NTNU

Norwegian University of
Science and Technology

Subspace Identification using Closed-Loop Data

Morten Bakke

Master of Science in Engineering Cybernetics
Submission date: June 2009
Supervisor: Tor Arne Johansen, ITK
Co-supervisor: Sigurd Skogestad, Institutt for kjemisk prosessteknikk

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The goal of this master thesis is to investigate MIMO system identification in closed loop using Subspace Identification Methods (SIMs). These methods are known for their numerical advantages, having the ability to cope with large data sets. Most of the theory on subspace methods assumes that datasets are collected in open-loop, and earlier results show that ordinary subspace methods fail when closed-loop data is applied, i.e. giving biased estimates of system parameters. The first goal of this thesis is to investigate former results of other researchers through a literature study on this topic. This includes a presentation of a few different subspace identification methods. Second, a selection of different subspace methods should be tested on MIMO LTI systems with different structures and characteristics in MATLAB. Third, the methods should be tested on a nonlinear industrial process using the simulation tool UniSim. The simulation work to be done is more specifically:

- Identify different idealized test systems in Matlab. It should be tested how different methods cope with a coupled system with a common denominator, as well as a higher order coupled, stiff system.
- Industrial case: Nonlinear debutanizer simulated in UniSim. Identify open-loop dynamics while the process is running in closed loop using different subspace methods.

Assignment given: 12. January 2009
Supervisor: Tor Arne Johansen, ITK

Abstract

The purpose of this thesis was to investigate how different subspace identification methods cope with closed loop data, and how the controller parameters affect the quality of the acquired models. Three different subspace methods were subject for investigation; the MOESP method, the N4SID method and the DSR_e method. It is shown through a simulation example that all three subspace methods will identify the correct open-loop model from closed-loop data if the data record is noise-free (deterministic identification with perfect data). This result is not a new one, but a confirmation of the results from other researchers. Among the three different subspace methods that were investigated, the DSR_e method developed by dr. David Di Ruscio gave the best overall results. This method is especially designed to cope with closed-loop data, different from the MOESP and N4SID methods.

Controller gain is shown to have a significant effect on the quality of the identified model when there is noise present in the loop. It is shown by simulations that up to a point, higher controller gain during the identification experiment actually gives more accurate open-loop models than models identified with lower controller gain. One of the reasons for this is that high gain tuning provides a higher signal to noise ratio through amplification of the reference signal, rendering the noise in the data used for identification less significant. Another reason may be that frequencies in the input signals will be more concentrated around the achievable bandwidth of the controller, which produces system outputs with more information of the frequency response around this bandwidth frequency. This is turn will reveal frequency information from the system that is important for control purposes.

PREFACE

The work presented in this thesis was carried out at the *Department for Engineering Cybernetics*, NTNU Norway, during the spring of 2009. The main topic of the thesis is system identification, hereunder identification for control.

The problem statement is made by the author, but several people deserve acknowledgement for helping out on both shaping the problem statement, and for great assistance during the work.

- I will thank my supervisor Tor Arne Johansen for inspiring conversations and useful comments.
- Sigurd Skogestad has been helpful with introducing me to the common opinions on the subject, and with giving me inspiration to use innovative thinking.
- Thanks to David Di Ruscio from HiT for providing me with the excellent Matlab toolbox DSR for system identification, and for insightful comments on my work.

Last, but not least, I would like to thank my fellow students Bjarne Grimstad, Henrik Tuvnes, Jonas Ingvaldsen, Jørgen Syvertsen and Martin Kvernland for a great semester with lots of fun.

" The times they are a-changin' "
- Bob Dylan

CONTENTS

LIST OF FIGURES

NOTATIONAL CONVENTIONS AND ABBREVIATIONS

$\mathbb{R}^n$	Eucledian space of dimension n
ARMAX	AutoRegressive Moving Average with eXogeneous inputs
CRLL	Cramér-Rao Lower Limit
CVA	Canonical Variate Analysis
DSR	combined Deterministic and Stochastic system identification and Realization
FIR	Finite Impulse Response
IV	Instrumental Variables
LTI	Linear Time Invariant
MIMO	Multiple Inputs Multiple Outputs
MOESP	Multivariable Output Error State sPace
N4SID	Numerical Subspace State Space System Identification
PRBS	Pseudo Random Binary Signal
RGA	Relative Gain Array
SIM	Subspace Identification Method
SISO	Single Input Single Output
SVD	Singular Value Decomposition

CHAPTER 1

INTRODUCTION

This chapter gives a short introduction to the structure of this thesis, as well as an introduction to the most relevant topics that are subject for investigation[1].

1.1 Structure of the thesis

Introduction part

The introduction chapter of this thesis is meant to give an introduction to identification for control in general, as well as the main issues with closed-loop data in an identification framework. In addition, a short presentation of state-of-the-art identification techniques is given.

Theory and literature review part

- Chapter 2 introduces some methods and concepts from system identification theory that are relevant for this thesis. This includes the model structure used, excitation signals used for identification, preconditioning of identification data, and definition of model performance metrics.

- Chapter 3 is included to give some insight into the effect of closing feedback loops, and issues with closed-loop identification.

- In chapter 4, a detailed discussion on subspace identification is given. The contents therein are definitions of mathematical tools used, an explanation

[1] It is assumed that the reader of this thesis has some background in system identification and control theory.

of different steps of the general subspace algorithm, some different implementations of the algorithm, and previous work of other researchers using subspace algortihms with closed-loop data.

Simulation part

- Chapter 5 consists of a discussion of the consequences caused by the most imporant issues from the theory part, and in addition an introduction to the simulation part of the thesis. It is in this chapter the problem description of the thesis is justified.
- In chapter 6, multivariable systems in closed loop with different characterizations are simulated in MATLAB. Identification experiments are performed in order to generate data for closed-loop identification using different subspace-methods.
- Chapter 7 discusses an industrial case study, in which a simulated debutanizer with cascaded PI composition control is subject for identification in closed-loop.

Discussion and conclusion part

- Chapter 8 will give a discussion on the results obtained in the simulation part. This includes how these results match up with results from other researchers, and a discussion on results that might contribute to answer the problem description of the thesis.
- Chapter 9 concludes the thesis by summarizing the most important results obtained, and giving suggestions to further work to be done on the topic.

1.2 Identification for control

In the field of engineering cybernetics, a mathematical model is used as a tool to develop *model-based controllers*. Examples of controllers that require a model are Model Predictive Controllers (MPC), Linear Quadratic (Gaussian) Regulators (LQR/LQG), $\mathscr{H}_2$ and $\mathscr{H}_\infty$-controllers. In addition, in order to find good tuning parameters for conventional P/PI/PID-controllers, some information from the system to be controlled is required (e.g. a first order model approximation). The term *identification for control* stems from the need of a model for either tuning, synthesis, updating or realization of these controllers. As pointed out by Ljung (1999), control design is actually one of the most important uses of identified models. When feedback control is used, a model that is reliable around the bandwidth frequency of the closed-loop system is important. The model may be mediocre for other frequencies, due to the "forgiving effect" of feedback control. This corresponds to the frequencies where the closed-loop sensitivity function is small.

Identification for control is also suitable for feed-forward design, i.e. identification of disturbance models. This requires knowledge of the disturbance effects, that is measurements of the disturbances.

1.3 Identification in closed-loop

In many cases the system to be identified is already operating under the presence of feedback, and controlled by some controller. The classical methodology is to break the feedback loop of the system, and perform an open-loop experiment on the system. Consider figure 1.1: The experiment design will involve manipulation of the variable u_m in order to identify the dynamics from u_m to y. One of the

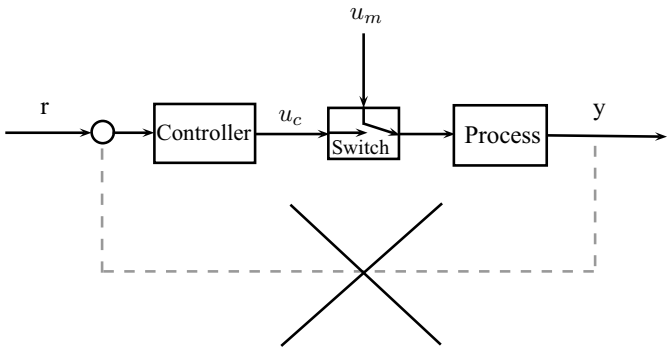


Figure 1.1: Feedback loop broken before identification experiment

main reasons why open-loop experiments are preferred is the lack of correlation between process- and measurement noise, and the input u to the system. Under closed-loop operation, the input to the process is the controller command u_c, which is a function of the output(s) of the system. Hence, the disturbances that affect the output variables y will also affect the controller command u_c through the feedback loop. A lot of the identification methods that are widely used today assumes output data that is uncorrelated with noise and disturbances.

Motivation for closed-loop identification

The classical procedure of system identification is shown in figure 1.2. This shows that some prior knowledge of the system to be identified is required to perform a successful identification experiment. This applies to e.g. choice of model set and experiment design. In fact, there are many systems where a bit of prior knowledge tells us that it is *practically impossible* to break the feedback loop in order to identify the open-loop dynamics. This yields in particular for processes with unstable behavior, and systems with inherited feedback effects (e.g. economic systems). It may also, in many cases, be too high a risk to break feedback loops for identifi-

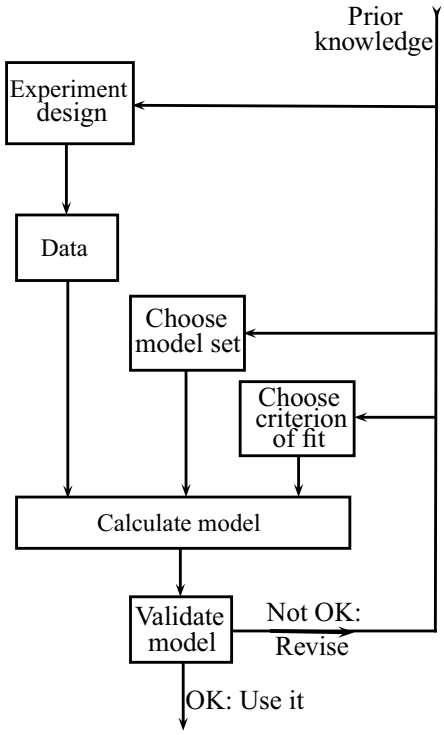


Figure 1.2: The identification procedure loop from Ljung (1999)

cation purposes, since the outputs may drift away from their nominal operating points during the identification experiment due to unknown disturbances.

Consider also a potential problem when a large process with many inputs and outputs is subject for identification. If the identification experiment is supposed to be performed in open-loop, is it even possible to predict what happens when all feedback loops are broken simultaneously? Also, if one feedback loop is broken at a time, and the process is interactive, the open-loop identification experiment may fail because the system characteristics change when one feedback loop is opened and others are running. An particular example is given in Jacobsen & Skogestad (1993), where an ill-conditioned plant with high steady-state RGA-values is subject for identification. It is shown that the prediction of process behavior is particularly poor under partial feedback control.

When a system/process is identified in *closed-loop*, the open-loop dynamics of the given system is estimated while the controllers already present are calculating the input u to the system. When a process G_0 is subject to feedback control by a controller C, the degree of freedom for the user to excite the system is usually moved from u to the controller reference signal r. Consider the block scheme in figure 1.3: Here the reference signal is denoted r_2, and an additional external input signal r_1 is directly added to the system input u. The signal r_1 may be a

controllable input to be used for identification purposes, but it may also consist solely of unknown disturbance effects that makes the controller command signal differ from the process input u, rendering an unknown process input. Throughout this thesis, it is assumed that this signal is an unknown signal in the form of process noise, and the reference r_2 is instrumental to externally excite the system dynamics. The common approach to closed-loop identification is to generate an

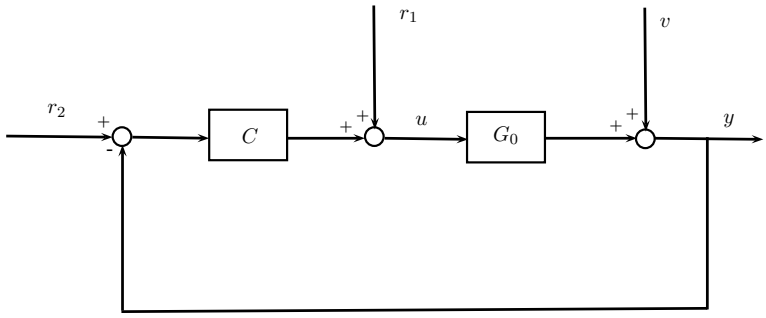


Figure 1.3: A system operating in closed-loop

input sequence to the system by varying the reference signal, and collect u and y data from the system to be used for identification. This will be further discussed in chapter 3.

1.4 State-of-the-art identification techniques

A traditional and well-known technique of system identification is the family of *prediction error identification methods* (PEM). Shortly explained, these are methods that find a parametrized model that minimizes the error between system output y and the predicted output $\hat{y}$ produced by some candidate models. This method of identification is an iterative approach, relying upon the solution of non-convex optimization problems. When the system to be identified is multivariable and has a complex structure, Viberg (1995) gives arguments that favors the use of *state-space models*. Even though PEMs are easily adapted to work with state-space models, the numerical optimization required to calculate the optimal estimate may be an impractically large problem. However, the PEM methods do cope well with closed-loop data, according to Ljung (1999) p. 430.

Since the late eighties a system identification technique based on linear algebra has emerged, known as *subspace identification*.[2]. A great advantage of subspace identification methods is that they are non-iterative, using well-understood algorithms with good numerical properties. They are also known to cope excellent with large data sets, rendering it possible to identify large systems in a fair

[2]The general subspace algorithm is given in appendix A, and a walk-through of the different steps in the algorithm is given in chapter 4.

amount of time. An example of this is given in Ben C. Juricek (2001), where the famous Tennessee Eastman challenge process is subject for identification. It is shown here that the only successful models are the state-space models produced by two different subspace algorithms.

Other more exotic approaches to system identification have also been investigated during the two last decades.

Kristinsson & Dumont (1992) use a *genetic algorithm* to identify a pole-zero model to be used by a pole placement adaptive controller. A closed-loop approach using a genetic algorithm is given in Whorton (2004). A great advantage of the genetic algorithm is that it is more likely to converge to a global optimum in the parameter space than gradient-following optimizers used with the ordinary PEM methods. A drawback is the computational effort required to run these algorithms, which may become impractically large for complex systems.

Fliess & Sira-Ramirez (2008) investigates the use of non-commutive ring theory and operational calculus in a closed-loop identification framework, identifying a double-bridge buck converter under bang-bang control. The investigated method was shown to be robust with respect to noisy data. The drawback is the lack of convergence analysis of the algorithm used with this identification technique.

1.5 The contributions provided by this thesis

In this thesis, a quantitative approach is taken to investigate the performance of three different subspace methods when closed-loop data is applied. The systems that are subject for investigation here are all multivariable, and operating under feedback from decentralized PI controllers.

The main contribution of the thesis is investigations on the performance of these subspace methods *when the controller parameters are varied.* Closed-loop identification of multivariable systems seems to the author as a pretty new research area, in particular analysis of the performance of identified models for different controller parameters when noise is present on both measurements and system inputs. In fact, no literature on this particular field is known to the author.

In order to investigate the significance of controller parameters in this closed-loop identification framework, both linear systems with different characteristics and a nonlinear simulated debutanizer process are studied.

The goal is not to find optimal tuning parameters for the controllers before an identification experiment, but to show the significance of controller tuning parameters. Insight into this could hopefully contribute to explain why closed-loop experiments in some cases fail, and provide knowledge on how the controllers should be tuned to give better identification conditions.

CHAPTER 2

BACKGROUND THEORY ON SYSTEM IDENTIFICATION

2.1 Linear state space models

A linear system, generally multiple inputs and multiple outputs (MIMO), may be described in state space form by

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} \end{aligned} \tag{2.1}$$

The discrete version reads

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A_d\boldsymbol{x}_k + B_d\boldsymbol{u}_k \\ \boldsymbol{y}_k &= C_d\boldsymbol{x}_k + D_d\boldsymbol{u}_k \end{aligned} \tag{2.2}$$

where the system quadruple (A, B, C, D) of (2.1) is discretized according to Chen (1999), p. 92. In the continuous case, $\boldsymbol{x}$ denotes states, $\boldsymbol{u}$ inputs, $\boldsymbol{y}$ outputs and $\dot{\boldsymbol{x}}$ the time derivative of $\boldsymbol{x}$. The frequency domain representation of (2.1) is obtained by taking the Laplace transform:

$$\begin{aligned} s\boldsymbol{x}(s) &= A\boldsymbol{x}(s) + B\boldsymbol{u}(s) \\ (sI - A)\boldsymbol{x}(s) &= B\boldsymbol{u}(s) \\ \boldsymbol{x}(s) &= (sI - A)^{-1}B\boldsymbol{u}(s) \\ \Rightarrow \boldsymbol{y}(s) &= C(sI - A)^{-1}B\boldsymbol{u}(s) + D\boldsymbol{u}(s) \\ &= \underbrace{\left(C(sI - A)^{-1}B + D\right)}_{\boldsymbol{G}(s)}\boldsymbol{u}(s) \end{aligned} \tag{2.3}$$

The relationship between $\boldsymbol{u}$ and $\boldsymbol{y}$ hence is

$$\boldsymbol{G}(s) = C(sI - A)^{-1}B + D \tag{2.4}$$

For a system with n states, m inputs and p outputs we have

$$\begin{aligned} \dim(A) &= n \times n \\ \dim(B) &= n \times m \\ \dim(C) &= p \times n \\ \dim(D) &= p \times m \end{aligned} \tag{2.5}$$

An important note is that there exists infinitely many state space representations of (2.4), because we can always perform a *similarity transform* of (2.1):

$$\boldsymbol{x} = Q\bar{\boldsymbol{x}} \tag{2.6}$$

where Q is the matrix of eigenvectors (or linear combinations of them) of A. Inserting this into (2.1) yields

$$Q\dot{\bar{\boldsymbol{x}}} = AQ\bar{\boldsymbol{x}} + B\boldsymbol{u} \tag{2.7}$$

$$\Rightarrow \dot{\bar{\boldsymbol{x}}} = \underbrace{Q^{-1}AQ}_{\bar{A}}\bar{\boldsymbol{x}} + \underbrace{Q^{-1}B}_{\bar{B}}\boldsymbol{u} \tag{2.8}$$

$$\boldsymbol{y} = \underbrace{CQ}_{\bar{C}}\bar{\boldsymbol{x}} + D\boldsymbol{u} \tag{2.9}$$

The state-space model structure is an excellent way of describing multivariable systems, due to their structural simplicity. The stability properties and dynamics are governed by the A-matrix, while the system gain is reflected by the B- and C-matrices. The D-matrix reflects wether any of the inputs have a direct effect on the outputs.

2.2 Persistence of Excitation

In order to generate consistent estimates of model parameters in system identification, there are some requirements to the input sequence used in the identification routine. The most important property of the input signal is that it reveals information from the system at hand, i.e. excites the dynamics in the system relevant for control purposes. *Persistence of excitation of order n* of an input signal u is a property that guarantees that the signal cannot be filtered to zero by an $(n-1)$th-order moving average filter on the form

$$M_n(q) = m_1 q^{-1} + \cdots + m_n q^{-n} \tag{2.10}$$

A more formal definition is given in Ljung (1999):

Definition 1. *A quasi-stationary signal $\{u(t)\}$, with spectrum $\Phi_u(\omega)$, is said to be persistently exciting of order n if, for all filters on the form* (2.10)*, the relation*

$$|M_n(e^{j\omega})|^2\Phi_u(\omega) \equiv 0 \text{ implies that } M_n(e^{j\omega}) \equiv 0 \tag{2.11}$$

2.3 PRBS-signals

A celebrated way of generating input signals in the system identification literature is by using *Pseudo-Random Binary Signals* (PRBS-signals). These signals are periodic, deterministic signals with properties of white noise. A great advantage of these signals is that they are easy to implement in practice, making them suitable for real identification experiments. Figure 2.1 shows an example of a PRBS-

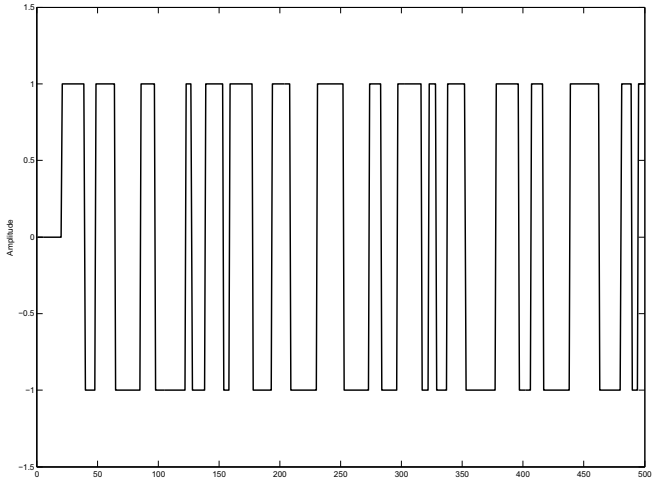


Figure 2.1: An example of a PRBS-signal

sequence. The signal is generated by the difference equation

$$u(t) = \text{rem}(A(q)u(t), 2) = \text{rem}(a_1 u(t-1) + \cdots + a_n u(t-n), 2) \tag{2.12}$$

The expression $\text{rem}(x, 2)$ means the remainder as x is divided by 2, and it assumed that the calculations are performed *modulo* 2. The value of u hence is either 0 or 1, but this value may of course be shifted after the signal is generated. Since the binary vector $\begin{bmatrix} u(t-1) & \dots & u(t-n) \end{bmatrix}$ only can assume 2^n different values, the sequence u must be periodic with a period of at most 2^n. The maximum period length is $M = 2^n - 1$ since n consecutive zeros would make further values of u zero, see equation (2.12). According to Ljung (1999) p. 421, PRBS-signals are persistently exciting of order $M - 1$.

2.4 Preconditioning of identification data

In order to generate good models using identification algorithms, it is imperative that the data used for identification is properly conditioned. It is not a given that

the data record collected is in shape for immediate use in the identification algorithm. Ljung (1999) mentions some possible deficiencies in the data that should be attended to

1. High-frequency disturbances in the data record, above the frequencies of interest to the system dynamics
2. Occasional bursts and outliers, missing data, non-continuous data records
3. Drift and offset, low-frequency disturbances, possibly of periodic character

In the case of off-line applications, Ljung (1999) recommends to always plot the data to investigate for these deficiencies.

Signal offsets

A dynamic model should be able to simulate the system dynamics of interest to the user. In a control setting, the steady-state properties are in many cases of minor importance, since integral action will take care of steady state errors. Consider a model on the form

$$A(q)y(t) = B(q)u(t) + v(t) \tag{2.13}$$

where q is the time delay operator. This model describes the dynamic properties between the input u and the output y. For constant signals $u(t) = \bar{u}$, the resulting steady-state value is noted $y(t) = \bar{y}$. The steady-state relation between the input and output hence is

$$A(1)\bar{y} = B(1)\bar{u} \tag{2.14}$$

The difference between (2.13) and (2.14) is that the first describes dynamic properties within different frequency bands, while the latter only describes the relation between signal magnitudes. These two equations have very little to do with each other, and according to Ljung (1999) equation (2.14) is an unecessary constraint for (2.13). To deal swith this problem, a reasonable approach is to subtract means from the data record. Let $y^{\text{raw}}(t)$ and $u^{\text{raw}}(t)$ be the raw measurements from the system. The data input to the identification algorithm will now be

$$y(t) = y^{\text{raw}}(t) - \bar{y} \tag{2.15}$$

$$u(t) = u^{\text{raw}}(t) - \bar{u} \tag{2.16}$$

where we calculate the near-equilibrium values $\bar{y}$ and $\bar{u}$ as

$$\bar{y} = \frac{1}{N}\sum_{t=1}^{N} y^{\text{raw}}(t) \tag{2.17}$$

$$\bar{u} = \frac{1}{N}\sum_{t=1}^{N} u^{\text{raw}}(t) \tag{2.18}$$

2.5 Metrics for evaluating model performance

In order to say something about model performance, it is imperative to define a performance measure metric. Oscar A.Z Sotomayor (2003) defines two different performance indicators, given by (2.19) and (2.20).

$$\text{MRSE} = \frac{1}{l}\sum_{i=1}^{l}\sqrt{\frac{\sum_{j=1}^{N}(y_i(j)-\hat{y}_i(j)^2}{\sum_{j=1}^{N}y_i(j))^2}} \tag{2.19}$$

$$\text{MVAF} = \frac{1}{l}\sum_{i=1}^{l}\left(1-\frac{\text{variance}(y_i-\hat{y}_i)}{\text{variance}(y_i)}\right) \tag{2.20}$$

The MRSE index given in equation (2.19) is used to measure the Mean Relative Squared Error between the real process outputs and the outputs produced by the model. As seen by equation 2.19, an MRSE index of 0 indicates a perfect model.

MVAF in equation (2.20) stands for Mean Variance Accounted For, and is a good measure for evaluating the dynamic properties of the produced models. If the ratio $\frac{\text{variance}(y_i-\hat{y}_i)}{\text{variance}(y_i)}$ is small, then the model has reproduced the dynamic properties of the real system well, and the MVAF index is close to 1.

