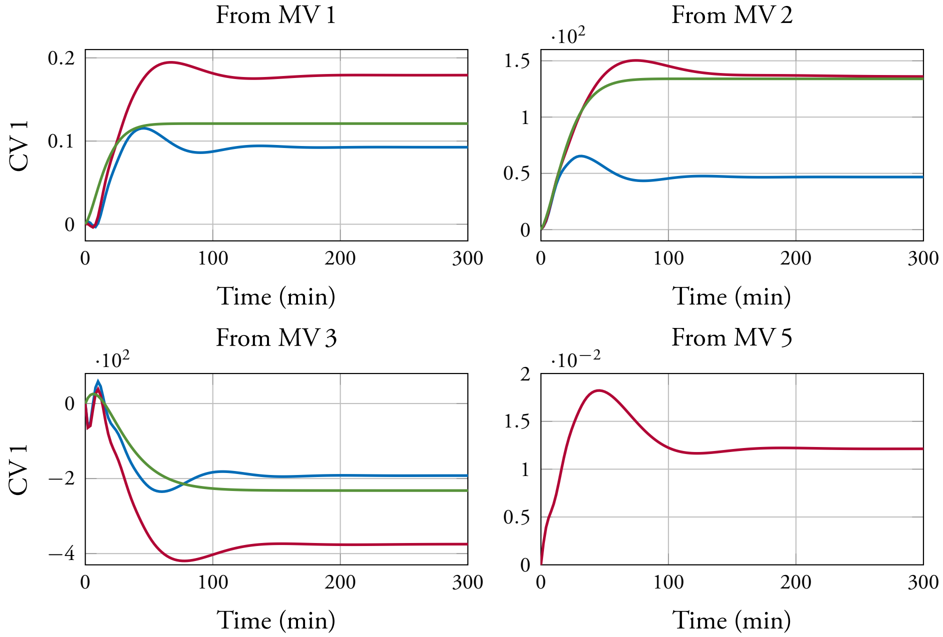


Figure 6.8 The step responses to CV 1 for the model without feed forward (——), model with feed forward (——) and the currently used model (——). Note that the model with feed forward is the only model which includes MV 5.

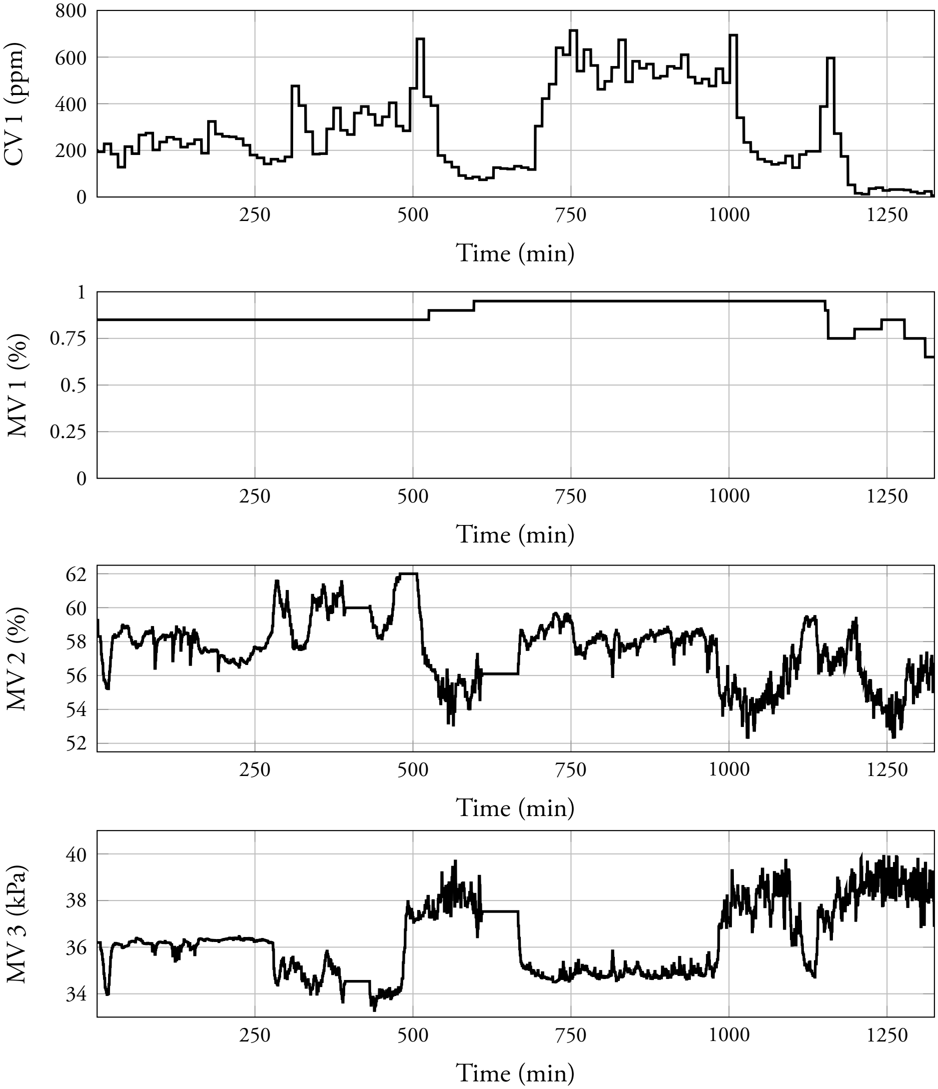


Figure 6.9 The excitation signals generated by MPC-X in the experiment closed-loop reidentification experiment of the plant. The two signals important for the control, MV 2 and 3, are excited while MV 1, which is typically not used by the controller, is not excited. The scale for MV 1 covers the full variable range in the MPC and the changes in the signal value are due to operator manipulations and not to MPC-X.

Step responses

The step responses from the manipulated variables to CV 1 of the models resulting from the MPC-X experiment are shown in Figure 6.10. MV2 and MV3 are the important signals for the control and the models for these channels are hence most important. The obtained model differs significantly from the model obtained from the open-loop experiments, both in terms of gains and response times of the model. At this point it is difficult to find the reasons for these results.

Feed forward effects

In Figure 6.10, the step response from a model obtained from MPC-X data generated without the feed forward from MV 5 is also shown. Due to the correlation between MV 5 and the inputs generated by MPC-X the resulting gains for MV 2 and MV 3 have the wrong sign. This issue is resolved when the feed forward is included in the model. This shows that including the effects of disturbances is not only beneficial for the control, it is also of importance for correct closed-loop modeling, as expected.

Residual analysis

The residual analysis in Appendix 6.B shows that the model obtained from MPC-X closed-loop reidentification has picked up the dynamics from all manipulated variables to CV 1 as all residuals in Figure 6.17 are within the confidence bounds. The residuals of the output are not completely white but show similar behavior as for the model from open-loop experiments.

Cross validation

To further validate the new model, it is used in simulation with MVs from open-loop operation as inputs. The resulting output is compared to the results using the model from open-loop identification. This is presented in Figure 6.11. The fit to data is 19 % for the MPC-X model and 32 % for the open-loop identification. The models are also tested on closed-loop data, as shown in Figure 6.11 where it is seen that the model from MPC-X fits closed-loop data much better than the open-loop model. The fit to data is 42 % for the MPC-X model and -200 % for the open-loop identification. The achieved fit to data is poor for all the models, which indicates that there is still a high degree of errors in the models. Note that more data than what is shown in the figures is used to calculate the numerical values of the fits.

Performance analysis

The model obtained from the MPC-X experiments is used in the MPC for controlling the depropanizer. The resulting performance is presented in Table 6.6. The main disturbance affecting the distillation is the C_4 level in the feed, MV 5, and a more fair comparison is made by normalizing the variance of CV 1, $p(\theta)$, by the variance of MV 5, which is shown in the last column of Table 6.6. The results show that including the extra measured disturbance has a positive effect on the control performance. The normalized variance reduces from 6.3×10^{-5} to 3.1×10^{-5} by using feed forward from MV 5.

Using the degraded model for control in MPC-X increases the normalized variance to 27×10^{-5}, almost a 9 fold increase. When the reidentified model from the closed-loop MPC-X experiment was implemented, the normalized variance is reduced to 9.7×10^{-5}. Although it is not possible to relate the normalized variance to the application cost and find a suitable upper bound, this is considered a significant performance improvement.

Fit to data and control performance

A curious result is that the model obtained from the MPC-X experiments results in worse closed-loop performance than when the model from open-loop identification is used. The normalized variance when using the model from MPC is 9.7×10^{-5} compared to 3.1×10^{-5} when the model from open-loop identification is used. This is unexpected since Figure 6.11 shows that the model from MPC-X has relatively good fit to both open- and closed-loop data while the model obtain from open-loop identification has a very poor fit to closed-loop data. Furthermore, the residual analysis of the models, see Appendix 6.B, indicates that the MPC-X model has captured more of the information in the data than the models from open-loop experiments. It is important to note that this does not indicate that there is a problem with the MPC-X formulation but rather that there seems to be some interplay between model and controller that is not completely understood at this moment. This should be investigated further.

Table 6.6 The variances of CV 1 and the main disturbance signal MV 5 for different cases of models used in the MPC controlling the depropanizer. Feed forward means that the model uses the feed composition signal, MV 5, as a measured disturbance

	Variance		
Model	CV 1	MV 5	CV 1 / MV 5
Open-loop	4.54×10^3	7.2×10^7	6.3×10^{-5}
Open-loop, feed forward	7.3×10^3	23×10^7	3.1×10^{-5}
Open-loop, feed forward degraded	95×10^3	34×10^7	27×10^{-5}
MPC-X, feed forward	36×10^3	37×10^7	9.7×10^{-5}

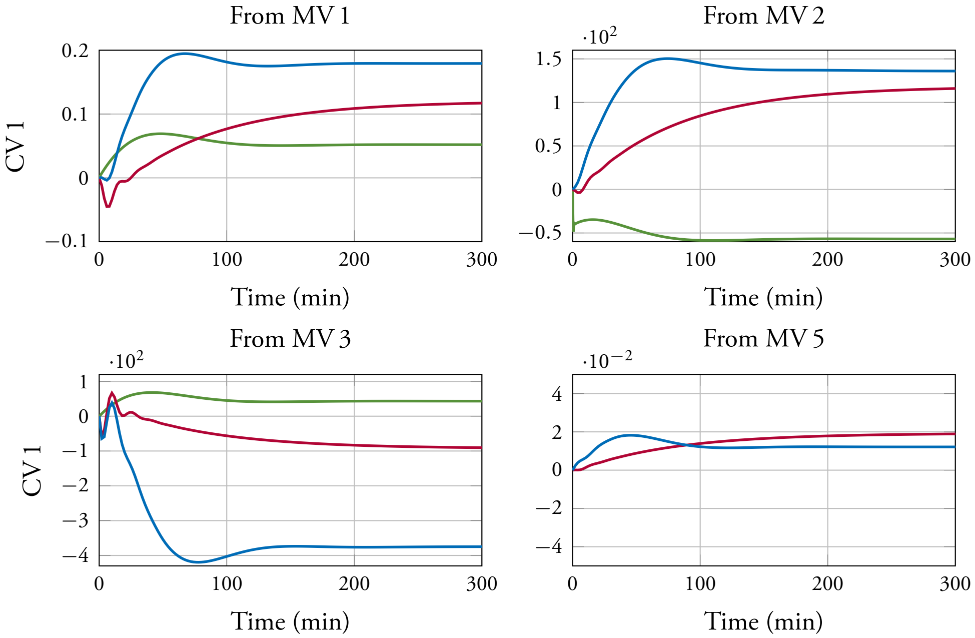
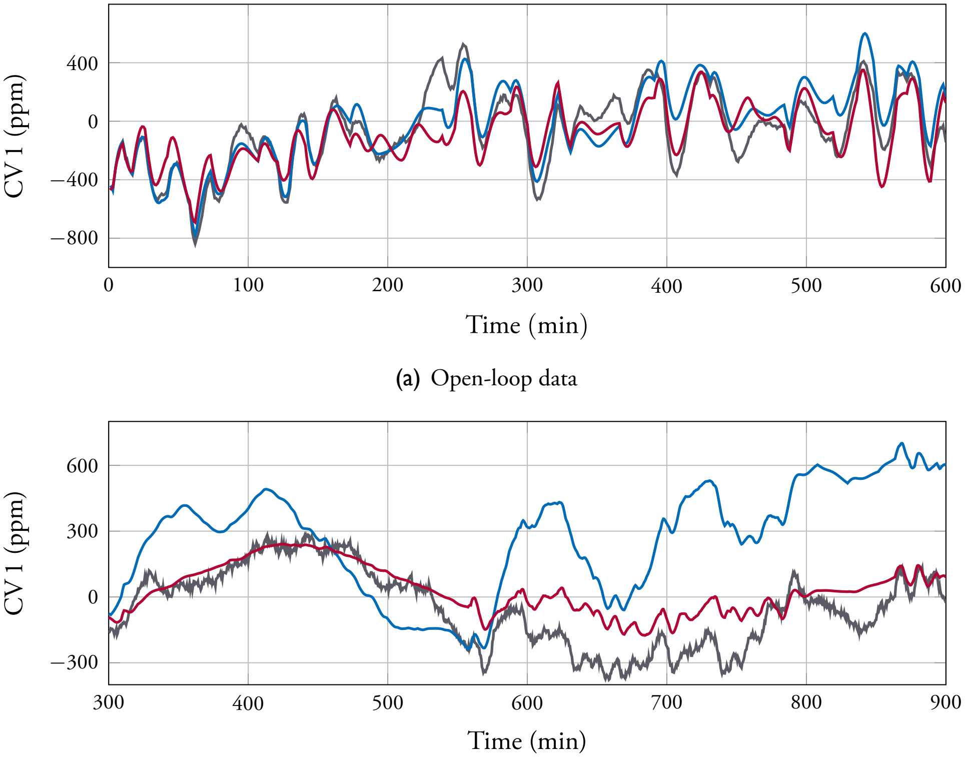


Figure 6.10 The step responses to CV1 for the model with feed forward (——) and the model obtained in the MPC-X experiment (——). One model obtained from MPC-X data without feed forward from MV 5 (——) is also shown, note that the gains from MV 2 and MV 3 have changed signs.



(a) Open-loop data

(b) Closed-loop data

Figure 6.11 The plant output (——) and data from simulations using the model with feed forward (——) and the model obtained in the closed-loop MPC-X experiment (——).

6.5 Summary

The MPC-X algorithm of Chapter 5 has been implemented and tested on an industrial process during regular production. An extensive experimental campaign evaluated the algorithm, both on a Honeywell UniSim simulator and on a depropanizer distillation column. The tests included initial MPC commissioning using open-loop identification and tuning of the controller. Initial identification was performed using binary random signals and the collected data were used to identify state space models of the simulator and plant. The identified models were used to construct MPC controllers, which were tuned for robust performance rejection. The initial models were changed to give a plant–model mismatch and MPC-X reidentification was used to restore performance.

The results show that MPC-X is able to generate an input signal that gives excitation to the plant while satisfying the signal constraints of the process. The models obtained from the closed-loop reidentification improve performance in all cases. On the simulator the performance after the model was updated satisfied the theoretical bound in both experiments. However, some issues regarding the implementation and use of the method were discovered. The experiments on the actual depropanizer emphasize the importance of correct noise modeling in closed-loop identification.

