

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Closed Loop System Identification of a Torsion System

Master Thesis in Vehicular Systems
at The Institute of Technology in Linköping

Andreas Myklebust

LITH-ISY-EX--09/4223--SE

Linköping 2009



Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Closed Loop System Identification of a Torsion System

Master Thesis in Vehicular Systems
at The Institute of Technology in Linköping


Andreas Myklebust

LITH-ISY-EX--09/4223--SE

Handledare: **David Banjerdpongchai**
Dept. of Electrical Engineering, Chulalongkorn University

Examinator: **Lars Eriksson**
isy, Linköpings University

Linköping, 27 March, 2009

	Avdelning, Institution Division, Department Division of Vehicular Systems Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden	Datum Date 2009-03-27
---	---	--

Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN _____ ISRN LITH-ISK-EX--09/4223--SE Serietitel och serienummer ISSN Title of series, numbering _____
--	---	---

URL för elektronisk version http://www.vehicular.isy.liu.se http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-17531	
--	--

Titel Title	Systemidentifiering av ett återkopplat torsionssystem Closed Loop System Identification of a Torsion System
Författare Author	Andreas Myklebust

Sammanfattning Abstract	<p>A model is developed for the Quanser torsion system available at Control Systems Research Laboratory at Chulalongkorn University. The torsion system is a laboratory equipment that is designed for the study of position control. It consists of a DC motor that drives three inertial loads that are coupled in series with the motor, and where all components are coupled to each other through torsional springs.</p> <p>Several nonlinearities are observed and the most significant one is an offset in the input signal, which is compensated for. Experiments are carried out under feedback as the system is marginally stable. Different input signals are tested and used for system identification. Linear black-box state-space models are then identified using PEM, N4SID and a subspace method made for closed-loop identification, where the last two are the most successful ones. PEM is used in a second step and successfully enhances the parameter estimates from the other algorithms.</p>
-----------------------------------	--

Nyckelord Keywords	nonlinear compensation, subspace, PEM
------------------------------	---------------------------------------

Abstract

A model is developed for the Quanser torsion system available at Control Systems Research Laboratory at Chulalongkorn University. The torsion system is a laboratory equipment that is designed for the study of position control. It consists of a DC motor that drives three inertial loads that are coupled in series with the motor, and where all components are coupled to each other through torsional springs.

Several nonlinearities are observed and the most significant one is an offset in the input signal, which is compensated for. Experiments are carried out under feedback as the system is marginally stable. Different input signals are tested and used for system identification. Linear black-box state-space models are then identified using PEM, N4SID and a subspace method made for closed-loop identification, where the last two are the most successful ones. PEM is used in a second step and successfully enhances the parameter estimates from the other algorithms.

Acknowledgments

This master thesis work was examined at the Division of Vehicular Systems under Department of Electrical Engineering at Linköping University. Whereas it has been carried out at the Division of Control Systems under the Department of Electrical Engineering at Chulalongkorn University in Bangkok.

I would like to thank Leif Johansson for informing me about and granting me the scholarship to go to Chulalongkorn. I also like to thank David Banjerdpongchai for taking me under his wings and making this thesis possible. I also have to thank Lars "Lasse" Eriksson for examining my thesis and valuable feedback. I want to thank all the funny moose at Suksit-Nives International House for taking my mind of the studies, sometimes a bit too much and often. More thanks go to my girlfriend Elin Julin for making me want to go back home to Sweden, otherwise I might still be working on the thesis. At last I like to thank anyone who feels forgotten.

Contents

1	Introduction	1
1.1	Objective	1
1.2	Earlier work	2
1.3	Outline	5
2	Torsion System	7
2.1	Physical System	7
2.1.1	Hardware Interface	7
2.1.2	Software Interface	8
2.2	Quanser Physical Model	8
2.3	Nonlinearities	10
2.3.1	Simulink Control Signal vs Real Input	10
2.3.2	Coulomb Friction	12
2.3.3	Asymmetric Sliprings	12
2.3.4	Motor/load slip	13
2.3.5	Saturations	13
3	Identification Algorithms	15
3.1	Stability and Closed Loop	15
3.2	Prediction Error Method	15
3.3	Numerical Algorithms for Subspace State Space System Identification	16
3.4	closed-loop Identification Algorithm	16
3.4.1	Notation	17
3.4.2	LQ-decomposition	18
3.4.3	Algorithm	20
4	Experiment Setup and Identification Process	23
4.1	Input Signal	23
4.2	Nonlinear Compensation	24
4.3	Controller and Saturation	25
4.3.1	PID controller	25
4.3.2	Control Signal Saturation	25
4.4	Identification Process	25
4.4.1	Model Fit	26
4.4.2	Identification Algorithm Verification	26