CHAPTER 3

LITERATURE ON CLOSED LOOP IDENTIFICATION

The purpose of this chapter is to give a brief introduction to closed loop identification, and to summarize some of the research that has been done on closed loop identification for control.

3.1 The goal of closed-loop identification

A problem with a lot of the literature on closed-loop identification, is that the ultimate goal of the identification is not stated. This might lead to confusion about what system identification in closed loop really is. In order to illustrate this, consider figure 3.1. The goals might be different, i.e.

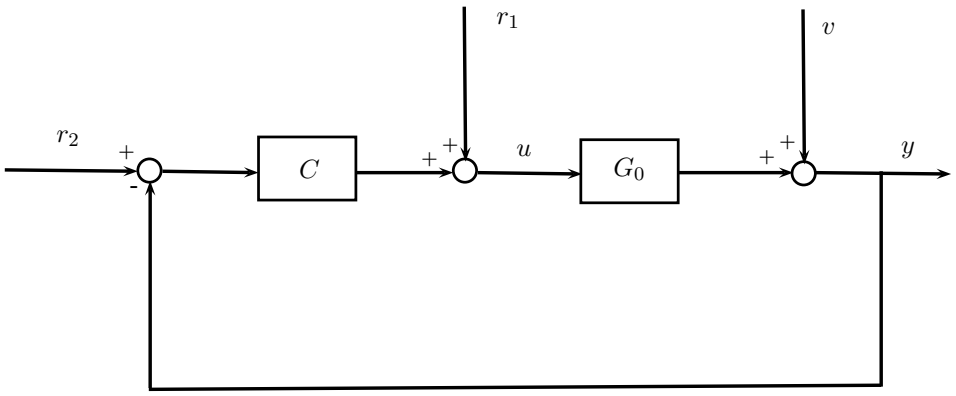


Figure 3.1: Closed-loop system

- Identification of the open-loop dynamics (estimate G_0) in order to tune the existing controller C using model-based tuning rules
- Identification of a disturbance model from the disturbance input signal r_1 to y in order to design a feed forward controller.
- Identification of the dynamics from the reference signal r_2 to y, i.e. identification of the transfer function $\frac{G_0C}{I+G_0C}$. This is a common approach, including the controller C in the loop and design a cascaded control system based on the derived model. Since the controller C now is a part of the system, the tuning of this controller will affect the identification. Possibly, it can render the model useless if the tuning of the controller C later is dramatically changed (changing the effective time constant of the system).
- Identification of the open loop dynamics with purpose of swapping the existing control structure with a new one.

The goal can also be combinations of these, i.e. estimating a disturbance model and the open loop dynamics with purpose of implementing an MPC controller, with or without the existing controller C in the bottom control layer. To avoid misunderstandings, the following definition is made to yield for this thesis:

Definition 2. *Closed-loop identification is defined as identification of the open-loop dynamics G_0 of figure 3.1 while the controller C tracks the reference signal r_2.*

3.2 The effect of closing controller loops

Consider a system G_0 with a controller C that has a zero reference, as depicted in figure 3.2. Suppose that the system G_0 is a simple stable first-order process,

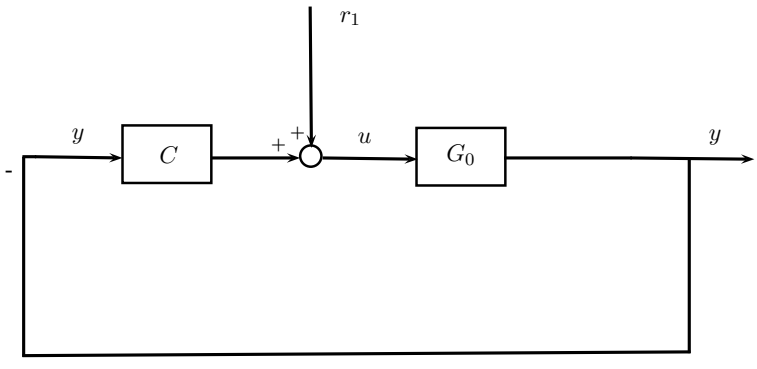


Figure 3.2: Closed-loop system without measurement noise and zero reference

described in state-space form as

$$\begin{aligned} \dot{x} &= -x + u \\ y &= x \end{aligned} \tag{3.1}$$

or $y = \frac{1}{s+1}u$ in the frequency domain. In addition, suppose that the controller C is a PI-controller

$$C(s) = K_c\left(\frac{\tau_i s + 1}{\tau_i s}\right) \tag{3.2}$$

The controller is tuned so that the closed loop time constant matches the open loop time constant, and so that the closed-loop gain equals the open-loop gain by using e.g. the SIMC tuning rules from Skogestad & Postlethwaite (2005). The tuning parameters are $K_c = \tau_i = 1$. In this case, the system gain and time constant are known parameters, and the controller is tuned based on this knowledge. Suppose that an impulse signal r_1 is injected into the closed-loop system. Now, the effect of closing the feedback loop is that that the open loop impulse response formerly seen on the output y *is moved* to the controller command signal u, with opposite sign because of the negative feedback. Figure 3.3 shows a simulation of the open loop impulse response on y and the controller command u in closed loop when the impulse signal is injected as shown in figure 3.2. The closed loop impulse re-

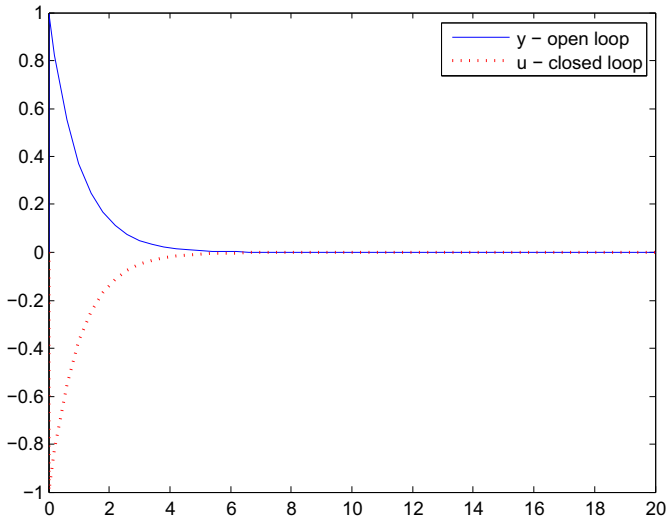


Figure 3.3: Impulse response moved from y in open-loop to u in closed loop

sponse on u will be a function of the controller tuning parameters. In this case it is possible to identify the open-loop dynamics G_0 solely from closed-loop u-data, but only because the controller is tuned based on system parameters assumed known beforehand. If the purpose is to determine the unknown transfer function

G_0, both u- and y-data are generally required. However, there are some fallacies using these data for identification purposes, as discussed in the next section.

3.3 Known issues with closed-loop data

There are some problems with closed loop identification that are not present in the open loop case. In this section some of the classical problems with closed-loop data will be presented. First of all, a very important aspect is that the very purpose of feedback is to make the closed loop system less sensitive to changes in the open loop system. Hence closed loop data typically has less information about the open loop system than open loop data has.

Effect of controller tuning

Rivera (1998) discusses how the controller tuning affect the information content of the controller command u. It is shown that aggressive tuning attenuates the low-frequency portion of the input signal, and amplifies the high-frequency portion. If the tuning is similar to open-loop (as in section 3.2), the controller will not introduce substantial bias into the input signal. Ljung (1999) mentions a scenario where the structure of the controller is *too simple* to identify the open-loop dynamics, even with persistently exciting controller commands. This shows that persistence of excitation of the controller command u is *not* a sufficient condition for closed-loop experiments.

Model structure requirements

Consider the feedback structure in figure 3.2, and suppose that u- and y-data is collected for identification purposes. I. Gustavsson (1977) mentions a persistent problem with identification methods that do not assume a causal model structure, namely the danger of identifying *the inverse of the controller*. This may be viewed as identifying the system from $-y$ to u through the the controller C, calculating the transfer from u to y as $-\frac{1}{C}$. This implies that the controller command at a specific time instant is known in advance of the output from the process at that specific time instance, which is improper. For identification methods based on *spectral analysis*, Ljung (1999) shows that the estimate of the open-loop transfer function will converge to

$$\hat{G}_0(j\omega) = \frac{G_0(e^{j\omega})\Phi_r(\omega) - C(e^{-j\omega})\Phi_v(\omega)}{\Phi_r(\omega) + |C(e^{j\omega})|^2\Phi_v(\omega)} \tag{3.3}$$

where $\Phi_r(\omega)$ and $\Phi_v(\omega)$ are the frequency spectra of the reference signal and noise signal respectively. If we assume that either the reference signal is identically zero, or that the magnitude of the controller is large compared to the system, the estimate of G_0 will converge to

$$\hat{G}_0(e^{j\omega}) = \frac{-C(e^{-j\omega})\Phi_v(\omega)}{|C(e^{j\omega})|^2\Phi_v(\omega)} = \frac{-C(e^{-j\omega})}{C(e^{-j\omega})C(e^{j\omega})} = -\frac{1}{C(e^{j\omega})} \tag{3.4}$$

This may be avoided by postulating a causal model structure to be identified, e.g. a state space structure.

Correlation as a consequence of feedback

The most famous argument against closed-loop identification is that model estimate becomes biased due to the correlation between disturbances and controller command u induced by feedback. This is mentioned in Forssell & Ljung (1999), and it is stressed that there always is a correlation between unmeasurable noise (both process and measurement noise) when the feedback mechanism is not identically zero. Standard subspace identification algorithms are mentioned as an example of methods that might fail when closed-loop data is used. This yields in particular for the methods that utilize the IV method, where it is assumed that the noise is uncorrelated with the instrumental variables. Since the instrumental variables usually are formed by past input and output data, and this data is corrupted by correlation to the process and measurement noise in closed loop, the assumption made in the subspace algorithm fail for closed-loop data. This does not necessarily mean that these methods will not work at all, but they are likely to give biased estimates of the open-loop transfer function.

Quality of the open-loop transfer function estimate

Ljung (1999) defines the covariance of the open-loop transfer function estimate as

$$\operatorname{Cov} \hat{G}_0 = \frac{n}{N} \frac{\Phi_v(\omega)}{|S_0|^2 \Phi_r(\omega)} \tag{3.5}$$

where S_0 is the sensitivity transfer function (the transfer from r_1 to u on figure 3.2), and Φ_v and Φ_r are the power spectra of the noise and reference signals respectively. n is the model order while N is the number of data. Equation (3.5) is, according to Ljung (1999), the asymptotic *Cramér-Rao lower limit* (CRLL), see Erlendur Karlsson (2000). This means that it tells us precisely the "value of information" of closed loop experiments. This may also be interpreted as the "noise to signal ratio", where lower noise to signal ratio gives lower uncertainty of the estimate of G_0. It is stated that the part of the input that originates from the feedback only (zero reference), has no information value when estimating the open-loop transfer function. Hence, it is the part of the input that stems from amplification of the reference signal that will reveal information from the system.

3.4 Motivation for performing closed-loop experiments

There are also benefits from performing identification experiments in closed-loop.

- It eliminates the need to put the control loop in manual during the identification experiment

- den Hof (1998) mentions that controllers may linearize possibly nonlinear plant behavior around a relevant operating point, enabling accurate linear modelling around this operating point.
- Closed-loop operation makes it possible to perform identification while keeping the plant within operating limits

The pioneering work of Ziegler & Nichols (1942) is actually based on identification in closed loop by tuning of the controller. By increasing the controller gain until the output response is persistently oscillating to a reference step, the controller may be tuned using the period of oscillation and the gain which produced the oscillations. This is maybe the most famous set of tuning rules in the control community, but not all are aware of the fact that this is a closed-loop identification experiment.

3.5 Approaches to closed-loop identification

When the feedback loop is closed, a number of different strategies may be applied in order to estimate the open-loop characteristics. The most common three will be discussed here.

The Direct Approach

Identification under closed loop using the so called ”direct approach” involves that issues with feedback is ignored, and the estimation is done using unaltered input/output signals. Hence, this is a simple approach. Ljung (1999) mentions a number of advantages with this approach:

- It works regardless of the complexity of the regulator, and requires no knowledge about the character of the feedback
- No special algorithms and software are required
- Given that the model structure contains the true system, consistency and optimal accuracy are obtained

The main problem with the direct approach is that the estimate may be biased due to correlation between disturbances and controllable inputs, see e.g. Katayama & Tanaka (2007).

The Indirect Approach

The indirect approach of closed loop identification assumes that the controller transfer function is known. The idea is to identify the closed loop transfer function

$$G_{\text{cl}} = \frac{GC}{1+GC} \tag{3.6}$$

by manipulating the reference signal (r_2 of figure 3.1). Since this is a open-loop problem, all the identification techniques that work for open-loop data may be applied. The drawback is that this approach demands a linear time-invariant controller. In industrial practice, this method is strongly disfavored due to nonlinearities that almost always exist in the controllers. Delimiters, anti-resetwindup functions and other nonlinearities are reasons why the controllers deviate from the LTI-form. In addition, estimates of the plant by the indirect approach are of higher order, and some model reduction procedure is needed.

The Joint Input-Output Approach

It is possible to view a closed-loop system as a system with input r, and two outputs u and y. The system is driven by the reference, producing outputs in the form of controller outputs and process outputs. The *joint input-output* technique use models of how both u and y are generated. Consider again figure 3.1, and define the transfer functions

$$G_{\text{ry}}(s) = \frac{G_0 C}{1 + G_0 C} = \text{the transfer from } r \text{ to } y \tag{3.7}$$

$$G_{\text{ru}}(s) = \frac{C}{1 + G_0 C} = \text{the transfer from } r \text{ to } u \tag{3.8}$$

By perfoming identification experiments and finding estimates $\hat{G}_{\text{ry}}$ and $\hat{G}_{\text{ru}}$, the open-loop transfer function may now be estimated as

$$\hat{G}_0 = \frac{\hat{G}_{\text{ry}}}{\hat{G}_{\text{ru}}} \tag{3.9}$$

Equations (3.8) and (3.7) shows that the denominator of G_{ry} and G_{ru} is equal, and ideally these should cancel out when performing the calculation in (3.9). The problem is that even small estimation errors from the identification of G_{ry} and G_{ru} will prevent this cancellation, since the estimates $\hat{G}_{\text{ry}}$ and $\hat{G}_{\text{ru}}$ will have slightly different denominators. A solution to this is to use e.g. the *normalized coprime factor method*, proposed by den Hof et. al (1995) to perform a model-reduction on the estimate $\hat{G}_0$.

CHAPTER 4

SUBSPACE IDENTIFICATION

During the two last decades the subspace identification method has gained much attention in the identification research community. There are several reasons for this, the most important being the fact that the algorithms spawned from this identification technique are non-iterative, removing any possible convergence problems. They also identify state space models directly, which is a well-understood model structure with numerous nice features. The numerical computations are taken care of by well-understood algorithms from linear algebra, such as the QR-decomposition and SVD. This assures computational effiency. As pointed out by Katrien De Cock (2003), this makes the subspace technique suitable for identification using large datasets. In this section, the different concepts and mathematical tools that make up the family of subspace identification algorithms will be revised. The theory in this section yields for combined deterministic and stochastic identification. Of special interest is the research that has been done on closed loop identification using subspace methods.

4.1 The basic idea behind subspace algorithms

The family of subspace algorithms seek to estimate the system matrices A, B C and D of the state space model from input-output data of the system. The subspace identification algorithm in its most general form is given in appendix A. All subspace identification methods consist of three main steps:

1. Estimating the predictable subspace for multiple future steps
2. Extract the state variables from the estimated subspace
3. Fitting the estimated states to a state space model (Calculating the matrices A_d, B_d, C_d and D_d of (2.2)).

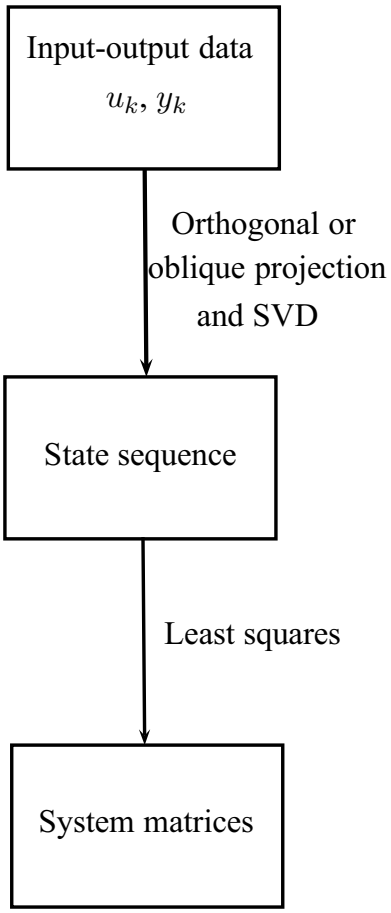


Figure 4.1: The subspace method

4.2 Block Hankel matrices

Socalled *block Hankel matrices* are important in subspace identification algorithms. They are constructed solely from recorded input-output data, see Katrien De Cock (2003):

$$U_{0|2i-1} = \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \dots & u_{j-1} \\ u_1 & u_2 & u_3 & \dots & u_j \\ \vdots & \vdots & \vdots & \dots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \dots & u_{i+j-2} \\ \hline u_i & u_{i+1} & u_{i+2} & \dots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \dots & u_{i+j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \dots & u_{2i+j-2} \end{array}\right] = \left(\frac{U_{0|i-1}}{U_{i|2i-1}}\right) = \left(\frac{U_p}{U_f}\right) \tag{4.1}$$

$$= \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \dots & u_{j-1} \\ u_1 & u_2 & u_3 & \dots & u_j \\ \vdots & \vdots & \vdots & \dots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \dots & u_{i+j-2} \\ u_i & u_{i+1} & u_{i+2} & \dots & u_{i+j-1} \\ \hline u_{i+1} & u_{i+2} & u_{i+3} & \dots & u_{i+j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \dots & u_{2i+j-2} \end{array}\right] = \left(\frac{U_{0|i}}{U_{i+1|2i-1}}\right) = \left(\frac{U_p^+}{U_f^-}\right) \tag{4.2}$$

$$Y_{0|2i-1} = \left[\begin{array}{ccccc} y_0 & y_1 & y_2 & \dots & y_{j-1} \\ y_1 & y_2 & y_3 & \dots & y_j \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{i-1} & y_i & y_{i+1} & \dots & y_{i+j-2} \\ \hline y_i & y_{i+1} & y_{i+2} & \dots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & y_{i+3} & \dots & y_{i+j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{2i-1} & y_{2i} & y_{2i+1} & \dots & y_{2i+j-2} \end{array}\right] = \left(\frac{Y_{0|i-1}}{Y_{i|2i-1}}\right) = \left(\frac{Y_p}{Y_f}\right) \tag{4.3}$$

$$= \left[\begin{array}{ccccc} y_0 & y_1 & y_2 & \dots & y_{j-1} \\ y_1 & y_2 & y_3 & \dots & y_j \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{i-1} & y_i & y_{i+1} & \dots & y_{i+j-2} \\ y_i & y_{i+1} & y_{i+2} & \dots & y_{i+j-1} \\ \hline y_{i+1} & y_{i+2} & y_{i+3} & \dots & y_{i+j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{2i-1} & y_{2i} & y_{2i+1} & \dots & y_{2i+j-2} \end{array}\right] = \left(\frac{Y_{0|i}}{Y_{i+1|2i-1}}\right) = \left(\frac{Y_p^+}{Y_f^-}\right) \tag{4.4}$$

The subscripts p and f means "past" and "future" respectively. The number of block rows i is a user-defined index which should be larger than the maximum order of the system to be identified. Since the identified system could be multi-variable, each block row contains m inputs, and the matrix $U_{0|2i-1}$ contains $2mi$ rows.

The number of columns j is typically $s-2i+1$, where s is the number of samples in the data record. The asymptotic requirement for estimating "perfect" noise models is that $s \to \infty$, but this is of course impossible in practice. For purely deterministic models the only requirement is that the input is persistently exciting of the same order as the model to be identified.

4.3 Geometric tools

In the following we assume that the matrices

$$A \in \mathbb{R}^{p\times j},\ B \in \mathbb{R}^{q\times j},\ C \in \mathbb{R}^{r\times j}$$

are given. The elements of a row of one of these given matrices can be considered as the coordinates of a vector on the j-dimensional *ambient space*. The rows of A, B and C thus defines a basis for a linear vector space in this ambient space. The operator Π_Γ is defined to have the property of projecting the row space of a matrix onto the row space of the matrix Γ,

$$\Pi_\Gamma = \Gamma^\top(\Gamma\Gamma^\top)^\dagger\Gamma \tag{4.5}$$

where $(\Gamma\Gamma^\top)^\dagger$ means the Moore-Penrose pseudo-inverse of the matrix $\Gamma\Gamma^\top$. The reason for this is that in general, the matrix $\Gamma\Gamma^\top$ may be singular.

Orthogonal projections

For simplicity, the notation

$$A/B = A\Pi_B = AB^\top(BB^\top)^\dagger B \tag{4.6}$$

is introduced, a shorthand for the projection of the row space of A onto the row space of B. An example of this projection when the ambient space is two-dimensional is given in figure 4.2. In addition we have the definition

$$A/B^\perp = A - A/B = A(I_j - B(BB^\top)^\dagger B) \tag{4.7}$$

The two projections Π_B and $\Pi_{B^\perp}$ decompose the matrix A into two matrices, of which the row spaces are orthogonal:

$$A = A\Pi_B + A\Pi_{B^\perp} \tag{4.8}$$

As proposed by Katrien De Cock (2003), these projections may be computed via QR decomposition of $\begin{pmatrix} B \\ A \end{pmatrix}$. This algorithm is described in detail in Strang (2006).

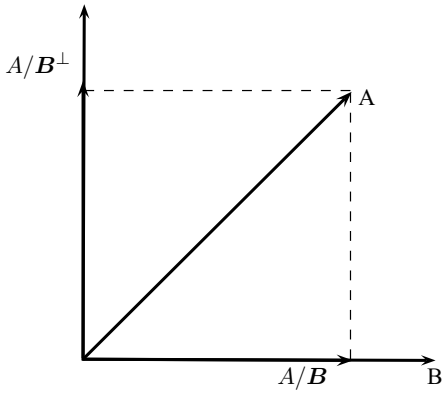


Figure 4.2: Orthogonal projection of a vector A in two dimensions

The purpose of using orthogonal (or oblique) projections is to remove noise from the data. In the general subspace algorithm stated in appendix A, it is shown that the projections are tools used to remove unknown terms in the identification objective equation. The use of these projections makes sure that an estimate of the matrix of *Markov parameters* (see equation A.3) is not needed.

Oblique projections

The orthogonal projection is achieved by decomposing the rows of matrix A as a linear combination of the rows of two orthogonal matrices, as in (4.8). Another possibility is to decompose the rows af A as a linear combination of the rows of two non-orthogonal matrices B and C, and of the orthogonal complement of these. The oblique projection of the row space of B into the row space of C is defined as

$$A/_B C = L_C C \tag{4.9}$$

and the equation for A is

$$A = L_B B + L_C C + L_{B^{\perp},C^{\perp}} \begin{pmatrix} B \\ C \end{pmatrix}^{\perp} \tag{4.10}$$

Figure 4.3 shows an example of an oblique projection of a vector A. This projection may also be computed numerically by QR decomposition, as proposed by Katrien De Cock (2003).

4.4 Calculation of the state sequence

With the geometric tools defined in section 4.3 and the block Hankel matrices at hand, it is possible to calculate the state sequence of the sought model in two

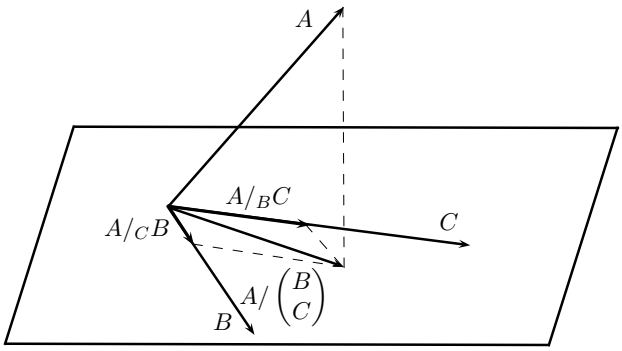


Figure 4.3: Oblique projection of a vector A in three dimensions

steps. First, the future output rowspace is projected along the future input row space into the joint row space of past inputs and past outputs

$$O_i = Y_f /_{U_f} \begin{pmatrix} U_p \\ Y_p \end{pmatrix} \tag{4.11}$$

$$O_{i-1} = Y_f^- /_{U_f^-} \begin{pmatrix} U_p^+ \\ Y_p^+ \end{pmatrix} \tag{4.12}$$

The following is required from the input-output data for these projections to give a perfect estimate of the state sequence:

(a) The process noise w_k and measurement noise v_k are uncorrelated with the input u_k

(b) The input u_k is persistently exciting of order $2i$, that is the input block Hankel matrix $H_{0|2i-1}$ is of full row rank.

(c) The sample goes to infinity, $j \to \infty$

(d) The process noise w_k and the measurement noise v_k are identically zero.

Note that it is possible to get a very good model even if these requirements are not completely fullfilled.