6.A Residual analysis of simulator models

The residuals are investigated to validate the models obtained from open-loop data and the closed-loop, MPC-X generated, data. The residuals are calculated using separate sets of validation data. The correlations for the model from open-loop data are plotted in Figure 6.12, and for the models from MPC-X data in Figures 6.13 and 6.14. The residuals are not white for any of the models, which can be explained by the fact that the output error model used cannot capture colored noise. It is also seen that there is still information about the dynamics from the MVs that has not been completely captured by the model. This is, in particular, true for MV 3 in the open-loop experiments. In the closed-loop experiments, however, MV 3 is much better modeled. This indicates that since this MV is important for the control, MPC-X has excited that channel appropriately.

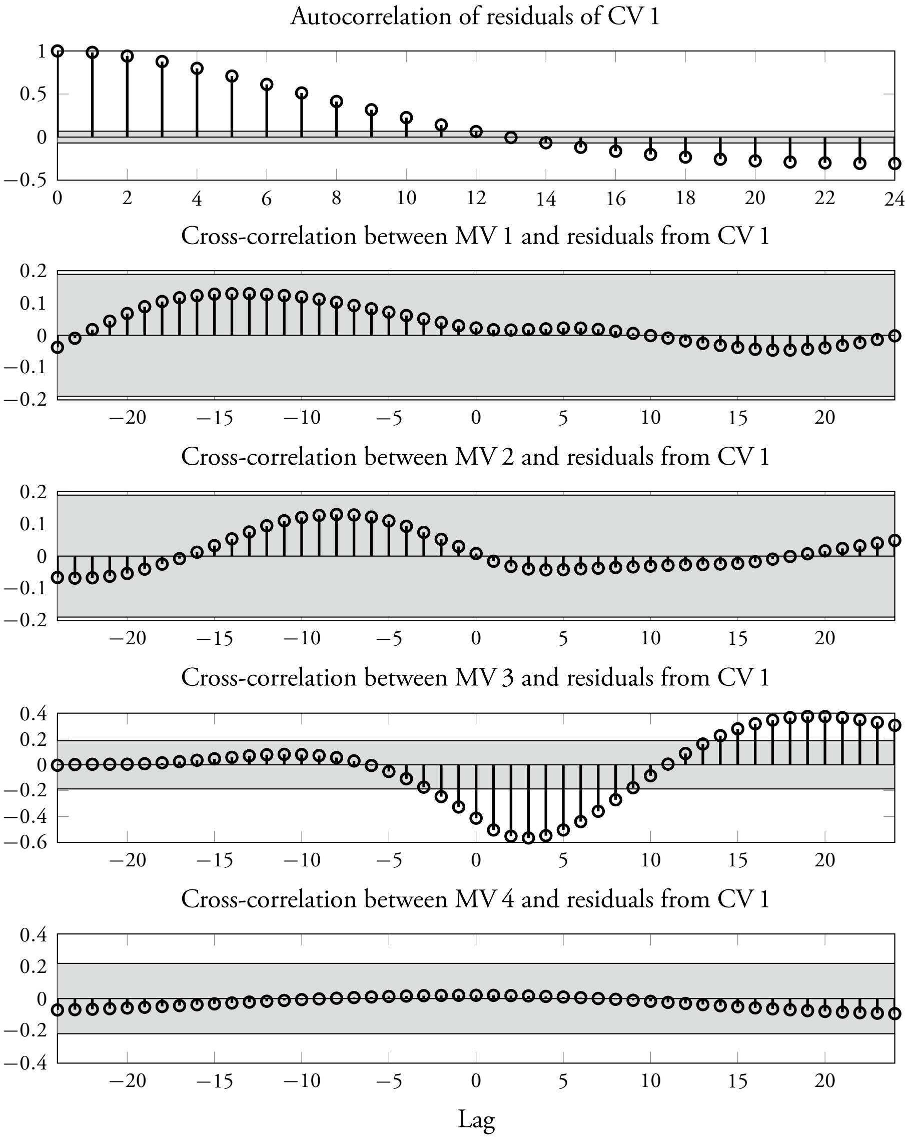


Figure 6.12 The autocorrelation of the residuals and the cross-correlation between manipulated variables and the residuals for the model obtained from open-loop experiments on the simulator. The residuals are not white, which can be explained by the fact that an output error model is used. The cross-correlations indicate that there are still effects from manipulated variables that are not captured by the model.

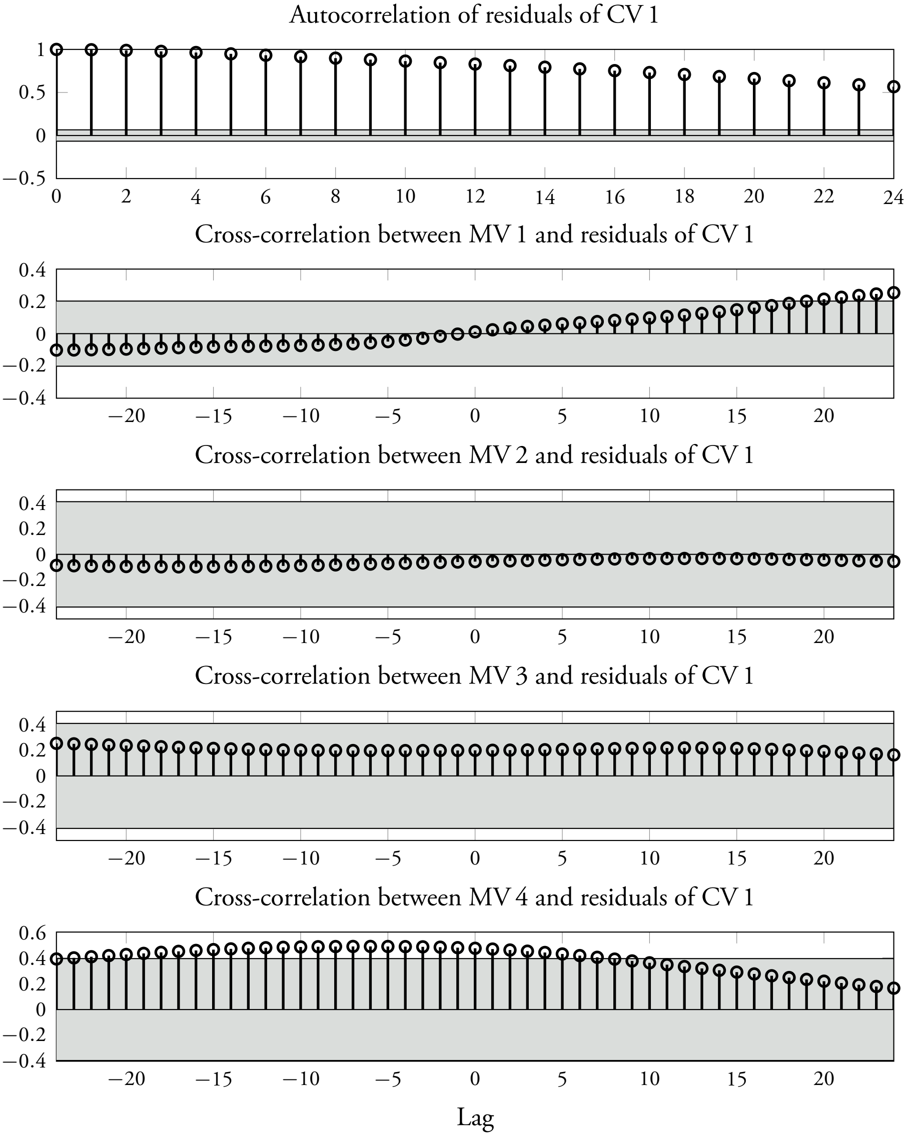


Figure 6.13 The autocorrelation of the residuals and the cross-correlation between manipulated variables and the residuals for the model obtained from MPC-X experiment 1. The residuals are not white, which can be explained by the fact that an output error model is used. The cross-correlations indicate that the effect of MV 3 is better captured than when using open-loop data.

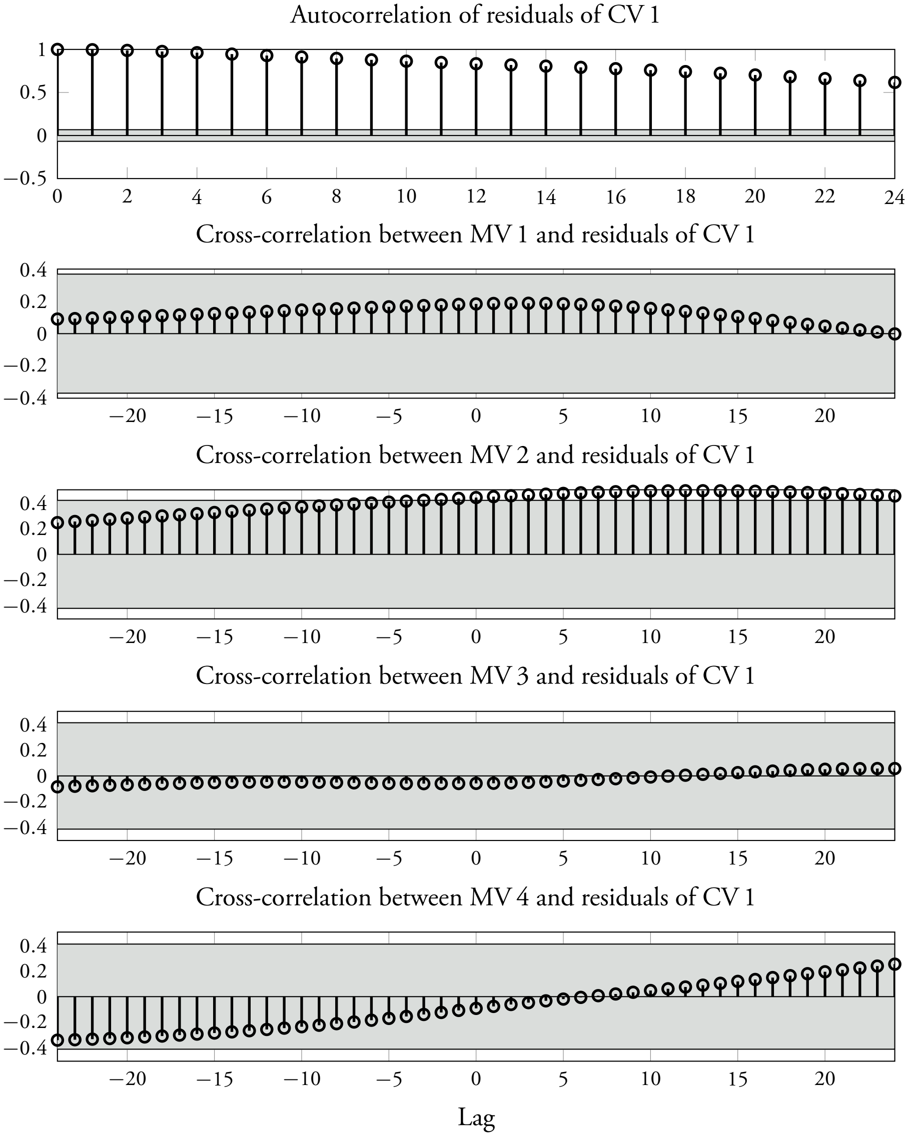
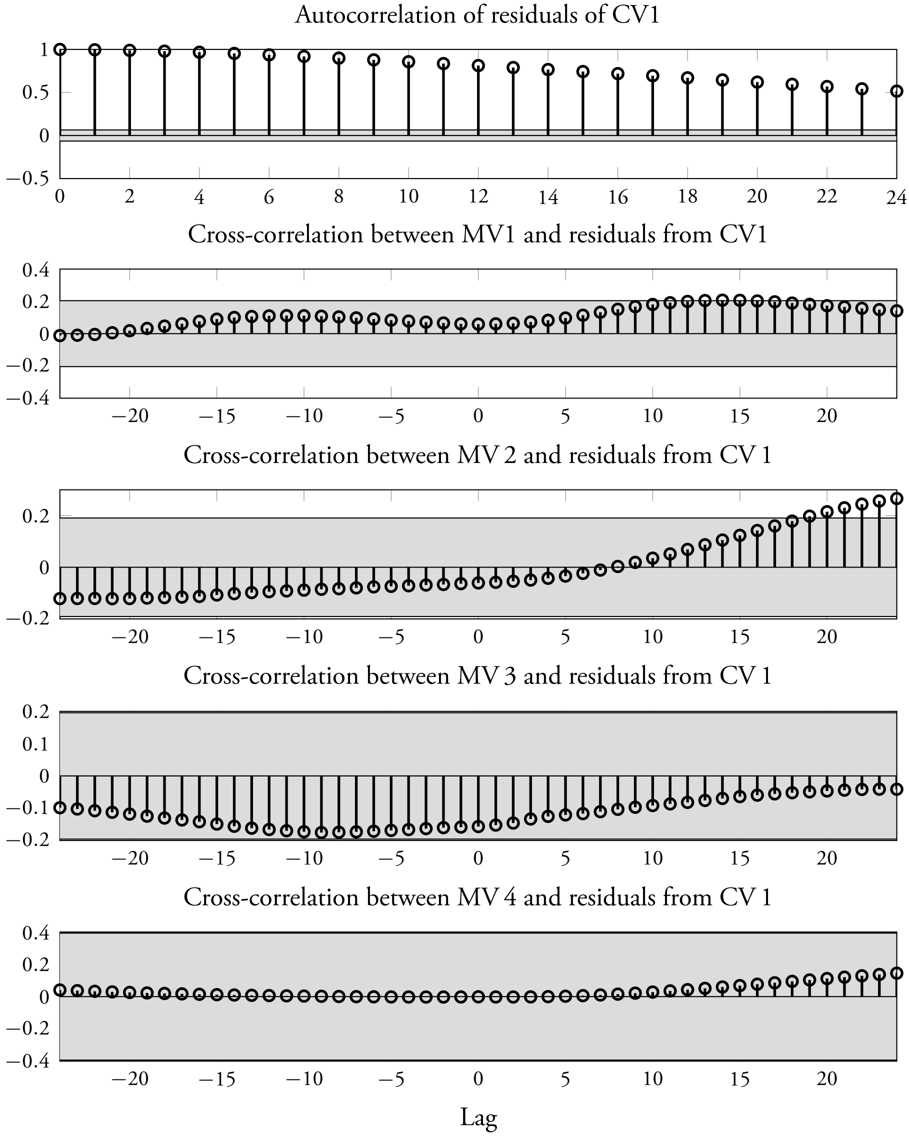


Figure 6.14 The autocorrelation of the residuals and the cross-correlation between manipulated variables and the residuals for the model obtained from MPC-X experiment 2. The residuals are not white, which can be explained by the fact that an output error model is used. The cross-correlations indicate that the effect of MV 3 is better captured than when using open-loop data.

6.B Residual analysis of plant models

The residuals are investigated to validate the models obtained from open-loop data and the closed-loop, MPC-X generated data. The residuals are calculated using separate sets of validation data. The correlations for the open-loop model without feed forward are plotted in Figure 6.15, for the open-loop model with feed forward in Figure 6.16, and for the MPC-X model in Figures 6.17. The residuals are close to being white, which shows that the inclusion of the process noise models helps in the identification. The cross-correlations are reasonable, in particular for the MPC-X data in Figure 6.17, where they indicate that most of the information in the data is explained by the model.