4.4.3	Input Determination	26
4.4.4	Parameter Determination	27
5	Experimental Results	29
5.1	Open Loop or Closed Loop	29
5.2	Nonlinear Compensation	31
5.3	Signal Type and Frequency	31
5.4	Magnitude of Input Signal	31
5.5	Controller	34
5.6	Sample Time	34
5.7	Number of States	36
5.8	VODMA Parameter i	37
5.9	Algorithm Comparison	39
6	Discussion and Conclusions	41
6.1	Identified Model	41
6.1.1	Linearization	41
6.1.2	Number of States	41
6.1.3	Model	42
6.2	Magnitude of Input Signal	44
6.2.1	Small Magnitudes	44
6.3	Algorithms	44
6.4	Summary	45
6.5	Future Work	45
	Bibliography	47
A	Torsion System Specifications	49
B	Algorithm Verification	51
B.1	Simulation and Results	51
B.2	Singular Values	53
C	Model for Torsion System with Increased Inertia	55
C.1	Model	55

Chapter 1

Introduction

A torsion system is a system where a motor of some kind creates torque, which in turn is used to rotate a load that is attached to the motor via an axle. The torsion system is called a flexible torsion system if any part, usually the axle, cannot be considered stiff, e.g. behaves like a torsion spring. With enough torque any axle will become flexible.

One example of this is found between the car engine and the tires of the car, the driveline. It consists of several parts which all are more or less flexible. For a sketch of a car driveline see figure 1.1. In fig 1.2 the flexibilities of the driveline can be seen as the differences in engine speed versus transmission speed and wheel speed. Moreover the driveline can be modeled in a similar fashion to that presented in section 2.2, where drive shafts, propeller shaft and transmission stands for the flexibilities, inertias and frictions. More reading concerning driveline modeling can be found in [3].

1.1 Objective

Since models are far from always accurate and system wear changes the system with time, controllers are required to be robust against model errors. With higher demands on controller performance more advanced controllers, that use as much system information as possible, are desirable. In order to develop new robust controllers and verify their robustness there must exist a model which can be used and altered. The aim of this thesis is to find sufficient models for a flexible torsion system test bench from Quanser, available at Control System Research Laboratory.

This particular system is meant for position control with relatively small angles, in other words when the angles are below $\pm 90^\circ$ but still clearly visible to the human eye. This has, of course, been kept in mind while carrying out the system identification.

As part of the work some different identification methods will be tested for their capabilities to correctly identify the torsion system, with a subsequent comparison of the different methods.