Second, an SVD is carried out to obtain the model order. This is shown in Henriksen (2001),

$$W_1 O_i W_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} = U_1 S_1 V_1^\top \tag{4.13}$$

By inspection of the magnitude of the singular values (diagonal components of S_1), it is possible to estimate the model order. This is usually the step where the

algorithm needs help from the user, by looking at a singular value plot and select an order that includes the dominant singular values.

The choice of two weighting matrices W_1 and W_2 is one of the most important distinctions between different subspace identification algortihms. This will be discussed further in section 4.6.

The extended observability matrix $\mathcal{O}_i$ may be calculated as

$$\mathcal{O}_i = U_1 S_1^{\frac{1}{2}} \tag{4.14}$$

Now, the state sequence and shifted state sequence may be estimated as

$$\tilde{X}_i = \mathcal{O}_i^{\dagger} O_i \tag{4.15}$$

$$\tilde{X}_{i+1} = \mathcal{O}_{i-1}^{\dagger} O_{i-1} \tag{4.16}$$

The matrix $\mathcal{O}_{i-1}$ denotes the matrix $\mathcal{O}_i$ without the last l rows, where l is the number of outputs of the system.

4.5 Calculation of the system matrices

The last step of the subspace identification routine is to estimate the system matrices A, B, C and D. This is done in a least squares sence, solving the following set of equations:

$$\begin{bmatrix} \tilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} \tilde{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \tag{4.17}$$

The matrices ρ_w and ρ_v are residual matrices, from which it is possible to estimate the covariances of the process and measurement noise. The two state estimates $\tilde{X}_i$ and $\tilde{X}_{i+1}$ however, is obtained with different initial conditions. This means that the set of equations (4.17) is not really theoretically consistent, giving slightly biased estimates of the system matrices. In order to guarantee unbiased estimates, one of the following conditions have to be satisfied:

1. $i \to \infty$
2. The system is purely deterministic, that is $v_k = w_k = 0 \, \forall \, k$
3. The input u_k is white noise

4.6 Different subspace methods

During the recent years a number of different subspace-algorithms have been developed. They all share the structure shown in figure 4.1, and utilize the mathematical and geometrical tools described in this section. One of the main differences between these is the choice of weighting matrices W_1 and W_2 used in the SVD, equation (4.13). There is also different ways of estimating the system matrices, either by using the state estimates or by using the extended observability

matrix $\mathcal{O}_i$. The general algorithm described in appendix A from Ljung (1999) uses the extended observability matrix to compute the system matrices. A handful of different realizations of the subspace algorithm will be discussed here.

MOESP

The Multivariable Output Error State Space (MOESP) family of subspace algorithms was proposed by Michel Verhaegen, and discussed in Verhaegen (1994) and Muharar (2006). As the name indicates, these algorithms assumes that the input-output relationship may be described in an *output-error* fashion, which means that the relationship between inputs and undisturbed outputs may be written as a linear difference equation. There are different kinds of MOESP algorithms (Muharar (2006)):

- *Ordinary MOESP* - Designet to cope with systems contaminated with white measurement noise only. Assumes that outputs are independent of past noise.
- *PI-MOESP* - Stands for "Past Input"-MOESP. This algorithm is designed to cope with systems contaminated by colored measurement noise (correlation between noise samples). Uses instrumental variables constructed from past inputs, hence the name.
- *PO-MOESP* - "Past Output"-MOESP. Different from the PI-MOESP algorithm, this algorithm cope with colored input noise and colored measurement noise. It uses both past inputs and past outputs to construct instrumental variables, so the name might be confusing.

All the MOESP algorithms use the extended observability matrix $\mathcal{O}_i$ to estimate the system matrices.

The weighting matrices used in the SVD in step 2 of the MOESP algorithm (see appendix A for an explanation of this step), are according to Ljung (1999)

$$W_1 = I \tag{4.18}$$

$$W_2 = (\frac{1}{N}\Phi\Pi^{\perp}_{\mathbb{U}^\top}\Phi^\top)^{-1}\Phi\Pi^{\perp}_{\mathbb{U}^\top} \tag{4.19}$$

The MOESP algorithm requires two inputs. The first is the embedded dimension used in the algorithm, that is the number of block Hankel rows used before the singular value decomposition (SVD). The second is the number of states of the final model, which is usually decided by inspection of the singular values from the SVD. An example where the two parameters of the MOESP algorithm are varied is shown in figure 4.4. The system to be identified here is a 2×2-plant, and the frobenius norm of the error between the real plant output and the outputs produced by the output is indicated on the z-axis. Since the MOESP algorithm assumes an output-error structure of the system to be identified, the algorithm typically produces good low order models for high signal-to-noise ratios in the identification

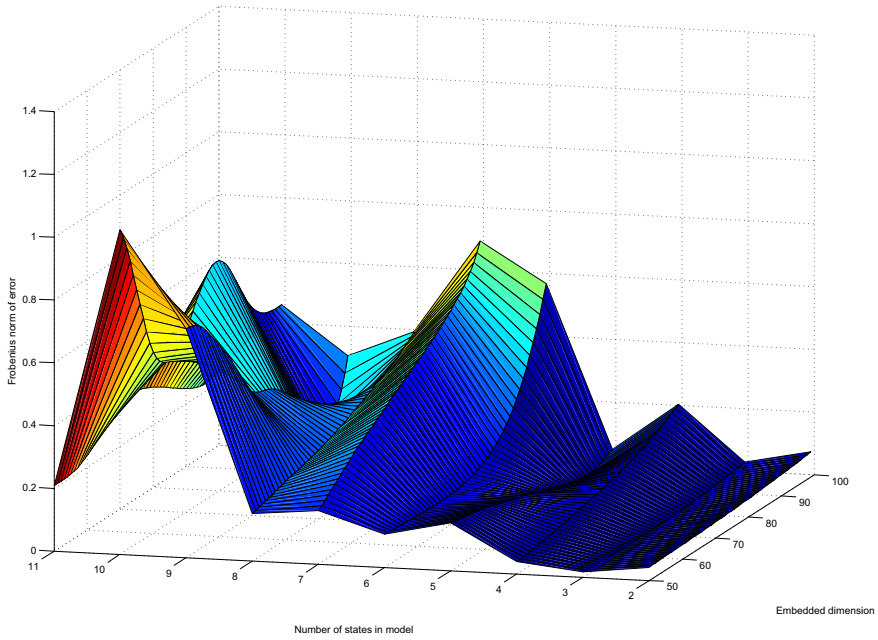


Figure 4.4: The error using the MOESP algorithm for varying embedded dimension and final states of the identified model. In this example, the best model was obtained by using an embedded dimension of 88, and 3 states in the final model.

data. This is also seen in figure 4.4, where the optimum appears for a third-order model. The reason for the low order is that the model structure is not trying to describe noise properties by adding more states to the model.

In this thesis, only deterministic identification is performed (only the A, B, C, D-quadruple, and no identifcation the noise properties). This means that the deterministic part of the system is to be found from both noise-free and noise-corrupted data.

The MOESP algorithm used in this thesis is the ordinary MOESP algorithm, and the implementation is made by Yi Cao at Cranfield University, UK.

N4SID

The *numerical algorithms for subspace state space system identification* (N4SID, read "enforce it") is a class of subspace algorithms first proposed by Overschee & Moor (1994). One of the main differences between the N4SID algortihms and the MOESP algortihsm, are the choice of weighting matrices used in the SVD step. The

general N4SID algoritms use the weighting matrices

$$W_1 = I \tag{4.20}$$

$$W_2 = (\frac{1}{N}\Phi\Pi^{\perp}_{\mathbb{U}^\top}\Phi^\top)^{-1}\Phi \tag{4.21}$$

It should be noted though, that the Matlab implementation of the algorithm offers the possibility to use weighting matrices similar to other subspace algorithms, like MOESP or CVA (Canonical Variate Analysis). In this thesis though, the original weighting is used in order to test the original algorithms against each other.

The ordinary N4SID algorithm is, unlike MOESP, not based on describing the input-ouput characteristic in an output-error form. A consequence of this is that N4SID tend to give more accurate models when a higher model order than MOESP is choosen during the identification routine. This is because the noise in the input-data is not removed before the state sequence is estimated, and a "full" state space model is estimated.

The only input parameter to the N4SID algorithm is the number of states in the final model, which makes it very convenient for easy identification of large systems. The number of states in the final model is derived in the same fashion as for the MOESP algorithm, by inspection of the singular values from the SVD step.

The N4SID algorithm used in this thesis is the version implemented in the Matlab Identification Toolbox. As stated above, this implementation is quite general, because it offers different choices of e.g. the weighting matrices of the SVD. The only input parameter given to the algorithm used in this thesis will be the number of states in the final model, and only deterministic identification is performed (as for the MOESP algorithm).

DSR and DSR_e

The DSR_e algorithm by David Di Ruscio is spawned from the combined deterministic and stochastic system identification and realization framework given in Ruscio (1996). This version of the subspace algorithm is designed to especially cope with closed-loop data. The general DSR algorithm uses orthogonal projections, see section 4.3.

The ordinary DSR algorithms are complete subspace methods, where both the deterministic part (A, B, C, D-quadruple) and the stochastic part (noise properties and kalman-gain) are estimated without solving the Riccati equation or using parameter error methods.

The DSR_e algorithm uses three input parameters, where one of them is related to the embedded dimension parameter of the MOESP algortihm. This parameter is denoted L, and the relation to the MOESP parameter is "$\mathrm{ED}_{\mathrm{MOESP}} = L + 1$", where ED means embedded dimension.

The second input parameter is a parameter J, that is the size of the past horizon used to define instrumental variables. This parameter should enforce the eigenvalues of $(A - BK)^J$ to approach zero, where K is the identified Kalman-gain.

J should hence be relatively large. This parameter is related to the stochastic part of the identification, but is also important to separate the noise components from the deterministic part of the model.

The third parameter is, as for the MOESP and N4SID algorithms, the number of states in the final model. This parameter is set after an inspection of the singular values from the SVD step, as for the other algorithms.

A brief explanation of the DSR_e method is given by David Di Ruscio, after personal communication with the author

> The DSR_e method is a two-stage method. In the first step, future inputs in the Hankel matrices are splitted into signal and noise. In order to achieve this, past inputs and outputs are used, where the horizon used is defined by the parameter J (which should be large). When the signal part of the data is found, the state space model is found by solving a deterministic subspace problem using the ordinary DSR algorithm.

4.7 Application of subspace methods using closed-loop data

In the standard subspace algorithm, one of the assumptions is that the noise in the data record is uncorrelated with the input data. In appendix A, the general subspace algorithm and criterions for good model identification are given. When the instrumental variables are formed to remove the noise terms in the data, it is to ensure that equation (A.14a) is satisfied. For noisy closed-loop data, the measurement noise as well as the input noise will be correlated with the system input through the feedback loop. This violates the assumption of uncorrelated input and noise data. Numerous researchers have proposed modified algorithms that tries to cope with this problem.

Verhaegen (1993) proposes a method that solves an open-loop problem by combining the models of process and controller, using a standard subspace identification method (MOESP) to solve the full scale problem. This approach also gives an estimate of the controller.

Subspace identification methods are primarily based on the estimates of k-step ahead output predictors. Ljung & McKelvey (1996) proposes the use of a high order ARX one-step ahead predictor model to recompute these k-step ahead output predictors. This way, closed-loop data may be handled. It is stressed by Ljung & McKelvey (1996) that the algortihm they proposed to cope with this problem is not the best in any sence, merely a feasible solution.

Ruscio (2003) shows that standard subspace identification algorithms (see appendix A) works perfectly with closed-loop data for purely deterministic systems, i.e. systems not corrupted by noise. An algorithm that uses the controller parameters in order to overcome the bias problem is proposed. In addition, the idea of

using a filter in the feedback loop to avoid that the process input is directly proportional to the measurement noise is introduced.

Katayama & Tanaka (2007) proposes a subspace identification method based on orthogonal decomposition for closed-loop identification. The idea of this approach is to project input and output data onto the space of exogenous inputs by using an LQ (similar to QR) decomposition to obtain the deterministic components of the system to be identified.

CHAPTER 5

PROBLEM STATEMENT AND METHODS

The purpose of this chapter is to introduce and justify the choice of methods used in the simulation parts (chapters 6 and 7), and to clearly state the questions to be answered in this thesis.

5.1 Choice of identification methods

The following subsections will justify the choice of identification techniques.

Closed-loop identification technique

The three different approaches to closed-loop identification discussed in section 3.5 use different methods of estimating the open-loop transfer function.

In order to keep the results produced in this thesis as practical applicable as possible, the *indirect approach* of closed-loop identification is ruled out. This is because knowledge of the controller parameters may not always be a sufficient condition for knowledge of the real controller. This approach is also limited to linear controllers only. Nonlinear effects like anti reset-windup and additional logic in the controllers may render the controller models useless for calculating the open-loop transfer function from equation (3.6).

As shown in section 3.5, the *joint input-output approach* has proven to perform well in numerous cases. The main problem with this method is the complexity, and the need to estimate two transfer functions instead of only one. When forming the transfer functions from the reference input to controller output, and from reference input to system output, a decision on the order of these transfer functions have to be made. This may be hard even when using singular value analysis during the subspace identification routine, since the controllers may require a high order approximation. After the model reduction step, this may leave a

simplified model where the controllers are dominating in the open-loop transfer function estimate. In addition, the inverse of the transfer function from reference inputs to controller outputs has to be calculated. If this transfer function is singular, or close to singular, the estimate will be very sensitive to estimation errors.

Being a simple approach, the *direct approach* is not using any knowledge of the controllers, operating solely on controller output and system output data. This strongly favors the use of this method when the purpose is to find an estimate of the open-loop transfer function only. If the purpose is to find a model of the controller as well, the direct approach may be used on controller inputs and outputs as well. In this thesis, the purpose is to find the open-loop transfer function only. Hence, the only approach to closed-loop identification will be the direct approach.

Choice of subspace methods

The subspace identification methods that will be compared are

- N4SID by Van Overschee and De Moor
- MOESP by Michel Verhaegen
- DSR_e by David Di Ruscio

The N4SID and MOESP methods are selected because they are among the most widely used subspace algorithms. MATLAB implementations of these algorithms are comercially available. As stated in section 4.6, these are primarly used with open-loop data. The DSR_e algorithm by David Di Ruscio is available on request. This algorithm is tailored to especially cope with closed-loop data, see section 4.6. The purpose is to investigate how the open-loop methods perform compared to the tailored closed-loop method. By comparing the methods, a better understanding of how the different methods work is sought.

5.2 Choice of simulated control structure

One of the most common control strategies in the industry is to use single loop PI-control. This means that each control loop is closed individually regardless of the multivariable nature of the system. As a consequence of this, the identification problem very often consists of finding the open-loop dynamics of a system operating under decentralized PI-control. It is also natural that one wants to replace such a control structure with e.g. MPC or LQR/LQG, that naturally handles multivariable systems.

The systems to be identified in the next chapters will hence operate under decentralized PI-control. To keep the structure relatively simple, the systems (or plants) discussed in chapters 6 and 7 will be restricted to 2×2-systems, controlled by a diagonal PI-controller.

A block diagram of the system structure used in chapter 6 is shown in figure 6.1. The focus of this chapter will be to identify systems with different dynamic properties, with and without measurement noise and unmeasured input disturbances.

In chapter 7, a more practical case study is revised. Here, a nonlinear plant is subject for identification. The focus here will be to identify a linear model around an operating point in closed-loop, with measurement and process noise present in the loop.

5.3 Problem statement

The main purpose of this thesis is twofold; First, the objective is to investigate the performance of the three subspace methods N4SID, MOESP and DSR_e when closed-loop data is applied. In order to indicate the performance of the different methods, the performance indices defined in section 2.5 will consequently be used. This will be studied in both chapter 6 and 7.

Second, these methods will be instruments to investigate how tuning of the controllers affect the identified open-loop model. This is also studied in both chapter 6 and 7.

CHAPTER 6

IDEALIZED CASE STUDIES

This chapter presents simulations and subspace identification of different systems using Matlab and Simulink. Three different subspace algorithms will be tested, namely

- DSR_e by David Di Ruscio
- MOESP by Michel Verhaegen
- N4SID by Van Overschee and De Moor

The performance of the subspace methods will be evaluated by the performance indices defined in section 2.5. The results are listed in appendix B, tables B.1 and B.2.

6.1 The system structure and identification configurations

As stated in section 5.2 , the systems at hand will all be linear and multivariable (2×2). The control structure is in all cases a decentralized PI-structure, and the system is depicted in figure 6.1. The controller equations are

$$C_1(s) = K_{p1}\frac{\tau_i s + 1}{\tau_{i1} s} \tag{6.1}$$

$$C_2(s) = K_{p2}\frac{\tau_i s + 1}{\tau_{i2} s} \tag{6.2}$$

The goal is to find an estimate $\hat{G}_0(s)$ of the matrix

$$G_0(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} \tag{6.3}$$

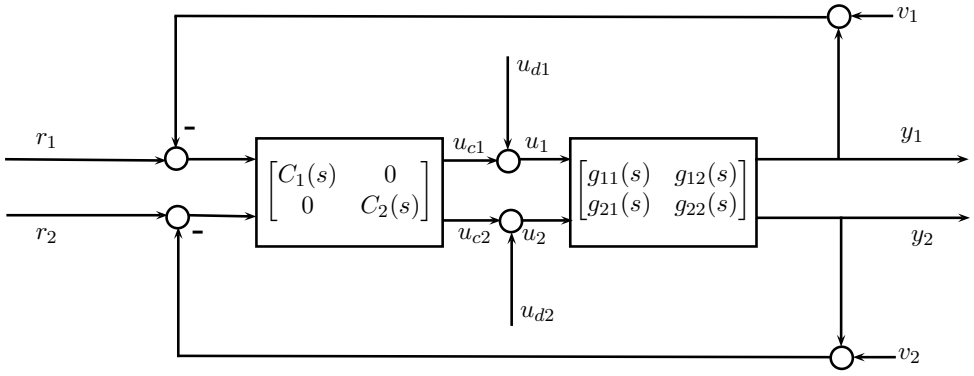


Figure 6.1: Closed-loop identification structure

The different signals in figure 6.1 are defined as follows[1]:

- $\{r_1, r_2\}$ - These are reference signals to be tracked by the controller. They are used to excite the system for identification in closed-loop.

- $\{u_{c1}, u_{c2}\}$ - Control signals calculated by the controller. These equal the signals $\{u_1, u_2\}$ in the special case of noise-free inputs.

- $\{u_{d1}, u_{d2}\}$ - These are unmeasured, external signals entering the closed-loop system. Due to these, the controller command u_c will differ from the input to the system u. These signals will be referred to as *process noise* signals.

- $\{u_1, u_2\}$ - The actual inputs to the system. These are the sum of $\{u_{c1}, u_{c2}\}$ and $\{u_{d1}, u_{d2}\}$.

- $\{v_1, v_2\}$ - Measurement noise entering the feedback loop. This signal may be a mixture of low-frequency and high-frequency components, i.e. measurement bias/drift and high-frequency noise.

- $\{y_1, y_2\}$ - Outputs of the system. Note that these are different from the measurements, since the measurements are in general corrupted with noise, while these are the "real" outputs.

The controller C in figure 6.1 is a socalled one degree-of-freedom controller, since the controller is operating on a single error signal (no feed-forward). It can be useful to define some transfer functions for the system at hand. First, we observe that the input to the process is $u = u_c + u_d$, and the controller command is given by

[1] In compressed notation, all the indices of the variables are dropped, e.g. $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$

the controller $u_c = C(r-y-v)$. Second, the open-loop transfer is given by $y = G_0 u$. By insertion, we get

$$\begin{aligned} y &= G_0 u = G_0(u_c + u_d) = G_0(C(r-y-v)) + G_0 v \\ &= G_0 C r - G_0 C y - G_0 C v + G_0 u_d \\ (I + G_0 C) y &= G_0 C r + G_0 u_d - G_0 C v \\ y &= \underbrace{(I+G_0C)^{-1} G_0 C}_{T} r + \underbrace{(I+G_0C)^{-1}}_{S} G_0 u_d - \underbrace{(I+G_0C)^{-1} G_0 C}_{T} v \\ &= Tr + SG_0 u_d - Tv \end{aligned} \tag{6.4}$$

The transfer functions S and T are defined as

$$S = (I + G_0 C)^{-1} : \text{ the } \textit{sensitivity} \text{ function} \tag{6.5}$$

$$T = (I + G_0 C)^{-1} G_0 C : \text{ the } \textit{complementary sensitivity} \text{ function} \tag{6.6}$$

Notice that the unmeasured disturbance u_d is filtered through the open-loop transfer function G_0 and the sensitivity function S. Now, the smaller the sensitivity function is in magnitude, the more will the unmeasured disturbance affect the process input u instead of the output y (recall section 3.2). If the controller gain is high, the magnitude of the sensitivity is reduced, and the disturbance effect from u_d on the output y will be reduced also.

6.2 Procedure

The identification procedure for identifying the different test systems will be explained. This includes input design, subspace algorithm parameter choices, and validation experiment.

Input design

Theorem 13.2 of Ljung (1999) states that a closed loop experiment is informative if and only if the reference signal r is persistently exciting. It is also mentioned that whatever choice of input experiment that is easier to implement accuratly is to be preferred. Due to these reasons, PRBS-signals will be generated in order to identify the closed-loop systems in this chapter. As mentioned in section 2.3, these signals are persistently exciting of a relatively high order (this order is given by the maximum period length M of the signal). These signals are also easy to generate, even for a real process. This is because the signal itself only consists of a series of steps up and down.

An implementation by David Di Ruscio of the algorithm from section 2.3 for generating PRBS-signals will be used. In this implementation, it is possible to adjust the minimum and maximum period of the signals. Some prior knowledge of the system to be identified is useful for deciding these parameters; if we know approximately how fast the slowest time constant of the system to be identified is,

we may set the maximum period of the PRBS-signal to be about four or five times this value to catch the steady-state properties. The minimum period parameter will be set to about ten times the sampling time of the system, in order to catch any possible fast time constants.

For all the systems in this chapter, the sampling time will be $0.1s$, and discrete time state space models are identified.

Since the systems to be identified in this section are known to be linear, it is possible to excite both references to the system at the same time. This is governed by the super position principle for linear systems, which let us sum up the contributions from different inputs to the systems, one at a time.

Subspace algorithm parameters

The three different subspace algorithms DSR_e, MOESP and N4SID require different input parameters. These are discussed in section 4.6. The number of states in the final model is needed for all three, and this decision is made after studying the magnitude of the singular values from the SVD of the Hankel matrices constructed from input-ouput data. The best dimension is not always easy to determine even from these values, and the performance indices will be used to find the best order of the models produced by each algorithm. When the performance indices for the different methods are compared, the model with the best performance indices from each of the methods is used. The performance is decided after running validation experiments on the produced models. The validation experiments are discussed in the next subsection.

The MOESP and DSR_e algorithms require additional input parameters which have to be decided. As shown in figure 4.4, it is possible to run a brute force experiment by varying different input parameters in search for the set of parameters that minimizes the error between system and model outputs. This will, however, not yield a global solution to the problem, since this optimum only corresponds to the data from the given validation experiment. In addition, these experiments are cumbersome and computationally nonefficient. Since there is no known easy way to determine these parameters, trial and error will be used to find sets of parameters that yield reasonable good results. Selection of optimal input parameters for subspace algorithms could actually be an interesting topic to investigate in future work.

Validation

The validation process consists of producing a set of inputs different from that used in the identification experiment, testing the identified models abilities to reproduce output data from the real system. In this thesis, the approach is simply to generate new reference input PRBS-signals to the real system, and record the output data. The controller output from the real system is then fed to the identified

open-loop system. At last, the outputs from the real system are compared to those produced by the models.

6.3 Simple, weakly coupled system

The system to be identified in this section is a simple system with a common denominator, given by

$$\mathscr{S}_1 = G_0(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} = \frac{1}{2s+1}\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \tag{6.7}$$

In the identification experiments, the order is assumed known.

In order to check how strongly coupled the system is, the steady-state RGA matrix for the system is calculated as (Skogestad & Postlethwaite (2005) p. 83)

$$\Lambda(G_0) = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{bmatrix};\ \lambda_{11} = \frac{1}{1-\frac{g_{12}g_{21}}{g_{11}g_{22}}} \tag{6.8}$$

where $(g_{11}, g_{12}, g_{21}, g_{22})$ are the steady state elements of $G_0(s)$. For the system at hand the value of the relevant RGA parameter is

$$\lambda_{11} = \frac{1}{1-\frac{1}{2}} = 2 \tag{6.9}$$

This indicates that there is some interaction in the system. This will be shown during the following subsection.

When the system is identified with noise, different simulation will yield different results. Hence, the system is identified multiple times, using the mean value of the performance indices as performance indications. This will be referred to as *Monte Carlo simulations*, since it is a (simplified) Monte Carlo experiment. The noise is bandlimited white noise for both process noise and measurement noise. The variance of the noise is 0.001 for both the process noise and the measurement noise. The system will be identified with no noise, with measurement noise only, with process noise only, and with both process and measurement noise. In the latter case, nine different tuning configurations will be tested, while only two different will be tested when the noise is either process or measurment noise. This is because the most realistic case is the one where the system is corrupted by both process- and measurement noise.