Autocorrelation of residuals of CV1

Cross-correlation between MV1 and residuals from CV1

Cross-correlation between MV 2 and residuals from CV 1

Cross-correlation between MV 3 and residuals from CV 1

Cross-correlation between MV 4 and residuals from CV 1

Figure 6.15 The autocorrelation of the residuals and the cross-correlation between manipulated variables and the residuals for the model obtained from open-loop experiments on the plant. The residuals are not white, which can be explained by the fact that an output error model is used. The cross-correlations indicate that there are still effects from manipulated variables that are not captured by the model.

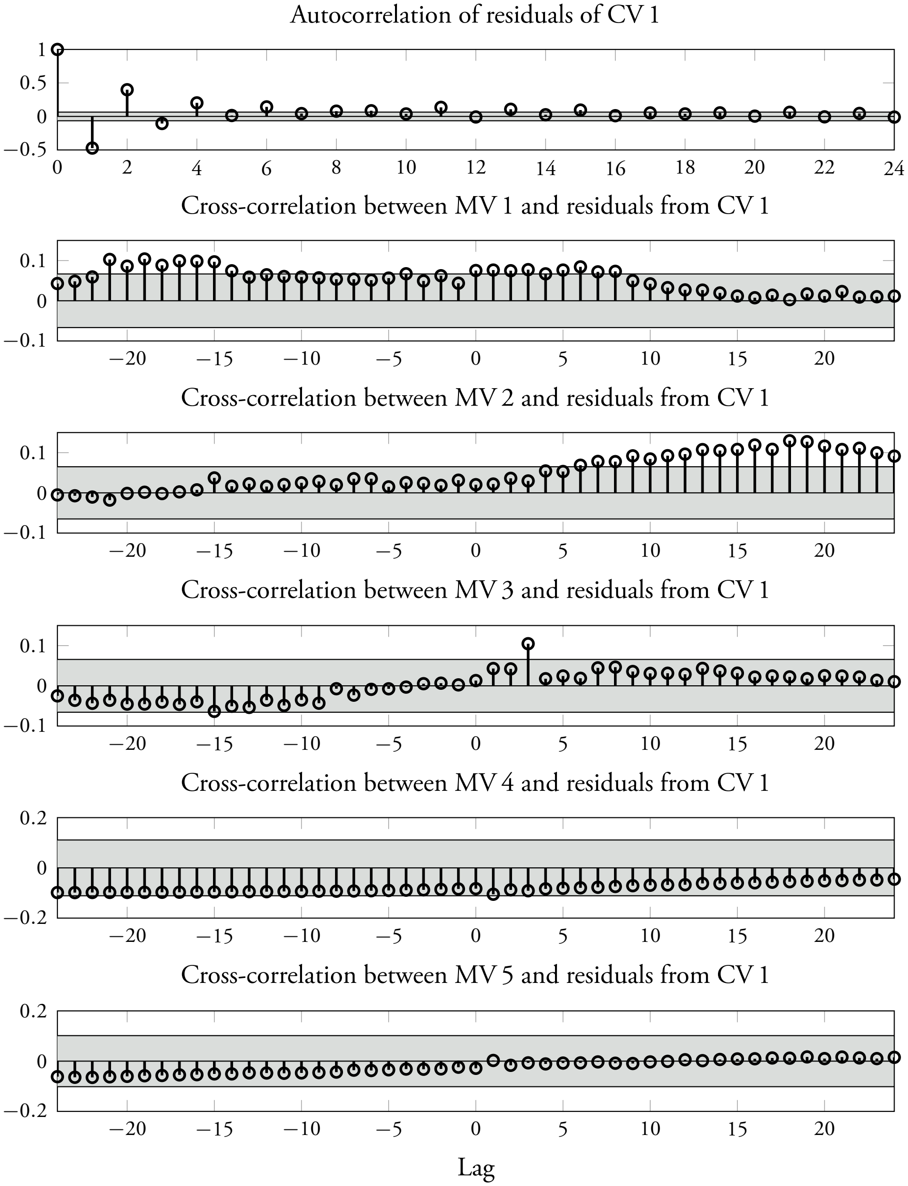


Figure 6.16 The autocorrelation of the residuals and the cross-correlation between manipulated variables and the residuals for the model obtained from open-loop identification with five inputs. The residuals are close to white because a disturbance model has been added. The cross-correlations indicate that there are still effects from manipulated variables that are not captured by the model.

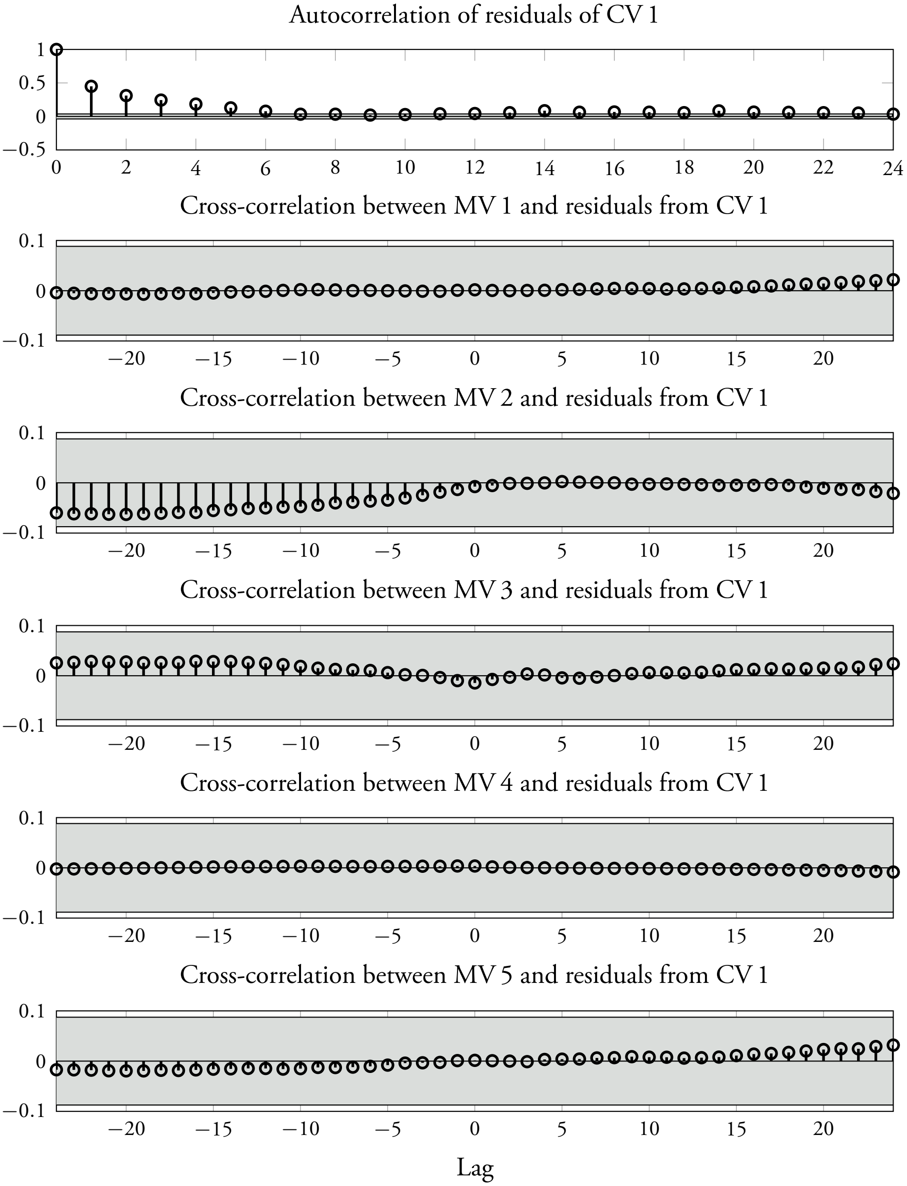


Figure 6.17 The autocorrelation of the residuals and the cross-correlation between manipulated variables and the residuals for the model obtain from closed-loop MPC-X experiments on the plant. The residuals are almost white and the model seems to capture the information available in the data.

Chapter 7
Signal generation for constrained systems

Generation of signals with specific autocorrelation properties is a problem which arises in many fields. For example, in system identification the choice of input signal can greatly impact the statistical properties of the estimates and the required experimental effort. Another example is radar applications where completely white signals are desirable, see He et al. (2012) for a discussion on this and further examples where white signals are desirable.

A complicating issue when generating signals in practice is that many systems have restrictions on signal amplitudes or rates of change. If these are not taken into account in the signal design the resulting signal may not have the right autocorrelation properties. This happens, for instance, if a Gaussian signal is filtered to have the right correlation and then "clipped" to get the right amplitude, shown by Hannan (1970). A more sophisticated method is to limit the design to binary signals where it is easy to enforce constraints on the amplitudes, for example, Liu and Munson (1982), Boufounos (2007), and Rojas et al. (2007) have proposed algorithms for this. Schoukens et al. (1991) and Pintelon and Schoukens (2001) instead propose using sums of sinusoids and then controlling the signal amplitude by adjusting the phases of the components.

In this chapter, an algorithm for designing a signal directly in the time domain is presented. The idea is to try to match the autocorrelation of the signal to a desired autocorrelation while guaranteeing constraint satisfaction. The problem is formulated as an optimal control problem and approximated using the receding horizon idea.

7.1 Problem formulation

The system and model assumptions are presented and the considered signal constraints are discussed. The constrained signal generation problem is also formalized.

System and model

Consider a discrete-time, dynamic, LTI system modeled by

$$\mathcal{M}(\theta) : \quad \begin{cases} x_{t+1} = A(\theta)x_t + B(\theta)u_t + v_t, \\ \quad z_t = C(\theta)x_t \\ \quad y_t = z_t + e_t, \end{cases} \tag{7.1}$$

where $x_t \in \mathbf{R}^n$, $u_t \in \mathbf{R}^m$, z_t,$y_t \in \mathbf{R}^p$. z_t is the system output while y_t is the output measured with noise. The noises v_t and e_t are zero mean, independent, identically distributed sequences. In general, no assumption on the noise distribution is needed, all that is required is that it is possible to draw samples of the noise, either from a random number generator or real data. The vector $\theta \in \mathbf{R}^{n_\theta}$ represents the parameters of the system, which may be uncertain. For example, the parameters used in the model could be estimated by system identification, which results in estimates that are random variables. Another example is that it is known, for instance from physical insight, that one parameter of the true system has a value between 0.5 and 0.7. In both cases, the uncertainties in the system can be represented by some probability distribution of the parameters. In the former case, the distribution for PEM estimates in (2.9) may be used and, in the latter case, a uniform distribution over $[0.5 \ 0.7]$ is suitable. It is assumed that, regardless of how the model parameters are obtained, it is possible to draw samples from their distribution.

Signal constraints

The input and output signals of the system (7.1) are subject to constraints such that

$$u_t \in \mathcal{U}, \tag{7.2a}$$

$$z_t \in \mathcal{Z}. \tag{7.2b}$$

The sets $\mathcal{U}$ and $\mathcal{Z}$ are prespecified convex sets representing the input and output constraints. Such constraints include, for example, signal amplitude constraints which are often encountered in practice.

In many cases, guaranteed satisfaction of the output constraint for every time instant is very conservative. In fact, in the case of noise with unbounded support, as for the common Gaussian noise case, it is impossible to guarantee hard output constraints; any hard output or state constraints will be violated almost surely. Therefore, a probabilistic constraint on the signal z_t is more suitable. For example, the constraint

$$\mathbb{P}\{z_t \notin \mathcal{Z}\} \leq \varepsilon, \tag{7.3}$$

which guarantees that the violation probability is small if ε is small. A complicating factor is that the chance constraint (7.3) is typically non-convex, even if the set $\mathcal{Z}$ is convex.

Signal generation

The goal for the signal generation algorithm is to generate N samples of the input vector

$$u_t = \begin{bmatrix} u_t^{(1)} & u_t^{(2)} & \cdots & u_t^{(m)} \end{bmatrix}^T,$$

such that the resulting signal has sample autocorrelation matching the first n_τ lags of the prescribed autocorrelation

$$R^d(\tau) = \mathbb{E}\left\{u_t u_{t-\tau}^T\right\}, \quad \tau = 0, 1, \ldots, n_\tau.$$

The motivation for matching a fixed number of autocorrelation lags n_τ is that, in many optimal input design problems, only a finite number of correlation lags affect the result, see, for example, Bombois et al. (2006) or Lindqvist and Hjalmarsson (2001).

Defining the (biased) sample autocorrelation at time t for time lag τ as

$$\begin{aligned} R_t(\tau) &= \frac{1}{t} \sum_{i=\tau+1}^{t} u_i u_{i-\tau}^T & (7.4) \\ &= \frac{t-1}{t} R_{t-1}(\tau) + \frac{1}{t} u_t u_{t-\tau}^T, & (7.5) \end{aligned}$$

the constrained signal generation can be formulated as the optimization problem

$$\begin{aligned} \underset{\{u_k\}_{k=1}^N}{\text{minimize}} \quad & \sum_{\tau=0}^{n_\tau} \|R_N(\tau) - R^d(\tau)\|_F^2 w(\tau), \\ \text{subject to} \quad & x_{t+1} = A(\theta)x_t + B(\theta)u_t + v_t, \\ & z_t = C(\theta)x_t, \\ & u_t \in \mathcal{U}, \\ & \mathbb{P}\{z_t \notin \mathcal{Z}\} \leq \varepsilon, \\ & t = 0, \ldots, N-1, \end{aligned} \tag{7.6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $w(\tau)$ are positive weighting factors that can be used to add flexibility in the algorithm. The idea of adding a weighting factor to the fit of the autocorrelations has also been suggested by Agüero et al. (2007). For simplicity, $w(\tau) = 1$ for all τ in the following treatment.

The signal generation can be seen as a constrained, stochastic optimal control problem. This problem, even though in principle solvable, is in practice a very complicated optimization problem, as is the case with many optimal control problems. One possibility is to consider the open loop control problem and search over a sequence of inputs. However, due to the uncertainties and the constraints, this will most likely result in a very conservative solution. Furthermore, the resulting cost function is a quartic polynomial in the decision variables $\{u_t\}_{t=0}^{N-1}$ and the chance constraint is in general non convex even if $\mathcal{Z}$ is convex.

For systems with only input constraints, a possibility is to use the result for the autocorrelation function of a "clipped" Gaussian process found in Hannan (1970). The result is presented here in the following theorem.

Theorem 7.1

Let $\{x_t\}$ be a strictly stationary Gaussian process with $\mathbb{E}\{x_t\} = 0$, $\mathbb{E}\left\{x_t^2\right\} = 1$ and autocorrelation $\mathbb{E}\{x_{t+\tau}x_n\}$. Then the autocorrelation of the "clipped" process $u_t = \operatorname{sgn} x_t$ is given by

$$\mathbb{E}\{u_{t+\tau}u_t\} = \frac{2}{\pi}\arcsin \mathbb{E}\{x_{t+\tau}x_t\}.$$

It is however known that a binary signal cannot have arbitrary covariance sequence, see, for example de Carvalho and Clark (1983), and hence the obtained signal spectrum can be suboptimal. Another approach is a receding horizon implementation of the problem (7.6). The receding horizon strategy has proven to be very successful for many other problems.