Controller configurations

In order to identify the system, the controller parameters have to be decided. The system will first be tuned relatively "smooth", which means that a reference step will result in a "smooth" controller command u to achieve $y = r$. This tuning will be referred to as *configuration 1*. Figure C.1 shows the step response of the closed-loop system with controller parameter configuration 1 given in table 6.1. Figure

C.1b exposes the interaction in the system, y_2 jumps up a bit at $t = 50$ as a result of the control command u_1, and y_1 jumps a bit down at $t = 150$ as a result of the control command u_2. These two different configurations are used to illustrate the difference between "smooth" and "tight" tuning of this system. When the system is corrupted by noise, several different tuning configurations will be compared. This

Table 6.1: "Smooth" and "tight" controller parameters for the simple, weakly coupled system

Parameter	*Value*	*Configuration*
K_{c1}	0.8	1
K_{c2}	0.8	1
τ_{i1}	5	1
τ_{i2}	5	1
K_{c1}	3	2
K_{c2}	3	2
τ_{i1}	2	2
τ_{i2}	2	2

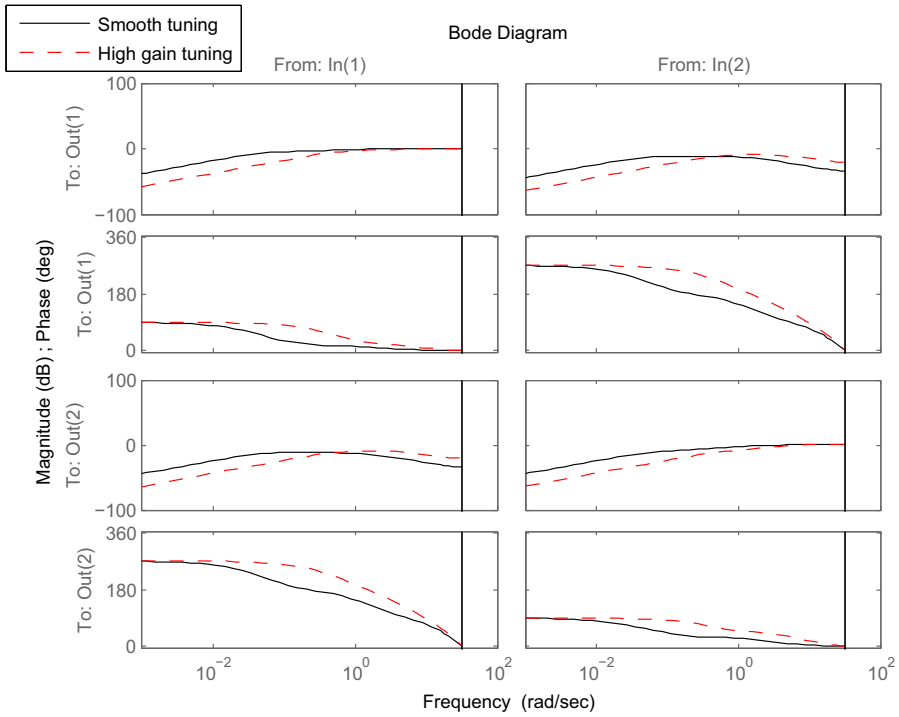


Figure 6.2: Sensitivity function for two different parameter configurations

is also seen in figure C.2, where controller parameter *configuration 2* is applied.

This tuning is relatively aggressive, as seen by the controller command response in figure C.2a. The closed-loop sensitivity function is plotted for these two tuning configurations, and shown in figure 6.2.

No noise

The technique of direct identification discussed in section 3.5 was applied to the system at hand. Shortly explained, it involves direct usage of u- and y-data from the closed-loop system for identification, ignoring feedback in the data record. The system is excited by a PRBS-sequence shown in figure 6.3. The PRBS-signal consists of steps with durations between $0.5s$ and $10s$. Figure 6.4 shows that the system is perfectly estimated with perfect data, even if the experiment is performed in closed-loop. The simulation was carried out with different tunings of the controller, but all tuning configurations gave the same result as shown in figure 6.4.

The subspace parameters used was

- MOESP - Embedded dimension = 60, Order = 2
- N4SID - Order = 2
- DSR_e - $L = 20$, $J = 20$ Order = 2

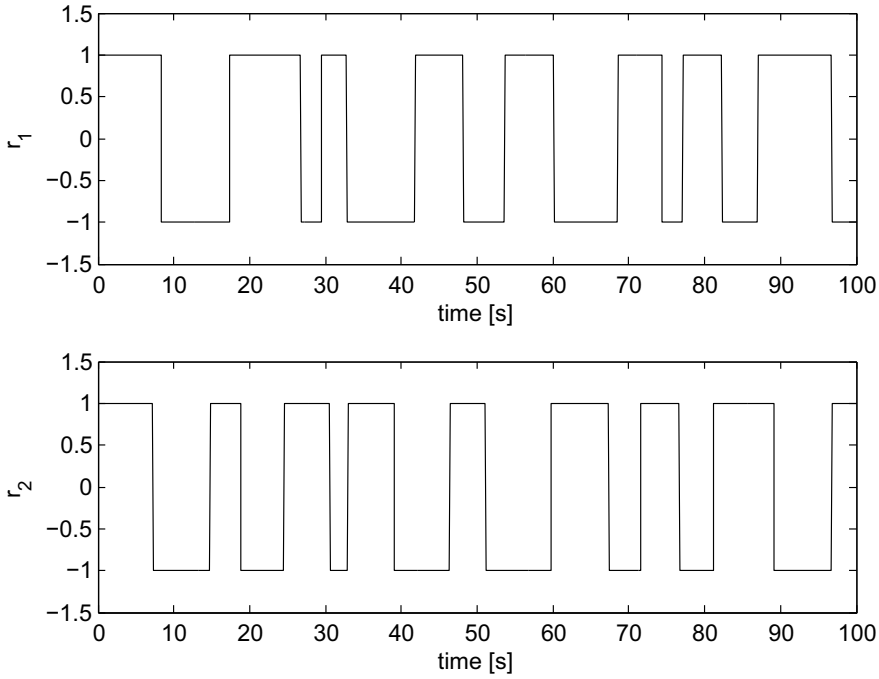


Figure 6.3: PRBS sequences used for direct identification

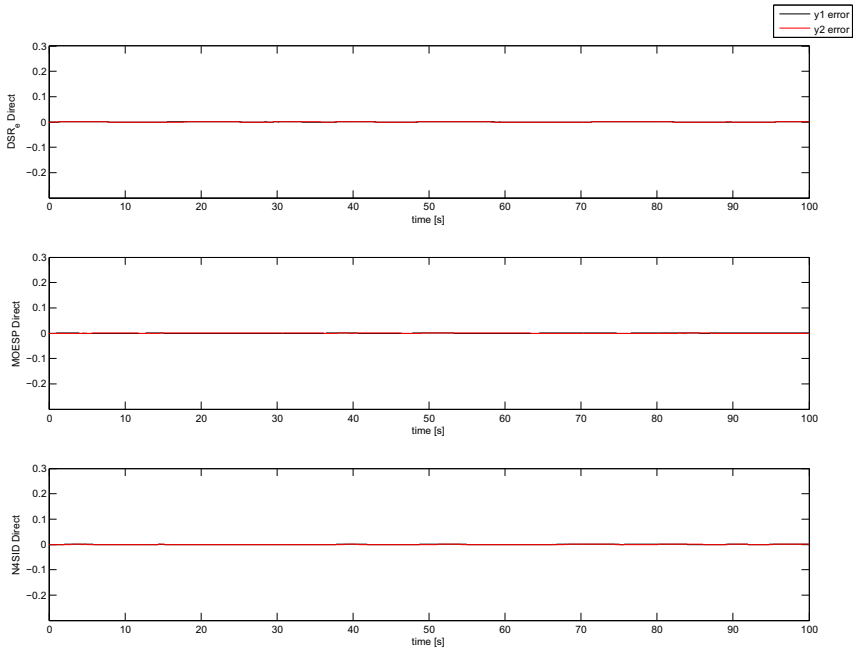


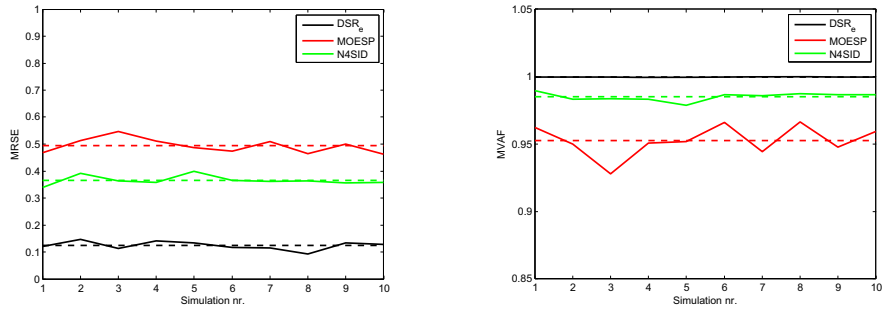
Figure 6.4: Identification error without process and measurement noise

Measurement noise only

Here the subspace algorithm parameters were chosen as follows

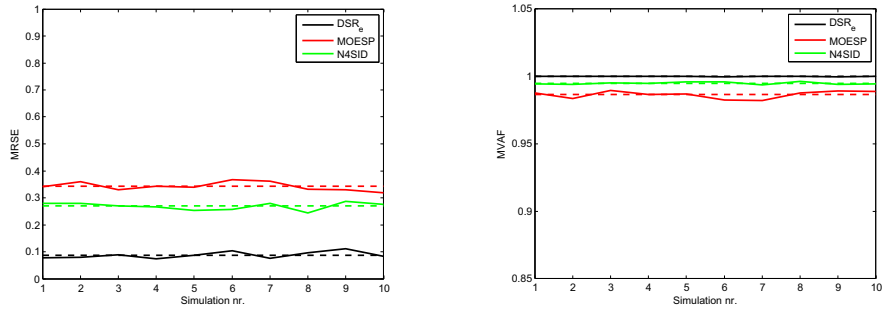
- MOESP - Embedded dimension = 60, Order = 2
- N4SID - Order = 2
- DSR_e - $L = 50$, $J = 50$ Order = 2

Figure 6.5 shows the result from the Monte Carlo simulation for tuning configuration 1, and figure 6.6 shows the result for tuning configuration 2. Notice that both



(a) MRSE (b) MVAF

Figure 6.5: Performance indices for tuning configuration 1, measurement noise only



(a) MRSE (b) MVAF

Figure 6.6: Performance indices for tuning configuration 2, measurement noise only

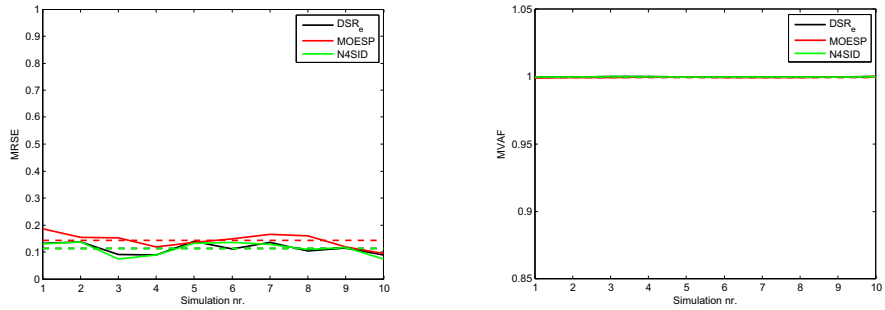
indices are better when the tight tuning configuration is applied to the controllers (configuration 2).

Process noise only

Here the subspace algorithm parameters were chosen as follows

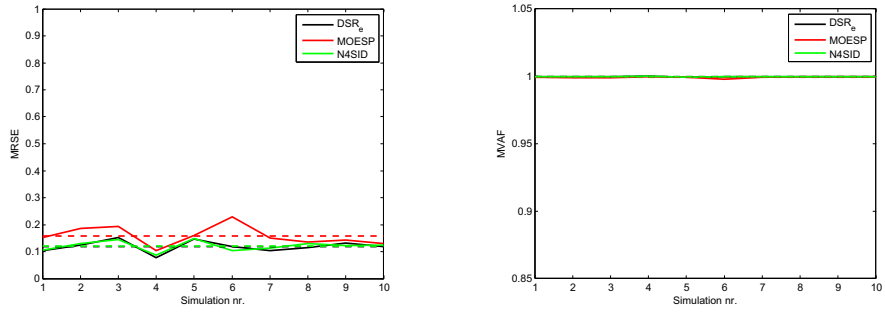
- MOESP - Embedded dimension = 60, Order = 2
- N4SID - Order = 2
- DSR_e - $L = 20$, $J = 20$ Order = 2

Figure 6.7 shows the result from the Monte Carlo simulation for tuning configuration 1, and figure 6.8 shows the result for tuning configuration 2. In this case the



(a) MRSE (b) MVAF

Figure 6.7: Performance indices for tuning configuration 1, process noise only



(a) MRSE (b) MVAF

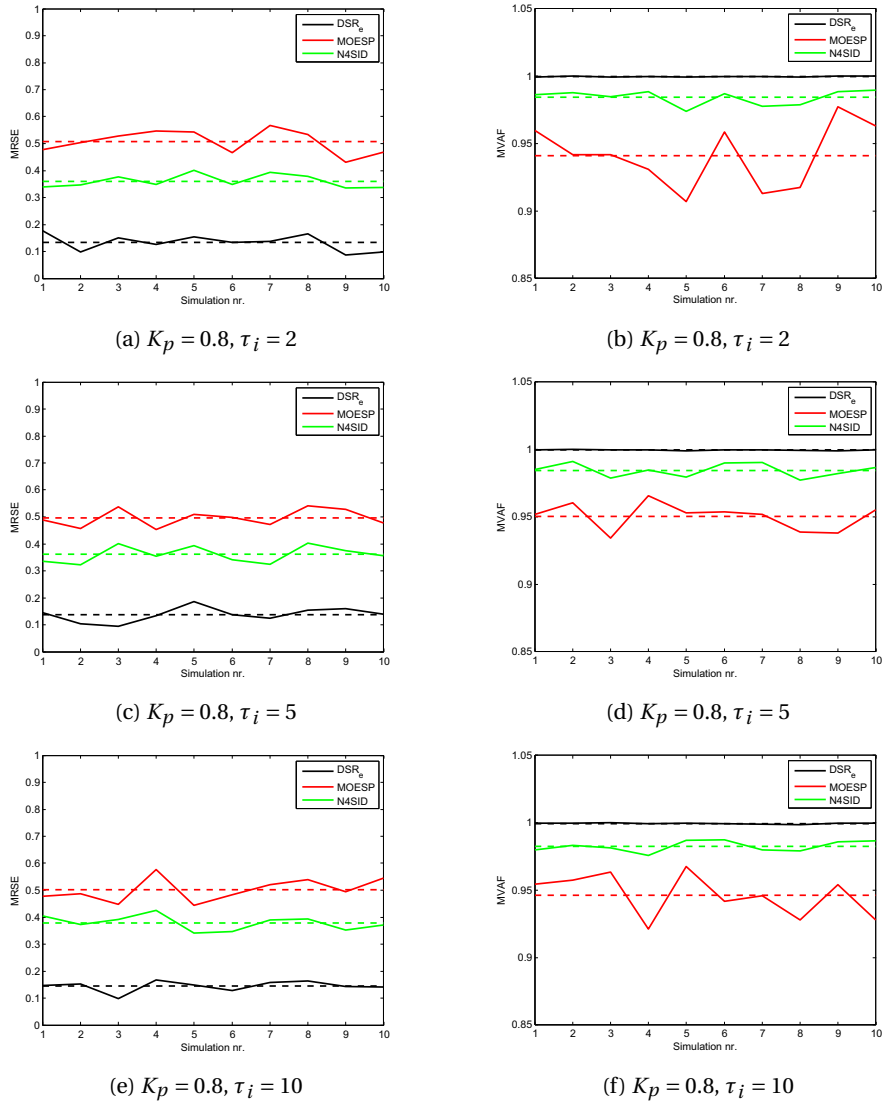
Figure 6.8: Performance indices for tuning configuration 2, process noise only

performance is almost equal for both tuning configurations. This is also seen in table B.1, where the numbers are listed.

Measurement and process noise

The subspace parameters used in this case was

- MOESP - Embedded dimension = 80, Order = 2
- N4SID - Order = 2
- DSR_e - $L = 50$, $J = 50$ Order = 2



(a) $K_p = 0.8$, $\tau_i = 2$

(b) $K_p = 0.8$, $\tau_i = 2$

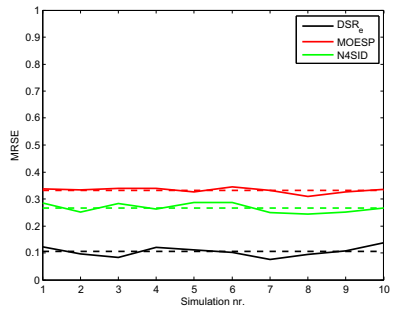
(c) $K_p = 0.8$, $\tau_i = 5$

(d) $K_p = 0.8$, $\tau_i = 5$

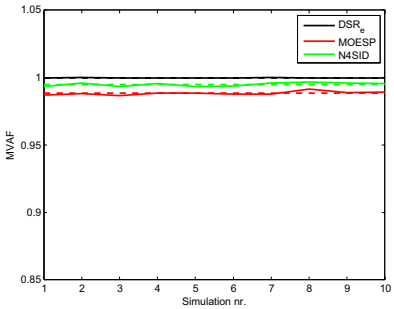
(e) $K_p = 0.8$, $\tau_i = 10$

(f) $K_p = 0.8$, $\tau_i = 10$

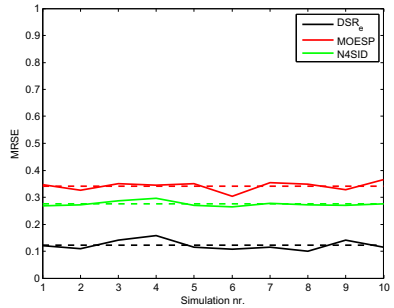
Figure 6.9: Performance indices for different values of $\tau_i = \tau_{i1} = \tau_{i2}$, $K_p = K_{p1} = K_{p2} = 0.8$



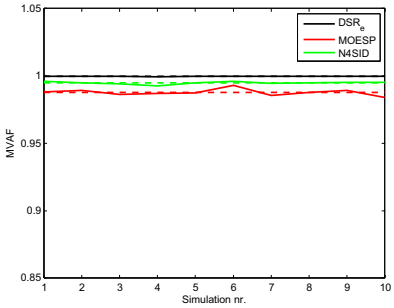
(a) $K_p = 3$, $\tau_i = 2$



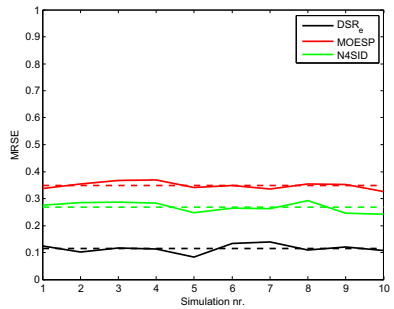
(b) $K_p = 3$, $\tau_i = 2$



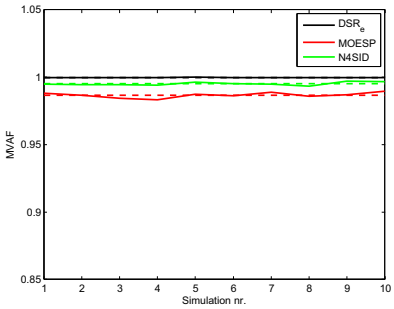
(c) $K_p = 3$, $\tau_i = 5$



(d) $K_p = 3$, $\tau_i = 5$



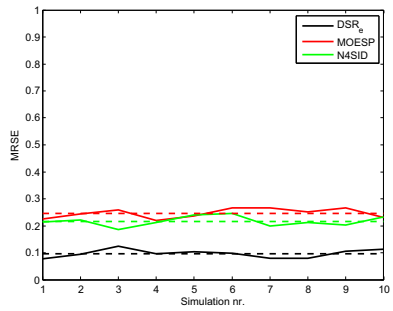
(e) $K_p = 3$, $\tau_i = 10$



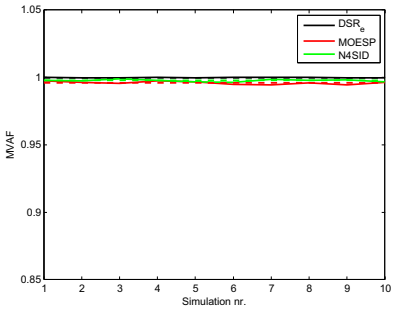
(f) $K_p = 3$, $\tau_i = 10$

Figure 6.10: Performance indices for different values of $\tau_i = \tau_{i1} = \tau_{i2}$, $K_p = K_{p1} = K_{p2} = 3$

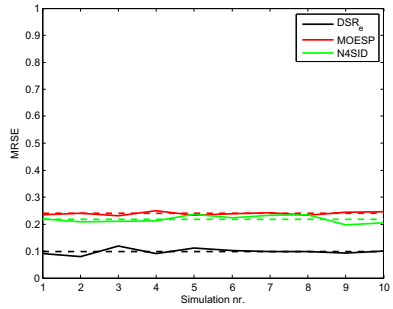
Figures 6.9, 6.10, and 6.11 shows clearly that the performance in this case is better with increased controller gain. The integral time seems to have little effect on the performance.



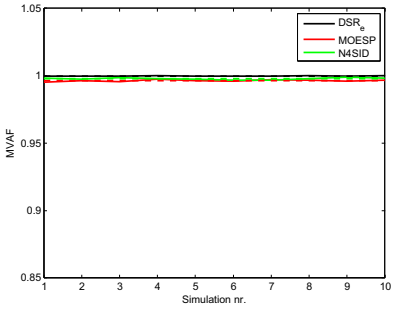
(a) $K_p = 10$, $\tau_i = 2$



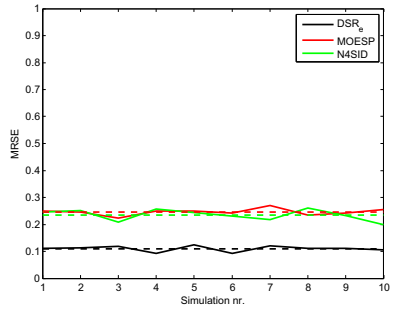
(b) $K_p = 10$, $\tau_i = 2$



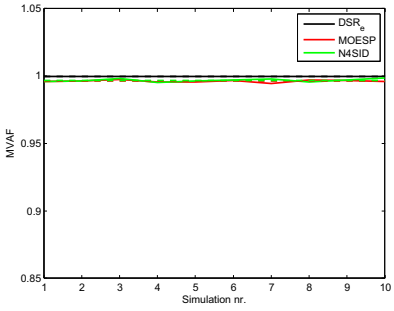
(c) $K_p = 10$, $\tau_i = 5$



(d) $K_p = 10$, $\tau_i = 5$



(e) $K_p = 10$, $\tau_i = 10$

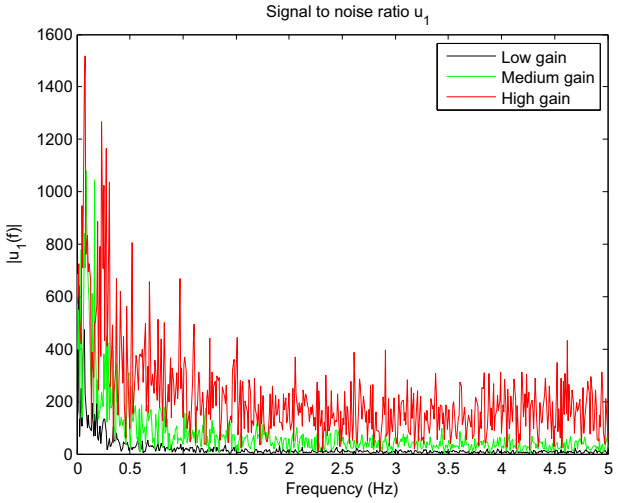


(f) $K_p = 10$, $\tau_i = 10$

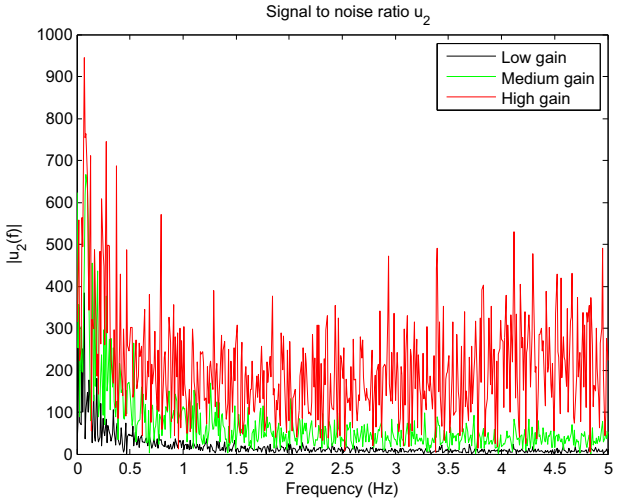
Figure 6.11: Performance indices for different values of $\tau_i = \tau_{i1} = \tau_{i2}$, $K_p = K_{p1} = K_{p2} = 10$

Signal to noise ratio for different controller gains

Figure 6.12 shows a simplified measure for the signal power compared to the noise power for frequencies up to the Nyquist frequency for system. In these plots, the power spectrums of the input signals u_1 and u_2 are found by FFT (Fast Fourier Transform), and the result is scaled with the power of the noise. These input signals are generated with both measurement and process noise present in the loop. Notice that the input power is amplified at high frequencies with increased controller gain, as discussed in section 3.3.