7.2 Signal generation algorithm

A computationally viable formulation for the signal generation problem (7.6), based on the algorithm by Rojas et al. (2007), is developed in the following sections. In this development, the model parameters θ can be uncertain but are not updated online in the algorithm. A fully adaptive formulation, where the model parameters and their probability distribution are re-estimated as more and more data are available, is developed in Section 7.3.

Receding horizon formulation

To simplify the optimization, a receding horizon approximation of the original problem (7.6) is formulated. The new problem is inspired by recent developments in stochastic model predictive control. The simplification comes from considering an open loop optimization problem over a shorter horizon which reduces the optimization to a significantly shorter sequence of inputs. Robustness and reduced conservativeness come from feedback of the current state, measured or estimated, in each time instant. To formalize, at sample t, the following problem is solved

$$\begin{aligned} \underset{\{u_k\}_{k=1}^{N_u}}{\text{minimize}} \quad & \sum_{\tau=0}^{n_\tau} \|R_{t+N_u-1}(\tau) - R^d(\tau)\|_F^2, \\ \text{subject to} \quad & x_{k+1} = A(\theta)x_k + B(\theta)u_k + v_k, \\ & z_k = C(\theta)x_k, \\ & x_1 = x_t, \\ & u_k \in \mathcal{U}, \\ & \mathbb{P}\{z_k \notin \mathcal{Z}\} \leq \varepsilon, \quad k = 1, \dots, N_y \end{aligned} \tag{7.7}$$

Here, N_u is the input horizon and N_y is the output horizon, that is, how far in the future we consider that the output should satisfy the constraints. The optimization is performed over the whole input horizon but only the first sample, $u_1^\star$, is implemented, that is $u_t = u_1^\star$, and the optimization is performed iteratively, in a receding horizon fashion. If the output horizon is longer than the input horizon, that is $N_y > N_u$, the input is set to $u_{k+N_u+1} = \cdots = u_{k+N_y} = 0$. The problem (7.7) can be seen as an MPC where the goal is to follow a reference for the autocorrelation while satisfying input and output constraints. When the state is not directly measurable, an estimate $\hat{x}_t$ is used instead.

Larsson et al. (2013b) proved convergence of the algorithm (7.7) for generation of white noise in the case of one input under certain conditions, which is summarized in the following theorem.

Theorem 7.1

For a sequence $\{u_i\}_{i=1}^t$ *generated by Algorithm* (7.7) *with,* $N_u = 1$, $m = 1$,

$$\begin{aligned} & r^d = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \end{bmatrix}^T, \\ & \mathcal{U} = \left\{ u_t : -u_{max} \leq u_t \leq u_{max}, \quad u_{max} \geq \sigma, \right\}, \\ & \mathcal{Z} = \mathbf{R}^p, \end{aligned}$$

that is, for generation of amplitude-constrained, pseudo-random white noise with variance σ^2, it holds that

$$\lim_{t\to\infty} \frac{1}{t} \sum_{i=\tau}^{t} u_i u_{i-\tau} = r^d(\tau), \quad \tau = 0, \dots, n.$$

Proof. The proof comes from Rojas et al. (2007). Introduce

$$\tilde{\mathbf{r}}_t = \begin{bmatrix} \tilde{r}_t(0) & \cdots & \tilde{r}_t(n) \end{bmatrix}^T,$$

where

$$\tilde{r}_t(\tau) = \sum_{i=\tau+1}^{t} (u_i u_{i-\tau} - r^d(\tau)), \quad 0 \le \tau \le \min(t-1, n_\tau),$$

are scalar because a single input is considered. Then

$$\tilde{r}_{t+1}(\tau) = \tilde{r}_t(\tau) + u_t u_{t-\tau} - r^d(\tau),$$

allowing the dynamics of $\tilde{\mathbf{r}}_t$ to be written as

$$\tilde{\mathbf{r}}_{t+1} = \tilde{\mathbf{r}}_t + u_t \mathbf{u}_t - \mathbf{r}^d,$$

where $\mathbf{u}_t = \begin{bmatrix} u_t & u_{t-1} & \dots & u_{t-n_\tau} \end{bmatrix}^T$, with initial condition $\tilde{\mathbf{r}}_0 = 0_{n_\tau+1}$. Note that $u_t = \arg\min_{u_t} \|\tilde{\mathbf{r}}_{t+1}\|_2^2$ and hence

$$\begin{aligned}
&\|\tilde{\mathbf{r}}_{t+1}\|_2^2 = \min_{u_t} \|\tilde{\mathbf{r}}_{t+1}\|_2^2 \\
&= \min_{u_t} \left\{ (\tilde{r}_t(0) + u_t^2 - \sigma^2)^2 + \sum_{\tau=1}^{n_\tau} (\tilde{r}_t(\tau) + u_t u_{t-\tau})^2 \right\} \\
&\le \min_{u_t \in \{-\sigma, \sigma\}} \left\{ (\tilde{r}_t(0) + u_t^2 - \sigma^2)^2 + \sum_{\tau=1}^{n_\tau} (\tilde{r}_t(\tau) + u_t u_{t-\tau})^2 \right\} \\
&= \tilde{r}_t(0)^2 + \min_{u_t \in \{-\sigma, \sigma\}} \sum_{\tau=1}^{n_\tau} (\tilde{r}_t(\tau) + u_t u_{t-\tau})^2 \\
&\le \|\tilde{\mathbf{r}}_t\|_2^2 + \sum_{\tau=1}^{n_\tau} \sigma^2 u_{\max}^2 \\
&\le \|\tilde{\mathbf{r}}_t\|_2^2 + n_\tau u_{\max}^4, \quad t > n_\tau.
\end{aligned} \tag{7.8}$$

Iterating (7.8) over $t \in \mathbf{N}$ using that $\tilde{\mathbf{r}}_0 = 0_{n+1}$ gives

$$\|\tilde{\mathbf{r}}_t\|_2^2 \leq n_\tau u_{\max}^4 t + c, \quad t > n_\tau, \tag{7.9}$$

where c is an upper bound on $\sum_{t=1}^{n_\tau} \|\tilde{\mathbf{r}}_t\|_2^2$. Noting that

$$\begin{aligned}
\|\tilde{\mathbf{r}}_t\|_2^2 &= \left(\sum_{i=1}^{t}(u_i^2 - \sigma^2)\right)^2 + \sum_{\tau=1}^{n_\tau}\left(\sum_{i=\tau+1}^{t}(u_i u_{i-\tau})\right)^2 \\
&\leq \left(\sum_{i=\tau+1}^{t} u_i^2\right)^2 + \left(\sum_{i=\tau+1}^{t} \sigma^2\right)^2 + \sum_{\tau=1}^{n_\tau}\left(t u_{\max}^2\right)^2 \\
&\leq t^2 u_{\max}^4 + n_\tau t^2 u_{\max}^4 = (n_\tau + 1)t^2 u_{\max}^4,
\end{aligned}$$

the bound, c, can be derived from

$$\sum_{t=1}^{n_\tau} \|\tilde{\mathbf{r}}_t\|_2^2 \leq (n_\tau + 1)u_{\max}^4 \sum_{t=1}^{n_\tau} t^2 = \frac{n_\tau(n_\tau + 1)^2(2n_\tau + 1)}{6} u_{\max}^4.$$

Dividing (7.9) by t^2 gives for $t \geq n_\tau + 1$,

$$\sum_{\tau=0}^{n_\tau}\left[\frac{1}{t}\sum_{i=\tau+1}^{t}(u_i u_{i-\tau} - r^d(\tau))\right]^2 \leq \frac{n_\tau u_{\max}^4}{t} + \frac{c}{t^2},$$

and consequently

$$\left|\frac{1}{t}\sum_{i=\tau+1}^{t}(u_i u_{i-\tau} - r^d(\tau))\right| \leq \sqrt{\frac{n_\tau u_{\max}^4}{t} + \frac{c}{t^2}} \longrightarrow 0, \text{ as } t \to \infty$$

by which it is concluded that

$$\lim_{t\to\infty} \frac{1}{t}\sum_{i=\tau+1}^{t} u_i u_{i-\tau} = r^d(\tau).$$

□

Remark 7.1 While the above theorem only guarantees convergence of the algorithm for generation of white noise with input constraints, simulations indicate convergence of the algorithm for most autocorrelation sequences, see the numerous simulations in Larsson et al. (2013b), Hägg et al. (2013) and Hägg et al. (2014), and the simulation study in Section 7.4.

Randomized algorithm

While the receding horizon formulation of the signal generation simplifies the objective function by considering a sequence of inputs over a reduced horizon, the constraint set is in general still non-convex due to the chance constraint (7.3). However, for a fixed value of the parameters, the constraint set is convex. From this observation, it seems appropriate to use the *scenario approach* by Calafiore and Campi (2004) to solve the chance constrained optimization. The idea is to draw samples (scenarios) of the noise sequences, system parameters, and the initial state and then solve the optimization under these sampled constraints. Each scenario correspond to one convex constraint set on the inputs u_k. If a sufficiently large number of samples are used, the original problem is satisfied with high probability. The scenario approach has previously been applied to stochastic MPC problems by, for example, Schildbach et al. (2013) and Calafiore and Fagiano (2013).

Assume that $i = 1, \dots, N_i$ independent and identically distributed samples, $\{v_1, v_2, \dots, v_{N_y}, \hat{x}_t, \theta_t\}^{(i)}$, of the process noise, the initial state and the parameter values are drawn from their joint distribution. These scenarios are used for the predictions of the system trajectories for $k = 1, \dots, N_y$ in the optimization. The scenario based signal generation formulation becomes

$$\begin{aligned} \underset{\{u_k\}_{k=1}^{N_u-1}}{\text{minimize}} \quad & \sum_{\tau=0}^{n_\tau} \|R_{t+N_u-1}(\tau) - R^d(\tau)\|_F^2, \\ \text{subject to} \quad & x_{k+1}^{(i)} = A(\theta_t^{(i)})x_k^{(i)} + B(\theta_t^{(i)})u_k + v_k^{(i)}, \\ & z_k^{(i)} = C(\theta_t^{(i)})x_k^{(i)}, \\ & x_1^{(i)} = \hat{x}_t^{(i)}, \\ & u_k \in \mathcal{U}, \quad k = 1, \dots, N_y, \\ & z_k^{(i)} \in \mathcal{Z}, \quad i = 1, \dots, N_i \end{aligned} \tag{7.10}$$

The problem (7.10) has $N_i + 1$ convex constraints on u_t. The accuracy of scenario based optimization has been studied by, for example, Calafiore and Campi (2006) and Campi and Calafiore (2009).

Sampling and discarding

In the proposed randomized algorithm (7.10), the chance constraint is replaced by a number of convex, deterministic constraints. This greatly simplifies the optimization. However, it is possible that the solution may sometimes show unwanted behavior due to unlikely scenarios being drawn. Calafiore and Fagiano (2013) suggested a sampling and discarding scheme for the scenarios to remedy this problem. The idea is to remove a small number of the drawn scenarios according to some deterministic removal rule. Unfortunately this typically increases the computational burden of the algorithm and therefore becomes prohibitive for the problem considered here.

Convex relaxation

The last step in achieving a computationally tractable problem is a convex relaxation of the objective. Since the sample autocorrelation $R(\tau)$ is quadratic in u, the optimization problem (7.10) corresponds to minimizing a constrained, multivariate, fourth-order polynomial under convex constraints. Larsson et al. (2013b) showed that, for the SISO case with input horizon $N_u = 1$, the optimization problem (7.10) can be solved analytically. However, in the general case, the problem has no analytic solution. However, by choosing N_u sufficiently small, the multivariate polynomial optimization can be solved numerically. Hägg et al. (2014) proposed two different numerical approaches to solve the optimization problem (7.10). One of the approaches, an LMI based relaxation scheme proposed by Lasserre (2000) is used here. This method has the benefit that it can certify that the global optimum has been found. In fact, the global optimum is guaranteed to be found but possibly at the price of very high computational burden.

State estimator

To be able to predict the future state trajectory, an estimate of the current state is needed. Furthermore, in the scenario based optimization, an estimate of the distribution of the state estimate is also needed. For linear systems and models, the standard filter by Kalman (1960) can be used. In the Kalman filter, the estimate of the state at time t is given by

$$\begin{aligned}
\hat{x}_{t|t-1} &= A(\theta)\hat{x}_{t-1|t-1} + B(\theta)u_t,\\
P_{t|t-1} &= A(\theta)P_{t-1|t-1}F(\theta)^T + \Lambda_v,\\
K_t &= P_{t|t-1}C(\theta)^T(C(\theta)P_{t|t-1}C(\theta)^T + \Lambda_e)^{-1},\\
\hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t(y_t - C(\theta)\hat{x}_{t|t-1}),\\
P_{t|t} &= P_{t|t-1} - K_tC(\theta)P_{t|t-1}.
\end{aligned}$$

Above, $\Lambda_v = \mathbb{E}\left\{v_t v_t^T\right\}$ is the covariance matrix of the process noise and $\Lambda_e = \mathbb{E}\left\{e_t e_t^T\right\}$ is the covariance matrix of the measurement noise. In addition to the state estimate $\hat{x}_{t|t}$, an estimate of the covariance of the estimated state, $P_{t|t} = \mathbb{E}\left\{(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T\right\}$, is also given. These two can be used to generate the samples of $\hat{x}_t$ needed for the scenarios in (7.10).

For linear systems with Gaussian noise sources, the Kalman filter gives the means square optimal estimate. For non-Gaussian noises or nonlinear systems, other estimators may give better results. For example, sequential Monte Carlo method based filters, such as particle filters could be used. Such techniques are discussed, for instance, by Del Moral et al. (2006).

Remark 7.2 In the above Kalman filter, it is assumed that the process and measurement noises are uncorrelated. Kalman filter equations for the correlated case also exist, but are somewhat more involved.

Scenario signal generation algorithm

The complete scenario based algorithm for generation of the input signal is summarized in Algorithm 7.1.

Algorithm 7.1 Scenario signal generation

```
Require: desired $R^d$
1:  $t \leftarrow 1$
2:  $R_t(\tau) \leftarrow 0, \quad \tau = 0, 1, \dots, n_\tau$
3:  while $t \le N$ do
4:      measure $y_t$
5:      calculate $\hat{x}_t$ and $P_{t|t}$
6:      draw $N_i$ scenarios of $v_1, \dots, v_{N_y}$, $\hat{x}_t$ and $\theta_t$.
7:      solve the optimization problem (7.10). If the solution is non unique, arbitrarily
        choose $u^\star$
8:      $u_t \leftarrow u_1^\star$
9:      update $R_t(\tau)$ using (7.5)
10:     $t \leftarrow t + 1$
11: end while
```

7.3 Adaptive signal generation

The method described in Algorithm 7.1 can be used to generate signals that satisfy constraints in a general setting for systems where only uncertain models are available. However, in a recursive application of the algorithm, more and more data are available

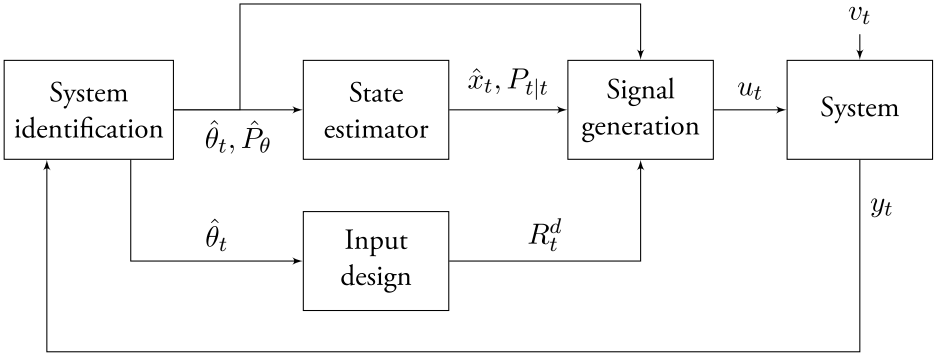


Figure 7.1 Adaptive signal generation. The parameters of the signal generation are updated online, based on the currently best available model information. The System identification block and the state estimator give model and state estimates along with the distributions needed for scenario generation. The input design can also be updated to reflect the new model information.

to the user. These data could be used to reduce the uncertainties and improve the performance of the algorithm. Specifically, in situations where little is known about the system initially, large uncertainty can lead to very conservative signals. This means that the systems is not excited as much as it could be which often means longer experiment times than necessary.