(a) u_1



(b) u_2

Figure 6.12: Power spectrum of identification inputs u_1 and u_2 scaled with noise power

6.4 Stiff, weakly coupled system

The system to be identified in this section is

$$G_0(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{2s-1}{(3s+1)(s+1)} & \frac{1}{2s+1} \\ \frac{1}{50s+1} & \frac{2s+1}{(70s+1)(30s+1)} \end{bmatrix} \tag{6.10}$$

This system is more complex than $\mathscr{S}_1$, since it has a zero in the right half plane, which will give inverse responses. As shown further down in this section, its eigenvalues are spread by a factor of 70, which means that this is a relatively stiff system.

As in section 6.3, the steady state RGA matrix is calculated using the formula

$$\lambda_{11} = \frac{1}{1 - \frac{g_{12}g_{21}}{g_{11}g_{22}}} \tag{6.11}$$

in order to determine how strongly coupled the multivariable system is. Recall that the elements of the transfermatrix used in equation (6.11) are the steady state elements. For system $\mathscr{S}_2$, the steady state RGA value is

$$\lambda_{11} = \frac{1}{1 - \frac{g_{12}g_{21}}{g_{11}g_{22}}} = \frac{1}{1 - \frac{1}{-1}} = 0.5 \tag{6.12}$$

This indicates that the system is slightly interactive.

A canonical, continuous state-space realization of this system reveals its eigenvalues through the A-matrix:

$$\begin{aligned} \dot{x} &= \underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.3333 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0333 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.01429 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 \end{bmatrix}}_{A} x + \begin{bmatrix} -2.121 & 0 \\ -1.581 & 0 \\ 0 & -0.2998 \\ 0 & -0.2255 \\ 0.0625 & 0 \\ 0 & 1 \end{bmatrix} u \\ y &= \begin{bmatrix} -0.7071 & 0.5270 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.0778 & -0.1077 & 0.32 & 0 \end{bmatrix} x \end{aligned} \tag{6.13}$$

Figure C.3 shows the unit step response of this system when both inputs are stepped. As seen by this response and the A-matrix, the system has both fast and slow dynamics. This is the reason why this particular system is selected, in order to test the subspace algorithms abilities to capture the different time constants of the system. As mentioned in Hugues Garnier & Young (2008), the main difficulty with stiff systems is to choose a proper sampling rate. Consider an exact discretization

of the linear system 6.13 with a sampling time of $0.1s$:

$$\dot{x} = \underbrace{\begin{bmatrix} 0.9048 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9672 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9967 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9986 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.998 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9512 \end{bmatrix}}_{A} x + \begin{bmatrix} -0.2019 & 0 \\ -0.1555 & 0 \\ 0 & -0.0299 \\ 0 & -0.2253 \\ 0.006244 & 0 \\ 0 & 0.09754 \end{bmatrix} u$$
$$y = \begin{bmatrix} -0.7071 & 0.5270 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.0778 & -0.1077 & 0.32 & 0 \end{bmatrix} x \tag{6.14}$$

As the system matrix A_d shows, the slowest pole of the system is very close to the unit circle (0.9986). The result of this is that even a slight estimation error may result in an unstable model (eigenvalues outside the unit circle).

Controller configurations

As for the system in section 6.3, identification will be performed using different controller parameters. In appendix C.1, the closed-loop step responses for three different parameter configurations is shown. Since the last section showed that the integral time was of minor importance compared to the controller gain, only the controller gain will be varied here. The different tuning configurations are

- **Configuration 1** - $K_{p1} = -0.2$, $K_{p2} = 0.5$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$
- **Configuration 2** - $K_{p1} = -0.4$, $K_{p2} = 1$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$
- **Configuration 3** - $K_{p1} = -1$, $K_{p2} = 1$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$

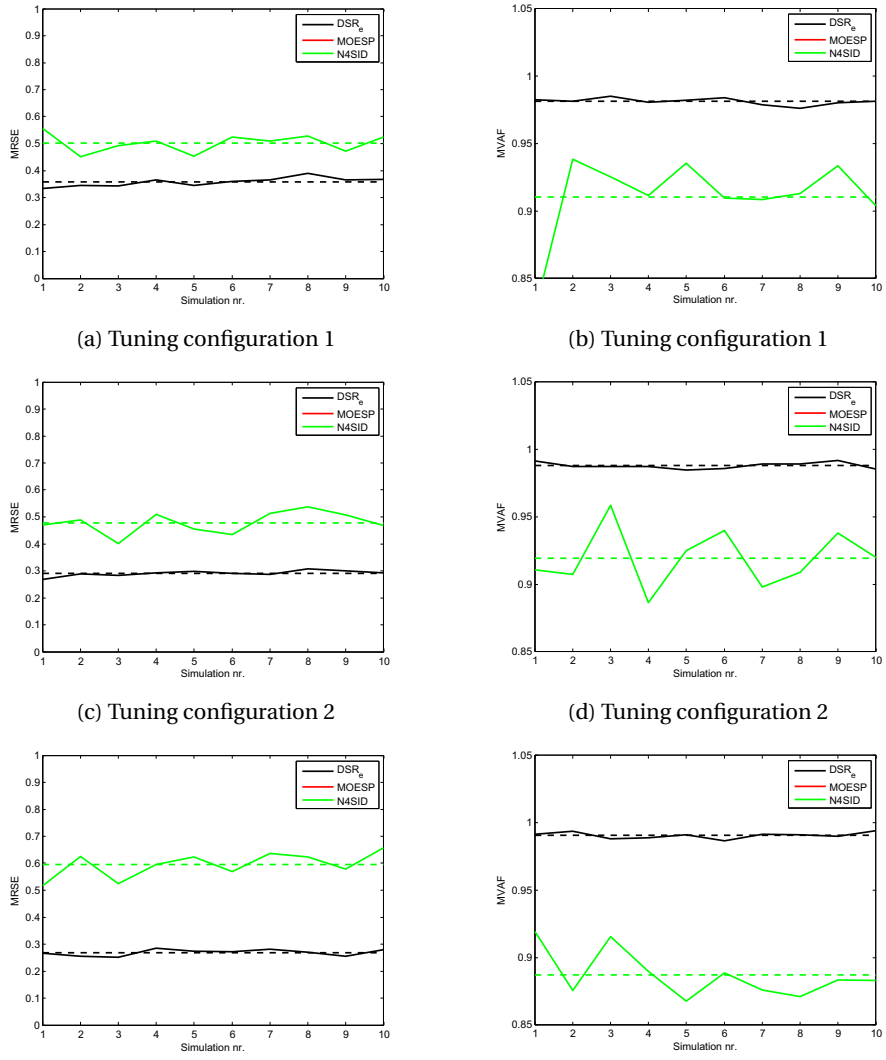
Notice that controller C_1 has negative gain, since the steady state value of the open-loop transferelement g_{11} is negative.

For this system, different inputs to the subspace algorithms were tested. There was only a minor difference on the performance by varying these parameters for the different controller and noise configuratinos for this system, so the parameters are, for all of the following simulations,

- MOESP - Embedded dimension = 90, Order = 6
- N4SID - Order = 6
- DSR_e - $L = 60$, $J = 40$ Order = 6

The order is, as in section 6.3, assumed known.

Measurement noise only



(a) Tuning configuration 1

(b) Tuning configuration 1

(c) Tuning configuration 2

(d) Tuning configuration 2

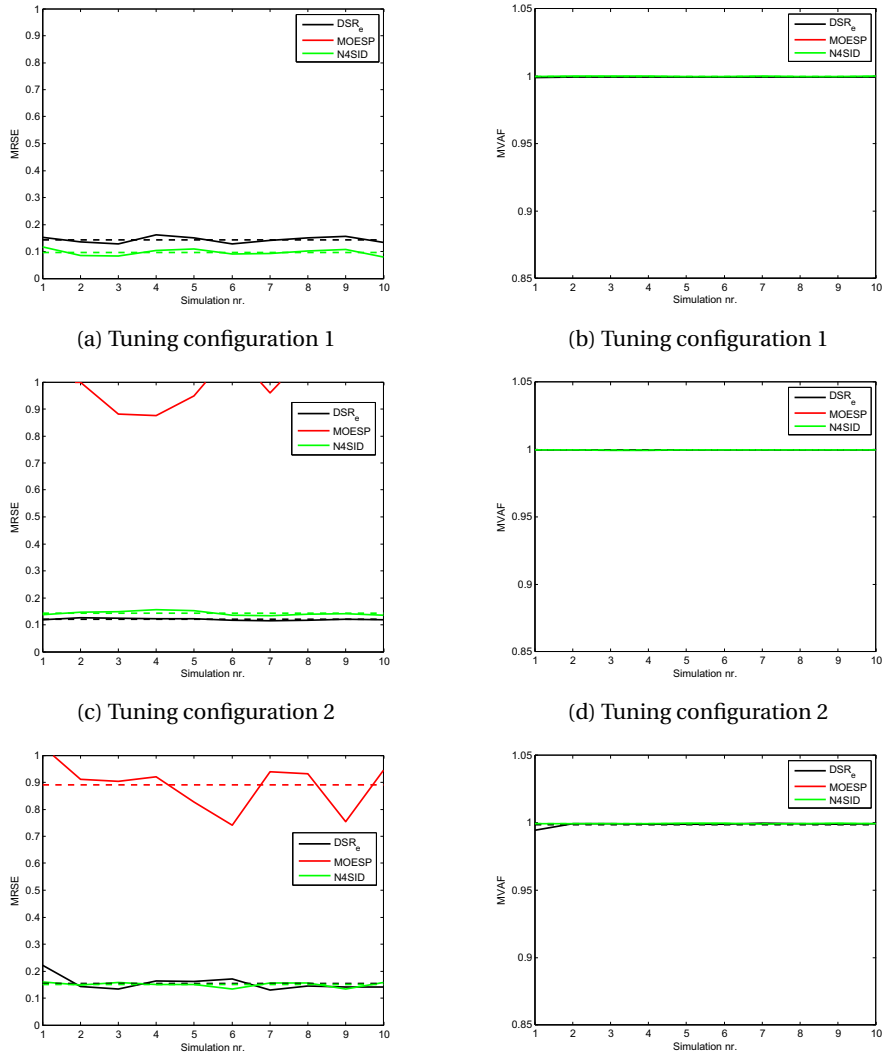
(e) Tuning configuration 3

(f) Tuning configuration 3

Figure 6.13: Performance indices for different values of controller gains

Remarks: The MOESP method failed to identify this system when measurement noise was present, giving a strongly biased model. DSR_e performed better with higher controller gain, while N4SID performed worst for tuning configuration 3.

Process noise only



(a) Tuning configuration 1

(b) Tuning configuration 1

(c) Tuning configuration 2

(d) Tuning configuration 2

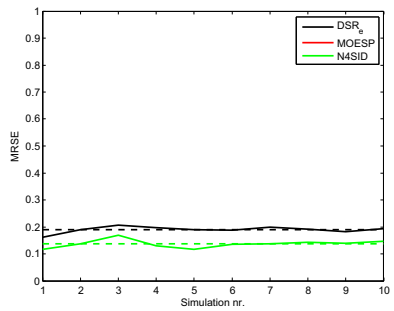
(e) Tuning configuration 3

(f) Tuning configuration 3

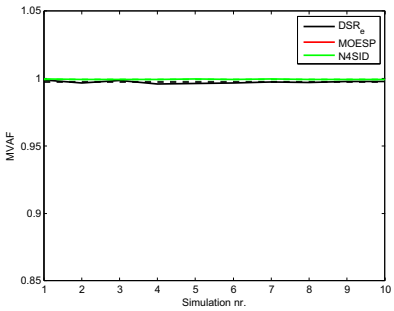
Figure 6.14: Performance indices for different values of controller gains

Remarks: The MOESP method gave better models with higher controller gain, but the models are still very biased. Tuning configuration 2 seems to work best for N4SID and DSR_e in this case, while configuration 3 worked best for MOESP.

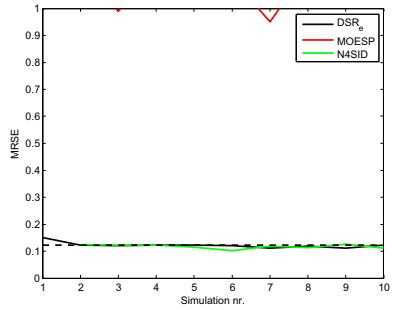
Measurement and process noise



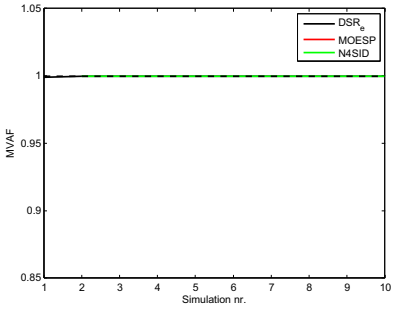
(a) Tuning configuration 1



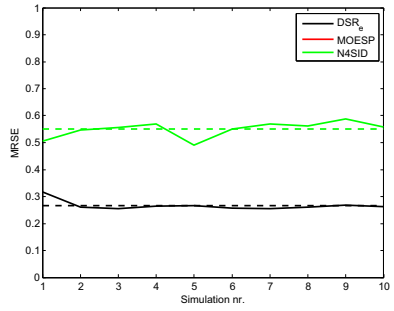
(b) Tuning configuration 1



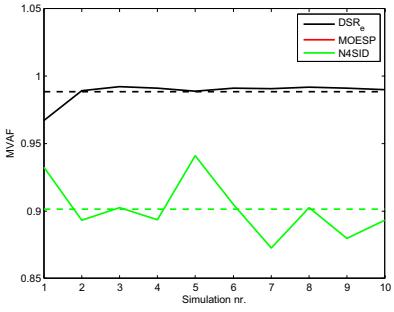
(c) Tuning configuration 2



(d) Tuning configuration 2



(e) Tuning configuration 3



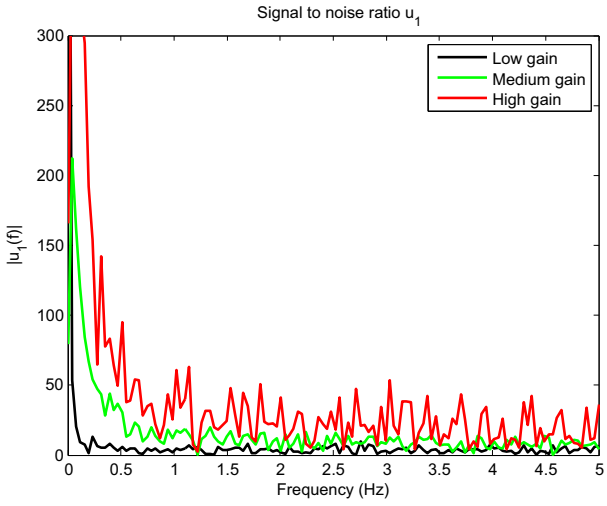
(f) Tuning configuration 3

Figure 6.15: Performance indices for different values of controller gains

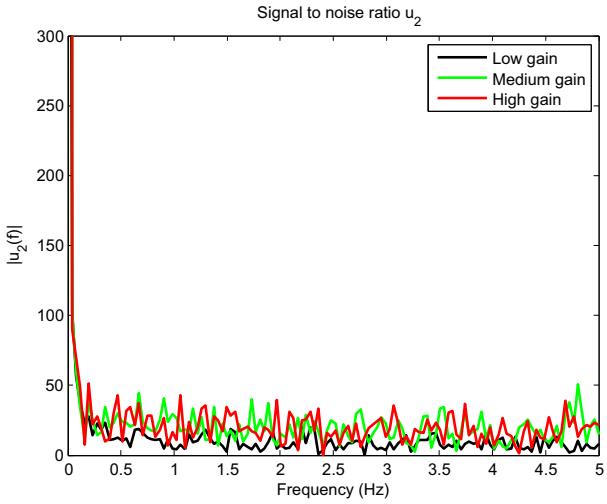
Remarks: The MOESP method gives strongly biased models also in this case. For tuning configuration 1, N4SID gave the model with best performance. Tuning configuration 2 seemed to work best overall in this case.

Signal to noise ratio for different controller gains

Figure 6.16 shows the signal to noise ratio for the identification inputs u_1 and u_2. Notice that the ratio here is lower than the signal to noise ratio for system $\mathscr{S}_1$ for all frequencies.



(a) u_1



(b) u_2

Figure 6.16: Power spectrum of identification inputs u_1 and u_2 scaled with noise power

CHAPTER 7

INDUSTRIAL CASE STUDY: DEBUTANIZER PROCESS

A well known process in the process control field is the *distillation* process. The idea is to separate out a given component of one or several feeds into a distillation column, using the physical properties of the components at hand. A distillation column, or more specifically a *debutanizer* will be used as an example in this chapter. The simulation tool UniSim is utilized for simulation purposes, and generating data for identification routines performed in Matlab. In this chapter the process and control configuration will be explained briefly, and the emphasis will be put on identification of the dynamics of this process.

7.1 The simulator and the process

The simulator

The following information is gathered from *UniSim Design Dynamic Modeling - Reference Guide*, UniSim (2005).

UniSim Design is a simulator used to simulate chemical processes. In addition to model processes using mass and energy balances, UniSim Design also includes vapourization, reactions and density changes in the modeling. For a distillation column, UniSim will model each tray of the column as a *hold-up*, which describes an important physical phenomenon. When changes are made to compositions, temperature, pressure or flow of streams that enters the vessel, there will be a time delay before changes are seen on the output streams. Because of this holdup, changes applied to the reflux flow or reboiler effect will not immediately be seen on the distillate or bottoms flow which leaves the column. Hence, the time delay that is present in a real process is modeled in this case.

The process

The model for this process was used as an example for teaching advanced process control in Bakke (2008). Figure 7.1 shows a Process Flow Diagram (PFD) for the debutanizer. The input feed streams consists of different components, listed in table 7.1. The compositions of the input streams are assumed to be constant, but the flowrates are variable.

The reboiler heats the input flow so that hot vapour is sent back into the column, in order to control the temperature at a specific stage in the column. The effect of this reboiler is calculated by the temperature controller TIC-100, which in turn is controlled by a master bottoms composition controller XIC-101. The reflux flow is controlled by the flow controller FIC-100, which is a slave controller to the distillate composition controller XIC-100. In addition, there are two pressure controllers that controls the column pressure in a bypass configuration. As seen by the process flow diagram, this pressure controller will induce a flow into the reflux drum, which is under level control. This means that pressure control action will act as an "unknown disturbance" on the reflux flow through the reflux drum level controller LIC-102, seen from the flow controller FIC-100. This will in turn have effect on the compositions, since the flow controller adjust the flow to achieve its setpoint, and change of reflux flow means change of compositions. This will also affect the column temperature, as there is a physical interaction between the reflux flow into the column and the temperature.

The debutanizer process is nonlinear, because the relation between temperatures in the column and the compositions is described by a nonlinear differential equation in UniSim. There are also nonlinear effects inside the column that are modeled in UniSim, e.g. the thermodynamics that describes heat transportion throughout the column. The valves though, are modeled to have linear characteristics, and these are fast compared to the rest of the system.

In addition, the process is interactive, since the reflux-flow will directly affect the bottoms composition, and the temperature inside the column will also affect the distillate composition. The split in this column is basically between i-pentane (heavy key component) and n-butane (light key component).

7.2 The control configuration

The control configuration used for this distillation column is the standard LV-configuration, which is recommended for most distillation columns by Skogestad (2007). In this configuration, the bottoms flow B and distillate flow D are used for reboiler level control and condenser level control respectively. The reflux flow L and vapour V from the reboiler are used for composition control. The control configuration shown in figure 7.1 is a cascaded PI-approach, letting two decentralized PI-controllers write setpoints to the reflux flow and temperature controllers respectively. This approach assumes that measurements of the key composition components in the bottoms and distillate flows are available, which is true in this

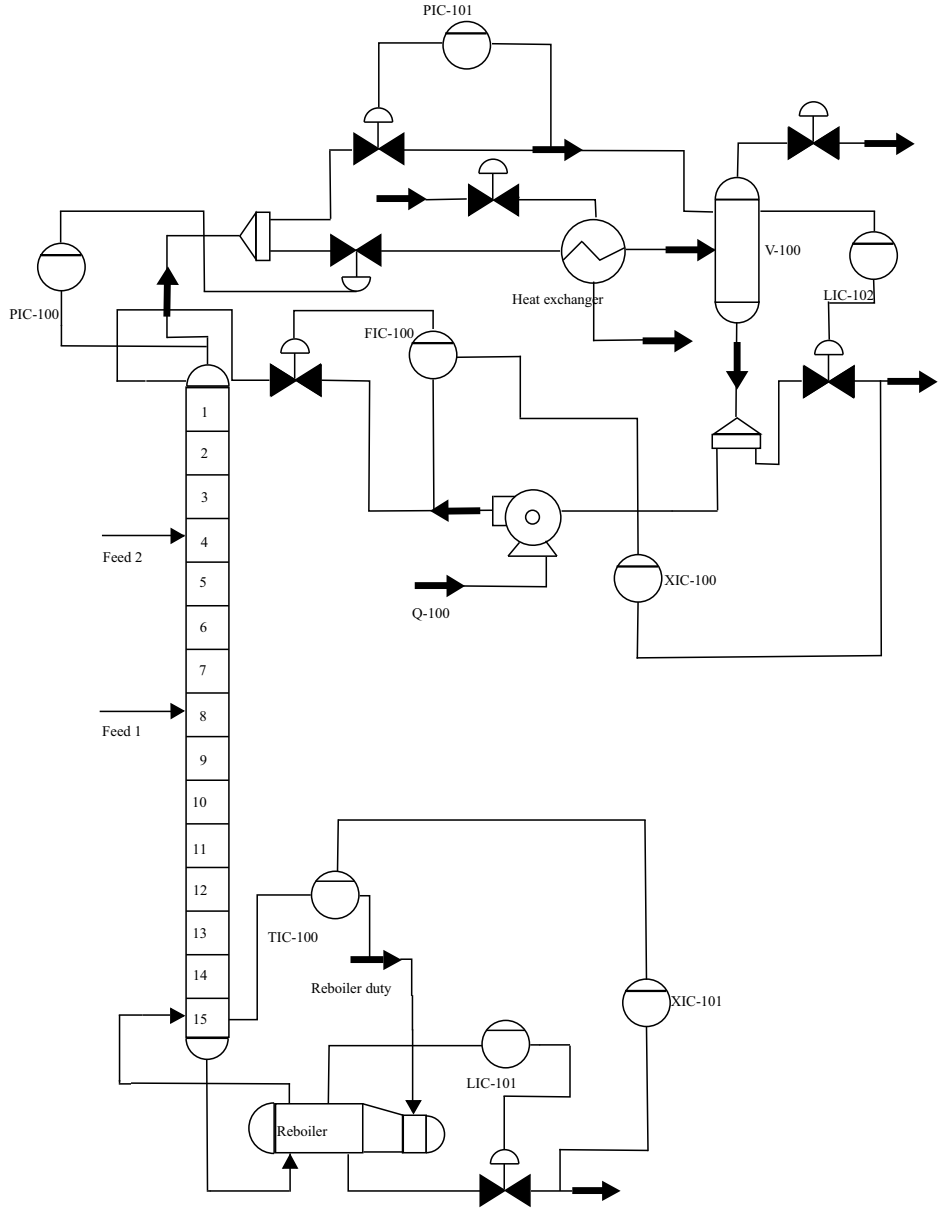


Figure 7.1: Process Flow Diagram for the debutanizer

Table 7.1

Feed 1		**Feed 2**	
Component	*Mole fraction*	*Component*	*Mole fraction*
Propane	0.019763	Propane	0.020000
i-Butane	0.212413	i-Butane	0.190000
i-Butene	0.010355	i-Butene	0.220000
n-Butane	0.212413	n-Butane	0.200000
i-Pentane	0.140921	i-Pentane	0.160000
n-Pentane	0.140921	n-Pentane	0.210000
n-Hexane	0.092701	n-Hexane	0.000000
n-Heptane	0.094220	n-Heptane	0.000000
n-Octane	0.076293	n-Octane	0.000000

simulation case. In a practical setting one would usually control temperatures inside the column only to achieve indirect control of the compositions. The following convention is defined to translate the process to "control language":

- r_1 - setpoint for the controller XIC-100 [Mole Fraction]
- r_2 - setpoint for the controller XIC-101 [Mole Fraction]
- u_1 - Setpoint to the flow controller FIC-100, which controls the reflux flow into the column $\left[\frac{m^3}{h}\right]$
- u_2 - Setpoint for TIC-100, which controls stage 8 temperature $[^\circ C]$
- d_1 - Mass flow of Feed 1 $\left[\frac{kg}{h}\right]$
- d_2 - Mass flow of Feed 2 $\left[\frac{kg}{h}\right]$
- y_1 - Composition of heavy key component (i-pentane) in the distillate flow [Mole Fraction]
- y_2 - Composition of light key component (n-butane) in the bottoms flow [Mole Fraction]

The subsystem of the process to be identified hence is multivariable, having two controllable inputs, two uncontrollable but measured inputs (disturbances), and two outputs. The system is sampled with a sampling time of 1min = 60s, because this is a relatively slow process.