Extending the original algorithm to an adaptive version is straightforward thanks to the recursive formulation that comes naturally in the receding horizon framework. Such an adaptive scheme is sketched in the block diagram in Figure 7.1. The algorithm generates input samples online using the best available model and state estimates. As more data are collected, the model is re-estimated using system identification, reducing the uncertainty. Model updates can be performed at every time instant or on batches of data, where the model would be updated when a new batch has been processed. This choice is up to the user and is dictated by the available computational power in relation to the time constants of the true system.

The following sections describe the particulars of each of the blocks in the adaptive signal generation algorithm.

Initialization

Initialization is formally not a block in the algorithm presented in Figure 7.1. Nevertheless, some parameters in the algorithm have to be initialized before the signal generation is started. An initial model $\mathcal{M}(\hat{\theta})$ together with the distribution of the model parameters is needed. In practice, this could come from a short, initial identification experiment. Another possibility could be to obtain a model from first principles modeling. The Kalman filter also needs to be initialized by choosing the initial state estimate $\hat{x}_{0|0}$ and the corresponding uncertainty $P_{0|0}$.

System Identification

The system identification block is used to estimate the distribution of the parameters of $\mathcal{M}(\theta)$ using data Z_t up to the current time t. Using PEM, both a parameter estimate, $\hat{\theta}_t$, and a corresponding covariance estimate, $\hat{P}_\theta$, can be found from data, see, for example, the book by Ljung (1999). Using the (asymptotic) Gaussian distribution of the estimate in (2.9), the scenarios needed in the optimization can be drawn.

State estimator

The state estimation block is used to estimate the distribution of the current state based on data Z_t up to the current time t and using the best available model. In the linear framework, the model $\mathcal{M}(\hat{\theta}_t)$ can be used in the Kalman filter equations (7.11).

Ideally, the knowledge that the model is uncertain should be used also in the state estimation. The process noise covariance Λ_v reflects the certainty in the model. Unfortunately, to the best of the author's knowledge, there is no good method of transforming the parameter uncertainty $\hat{P}_\theta$ to a corresponding process noise covariance. One possibility would be to use a Kalman filter that is robust to model uncertainties, such a filter is discussed by Ruckdeschel (2010).

Signal generation

The signal generation block is the main block of the algorithm, where an input sample to be applied to the system using Algorithm 7.1 is generated. The algorithm uses the estimated model of the system, the states and their respective uncertainties from the state estimator and the system identification blocks. The desired autocorrelation of the generated signal $R_t^d(\tau)$ comes from either a user choice, an initial input design, or from an adaptive input design.

Input Design

The input design block is used to calculate the desired input autocorrelation sequence, $R_t^d(\tau)$. This can be done using the application-oriented ideas used in Chapters 3-5.

This is, however, not necessary as any input design algorithm that delivers a desired input autocorrelation sequence can be used.

One inherent problem with input design is that the solution typically depends on the unknown true system. Here, this is solved by using the latest identified model as a substitute for the true system in the design. The desired autocorrelation, $R_t^d(\tau)$, is calculated from the identified model, $\mathcal{M}(\hat{\theta}_t)$.

Termination

The termination criterion is left to the user as it often depends on the application. Sometimes a given number of samples should be generated. Other times, the algorithm should run until the estimated parameters are sufficiently accurate. In the applications oriented input design framework, for example, this would mean that the achieved information matrix is large enough.

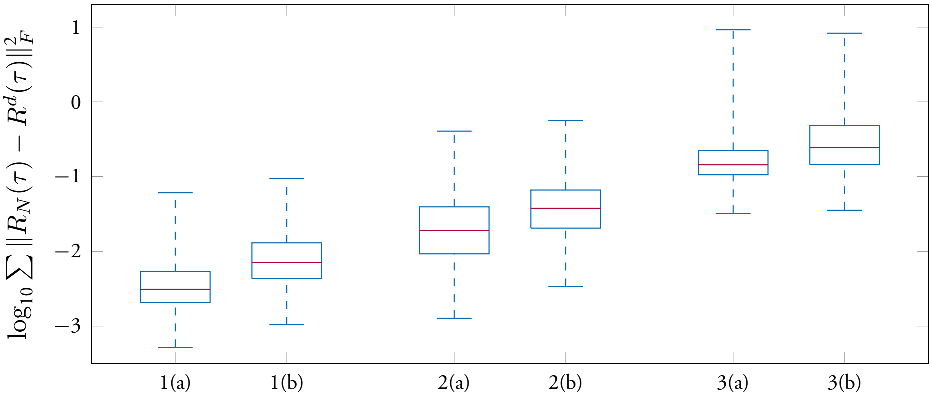


Figure 7.2 Error between the sample autocorrelation of the generated signal and the desired autocorrelation for 3,300 randomly chosen test spectra in. Group 1 is the presented algorithm without constraints, group 2 is the algorithm with input constraints, which forces the signal to be binary, and group 3 is filtered white noise. In (b), spectra with at least one pole of magnitude greater than 0.95 are considered, in (a) the remaining spectra are considered.

7.4 Comparison with filtered noise

Algorithm 7.1 is tested on 3,300 randomly chosen test spectra. The spectra are found by generating random 2 × 2 systems of random order between 1 and 25 using the `drss` command in Matlab; systems with poles on the unit circle are discarded. The systems are approximated by an FIR system with 26 coefficients and the autocorrelations corresponding to the resulting FIR spectra, normalized to have variance 1, are used.

For each test spectrum, 1,000 input samples are generated using three different methods. In the first case, the proposed algorithm is used without any constraints. In the second case, the proposed algorithm is used with the input constrained between -1 and 1, which forces the signal to be binary. Finally, in the third case, a spectral factor is used to filter white Gaussian distributed noise to get the correct autocorrelation. The results are presented in Figure 7.2. It can be seen that the proposed algorithm performs better than if white noise is filtered, even under input amplitude constraints. For 1,000 samples, the difference in the error between the sample autocorrelation and the desired autocorrelation is about one order of magnitude, in favor for the new algorithm.

7.5 Examples

In this section, simulation examples are used to illustrate the key points of the algorithm. In a first example, the convergence of the algorithm for generation of white noise is illustrated. In the second example, the algorithm is applied in an adaptive setting for estimating the gain of a system using the optimal input. The last example shows an experimental application of the algorithm on a four tank laboratory process.

Example 1 — White noise generation

Two properties of the algorithm are illustrated here. The convergence for white noise generation in the unconstrained and constrained cases is illustrated. The importance of the output horizon for satisfaction of output constraints is also shown.

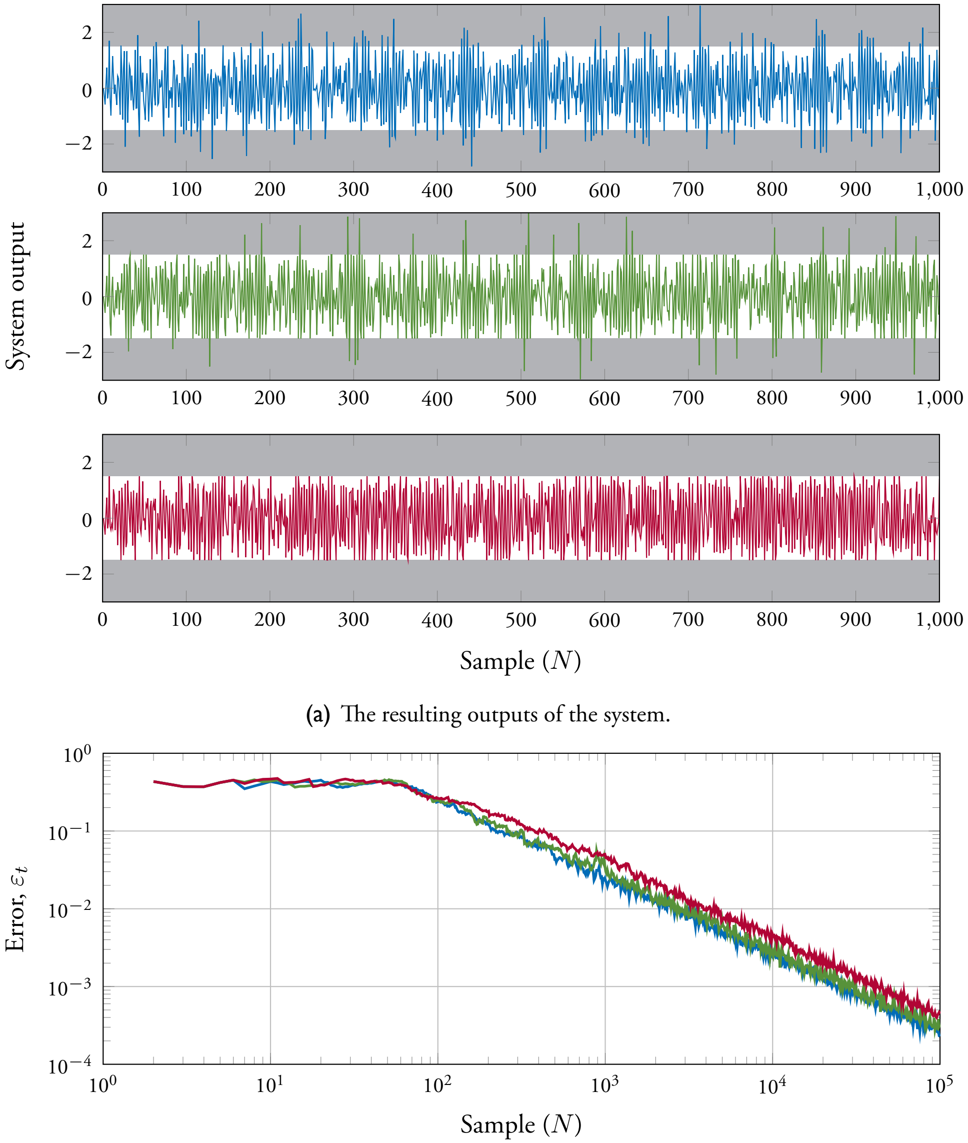
The goal is to apply white noise to a system while satisfying constraints on the output. The system is randomly generated using the `drss` command in Matlab, the direct term from input to output is removed and the system is scaled to have unit gain. The input is constrained to the interval $[-2\ 2]$ and the output to the interval $[-1.5\ 1.5]$. The input horizon is set to $N_u = 1$, the first $n_\mathcal{T} = 50$ autocorrelation coefficients are matched, and $N = 10^5$ samples are generated. Three output horizons are used: $N_y = 0$, $N_y = 1$, and $N_y = 5$. For $N_y = 0$, output constraints are not taken into account, which correspond to the situation in Theorem 7.1. The system is considered known and noise free, therefore, no scenarios, state estimation or adaptation is used.

The resulting outputs and the convergence of the algorithm are shown in Figure 7.3. For the first case, $N_y = 0$, the constraint is violated in 130 of the first 1000 samples, for $N_y = 1$ constraints are violated 70 times and for the horizon, $N_y = 5$ there are no constraint violations.

Convergence is measured in terms of the error

$$\varepsilon_t = \sqrt{\sum_{\tau=0}^{n_\mathcal{T}} \left(r_u^t(\tau) - r^d\right)^2}$$

where $r_u^t(\tau)$ is the autocorrelation sequence calculated with data up to time t. It is seen in Figure 7.3 that the cost of using a longer output horizon is a slower convergence rate. However, in this case the cost is quite low and a longer horizon results in satisfaction of the output constraints.



(a) The resulting outputs of the system.

(b) The convergence rates.

Figure 7.3 Performance of the algorithm for white noise generation. The signal generation algorithm is used to generate pseudo-random white noise while maintaining input and output constraints on a randomly generated system. Three different output horizons, $N_y = 0$ (——), $N_y = 1$ (——) and $N_y = 5$ (——), are used. The output constraints are only satisfied in the last case, when a longer output prediction horizon is used.

Example 2 — $\mathcal{L}_2$-gain estimation

The convergence of the algorithm when used in adaptive input design is illustrated. Gerencsér et al. (2009) presented the example of adaptive input design for estimation of the $\mathcal{L}_2$-gain of a dynamic system. The performance of the adaptive signal generation scheme, compared to what would be obtained if the optimal autocorrelation had been known at the beginning of the experiment, is studied.

An identification experiment of length $N = 200$ is used to estimate the $\mathcal{L}_2$-gain of the fourth order FIR system given by

$$\begin{aligned} \varphi_t &= \begin{bmatrix} u_{t-1} & u_{t-2} & u_{t-3} & u_{t-4} \end{bmatrix}^T, \\ \theta &= \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T, \\ y_t &= \varphi_t^T \theta + e_t, \end{aligned} \tag{7.12}$$

where e_t is zero mean white Gaussian noise with variance σ^2. The input is designed to have the lowest possible power that guarantees the variance of the gain estimate to be less than γ at the end of the experiment. Gerencsér et al. (2009) showed that the autocorrelation of this input signal is given by the solution to the optimization problem

$$\begin{aligned} \underset{r^d}{\text{minimize}} \quad & r^d(0) \\ \text{subject to} \quad & r^d \text{ defines a spectrum,} \\ & \begin{bmatrix} R(r^d) & 2\theta \\ 2\theta^T & \frac{\gamma N}{\sigma_e^2} \end{bmatrix} \succeq 0, \end{aligned} \tag{7.13}$$

where $R(r^d)$ is a Toeplitz matrix with first row given by $[r^d(0) \ \cdots \ r^d(n_\tau)]$. This problem can be transformed into an SDP, as shown by Gerencsér et al. (2009).

The model structure is assumed to be known but the parameters are recursively estimated using the `rarx` command from the System Identification Toolbox (Ljung, 2012). The noise variance is estimated as

$$\hat{\sigma}_t^2 = \frac{t-1}{t}\hat{\sigma}_{t-1}^2 + \frac{1}{t}(y_t - \varphi_t^T \hat{\theta}_t)^2.$$

The parameters of the true system are set to $\theta = [0.9 \ \ 0.6 \ \ 0.2 \ \ 0.3]^T$, the variance bound to $\gamma = 0.001$, and the number of autocorrelation lags is set to $n_\tau = 3$. The optimization (7.13) is solved in each sample using the newly identified model and noise variance. In this case there are no input or output constraints, which means that

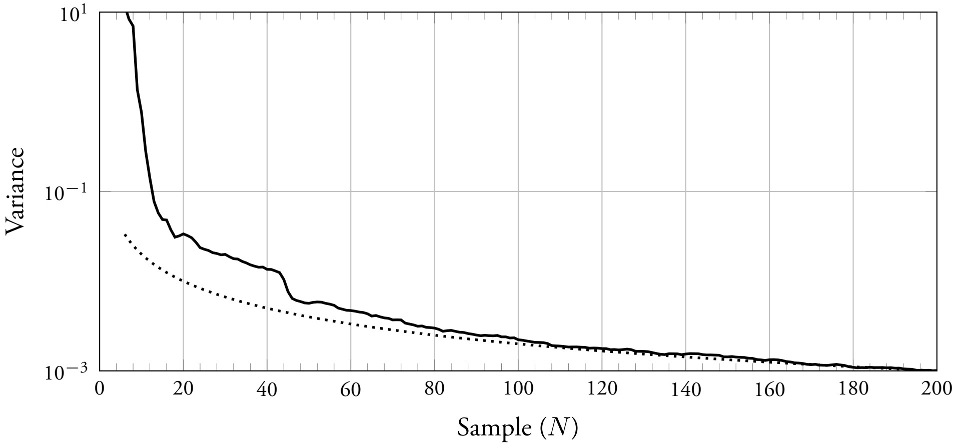


Figure 7.4 Variance of the estimated $\mathcal{L}_2$-gain of the FIR system (7.12), averaged over 200 Monte-Carlo simulations (——) and the theoretical, asymptotic variance if the true optimal input was known (······).

a state estimator is not needed. The performance of the adaptive identification scheme is evaluated in a Monte Carlo simulation with 200 trials.

Figure 7.4 shows the sample variance of the squared estimate of the $\mathcal{L}_2$-gain, calculated over the Monte Carlo trials. The input was design to give a variance of the estimated gain below $\gamma = 0.001$ after $N = 200$ samples. The sample variance at $N = 200$ is indeed very close to the desired variance. In Figure 7.4 also shows the theoretical asymptotic variance if the true optimal input signal is known from the beginning. After around 100 samples, the variance using the adaptive signal generation is very close to the optimal variance.

Example 3 — Identification of water tank system

The algorithm with adaptive signal generation and identification is tested on the four tank laboratory process suggested by Johansson et al. (1999). The layout of the process is illustrated in Figure 7.5. The process consists of four interconnected tanks, two pumps and a water reservoir. Tubes connect pump 1 to tanks 1 and 4 while pump 2 is connected to tanks 2 and 3. The ratio of water flow into the upper and lower tanks is decided by the parameters γ_1 and γ_2 for the two pumps, respectively. The water flows freely through holes at the bottom of each tank, from the upper tanks to the lower tanks and from the lower tanks into the reservoir. The system is controlled using the voltages applied to the two pumps as inputs and the water levels in the two lower tanks, tanks 1 and 2, are the system outputs.

A system identification experiment is used to identify a linear model of the system around a given operating point. The input used in the experiment is designed to be white while satisfying constraints on the input and output.

Model

The process can be modeled in continuous time by a linear model around an equilibrium point $h^0 = \begin{bmatrix} h_1^0 & h_2^0 & h_3^0 & h_4^0 \end{bmatrix}$ as

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} -\tau_1 & 0 & \tau_3 & 0 \\ 0 & -\tau_2 & 0 & \tau_4 \\ 0 & 0 & -\tau_3 & 0 \\ 0 & 0 & 0 & -\tau_4 \end{bmatrix} x(t) + \begin{bmatrix} \frac{k_1\gamma_1}{A} & 0 \\ 0 & \frac{k_2\gamma_2}{A} \\ 0 & \frac{k_2(1-\gamma_2)}{A} \\ \frac{k_1(1-\gamma_1)}{A} & 0 \end{bmatrix} u_t + v(t), \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x(t) + e(t), \end{aligned}$$

where $v(t)$ is process noise and $e(t)$ measurement noise and τ_i is given by

$$\tau_i = \frac{a_i}{A}\sqrt{\frac{g}{2h_i^0}}. \tag{7.14}$$

The system is sampled at 1 Hz using zero order hold and the corresponding discrete time model is used in the identification. The resulting discrete time noises are assumed to be Gaussian distributed, white noise with covariances Λ_v and Λ_e.

The parameters of the model are explained in Table 7.1. In this setup the parameter vector is $\theta = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & k_1 & k_2 \end{bmatrix}^T$ and the noise covariances Λ_v and Λ_e are unknown and also need to be estimated.

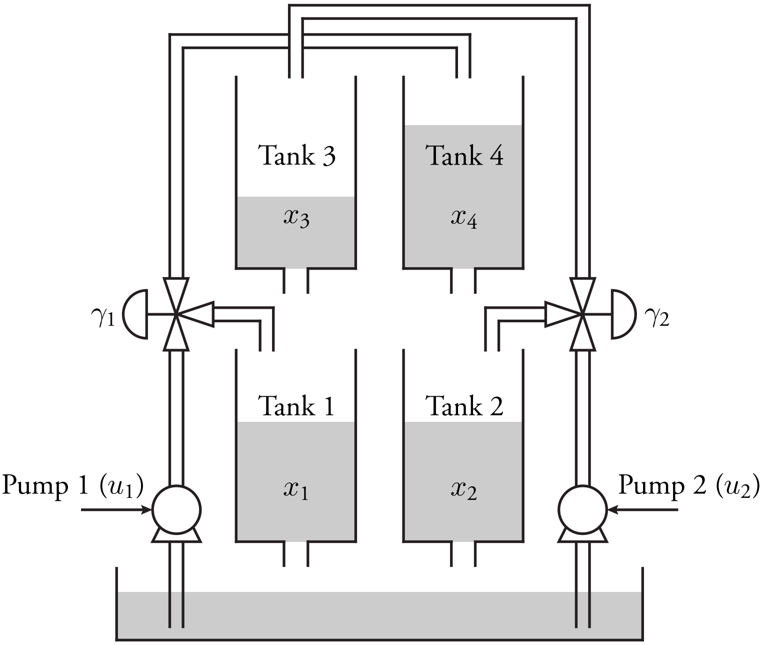


Figure 7.5 The four tank process. Water is pumped from the lower reservoir into the four tanks. The flow from pump 1 fills tanks 1 and 4 while the flow from pump 2 fills tanks 2 and 3. The flow is divided between the tanks according to the settings of the two valves, γ_1 and γ_2. The water levels of the tanks are $x_i,\ i = 1, 2, 3, 4$. The inputs to the process are the voltages u_1 and u_2.

Input design

The system is excited using a white input signal with variance 0.16 and the first $n_\tau = 11$ autocorrelation matrices of the input are matched in the signal generation algorithm. This gives the desired autocorrelation

$$R^d(0) = \begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix} \qquad R^d(\tau) = 0, \quad \tau = 1, \dots, 10.$$

It is also desired that the water levels in the two lower tanks are kept within ±0.5 cm of the operating point. The input is also constrained and should be within ±3 V of the operating point.

Table 7.1 Parameters of the four tank process and the numerical values of known parameters. The parameters without numerical values are unknown and need to be estimated in the system identification experiment.

Parameter	Value	Description
a_i	—	outlet area of tank $i = 1, 2, 3, 4$.
k_i	—	volumetric flow constant of pump $i = 1, 2$.
A	$15.15\,\mathrm{cm}^2$	cross-sectional area of tanks
g	$10^3\,\mathrm{cm}/\mathrm{s}^2$	gravitational constant
γ_1	0.6	flow ratio of tank 1 and 4
γ_2	0.6	flow ratio of tank 2 and 3

Identification experiment

As initial model for the algorithm, a known model of the water tank process with increased parameter uncertainties is used. This mimics the case with a short initial identification experiment. The estimated model and noise covariances are used to find a stationary Kalman filter for state estimation and to calculate the needed stationary estimation error covariance P. The operating point during the experiment is

$$h^0 = \begin{bmatrix}10.2 & 15.4 & 15 & 10\end{bmatrix}^T \mathrm{cm},$$
$$u^0 = \begin{bmatrix}3.75 & 3.75\end{bmatrix}^T \mathrm{V}.$$

The horizons in algorithm are chosen as $N_u = 1$ and $N_y = 6$, and $N_i = 200$ scenarios are used in the optimization. The sampling time is chosen as $T_s = 1\,\mathrm{s}$ and a total of $N = 300$ samples are generated.

The identification of a new model takes about 2-3 samples to perform and therefore it is not possible to reidentify the model in every sample. Instead, the experiment is divided into 3 batches of 100 samples each. After each batch, the model is reidentified and the Kalman filter is updated accordingly. The full state trajectory is also re-estimated using the new model. During the 3 samples that it takes to do the reidentification, it is not possible in this setup to change the input or measure the output. This is a limitation of the experimental equipment and not the algorithm. During this outage, the input is set to zero (around the operating point) and a multi step prediction of the output is used instead of the measured output.

Results

The generated input signals and the resulting measured outputs from the experiment are shown in Figure 7.6. It is seen that all the constraints are satisfied. During the first 100 samples of the experiment, the initial model is used and the algorithm is cautious, keeping the output further from the constraints. After the first reidentification, the process is operated closer to the constraints on the outputs. The model is further improved after the second reidentification, although this is hard to see in the resulting outputs. In Figure 7.6 the 99 % confidence interval of the one step ahead predictions from the Kalman filter is shown. Note that the uncertainties in the model are not reflected by this confidence interval. However, the initial conservativeness of the algorithm shows that model uncertainties are indeed taken into account.

The final covariance matrix of the generated input is

$$R_N(0) = \begin{bmatrix} 0.158 & 0.000 \\ 0.000 & 0.156 \end{bmatrix},$$

and the total error between the achieved autocorrelation and the desired sequence is

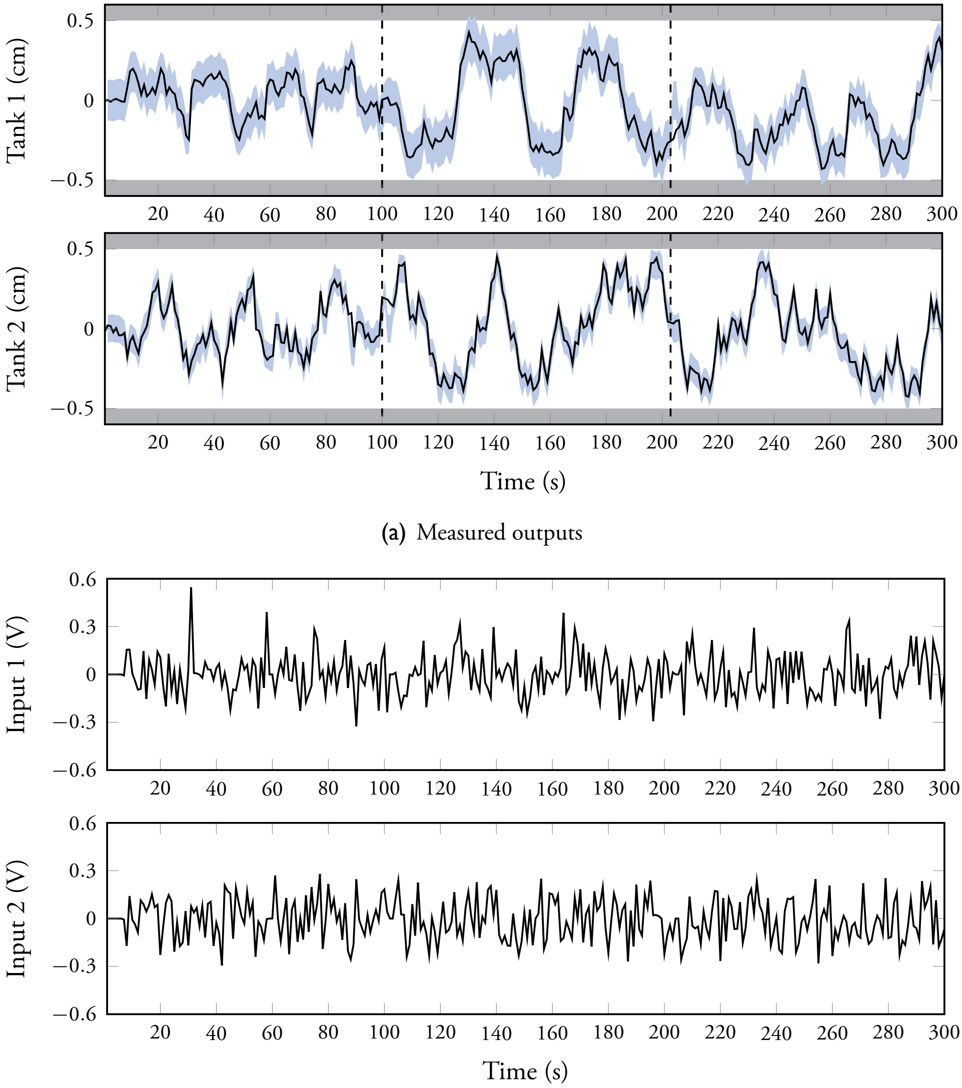
$$\sum_{\tau=0}^{n_\tau} \|R_N(\tau) - R^d(\tau)\|_F^2 = 8.25 \cdot 10^{-5}.$$

The whiteness of the generated input is also reflected by the periodograms of the two generated input signals, plotted in Figure 7.7.

For reference, an experiment where a pseudo random, white Gaussian noise signal with the desired variance is used as excitation is performed. The measured output during this experiment is shown in Figure 7.8. There is no way to account for signal constraints and several violations can be observed. Identifying a model using this input signal gives a model that is similar to the model identified with data generated by Algorithm 7.1. This is expected since the autocorrelation properties of the input signals are almost equal. The pseudo random, white signal achieves a total error

$$\sum_{\tau=0}^{n_\tau} \|R_N(\tau) - R^d(\tau)\|_F^2 = 3.3 \cdot 10^{-3}.$$

In conclusion, the proposed signal generation algorithm offers no apparent performance loss compare to using random signals while it offers the possibility to take both input and output constraints into consideration during the identification experiment.



(a) Measured outputs

(b) Generated inputs

Figure 7.6 The measured outputs and generated inputs. The signals from the experiment are plotted as (——). The 99 % confidence interval for the one step ahead prediction from the Kalman filter is also plotted as (🟦) around the measured outputs. The constraints are shown as (▨) and the reidentification times are shown as (- - -). The effect of model uncertainties is not reflected by the confidence intervals.