Figure 7.2 shows a block diagram for the system at hand. The purpose is to identify a relatively low order linear model that is suitable around the nominal operating for the debutanizer. A full state space of the model would require many states, since each of the trays in the column is modeled to have holdups between

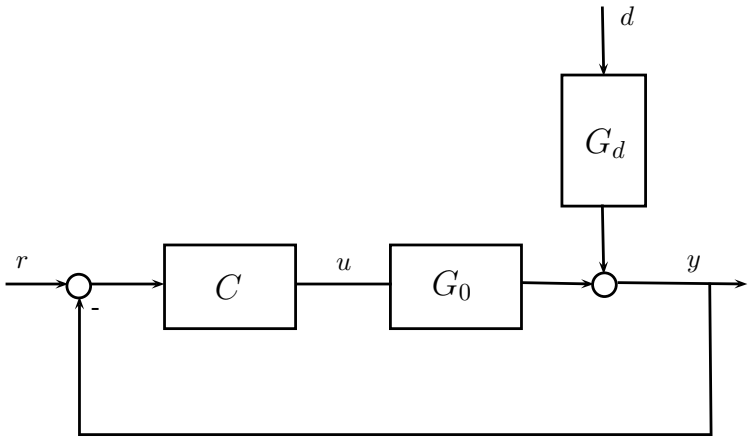


Figure 7.2: Block diagram to show signal flow in the process

them. The states would then describe the heat transfer between the trays inside the column, as well as flows between the trays. In addition, a description of the pressure dynamics would require additional states. For the subsystem of the full process that is identified here, the effects from the rest of the process that is not measured during the identification experiment will act as unknown disturbances to the system, so a perfect model is very hard to achieve. A model that describes the input-ouput properties reasonbly good is sought, i.e. models that are capable of reproducing outputs from the real process with the same input data.

7.3 The model

The goal is to identify the transfer function G_0 of figure 7.2. Identification of G_d is not considered here, since this is an open-loop problem.

Open-loop model

In the frequency domain, the open loop model is

$$y(s) = G_0(s)u(s) + G_d(s)d(s) \tag{7.1}$$

Assuming linearity around an operating point $u(s) = u^* = \text{const}$, $d(s) = d^* = \text{const}$, it is possible to treat the two contributions from $u(s)$ and $d(s)$ separatly by the super position principle. The control signal $u(s)$ is given by the control law

$$u(s) = \begin{bmatrix} C_1(s) & 0 \\ 0 & C_2(s) \end{bmatrix}(r-y) = \begin{bmatrix} k_1\frac{\tau_{i1}s+1}{\tau_{i1}s} & 0 \\ 0 & k_2\frac{\tau_{i2}s+1}{\tau_{i2}s} \end{bmatrix}(r-y) \tag{7.2}$$

The idea now is to identify the two transfer functions using the subspace methods described in section 4.6.

Controller configurations

The decentralized controller will be tuned with two different tuning configurations, where the gains of the controllers will be varied. Three different tuning configurations are used, where only the controller gains are varied. The parameters for the different configurations are

-
- **Configuration 1** - $K_{c1} = 1.29$, $K_{c2} = 0.319$
- **Configuration 2** - $K_{c1} = 1.935$, $K_{c2} = 0.4785$
- **Configuration 3** - $K_{c1} = 2.58$, $K_{c2} = 0.638$

In all cases, the integral times are $\tau_{i1} = 38.3$[min] and $\tau_{i2} = 13.5$[min].

Noise properties

The identification data is contaminated by white noise with variance 0.02% of the respective signal values. This yield for the input-output data used in the subspace algorithms for different controller configurations, i.e. u_1, u_2, y_1 and y_2. When the model order is decided, noisefree data is used to find the best input parameters for the subspace algorithms.

The validation data is generated with no noise in order to evaluate the performance of the deterministic models. It as also generated with a different PRBS sequence than the identification data. The outputs of these deterministic models are compared with outputs with the noisefree process outputs, where the u-data generated by the closed-loop process are fed to the models identified by the subspace methods.

7.4 Identification experiment

As explained in section 7.1, the true process has nonlinear dynamics, and it contains time delays that need to be approximated by the identified state space models. In additino, the data is contaminated by noise. Because of this, it is expected that the outputs from the identified models will differ some from the real process outputs.

In order to generate informative data for identification of the debutanizer, a PRBS sequence is generated for both external inputs r_1 and r_2. The identification PRBS-sequences are designed so that the longest steps will give the slowest time constants of the system time to settle. The disturbances d_1 and d_2 are left constant during the identification experiment.

The references are stepped with ± 0.0005 mole fractions around their nominal operiting point, which corresponds to steps of $\pm 2.5\%$ of their absolute values. Figures 7.3 and 7.4 show the outputs and references, and controller commands.

This is noisefree data generated with control configuration 2, and it will be used to decide on the model order.

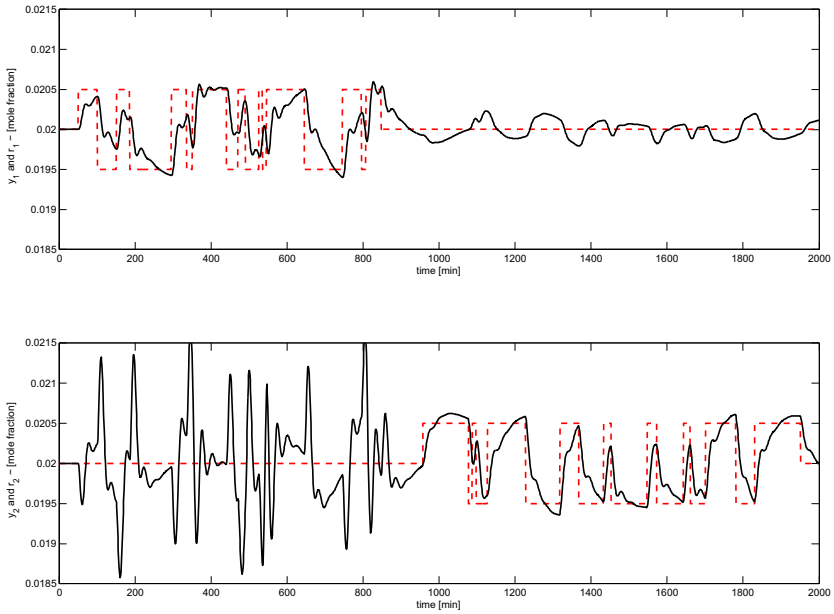


Figure 7.3: The outputs y_1 and y_2 along with the references r_1 and r_2, no noise. The broken lines are the references, and solid lines the outputs

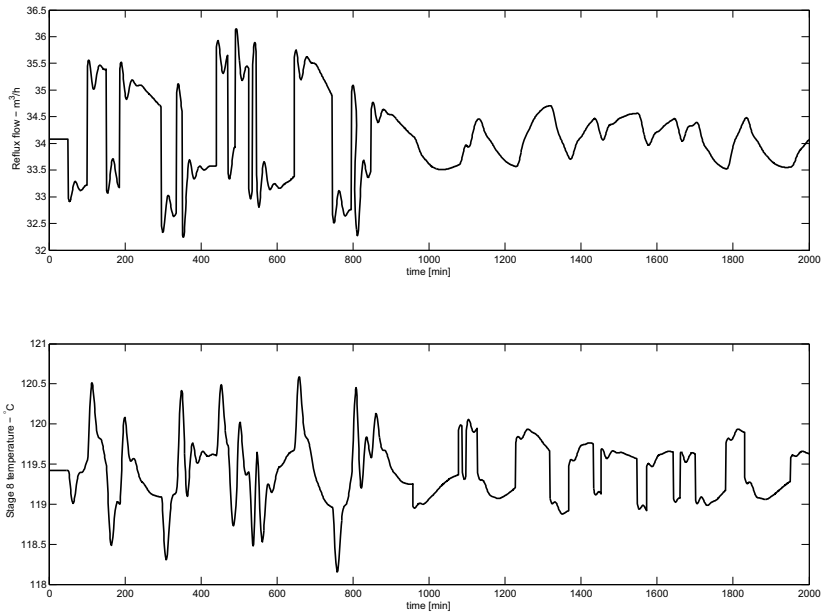


Figure 7.4: The controller commands u_1 and u_2, no noise

7.5 Preconditioning of identification data

Since this model is operating around a nominal operating point, the means in the data record will be removed to capture the dynamics of the process. The reason for this is explained in section 2.4. When the model order is to be decided, the input and output data is scaled to be in the range 0 to 1 in order to normalize the singular value plot.

7.6 Determination of the model orders

In order to decide on the model order, the singular values from the SVD step in the DSR_e algorithm is utilized. Figure 7.5 shows the singular values along with the condition number from this step. The idea is to select an order that includes the

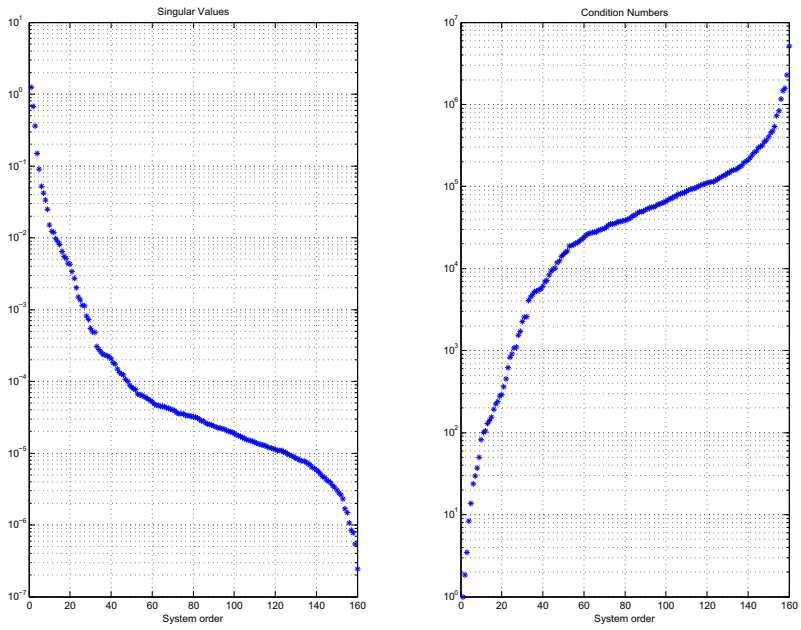


Figure 7.5: Singular values to decide on model order

dominant singular values shown in this plot. In this case, the most dominating singular values seems to corresponds to model orders between 3 and 9. When the model order was decided here, these different orders was fed to the subspace algorithms to see which order that gave the best performance indices. For the DSR_e method, the model order that gave the best performance was 6. The N4SID gave the best results for a model order of 7, while the MOESP method performed

best for only 4 states in the final model. These orders will be used as inputs to the algorithms during the next section, when the controller gains are varied.

7.7 Identification of debutanizer process with different controller gains

As for chapter 6, the performance of the different algorithms will be evaluated when the controller gains are varied. The three different controller configurations given in section 7.3 are used. The input parameters to the subspace algorithms that seemed to give best results for each of the algorithms were:

- MOESP - Embedded dimension = 90, Order = 4
- N4SID - Order = 7
- DSR_e - $L = 80$, $J = 80$, Order = 6

Table 7.2 shows the performance indices for each of the subspace methods for each of the three different controller configurations. These indices show clearly

Table 7.2: Performance indices for the subspace methods used for identification of the debutanizer

Configuration	Method	MRSE	MVAF
1	DSR_e	70.11%	76.28%
	MOESP	89.46%	49.83%
	N4SID	73.13%	69.93%
2	DSR_e	56.32%	89.88%
	MOESP	87.9%	43.74%
	N4SID	60.01%	86.22%
3	DSR_e	49.76%	93.68%
	MOESP	91.7%	46.55%
	N4SID	56.43%	88.67%

that the performance of the models was better with higher controller gain for the two methods DSR_e and N4SID. For the MOESP method, the indices do not reflect any clear improvement for higher controller gains. This method though, was shown to return very biased and unreliable models for noisy closed-loop data in section 6.4, hence the results from this method will not be dwelled much upon.

An important note though, is that an experiment with even higher controller gain than configuration 3 was also tested. In this case, the two open-loop methods N4SID and MOESP produced unstable models, and DSR_e produced a model with worse performance than all the three other configurations. See figure 7.6 for a validation test of these models that shows the bad performance.

To show the validation results more visually, figures 7.7, 7.8, 7.9 and 7.10 show each of the process output plotted along with the outputs from the models found using each of the three controller configurations. The DSR_e method is used in this example, since the results from the N4SID method is equal, except for slightly worse performance. The plots for each output are splitted in half to give a better view of the differences between the three controller configurations. Notice that all the variables are scaled to be in the range 0 to 1. Plots of the controller outputs (u_1 and u_2) are given in section 7.8.

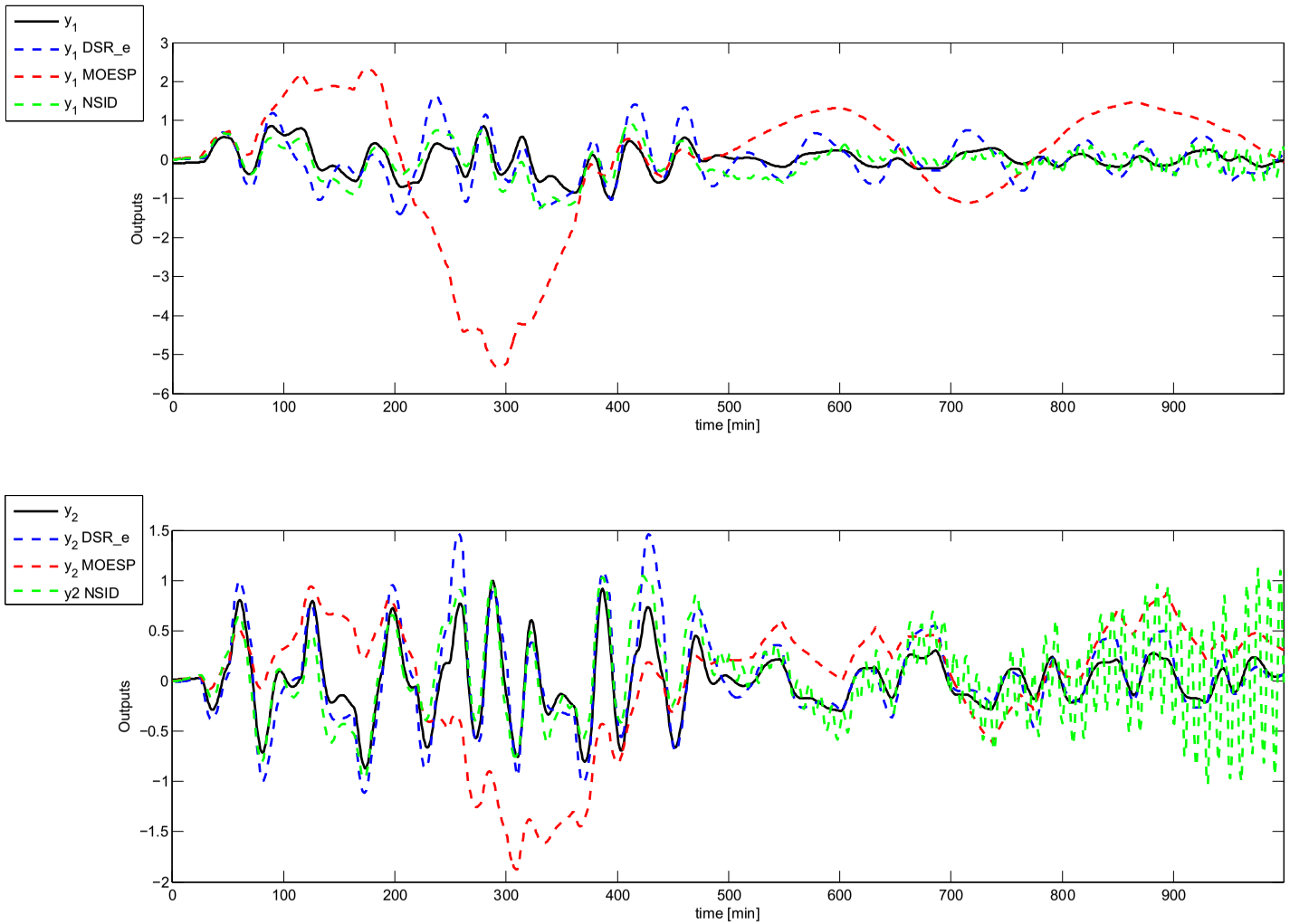


Figure 7.6: Models produced by the three algorithms when the controller gains were too high. Broken lines are model outputs and blacksolid lines are the outputs of the real process.

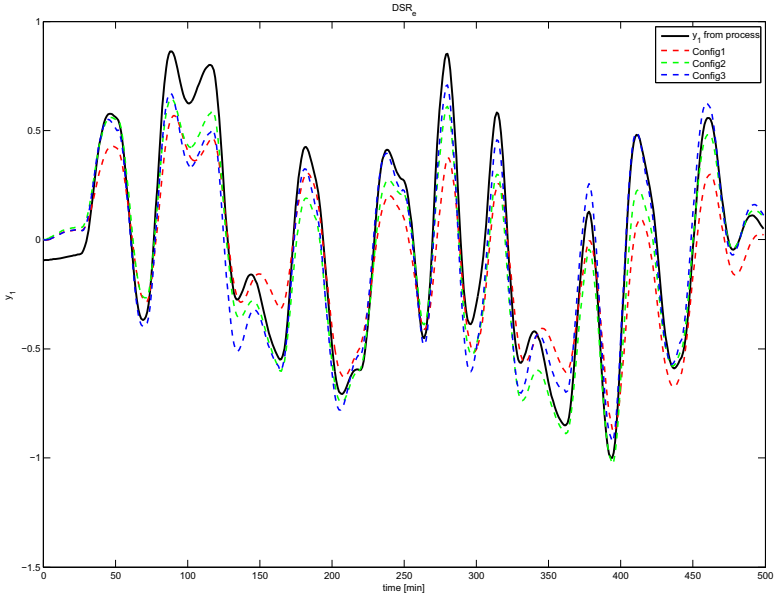


Figure 7.7: First half of validation sequence for y_1

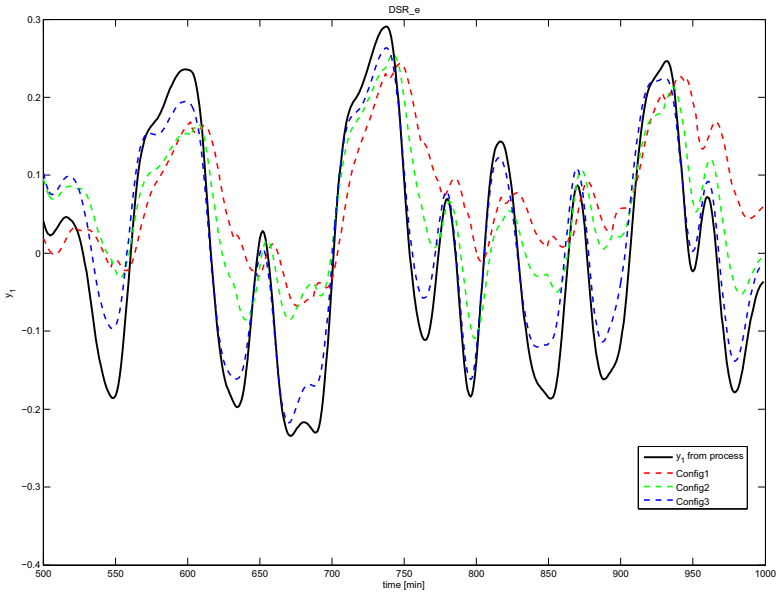


Figure 7.8: Second half of validation sequence for y_1

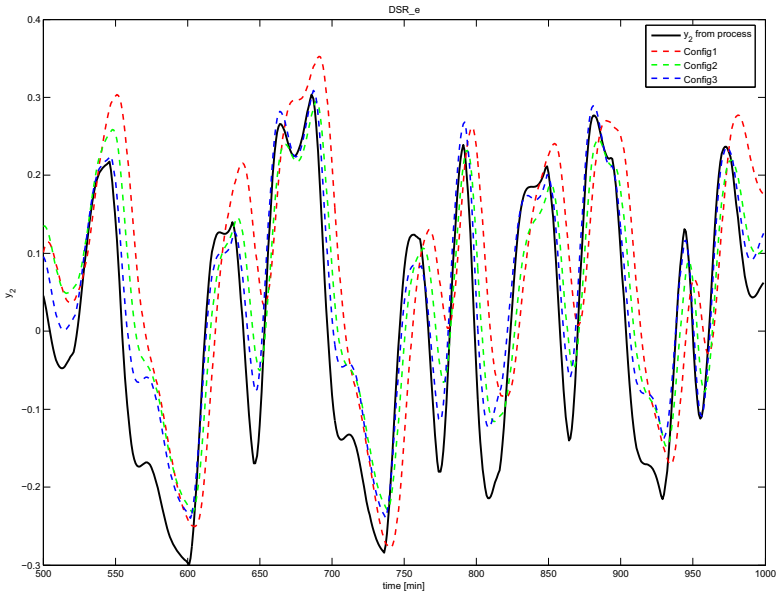


Figure 7.9: First half of validation sequence for y_2

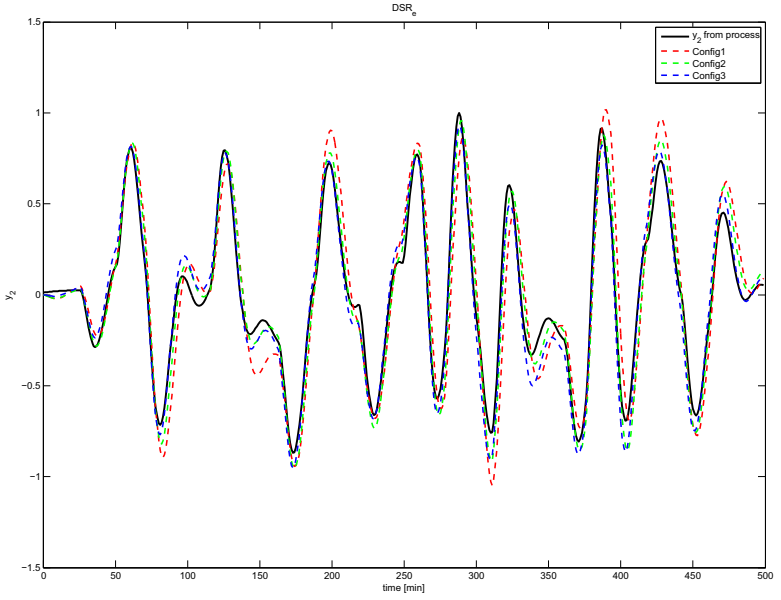


Figure 7.10: Second half of validation sequence for y_2

7.8 Controller outputs in the time domain

Figures 7.11 and 7.12 show the controller outputs u_1 and u_2 for different tuning configurations in the time domain. These are the inputs used during the identification in section 7.7. As seen by the plots of u_1, the difference is not big between the signals generated for different gains. For u_2 on the other hand, the difference between the signals is seen more visually. In the next section, the power of these signals is shown in the frequency domain

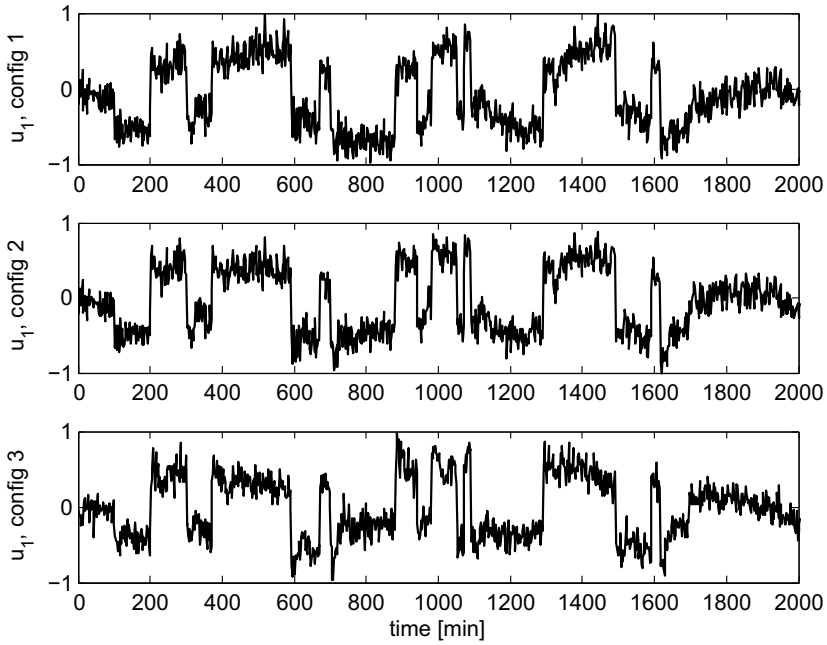


Figure 7.11: Controller output u_1 for varying gains. The gain increases from top to bottom of this figure

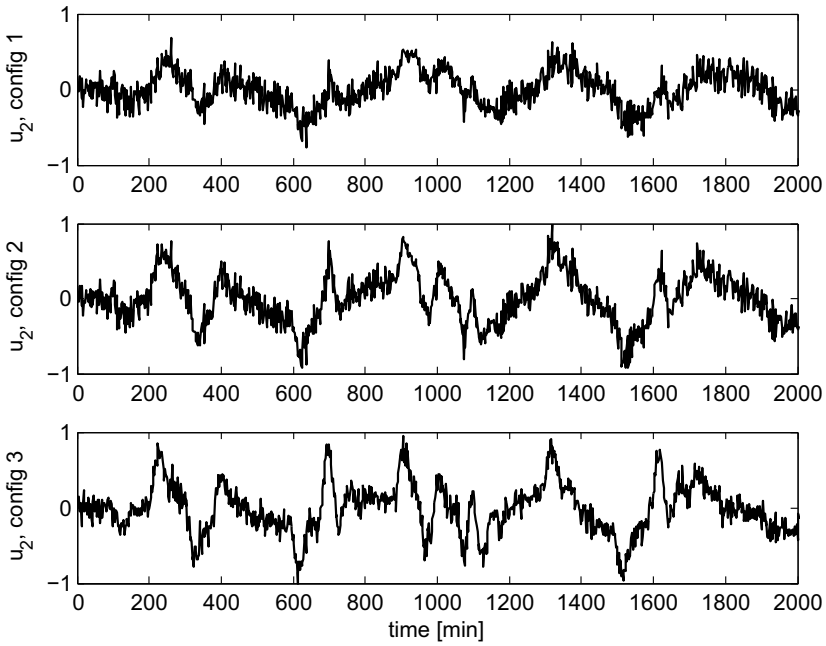


Figure 7.12: Controller output u_2 for varying gains. The gain increases from top to bottom of this figure

7.9 Signal to noise ratios

Figures 7.13 and 7.14 show the simplified signal to noise ratios for the controller outputs u_1 and u_2. In this case, the signal to noise ratio of u_1 is a bit lower for low frequencies when the gain of both controllers are increased, while the signal to noise ratio for u_2 is significantly higher for the first half of the frequency spectrum (for increased gain).