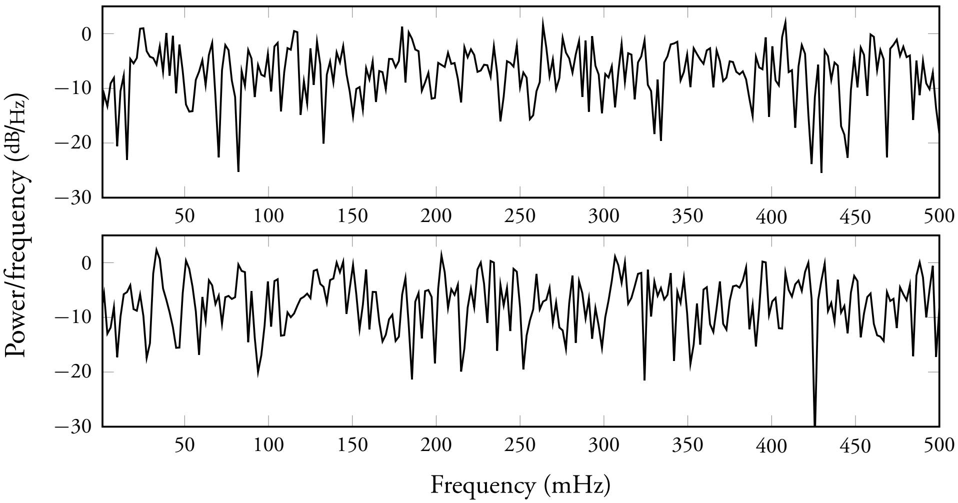


Figure 7.7 Frequency content in the inputs. The periodograms of the generated signals, $u_1(t)$ (above) and $u_2(t)$ (below), show that both signals have power evenly distributed over the whole frequency range.

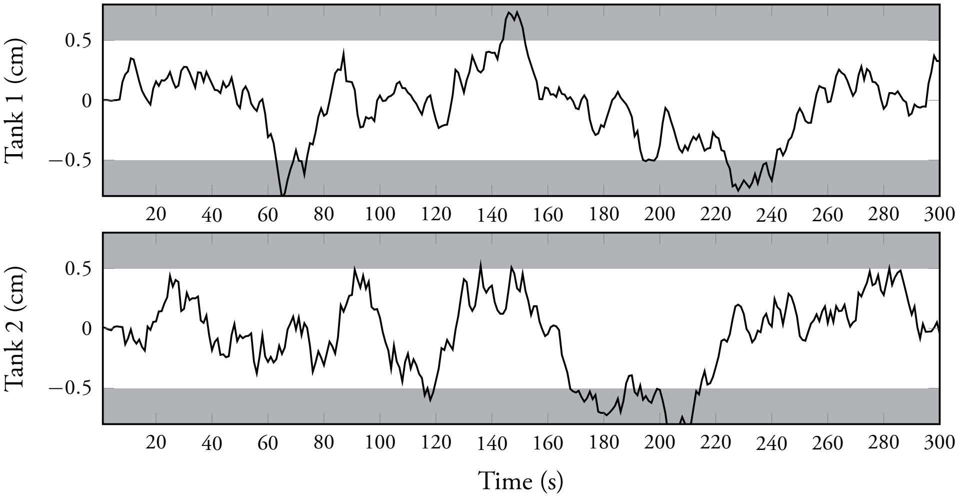


Figure 7.8 White noise experiment. The measured outputs around the equilibrium point when a pseudo-random, white Gaussian input is used. The outputs are plotted as (——) and the constraints as (▭). The constraints cannot be taken into account and several violations are observed.

7.6 Conclusions

A framework for generating an input signal with prespecified autocorrelation under time domain signal constraints has been formulated. Using a model of the system the signal is to be applied to, the algorithm can account for both input and output constraints. The method is based on stochastic optimal control and the optimization is solved using techniques from stochastic MPC. In particular, the scenario approach is used to find a relaxed, deterministic optimization problem. The framework is general and allows for use of arbitrary state estimators and identification methods. The only requirement is that distributions of the uncertain elements are available to sample from.

The framework can be used for generation of white signals with amplitude constraints with proven convergence, asymptotically in the number of generated samples. For generation of signals with short samples sizes, the methods by He et al. (2012) are often more suitable. In simulations, the algorithm shows good convergence properties also for other autocorrelations and under more general constraints.

Furthermore, examples show the usefulness of the framework as the signal generation component of adaptive input design in system identification. The full robust and adaptive version of the framework was also successfully applied to a laboratory process in a real experiment.

Chapter 8
Conclusions

Numerous success stories where MPC has improved product quality, increased process efficiency and raised profits have been reported by industry. Unfortunately, it is also often reported that, even if an MPC implementation initially performs well, after a while the control performance of MPC is significantly reduced to the point that the controller has to be turned off. This issue has been addressed in this thesis by considering the problem of updating the model used by the controller to reflect plant changes and thereby maintaining high performance. The developed ideas are summarized in the first section of this chapter and ideas for future research directions are discussed. In the second section of this chapter, the method for generating signals presented in Chapter 7 is summarized and some future research directions are suggested.

8.1 Experiment design for MPC

Model predictive control is an advanced control strategy. The controller uses an internal model of the controlled process to predict future behavior. Optimization is used to find the best control input with respect to a given control cost while respecting signal and state constraints. Feedback is introduced by using a receding horizon implementation. As a result, MPC is often a robust and flexible control strategy but also complex and difficult to analyze theoretically. From an experiment design perspective, the correlation between input and disturbances, which is introduced by the feedback and the nonlinear nature of the controller are the complicating factors. The general application-oriented experiment design problem for MPC was formulated as an optimal control problem with constraints.

MDP formulation

A matrix constrained Markov decision process formulation of the application-oriented experiment design for MPC was formulated. The system and controller models assume finite state and action spaces, which typically is an approximation since many industrial processes have states and actions in infinite spaces. The formulation can be of theoretical interest as it allows for study of the experiment design problem for general noise structures and may offer some insight into the solution to the optimal control problem. The resulting solution can also be used as a benchmark for alternative, approximate solutions. The problem was solved using an SDP, which is an extension of the linear programming solution that is used for CMDPs.

In practice, the formulation suffers from the curse of dimensionality whenever state and action spaces have to be discretized. However, it may be an interesting route for experiment design for systems that inherently have a finite number of states and actions.

As a final comment, similar problems are considered in the reinforcement learning community. There the controller tries to learn the behavior of the system while it is controlling it. The methods used are, however, very different. An interesting future research direction could be to merge the learning approaches and the application-oriented paradigm.

MPC formulations

Two controllers based on MPC where the application-oriented experiment design is included as an explicit constraint were developed. This results in an MPC with dual properties that is called MPC-X. These controllers are restricted to OE models, contrary to the MDP based controller. However, MPC-X is much less computationally demanding as it results in significantly smaller optimization problems. Therefore, MPC-X is implementable on much larger systems.

The restriction to OE models is somewhat limiting in a closed-loop experiment design setting. The inevitable correlation between the input from the controller and the disturbances can result in very bad estimates if it is disregarded in the experiment design. This becomes clear in the industrial experimental validation of MPC-X presented in Chapter 6. A possible route to extending MPC-X to more general noise models is to consider a stochastic MPC formulation. This is interesting for future research.

The choice of the scaling parameter κ, which controls the level of excitation remains a difficulty and a badly chosen value may result in an infeasible optimization problem. This problem is shared by MPC and the persistently exciting MPC algorithms. The problem of infeasibility can be avoided by softening the requirements on excitation, as is done in the minimum time MPC-X. However, then there are no guarantees on excitation in the input. Suitable choices of κ should be further investigated.

Finally, MPC-X, as it is formulated in this thesis, generates excitation such that a good model can be estimated after a batch of data has been collected. However, changing the algorithm to allow for a truly adaptive MPC formulation seems fairly straightforward. This is a very interesting idea of future research, well in line with current trends in the MPC community.

Industrial validation

The MPC-X algorithm developed in the thesis was implemented and evaluated on an industrial depropanizer in an oil refinery during normal operation of the plant. The experiments were performed on a simulator of the depropanizer as well as on the real production unit. This seems to be the first time that an MPC with dual properties has been tested on a full scale industrial process.

On the simulator, the resulting accuracy, or γ values, is low compared to the achieved performance. This could partly be explained by the fact that the information matrix is calculated based on the model used by the MPC, which is sampled at a 2 minutes interval. For the identification however, 30 seconds sampling is used, resulting in more samples. This should have a positive effect on the model quality and hence also on the performance when the model is used.

Choosing the scaling κ is a major issue in the presented formulation, a too fast increase easily results in an infeasible optimization problem while a too slow increase gives poor excitation, which wastes experimentation time. The suggested scaling with a linear increase in each sample was not possible to use in practice and instead, κ was manually adjusted after a batch of data had been collected. The resulting information available in the data was not as high as wanted, resulting in low values for γ. This explains why the performance after reidentification is not as good as the initial performance.

The experiments on the actual depropanizer showed the importance of correct noise modeling in closed loop identification. The initial attempts with a model without feed forward from the C_4 content in the feed resulted in models that were completely wrong. The reason for this was that not including this signal results in an input which is highly correlated to the disturbance since the controller is trying to counteract the effect of changes in the feed composition through feedback. The high correlation means that it is virtually impossible to separate the effect of the input and the disturbance, resulting in useless models. Including the disturbance in the model allows for MPC-X to take the correlation into account through the information matrix. This is an indication that MPC-X should be extended to models with more complex noise models than output error.

Changes in operating point and the level of the main disturbance, the feed composition, makes fair comparison of the cases on the actual depropanizer more challenging. This also shows that one should probably include the effect of disturbances in a more direct way in the application cost.

The results emphasize the need for extending MPC-X to more general model classes where the correlation between input and noise can be taken into account and this should be further investigated. The difficulty in choosing the level of excitation through the scaling factor also shows the merits of a scheme like the minimum time MPC-X in Section 5.4. A similar experiment using that algorithm would be of interest to conduct. Finally, the observed results, where good model fit does not correspond to better controller performance, indicate an intricate interplay between models and controllers that should be better understood.

8.2 Signal generation

A method for generating signals with prescribed autocorrelation under time domain constraints was developed in Chapter 7. The motivation was that many optimal experiment design methods give the optimal spectrum of the input to be used during the identification. Therefore, any time domain constraints need to be accounted for during the signal generation. The method was formulated as an optimization problem which was simplified using the receding horizon principle. Consequently, the experiment design problem can be seen as a nonlinear, stochastic MPC. The stochastics were handled using scenario optimization and the resulting optimization problem was approximately solved using an LMI based approximation technique.

The algorithm has proven convergence, as the sample size grows, for generation of white noise. However, no proof exists for more general autocorrelations. Simulations show good convergence also for the general case. Future research should focus on investigating for which autocorrelations the algorithm converges. Formal proofs of more general convergence would also be useful.

Bibliography

E. Aggelogiannaki and H. Sarimveis (2006). Multiobjective constrained MPC with simultaneous closed-loop identification. *International Journal of Adaptive Control and Signal Processing*, 20(4): 145–173.

J. C. Agüero and G. C. Goodwin (2007). Choosing between open- and closed-loop experiments in linear system identification. *IEEE Transactions on Automatic Control*, 52(8): 1475–1480.

J. C. Agüero, J. S. Welsh, and O. J. Rojas (2007). On the optimality of binary-open-loop experiments to identify hammerstein systems. In *Proceedings of the 2007 American Control Conference*. New York, NY.

E. Altman (1999). *Constrained Markov Decision Processes*. Chapman and Hall/CRC.

M. Annergren, D. Kauven, C. A. Larsson, M. G. Potters, Q. N. Tran, and L. Özkan (2013). On the Way to Autonomous Model Predictive Control: A Distillation Column Simulation Study. In *Proceedings of the 10th IFAC International Symposium on Dynamics and Control of Process Sytsems*. Mumbai, India.

M. Barenthin, H. Jansson, and H. Hjalmarsson (2005). Applications of Mixed $\mathcal{H}_\infty$ and $\mathcal{H}_2$ Input Design in Identification. In *Proceedings of the 16th IFAC World Congress*. Prague, Czech Republic.

A. Barto, R. Sutton, and C. Anderson (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(5): 834–846.

M. Bauer and I. K. Craig (2008). Economic assessment of advanced process control: A survey and framework. *Journal of Process Control*, 18(1): 2–18.

D. A. Berry and B. Fristedt (1985). *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall Ltd.

D. Bertsekas (1975). Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3): 415–419.

R. Bitmead (1984). Persistence of excitation conditions and the convergence of adaptive schemes. *IEEE Transactions on Information Theory*, 30(2): 183–191.

X. Bombois, M. Gevers, and G. Scorletti (2005). Open-loop versus closed-loop identification of Box–Jenkins models: a new variance analysis. In *Proceedings of the 44th IEEE Conference on Decision and Control*, 3117–3122. Sevilla, Spain.

X. Bombois and G. Scorletti (2012). Design of least costly identification experiments. The main philosophy accompanied by illustrative examples. *Journal Européen des Systèmes Automatisés*, 46(6-7): 587–610.

X. Bombois, G. Scorletti, M. Gevers, P. M. J. Van den Hof, and R. Hildebrand (2006). Least costly identification experiment for control. *Automatica*, 42: 1651–1662.

X. Bombois, G. Scorletti, P. M. J. Van den Hof, and M. Gevers (2004). Least costly identification experiment for control. A solution based on a high-order model approximation. In *Proceedings of the 2004 American Control Conference*. Boston, MA.

P. Boufounos (2007). Generating Binary Processes with all-Pole Spectra. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2007*.

S. Boyd and L. Vandenberghe (2003). *Convex Optimization*. Cambridge University Press.

G. Calafiore and M. C. Campi (2004). Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1): 25–46.

G. C. Calafiore and M. C. Campi (2006). The Scenario Approach to Robust Control Design. *IEEE Transactions on Automatic Control*, 51(5): 742–753.

G. C. Calafiore and L. Fagiano (2013). Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, 58(1): 219–224.

E. Camacho and C. Bordons (2004). *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London.

M. Campi and G. Calafiore (2009). Notes on the Scenario Design Approach. *IEEE Transactions on Automatic Control*, 54(2): 382–385.

W. M. Canney (2003). The future of advanced process control promises more benefits and sustained value. *Oil & Gas Journal*, 101(16): 48–54.

W. M. Canney (2005). Are you getting the full benefit from your advanced process control system? *Hydrocarbon processing*, 84(6): 55–58.

C. Chow and J. Tsitsiklis (1991). An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Transactions on Automatic Control*, 36(8): 898–914.

D. W. Clarke, C. Mohtadi, and P. S. Tuffs (1987). Generalized predictive control — parts I and II. *Automatica*, 23(2): 137–160.

C. R. Cutler and B. C. Ramaker (1980). Dynamic matrix control — a computer control algorithm. In *Proceeding of the 1980 American Control Conference*. San Francisco, CA.

J. L. M. de Carvalho and J. M. C. Clark (1983). Characterizing the autocorrelations of binary sequences. *IEEE Transactions on Information Theory*, 29(4): 502–508.

R. De Keyser and A. Van Cauwenberghe (1979). A self-tuning multistep predictor application. *Automatica*, 17(1): 167–174.

R. De Keyser, P. Van de Velde, and F. Dumortier (1988). A comparative study of self-adaptive long-range predictive control methods. *Automatica*, 24(2): 149–163.

P. Del Moral, A. Doucet, and A. Jasra (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3): 411–436.

A. Ebadat, M. Annergren, C. A. Larsson, C. R. Rojas, B. Wahlberg, H. Hjalmarsson, M. Molander, and J. Sjöberg (2014). Application set approximation in optimal input design for model predictive control. In *Proceedings of the 2013 European Control Conference*. Strasbourg, France.

V. V. Fedorov (1972). *Theory of Optimal Experiments*. Academic Press, New York.

A. A. Feldbaum (1960–61). Dual control theory. I–IV. *Automation and Remote Control*, 21,22.

N. Filatov and H. Unbehauen (2000). Survey of adaptive dual control methods. *IEE Proceedings – Control Theory and Applications*, 147(1): 118–128.