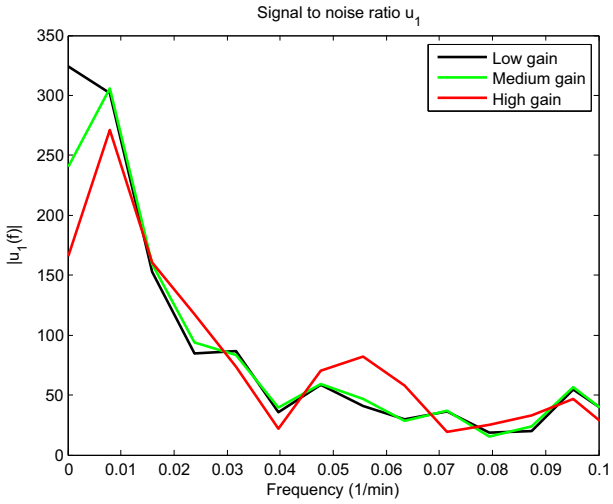


Figure 7.13: Signal to noise ratio for u_1

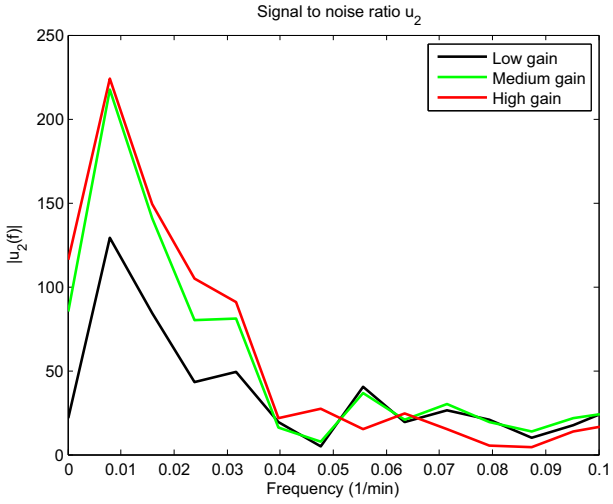


Figure 7.14: Signal to noise ratio for u_2

CHAPTER 8

DISCUSSION

In this chapter, a discussion on the results from the simulation parts will be given (chapters 6 and 7).

8.1 Matlab case studies

In chapter 6, two different multivariable LTI systems under decentralized PI control were simulated. The open-loop systems are

$$\mathscr{S}_1 = \frac{1}{2s+1}\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \tag{8.1}$$

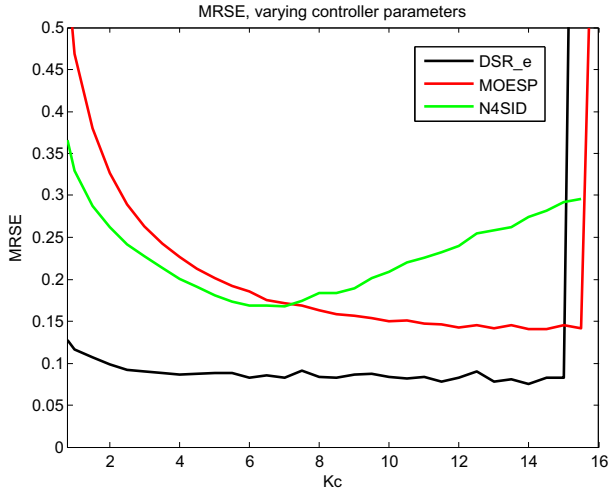
$$\mathscr{S}_2 = \begin{bmatrix} \frac{2s-1}{(3s+1)(s+1)} & \frac{1}{2s+1} \\ \frac{1}{50s+1} & \frac{2s+1}{(70s+1)(30s+1)} \end{bmatrix} \tag{8.2}$$

System $\mathscr{S}_1$

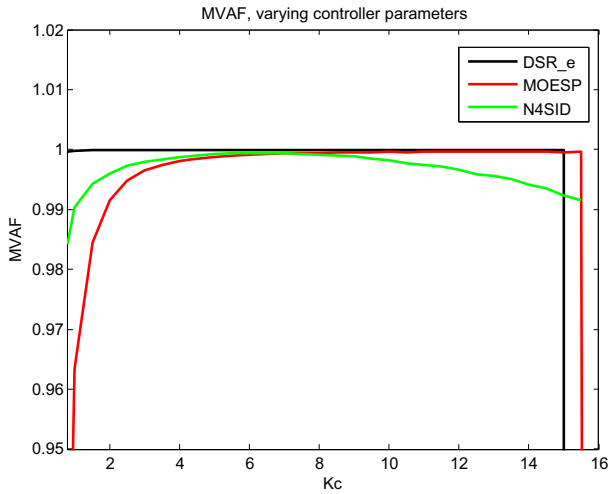
The identification experiments performed on system $\mathscr{S}_1$ showed clearly that increasing controller gains gave better performance of the identified models. This is for the cases when measurement noise was present in the loop. When only process noise was present, the results did not show big differences between the different configurations. The signal to noise ratio illustrated in figure 6.12 shows that the input power was significantly increased with the controller gain, which gives higher signal to noise ratio.

The open-loop methods MOESP and N4SID were affected most by increased controller gains, but the closed-loop method DSR_e also gave models with slightly better performances when the controller gains were increased. To show this more clearly, a brute force experiment was made by varying the controller gains in steps of 0.5 ($K_{c1} = K_{c2}$), and running 30 simulations per gain configuration. These data

are generated with both measurement noise and process noise in the loop, hence the number of simulations. The result is shown in figure 8.1. The trend is a bit



(a) MRSE



(b) MVAF

Figure 8.1: Performance of identified models for system $\mathscr{S}_1$ for increasing controller gains, $K_c = K_{c1} = K_{c2}$

different for the three methods at hand. This may be explained by the fact that this experiment also depends on the algorithm input parameters, which are fixed in this case.

For the MOESP method, the performance keeps getting better with increasing controller gain, but at one point the gain is too high ($Kc1 = Kc2 = 15.5$). As a consequence, both performance indices are dramatically worse at this point. After

this point, the method kept returning unstable models.

For the N4SID method, there seems to exist a set of equal controller parameters that minimizes the MRSE index, and maximizes the MVAF index, in this case at about $K_{c1} = K_{c2} = 6.5$. Same as for the MOESP method, the method returned unstable models for gains at 15.5 or higher.

The DSR_e method was not as sensitive as the other methods to the varying gain, but the plot shows some improved performance for higher controller gains. As for the other methods, too high controller gains resulted in unstable models.

System $\mathscr{S}_2$

For system $\mathscr{S}_2$, the main problem seemed to be to identify the slow time constants of the system. An example of this is shown in figure 8.2, where the model outputs are plotted along with the outputs from the system. For the example at hand, the high gain configuration for system $\mathscr{S}_2$ was applied, and both process noise and measurement noise was present in the identification data. As seen by the figure, the MOESP method failed to identify the system, while the N4SID method gave pretty bad results in terms of reproducing output y_2. Nevertheless, N4SID managed to reproduce output y_1 pretty good. DSR_e reproduced y_1 very good, and y_2 pretty good considering the relatively low signal to noise ratio.

When the performance indices are calculated, the error between all system outputs and all model outputs are calculated into a scalar value, and all output errors are equally worthy. For this example, it is seen by examining figure 8.2 that it was reproducing the slow output y_2 that rendered problems for the two methods that managed to identify the system.

When there was no noise in the loop, all methods gave perfect performance indices. This contributes to confirm that all correct subspace methods give perfect deterministic models for noise-free closed-loop data, provided that the identification experiment is informative.

The Monte Carlo simulations performed on this system showed the following:

- The MOESP method failed to identify a model with satisfying performance for all controller configurations when there was either process or measurement noise in the loop. This shows that this method is not trustworthy for closed-loop identification when noise is present in the data record, and the signal to noise ratio is relatively low.

- The N4SID method, being an open-loop method, actually managed to produce models with good performance when there was only process noise in the loop. When there was measurement noise in the loop as well, the results were also very good for this method, at least for controller configurations 1 and 2. However, the N4SID method tended to produce unstable models for some noise realizations, because the slowest time constant was estimated to be slightly outside the unit circle.

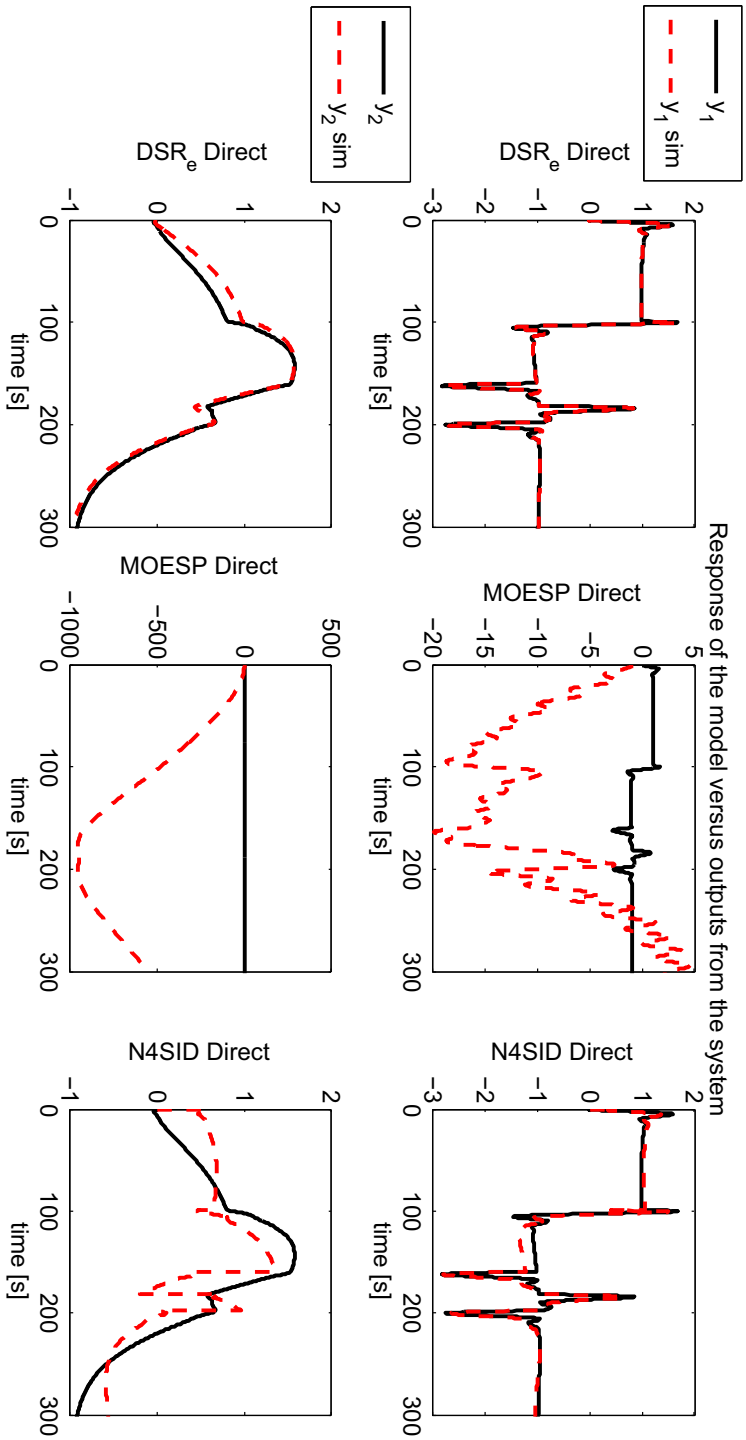


Figure 8.2: Example that shows the problem of identifying slow time constants of system $\mathscr{S}_2$

- The DSR_e method returned models with satisfying performance for all noise configurations. This method was least affected by the controller parameters, but it is still observed that the performance is increased from tuning configuration 1 to 2 (increased controller gain). For tuning configuration 1 with both measurement noise and process noise, the N4SID produced a better model than DSR_e, but the DSR_e method was most consistent on producing reliable models for the other controller and noise configurations.

- Tuning configuration 2 (medium gain) gave models with the best performance in this case. As shown by the signal to noise ratio in figure 6.16, this configuration gave higher signal to noise ratio for u_2 than configuration 3 for a large portion of the frequency spectrum. This is true even though the controller gain was higher for controller C_1 in configuration 3 than configuration 2.

8.2 Debutanizer case study

In this chapter, a nonlinear process was subject for identification in closed loop. The input design consisted of PRBS sequences on the references to the controllers, where one input was stepped at a time. The identification was performed when there was white noise present on both output measurements and controller outputs. The validation experiment was generated without noise, so that a comparison between the noisefree outputs from the "real" process and the outputs from the identified models was rendered possible.

As shown by the performance indices for the different models, the performance of the identified models was better for increasing gain, and best for tuning configuration 3. When the gain was too high though, both gains the double of configuration 3, the performance of the identified models was drastically reduced. The open-loop methods MOESP and N4SID produced unstable models, while the DSR_e method produced a model with worse performance than any of the other three configurations. As for the systems $\mathscr{S}_1$ and $\mathscr{S}_2$, there seems to be a limit to how much the gain can be increased before the performance of the identified models is reduced.

The DSR_e method gave models with best performances in all cases, which shows that this method has an advantage to the other two methods in being able to remove the noise terms that is correlated with the inputs in the data.

8.3 Comparing the results

The results obtained from studying the three different systems in chapter 6 and 7 yield some similar results. When controller gains are increased, and the identification data is corrupted by process- and measurement noise, the identification performance using closed-loop data is increased for the DSR_e method and the

N4SID method. The MOESP method also gives better performance in some cases, but these results are not as consistent as for the other two methods. These results however, are true up to certain values for the controller gains.

As shown for both system $\mathscr{S}_1$, $\mathscr{S}_2$, and the debutanizer process, the identification performance was degraded for too high a value for the gains. A possible explanation to this is that when the controller gains are increased, frequencies in the input signals will be more concentrated around the achievable bandwidth of the controller, which produces system outputs with more information of the frequency response around this bandwidth frequency. If the controller gains are pushing this limit, the controller will be so aggressive that it dominates the open-loop dynamics, and more of the controller is seen on the identified models.

In addition, the signal to noise ratios will come into play when there is noise present in the loop. As discussed in section 3.3, Ljung (1999) mentions on p. 434 that the part of the system input that originates from the feedback has no information value when it comes to identifiying the open-loop dynamics. On the other hand, it is the part of u (controller output, system input) that stems from the reference signal that will reveal information from the open-loop system, and give a lower signal to noise ratio. In this context, it is explained why the models identified with high gains in the feedback loops performed better than those identified with the low gains in the loops. Consider the controller equation that have been used throughout the thesis:

$$u(s) = \begin{bmatrix} C_1(s) & 0 \\ 0 & C_2(s) \end{bmatrix} (r - y) \tag{8.3}$$

The same reference sequences are used when comparing all the methods, for all controller parameter configurations. Notice that u consists of two different signals, namely the part that stems from the feedback (y), and the part that stems from the reference signal r. When the reference signal r is manipulated, it is the controller gain that determines how much the external reference signal is amplified. As seen by equation (3.5), the amplification of the reference signal will directly lower the "noise to signal ratio", or equivalently increase the signal-to-noise ratio. As stated in Ljung (1999), it is this ratio that determines how well the open-loop transfer function may be estimated when there is noise present in the loop.

CHAPTER 9

CONCLUSIONS AND FURTHER WORK

In this chapter, conclusions on the work of this tesis will be drawn, and suggestions to further work will be stated.

9.1 Conclusions

The purpose of this thesis is to investigate the performance of different subspace identification methods when the input-output data used in the algorithms is collected in closed-loop. In addition, the significance of varying controller parameters are investigated in order to compare performances of the identified models. Considering the subspace methods, two are originally intended for open-loop data (MOESP and N4SID), and one is designed to cope with closed-loop data (DSR_e).

As expected from the theory on the subspace methods, the closed-loop method gives in general more consistent and reliable models for different experimental conditions than the open-loop methods. It is shown though, that the open-loop methods can perform better if the gain of the existing controllers are increased.

The results from this work show quantitatively that tuning of exisiting controllers have a significant impact on the performance of the identified models. It is shown that when controller gains are increased, controller outputs with higher signal to noise ratios are generated, and the information content of these signals is higher because of the amplification of the input reference signals. This is believed to be the main reason why data produced by the relative high gain systems yield identified models with better performance than for systems with lower gain, since noise that corrupts the system will be dominated by the power of these controller outputs. It is also shown by simulations that there exists a limit to how much the gains can be increased before the performance of the identified models are de-

graded. To find this limit analytically, or by numerical approximations, though, is considered to be a hard problem. Luckily, numerous simulations tools for simulating dynamic systems exists, and may be used for running multiple identification experiments with different controller gains for comparison.

The field of closed-loop identification has drawn much attention during the last decades. One of the main benefits of running identification experiments on processes without breaking the already existing feedback loops, is that processes can operate within normal operating conditions during the experiment. This ensures both safety and maintenance of economic objectives. The author believes that closed-loop identification deserves to be looked even more into in the future, motivated by the need for model-based controllers for large, complicated systems.

9.2 Further work

As stated earlier in the thesis, finding good input parameters to the different subspace algorithms can be a cumbersome problem. During this work, the method of trial and error has been applied to find which input parameters that gave relatively good performance for the different methods. An interesting topic for further work could be to analyse the effect of input parameters to the different subspace algorithms for fixed data sets. The goal would be to develop easy ”rules of thumb” on how to choose these parameters given the input-output data.

As for the effect of increasing controller gain, it would be interesting to run experiments on a relatively simple, physical plant where the noise properties are unknown, to verify the results from this thesis.

BIBLIOGRAPHY

Bakke, M. (2008). A simulated debutanizer case study for teaching advanced process control. Master's thesis, NTNU.

Ben C. Juricek, Dale E. Seborg, W. E. L. (2001). Identification of the Tennessee Eastman challenge process with subspace methods. *Control Engineering Practice*, 9, 1337–1351.

Chen, C.-T. (1999). *Linear system theory and design*. Oxford University Press.

den Hof, P. V. (1998). Closed-loop issues in system identification. *Control*, 22, 173–186.

den Hof et. al, P. V. (1995). Identification of normalised coprime plant factors from closed-loop experiment data. *European Journal of Control*, 1, 62–74.

Erlendur Karlsson, Torsten Söderström, P. S. (2000). The cramér-rao lower bound for noisy input-output systems. *Signal Processing*, 80, 2421–2447.

Fliess, M. & Sira-Ramirez, H. (2008). *Identification of Continuous-time Models from Sampled Data*, chapter 13, (pp. 363–391). Springer London.

Forssell, U. & Ljung, L. (1999). Closed-loop identification revisited. *Automatica*, 35, 1215–1241.

Henriksen, R. (2001). Subspace identification of linear systems. Department of electrical and computer engineering, Iowa state university.

Hugues Garnier, L. W. & Young, P. C. (2008). *Direct identification of continuous-time models from sampled data: issues, basic solutions and relevance*, chapter 1, (pp. 1–29). Springer London.

I. Gustavsson, L. Ljung, T. S. (1977). Identification of processes in closed loop - identifiability and accuracy aspects. *Automatica*, 13, 59–75.

Jacobsen, E. W. & Skogestad, S. (1993). Identification of dynamic models for ill-conditioned plants - a benchmark problem. *European Control Conference*.

Katayama, T. & Tanaka, H. (2007). An approach to closed loop supspace identification by orthogonal decomposition. *Automatica*, 43, 1623–1630.

Katrien De Cock, Bart De Moor, K. L. (2003). Subspace identification methods. *Department of electrical engineering, ESAT-SCD Leuven Belgium*.

Kristinsson, K. & Dumont, G. A. (1992). System identification and control using genetic algorithms. *IEEE Transactions on systems, man and cybernetics*, 22.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall.

Ljung, L. & McKelvey, T. (1996). Subspace identification from closed loop data. *Signal Processing*, 52, 209–215.

Muharar, R. (2006). Linear time invariant systems identification by a subspace method: MOESP algorithm. *Tahun*, 5.

Oscar A.Z Sotomayor, Song Won Park, C. G. (2003). Multivariable identification of an activated sludge process with subspace-based algorithms. *Control Engineering Practice*, 11, 961–969.

Overschee, P. V. & Moor, B. D. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30, 75–93.

Rivera, D. E. (1998). Closed-loop and MIMO identification. In *Intro to system identification*.

Ruscio, D. D. (1996). Combined deterministic and stochastic system identification and realization: DSR - a subspace approach based on observations. *Modelling, identification and control*, 17, 193–230.

Ruscio, D. D. (2003). Subspace system identification of the Kalman filter. *Modelling, identification and control*, 24, 125–157.

Skogestad, S. (2007). The dos and don'ts of distillation column control. *Trans IChemE Part A*, 85, 13–23.

Skogestad, S. & Postlethwaite, I. (2005). *Multivariable Feedback Control - Analysis and design*. John Wiley & Sons, Ltd.

Strang, G. (2006). *Linear algebra and its applications*. Thomson Brooks/Cole.

UniSim (2005). *UniSim Reference Guide*. Honeywell, R350 edition.

Verhaegen, M. (1993). Application of a subspace model identification technique to identify LTI systems operating in closed-loop. *Automatica*, 29, 1027–1040.

Verhaegen, M. (1994). Identification of the deterministic part of MIMO state space models in innovations form from input-output data. *Automatica*, 30, 61–74.

Viberg, M. (1995). Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31, 1835–1851.

Whorton, M. S. (2004). Closed-loop system identification with genetic algorithms. *TD54 Guidance, Navigation and control systems group.*

Ziegler, J. & Nichols, N. (1942). Optimum settings for automatic controllers. *The American Society of mechanical Engineers.*

APPENDIX A

THE GENERAL SUBSPACE ALGORITHM

This algorithm is a summary of the algorithm given in Ljung (1999), p. 340-351.