U. Forssell and L. Ljung (1998). Identification for control: some results on optimal experiment design. In *Proceedings of the 37th IEEE Conference on Decision and Control.* Tampa, FL.

U. Forssell and L. Ljung (1999). Closed-loop identification revisited. *Automatica*, 35(7): 1215–1241.

U. Forssell and L. Ljung (2000). Some results on optimal experiment design. *Automatica*, 36(5): 749–756.

H. Genceli and M. Nikolaou (1996). New approach to constrained predictive control with simultaneous model identification. *AIChE Journal*, 42(10): 2857–2868.

L. Gerencsér, H. Hjalmarsson, and J. Mårtensson (2009). Identification of ARX systems with non-stationary inputs — asymptotic analysis with application to adaptive input design. *Automatica*, 45(3): 623–633.

M. Gevers, A. Bazanella, X. Bombois, and L. Miskovic (2009). Identification and the information matrix: How to get just sufficiently rich? *Automatic Control, IEEE Transactions on*, 54(12): 2828–2840.

M. Gevers and L. Ljung (1986). Optimal experiment designs with respect to the intended model application. *Automatica*, 22(5): 543–554.

A. González, A. Ferramosca, G. Bustos, J. Marchetti, and D. Odloak (2013). Model predictive control suitable for closed-loop re-identification. In *Proceeding of the 2013 American Control Conference*, 1709–1714.

G. C. Goodwin and R. L. Payne (1977). *Dynamic System Identification: Experiment Design and Data Analysis.* Academic Press, New York.

P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski (2006). Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4): 523–533.

M. Grant and S. Boyd (2011). CVX: Matlab Software for Disciplined Convex Programming, version 1.21. `http://cvxr.com/cvx`.

M. Green and J. B. Moore (1986). Persistence of excitation in linear systems. *Systems & Control Letters*, 7(5): 351–360.

H. Guidi, C. A. Larsson, Q. N. Tran, L. Ozkan, and H. Hjalmarsson (2013). Deliverable 6.4 — results on the Sasol validation case. Autoprofit project.

P. Hägg, C. Larsson, A. Ebadat, B. Wahlberg, and H. Hjalmarsson (2014). Input signal generation for constrained multiple-input multiple-output systems. In *Proceedings of 19th IFAC World Congress*. Cape Town, South Africa.

P. Hägg, C. A. Larsson, and H. Hjalmarsson (2013). Robust and adaptive excitation signal generation for input and output constrained systems. In *Proceedings of the 2013 European Control Conference*. Zürich, Switzerland.

J. Hammersley and D. Handscomb (1964). *Monte Carlo Methods*. Chapman and Hall Ltd.

E. Hannan (1970). *Multiple Time Series*. Wiley, New York.

T. J. Harris (1989). Assessment of control loop performance. *The Canadian Journal of Chemical Engineering*, 67(5): 856–861.

H. He, J. Li, and P. Stoica (2012). *Waveform Design for Active Sensing Systems: A Computational Approach*. Cambridge University Press.

T. A. Heirung, E. B. Ydstie, and B. Foss (2013). An MPC approach to dual control. In *Proceedings of the 10th IFAC International Symposium on Dynamics and Control of Process Systems*. Mumbai, India.

R. Hildebrand and M. Gevers (2013). Central extensions in closed-loop optimal experiment design. In *Proceedings of the 52nd IEEE Conference on Decision and Control*. IEEE, Florence, Italy.

R. Hildebrand, M. Gevers, and G. Solari (2010). Closed-loop optimal experiment design: The partial correlation approach. In *Proceedings of the 49th IEEE Conference on Decision and Control*. Atlanta, GA.

H. Hjalmarsson (2005). From experiment design to closed-loop control. *Automatica*, 41(3): 393–438.

H. Hjalmarsson (2009). System identification of complex and structured systems. *European Journal of Control*, 15(4): 275–310.

H. Hjalmarsson, M. Gevers, and F. De Bruyne (1996). For model based control design criteria, closed loop identification gives better performance. *Automatica*, 32: 1659–1673.

H. Hjalmarsson and H. Jansson (2008). Closed loop experiment design for linear time invariant dynamical systems via LMIs. *Automatica*, 44(3): 623–636.

H. Hjalmarsson and C. A. Larsson (2012). Deliverable 3.1 — Novel algorithms for least-costly closed-loop re-identification. Autoprofit project.

H. Hjalmarsson, C. A. Larsson, and M. Annergren (2013a). Deliverable 3.2 — Matlab software toolbox. Autoprofit project.

H. Hjalmarsson, C. A. Larsson, P. Hägg, and A. Ebadat (2013b). Deliverable 3.3 — Novel algorithms for productivity preserving testing. Autoprofit project.

P. Hokayem, E. Cinquemani, D. Chatterjee, F. Ramponi, and J. Lygeros (2012). Stochastic receding horizon control with output feedback and bounded controls. *Automatica*, 48(1): 77–88.

A. Hordijk and L. C. M. Kallenberg (1984). Constrained undiscounted stochastic dynamic programming. *Mathematics of Operations Research*, 9(2): 276–289.

B. Huang, S. Lu, and F. Xu (2006). Towards optimal MPC performace: Industrial tools for multivariate control monitoring and diagnosis. Presented at International Workshop on Solving Industrial Control and Optimization problems, Cramado, Brazil.

H. Jansson and H. Hjalmarsson (2005). Input Design via LMIs Admitting Frequency-wise Model Specifications in Confidence Regions. *IEEE Transactions on Automatic Control*, 50(10): 1534–1549.

K. Johansson, A. Horch, O. Wijk, and A. Hansson (1999). Teaching multivariable control using the quadruple-tank process. In *Proceedings of the 38th IEEE Conference on Decision and Control*, December, 807–812. Phoenix, AZ.

L. P. Kaelbling, M. L. Littman, and A. W. Moore (1996). Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4: 237–285.

R. E. Kalman (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME — Journal of Basic Engineering*, 82(Series D): 35–45.

S. Karlin and W. J. Studden (1966). Optimal experimental designs. *The Annals of Mathematical Statistics*, 37(4): 783–815.

J. Kiefer (1959). Optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 21(2): 272–319.

J. Kiefer and J. Wolfowitz (1959). Optimum designs in regression problems. *The Annals of Mathematical Statistics*, 30(2): 271–294.

G. Kotze, C. Chetty, and S. Simbanegavi (2011). *Advanced Control and Optimization (ACO) for SCC FT Depropanizer. Advanced Process Control: AS-built document.* SASOL Synthetic Fuels (Pty) Ltd., Honeywell Automatic and Control Solutions South Africa (Pty) Ltd., PO Box 6674, Midrand 1685, South Africa.

T. Lai and C. Wei (1982). Least-squares estimates in stochastic regression-models with applications to identification and control of dynamic systems. *The Annals of Statistics*, 10(1): 154–166.

C. A. Larsson (2011). *Toward applications oriented optimal input design with focus on model predictive control.* Licentiate thesis, KTH, Automatic Control.

C. A. Larsson, M. Annergren, and H. Hjalmarsson (2011a). On Optimal Input Design for Model Predictive Control. In *Proceedings of the 50th IEEE Conference on Decision and Control.* Orlando, FL.

C. A. Larsson, M. Annergren, H. Hjalmarsson, C. R. Rojas, X. Bombois, A. Mesbah, and P. E. Modén (2013a). Model predictive control with integrated experiment design for output error systems. In *Proceedings of the 2013 European Control Conference*. Zürich, Switzerland.

C. A. Larsson, A. Ebadat, C. R. Rojas, and H. Hjalmarsson (2014a). An application-oriented approach to optimal control with excitation for closed-loop identification. *European Journal of Control.* To be submitted.

C. A. Larsson, P. Hägg, and H. Hjalmarsson (2013b). Recursive generation of amplitude constrained signals with prescribed autocorrelation sequence. In *Proceedings of the 2013 American Control Conference*. Washington DC, USA.

C. A. Larsson, P. Hägg, and H. Hjamarsson (2014b). Generation of signals with specified second order properties for constrained systems. *International journal of adaptive control and signal processing.* Submitted.

C. A. Larsson, C. R. Rojas, X. Bombois, and H. Hjalmarsson (2014c). Experimental evaluation of model predictive control with excitation (MPC-X) on an industrial depropanizer. *Journal of Process Control.* Submitted.

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

J. B. Lasserre (2000). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3): 796–817.

F. Lewis, D. Vrabie, and K. Vamvoudakis (2012). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems*, 32(6): 76–105.

K. Lindqvist and H. Hjalmarsson (2001). Identification for control: Adaptive input design using convex optimization. In *Proceedings of the 40th IEEE Conference on Decision and Control*. Orlando, FL.

B. Liu and J. Munson, D. (1982). Generation of a random sequence having a jointly specified marginal distribution and autocovariance. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 30(6): 973–983.

L. Ljung (1985). Asymptotic variance expressions for identified black-box transfer function models. *IEEE Transactions on Automatic Control*, 30(9): 834–844.

L. Ljung (1999). *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition.

L. Ljung (2012). *System Identification Toolbox: User's Guide 2012b*. The MathWorks, Inc.

L. Ljung and P. Caines (1979). Asymptotic normality of prediction error estimation for approximate system models. *Stochastics*, 3(1): 29–46.

L. Ljung and T. Söderström (1983). *Theory and Practice of Recursive Identification*. Signal Processing, Optimization and Control. The MIT Press, Cambridge, MA.

J. Lunze (1998). On the markov property of quantised state measuirement sequences. *Automatica*, 34(11): 1439–1444.

Z. Q. Luo, W. K. Ma, A. So, Y. Ye, and S. Zhang (2010). Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3): 20–34.

J. M. Maciejowski (2002). *Predictive Control with Constraints*. Prentice Hall, Edinburgh Gate, Harlow, Essex, England.

I. R. Manchester (2010). Input Design for System Identification via Convex Relaxation. *ArXiv e-prints*.

G. Marafioti (2010). *Enhanced Model Predictive Control: Dual Control Approach and State Estimation Issues*. Ph.D. thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics.

D. Mayne, J. Rawlings, C. Rao, and P. Scokaert (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6): 789–814.

R. K. Mehra (1974). Optimal Inputs for Linear System Identification. *IEEE Transactions on Automatic Control*, 19(3): 192–200.

G. Menga and E. Mosca (1980). MUSMAR: Multivariable adaptive regulator based on multistep cost functionals. In *Advances in Control*, 334–341. Dordrecht.

P. E. Modén and M. Lundh (2013). Performance monitoring for model predictive control maintenance. In *Proceedings of the 2013 European Control Conference*.

M. Morari and J. H. Lee (1997). Model Predictive Control: Past, Present and Future. *Computers and Chemical Engineering*, 23: 667–682.

R. Munos and A. Moore (2002). Variable resolution discretization in optimal control. *Machine Learning*, 49(2-3): 291–323.

F. Pieterse, G. Kotze, and S. Simbanegavi (2011). *Advanced Control and Optimization (ACO) for C3 distribution. FT Depropanizer APC: Operator Training Manual.* SASOL Synthetic Fuels (Pty) Ltd., Honeywell Automation and Control Solutions South Africa (Pty) Ltd., PO Box 6674, Midrand 1685, South Africa.

R. Pintelon and J. Schoukens (2001). *System Identification: A Frequency Domain Approach*. IEEE Press, New York.

M. L. Puterman (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

S. J. Qin and T. A. Badgwell (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7): 733 – 764.

J. Rathouský and V. Havlena (2011). MPC-based approximation of dual control by information maximization. In *Proceedings of the International Conference on Process Control*, 247–252. Tatranská Lomnica, Slovakia.

J. Rathouský and V. Havlena (2013). Mpc-based approximate dual controller by information matrix maximization. *International Journal of Adaptive Control and Signal Processing*, 27(11): 974–999.

J. Richalet, A. Rault, J. Testud, and J. Papon (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5): 413 – 428.

C. R. Rojas, J. C. Agüero, J. S. Welsh, and G. C. Goodwin (2008). On the equivalence of least costly and traditional experiment design for control. *Automatica*, 44(11): 2706–2715.

C. R. Rojas, J. S. Welsh, and G. C. Goodwin (2007). A Receding Horizon Algorithm to Generate Binary Signals with a Prescribed Autocovariance. In *Proceedings of the 2007 American Control Conference*, 122 –127.

P. Ruckdeschel (2010). Optimally robust Kalman filtering. *Berichte des Fraunhofer ITWM*, 185.

G. Schildbach, L. Fagiano, C. Frei, and M. Morari (2013). The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *ArXiv e-prints.*

J. Schoukens, P. Guillaume, and R. Pintelon (1991). Design of multisine excitations. In *Proceedings of the International Conference on Control 1991*, 638–643.

N. Z. Shor (1987). Quadratic optimization problems. *Soviet Journal of Circuits and Systems Sciences*, 25(6): 1–11.

S. Skogestad (1997). Dynamics and control of distillation columns: A tutorial introduction. *Chemical Engineering Research and Design*, 75(6): 539–562.

S. Skogestad, M. Morari, and J. Doyle (1988). Robust control of ill-conditioned plants: high-purity distillation. *IEEE Transactions on Automatic Control*, 33(12): 1092–1105.

T. Söderström (2002). *Discrete-Time Stochastic Systems: Estimation and Control.* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

T. Söderström, L. Ljung, and I. Gustavsson (1976). Identifiability conditions for linear multivariable systems operating under feedback. *IEEE Transactions on Automatic Control*, 21(6): 837–840.

K. C. Toh, M. J. Todd, and R. H. Tütüncü (1999). SDPT3 — a MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11: 545–581.

Q. N. Tran, L. Özkan, and A. C. P. M. Backx (2012). MPC tuning based on impact of modelling uncertainty on closed-loop performance. In *Proceedings of the AIChE annual meeting*. Pittsburg, PA.

P. Van Overschee and B. De Moor (1996). *Subspace Identification for Linear Systems — Theory – Implementation – Applications*. Springer US.

E. Žáčeková, S. Privara, and J. Komarek (2012). On dual control for buildings using persistent excitation condition. In *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, HI.

E. Žáčeková, S. Privara, Z. Vana, J. Cigler, and L. Ferkl (2013). Dual control approach for zone model predictive control. In *Proceedings of the 2013 European Control Conference*. Zürich, Switzerland.

C. J. C. H. Watkins (1989). *Learning from Delayed Rewards*. Ph.D. thesis, King's College, Cambridge, UK.

B. Wittenmark (1995). Adaptive dual control methods: An overview. In *In 5th IFAC symposium on Adaptive Systems in Control and Signal Processing*, 67–72.

V. Yakubovich (1962). Solution of certain matrix inequalities occurring in the theory of automatic control. *Proceedings of the USSR Academy of Sciences*, 1304–1307.

E. B. Ydstie (1984). Extended horizon adaptive control. In *Proceedings of the 9th IFAC World Congress*. Budapest, Hungary.

M. B. Zarrop (1979). *Optimal Experiment Design for Dynamic System Identification*, volume 21 of *Lecture Notes in Control and Information Sciences*. Springer Verlag, Berlin Heidelberg, Germany.

Y. Zhu (2009). System identification for process control: recent experience and outlook. *International Journal of Modelling, Identification and Control*, 6(2): 20–32.