From the general expression for a discrete state space system, we may write the k-step ahead expression for the output as

$$\begin{aligned} y(t+k) &= CA^k x(t) + CA^{k-1}Bu(t) + CA^{k-2}Bu(t+1) + \dots \\ &\quad + CBu(t+k-1) + Du(t+k) \\ &\quad + CA^{k-1}w(t) + CA^{k-2}w(t+1) + \dots \\ &\quad + Cw(t+k-1) + v(t+k) \end{aligned} \tag{A.1}$$

The first task in the subspace algorithm is to generate an estimate of the *extended observability matrix* of the system:

$$\mathcal{O}_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} \tag{A.2}$$

In order to show how $\mathcal{O}_r$ is approximated, it is useful to introduce some new matrices and vectors to keep the notation compressed. Define the matrix S_r as

$$S_r = \begin{bmatrix} D & 0 & \dots & 0 & 0 \\ CB & D & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{r-2}B & CA^{r-3}B & \dots & CB & D \end{bmatrix} \tag{A.3}$$

and the vector $V(t)$ defined by its k:th block components as

$$\begin{aligned} V_k(t) &= CA^{k-2}w(t) + CA^{k-3} + \dots \\ &= Cw(t+k-2) + v(t+k-1) \end{aligned} \tag{A.4}$$

Writing equation (A.1) in compressed form using these matrices yields

$$Y_r(t) = \mathcal{O}_r x(t) + S_r U_r(t) + V(t) \tag{A.5}$$

The input and output vectors are defined as

$$\begin{aligned} U_r(t) &= \begin{bmatrix} u(t) & u(t+1) & \dots u(t+r-1) \end{bmatrix}^\top \\ Y_r(t) &= \begin{bmatrix} y(t) & y(t+1) & \dots y(t+r-1) \end{bmatrix}^\top \end{aligned} \tag{A.6}$$

The idea is to correlate both sides of (A.5) with quantities that eliminate the term with $U_r(t)$ and make the noise influence from V disappear asymptotically. Further, define the matrices

$$\begin{aligned} \mathbb{Y} &= \begin{bmatrix} Y_r(1) & Y_r(2) & \dots & Y_r(N) \end{bmatrix} \\ \mathbb{X} &= \begin{bmatrix} x(1) & x(2) & \dots & x(N) \end{bmatrix} \\ \mathbb{U} &= \begin{bmatrix} U_r(1) & U_r(2) & \dots & U_r(N) \end{bmatrix} \\ \mathbb{V} &= \begin{bmatrix} V(1) & V(2) & \dots & V(N) \end{bmatrix} \end{aligned} \tag{A.7}$$

It is possible to write equation (A.5):

$$\mathbb{Y} = \mathcal{O}_r \mathbb{X} + S_r \mathbb{U} + \mathbb{V} \tag{A.8}$$

The $N \times N$ matrix

$$\Pi^\perp_{\mathbb{U}^\top} = I - \mathbb{U}^\top (\mathbb{U}\mathbb{U}^\top)^\dagger \mathbb{U} \tag{A.9}$$

defines the orthogonal projection of the matrix $\mathbb{U}$. Note that $\mathbb{U}\Pi^\perp_{\mathbb{U}^\top} = 0$. This is used to remove the $\mathbb{U}$-term from equation (A.5), avoiding the need of an estimate of the matrix S_r. This is seen by multiplication from the right by $\Pi^\perp_{\mathbb{U}^\top}$

$$\mathbb{Y}\Pi^\perp_{\mathbb{U}^\top} = \mathcal{O}_r \mathbb{X}\Pi^\perp_{\mathbb{U}^\top} + \mathbb{V}\Pi^\perp_{\mathbb{U}^\top} \tag{A.10}$$

In order to also remove the noise term from equation (A.5) another matrix $\mathbf{\Phi}$ is defined as

$$\mathbf{\Phi} = \begin{bmatrix} \varphi_s(1) & \varphi_s(2) & \dots & \varphi_s(N) \end{bmatrix} \tag{A.11}$$

The vector $\varphi_s(t)$ is a vector of *instrumental variables* (IV), and a typical choice for this is

$$\varphi_s(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-s_1) \\ u(t-1) \\ \vdots \\ u(t-s_2) \end{bmatrix} \tag{A.12}$$

By multiplying equation (A.10) from the right by $\mathbf{\Phi}^\top$ and normalizing by N, G is defined as:

$$G = \frac{1}{N}\mathbb{Y}\Pi^{\perp}_{\mathbb{U}^\top}\mathbf{\Phi}^\top = \mathcal{O}_r \underbrace{\frac{1}{N}\mathbb{X}\Pi^{\perp}_{\mathbb{U}^\top}\mathbf{\Phi}^\top}_{\tilde{T}_N} + \underbrace{\frac{1}{N}\mathbb{V}\Pi^{\perp}_{\mathbb{U}^\top}\mathbf{\Phi}^\top}_{V_N} \tag{A.13}$$

The goal is now that the vector $\varphi_s(t)$ assures

$$\lim_{N\to\infty} V_N = \lim_{N\to\infty} \mathbb{V}\Pi^{\perp}_{\mathbb{U}^\top}\mathbf{\Phi}^\top = 0 \tag{A.14a}$$

$$\lim_{N\to\infty} \tilde{T}_N = \lim_{N\to\infty} \Pi^{\perp}_{\mathbb{U}^\top}\mathbf{\Phi}^\top = \tilde{T} \text{ has full rank } n \tag{A.14b}$$

The family of subspace algorithms may be summarized by the following (Ljung (1999)):

1. From the input-output data, form

$$G = \frac{1}{N}\mathbb{Y}\Pi^{\perp}_{\mathbb{U}^\top}\mathbf{\Phi}^\top \tag{A.15}$$

   with matrices defined by (A.6), (A.7), (A.9), (A.11) and (A.12).

2. Select two weighting matrices, W_1 with dimension $rp \times rp$ and invertible, and W_2 with dimension $(ps_1 + ms_2) \times \alpha$ and perform SVD

$$\hat{G} = W_1 G W_2 = USV^\top \approx U_1 S_1 V_1^\top \tag{A.16}$$

   The approximation in (A.16) is due to the fact that only the n most significant singular values in S is kept, and the rest is set to 0.

3. Select a full rank matrix R and define the $rp \times n$ matrix $\hat{\mathcal{O}}_r = W_1^{-1}U_1 R$. To obtain estimates of the system matrices A and C, solve

$$\hat{C} = \hat{\mathcal{O}}_r(1:p, 1:n) \tag{A.17a}$$

$$\hat{\mathcal{O}}_r(p+1:pr, 1:n) = \mathcal{O}_r(1:p(r-1), 1:n)\hat{A} \tag{A.17b}$$

   for $\hat{C}$ and $\hat{A}$. Notice that the notation used here is MATLAB-like.

4. Estimate $\hat{B}$, $\hat{D}$ and $\hat{x}_0$ from the linear regression problem

$$\underset{B,D,x_0}{\arg\min} \frac{1}{N}\sum_{t=1}^{N} ||y(t) - \hat{C}(qI - \hat{A})^{-1}Bu(t) - Du(t) \tag{A.18}$$

$$-\hat{C}(qI - \hat{A})^{-1}x_0\delta(t)||^2 \tag{A.19}$$

5. In order to add a noise model, form $\hat{\mathbb{X}}$ as $\hat{\mathbb{X}} = R^{-1}U_1^\top\hat{\mathbb{Y}} = \begin{bmatrix}\hat{x}_1 & \hat{x}_2 & \dots & \hat{x}_N\end{bmatrix}$ and estimate the noise contribution from

$$\begin{aligned} w(t) &= \hat{x}(t+1) - \hat{A}\hat{x}(t) - \hat{B}u(t) \\ v(t) &= y(t) - \hat{C}\hat{x}(t) - \hat{D}u(t) \end{aligned} \tag{A.20}$$

APPENDIX B

IDENTIFICATION PERFORMANCE

The subspace identification methods are evaluated by the two performance indices given in section 2.5, MRSE and MVAF. The mean values of these indices from the different Monte Carlo simulations performed in chapter 6 are listed in tables B.1 and B.2.

The identification experiment is considered failed if both indices are outside the range used in the plots in chapter 6, i.e. MRSE $\in (0\%, 100\%)$ and MVAF $\in (85\%, 100\%)$. If one of these indices are inside the range, but not the other one, the index outisde this given range will be listed as OOR (Out Of Range). In the cases of only measurement noise and only process noise, only two different controller gains were used to illustrate the effect of increasing the gain.

Table B.1: Performance indices from identification of system $\mathscr{S}_1$ in chapter 6. PN = Process Noise, MN = Measurement Noise, LG = Low Gain, MG = Medium Gain, HG = High Gain.

PN	MN	Method	MRSE/MVAF (LG)	MRSE/MVAF (MG)	MRSE/MVAF (HG)
No	No	DSR_e	0%/100%	0%/100%	0%/100%
		MOESP	0%/100%	0%/100%	0%/100%
		N4SID	0%/100%	0%/100%	0%/100%
No	Yes	DSR_e	12.5%/99.9%	8.9%/100%	—
		MOESP	49.5%/95.7%	34.7%/98.3%	—
		N4SID	37%/98.02%	27%/99.4%	—
Yes	No	DSR_e	11.5%/100%	11.4%/100%	—
		MOESP	14.5%/99%	15.5%/99.5%	—
		N4SID	11.5%/100%	11.5%/100%	—
Yes	Yes	DSR_e	13%/99.99%	10.5%/100%	10%/100%
		MOESP	51%/94%	33.5%/98%	24.5%/99%
		N4SID	36.5%/98%	27%/99%	22%/99.4%

Table B.2: Performance indices from identification of system $\mathscr{S}_2$ in chapter 6. PN = Process Noise, MN = Measurement Noise, OOR = Out Of Range LG = Low Gain, MG = Medium Gain, HG = High Gain.

PN	MN	Method	MRSE/MVAF (LG)	MRSE/MVAF (MG)	MRSE/MVAF (HG)
No	No	DSR_e	0%/100%	0%/100%	0%/100%
		MOESP	0%/100%	0%/100%	0%/100%
		N4SID	0%/100%	0%/100%	0%/100%
No	Yes	DSR_e	36.5%/97.4%	29.3%/98.1%	27.2%/98.4%
		MOESP	failed	failed	failed
		N4SID	50%/91.8%	48%/92.2%	59.8%/88%
Yes	No	DSR_e	22.5%/99.1%	11.6%/99.99%	16.2%/99.6%
		MOESP	failed	failed	89.3%/OOR%
		N4SID	14.6%/99.96%	13.7%/99.96%	16.2%/99.8%
Yes	Yes	DSR_e	19%/99.1%	12.1%/99.9%	27%/98.1%
		MOESP	failed	failed	failed
		N4SID	14.2%/99.8%	12.1%/99.9%	55.3%/90.4%

APPENDIX C

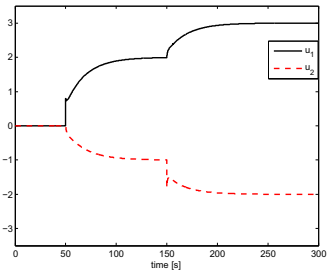
AUXILLIARY PLOTS FOR CHAPTER 6

C.1 System $\mathscr{S}_1$

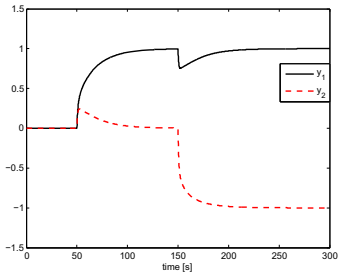
This section shows closed loop stepresponses for system $\mathscr{S}_1$ from section 6.3. Two different tuning configurations are shown for the closed-loop case to illustrate the difference between "smooth" and "tight" tuning configuration. The different configurations are

- **Smooth tuning** - $K_{c1} = 0.8$, $K_{c2} = 0.8$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$
- **Tight tuning** - $K_{c1} = 3$, $K_{c2} = 3$, $\tau_{i1} = 2$ and $\tau_{i2} = 2$

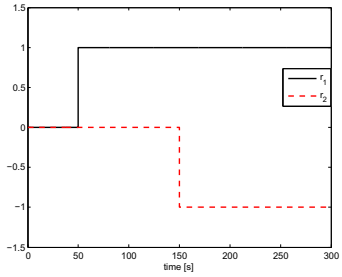
See figures C.1 and C.2 for the different responses.



(a) Controller command

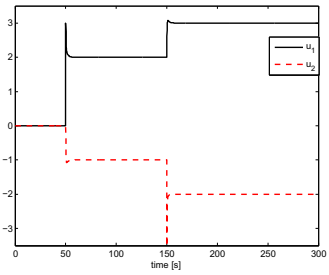


(b) Outputs

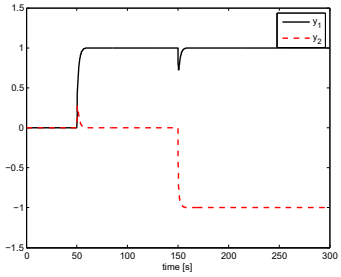


(c) References

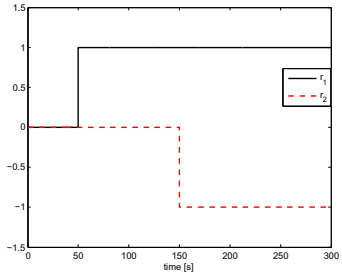
Figure C.1: Step response of the system $\mathscr{S}_1$ with tuning configuration 1



(a) Controller command



(b) Outputs



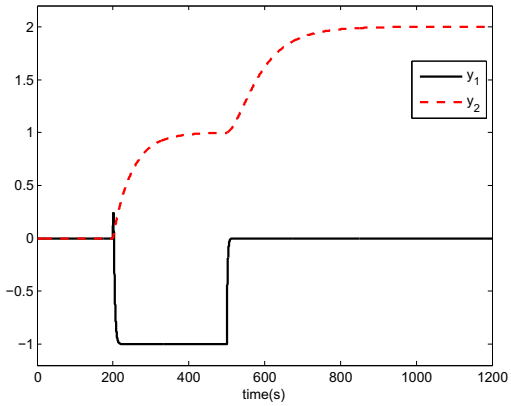
(c) References

Figure C.2: Step response of the system $\mathscr{S}_1$ with tuning configuration 2

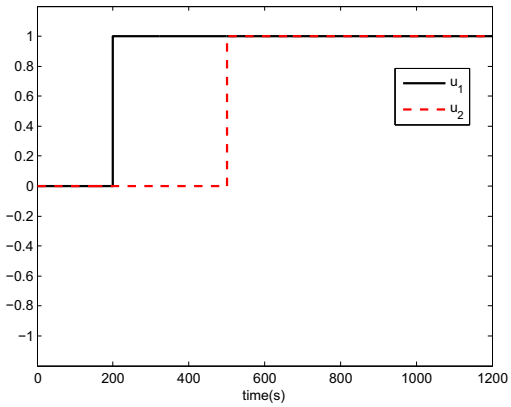
C.2 System $\mathscr{S}_2$

In this section, both open- and closed-loop stepresponses are shown for the system $\mathscr{S}_2$ from section 6.4. The tuning configurations for the closed-loop case are

- **Configuration 1** - $K_{c1} = -0.2$, $K_{c2} = 0.5$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$
- **Configuration 2** - $K_{c1} = -0.4$, $K_{c2} = 1$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$
- **Configuration 3** - $K_{c1} = -1$, $K_{c2} = 1$, $\tau_{i1} = 5$ and $\tau_{i2} = 70$

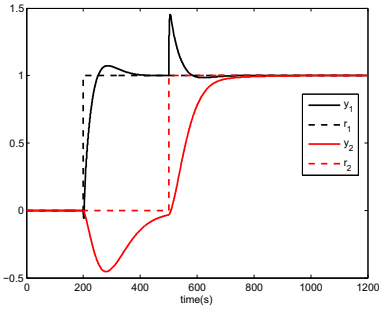


(a) Outputs y_1 and y_2

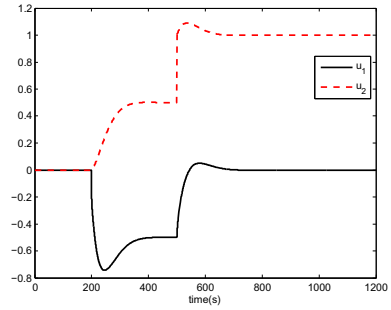


(b) Inputs u_1 and u_2

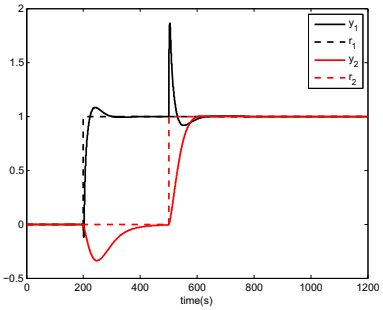
Figure C.3: Open-loop step response for system $\mathscr{S}_2$



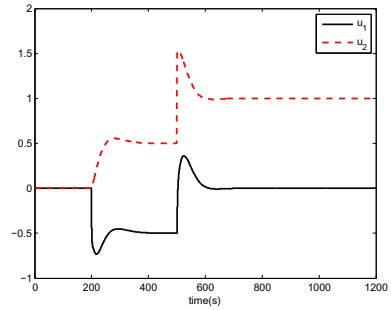
(a) Outputs and references, configuration 1



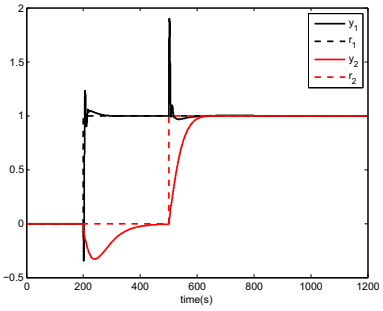
(b) Controller command, configuration 1



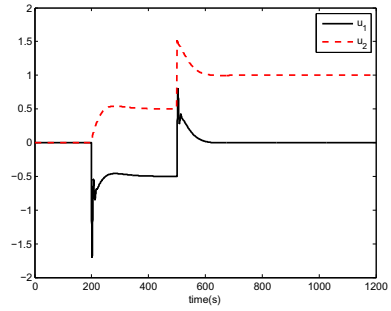
(c) Outputs and references, configuration 2



(d) Controller command, configuration 2



(e) Outputs and references, configuration 3

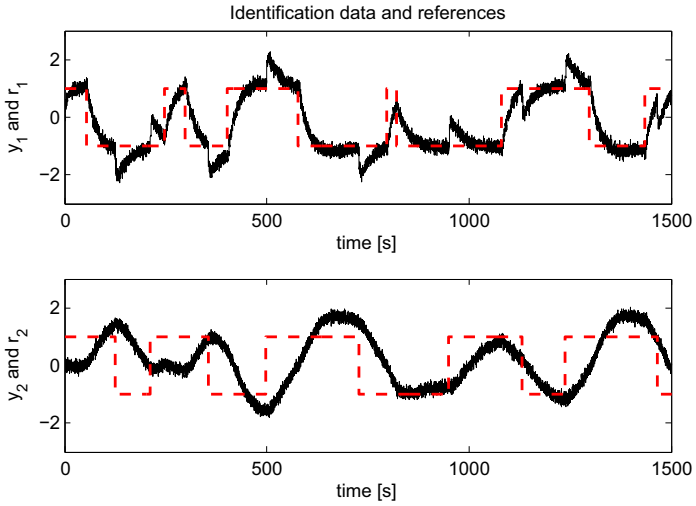


(f) Controller command, configuration 3

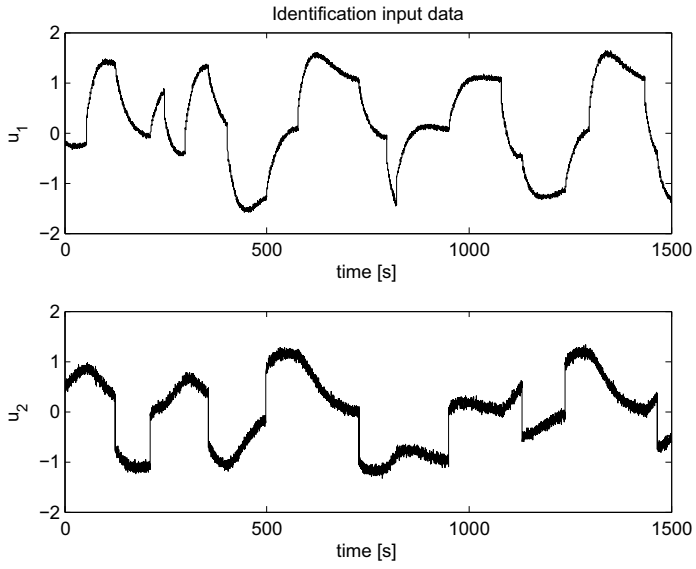
Figure C.4: Closed-loop step responses of the system $\mathscr{S}_2$ with different tuning configurations

C.3 Identification data for system $\mathscr{S}_2$

In this section, identification data from the experiments performed on system $\mathscr{S}_2$ is given. The data is generated with both process and measurement noise, and data for the three tuning configurations is shown in in figures C.5, C.6 and C.7. The purpose is to show that the different tuning configurations generate different input-output data.

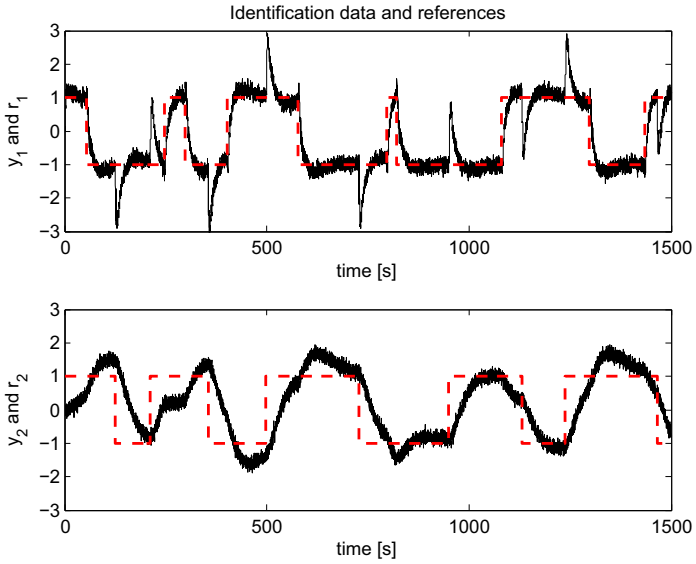


(a) Outputs and references, configuration 1

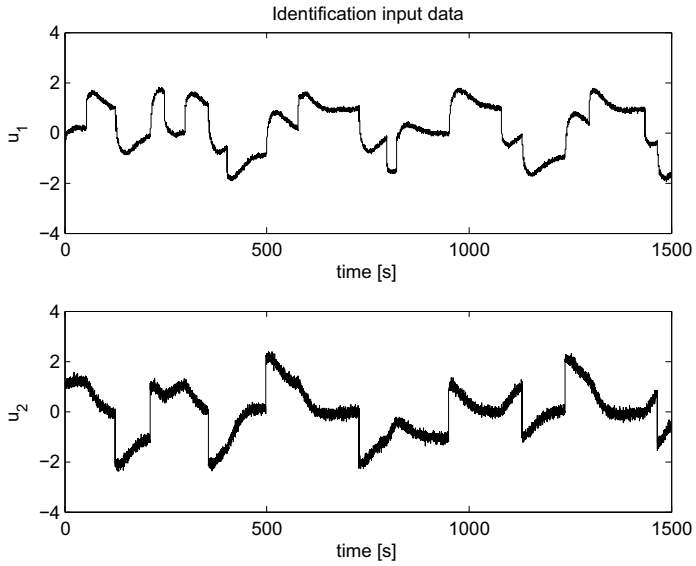


(b) Controller commands, configuration 1

Figure C.5: Identification data for tuning configuration 1

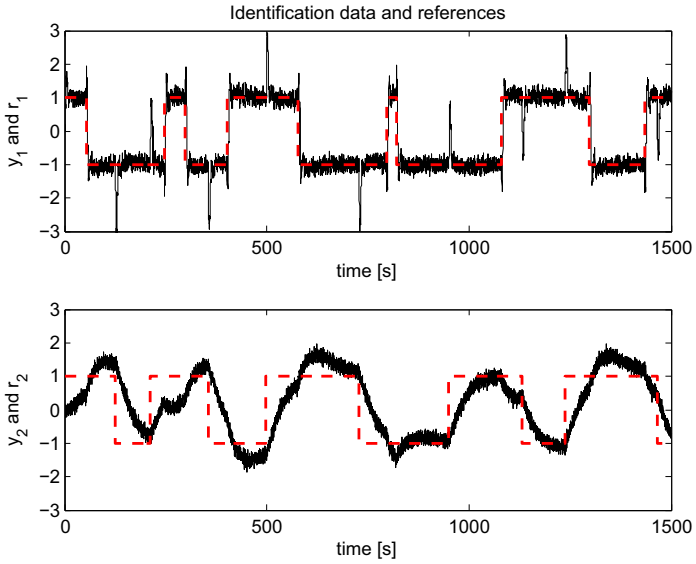


(a) Outputs and references, configuration 2

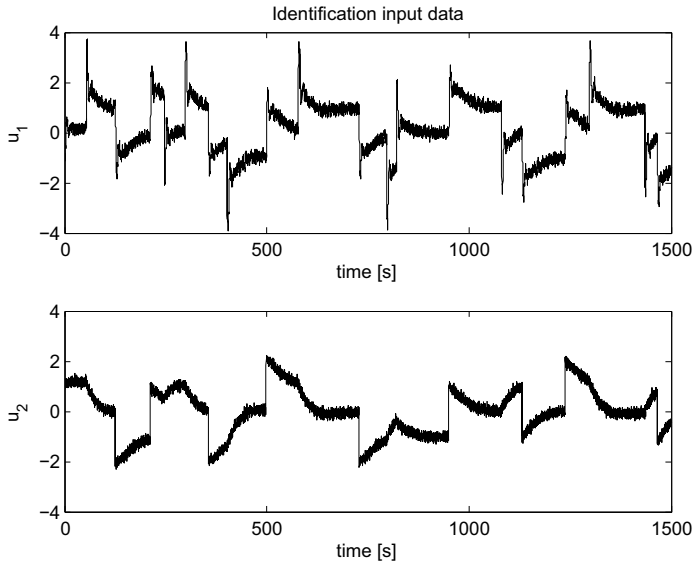


(b) Controller commands, configuration 2

Figure C.6: Identification data for tuning configuration 2



(a) Outputs and references, configuration 3



(b) Controller commands, configuration 3

Figure C.7: Identification data for tuning configuration 3