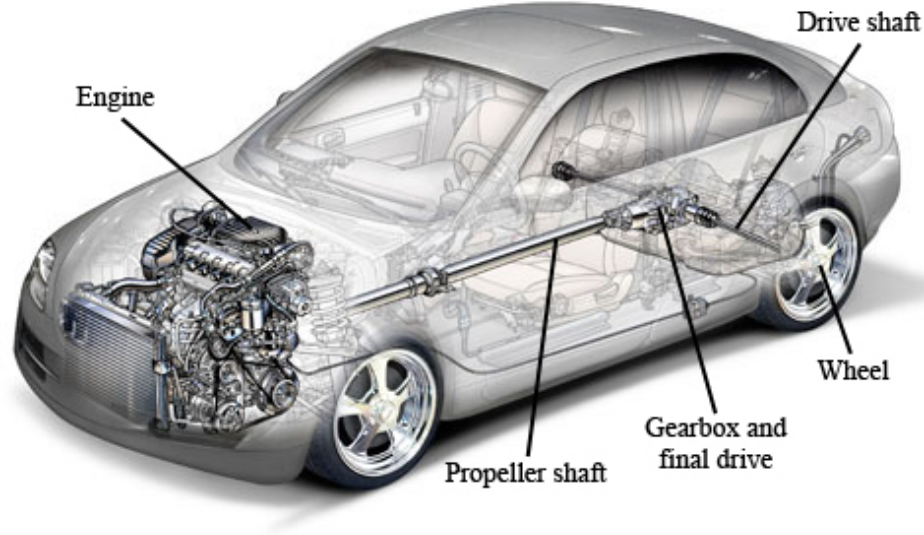


Figure 1.1. The different parts of a driveline and how they are built into a rear-wheel-drive car. The driveline is an example of a torsion system. All marked parts have inertia and friction. The parts between the engine and the wheels are flexible. Courtesy Beau and Alan Daniels, [2], explaining texts have been added afterwards.

1.2 Earlier work

Most work concerning closed-loop system identification were made in the late 1990's, where some of the big names were Lennart Ljung, Bart de Moor, Peter Van Overschee and Michel Verhaegen. In this thesis extensive use has been made of Van Overschee and De Moor's paper of 1997, [12], on system identification of closed-loop systems as one of the algorithms has been implemented. Apart from the just mentioned paper, Van Overschee and De Moor's earlier work of 1994, [10], about the same algorithms but this time for open-loop identification, has been used to increase the understanding of the algorithms. More technical details related to the methods presented in [10] and [12] can be found in the technical reports [9] and [11].

Other algorithms have been considered before the implementation of Van Overschee and De Moor's algorithm. For instance, there is Jitendra Tugnait and Yi Zhou's, [8], polyspectral solution of the closed-loop identification problem from 1998. Yet another solution by creating modified Output-Error and Box-Jenkins models were proposed in a paper by Urban Forssell and Lennart Ljung, [4], in year 2000.

Besides, the above mentioned papers some of the problems with closed-loop identification and how they affect earlier algorithms are described in Lennart Ljung and Tomas McKelvey's paper of 1996, [5].

In addition to all the earlier work made in the general area of closed-loop system

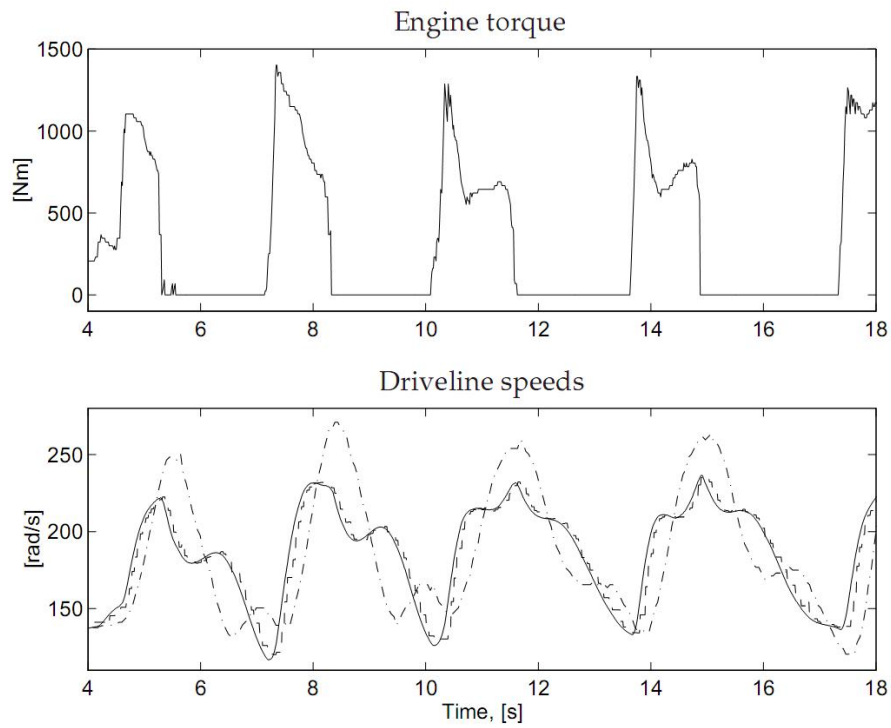


Figure 1.2. Logged data on the CAN-bus during tests with a Scania 144L truck. The transmission speed (dashed) and the wheel speed (dash-dotted) are scaled to engine speed in solid. The differences in speed indicates torsional effects in the driveline. Courtesy Eriksson and Nielsen, [3].

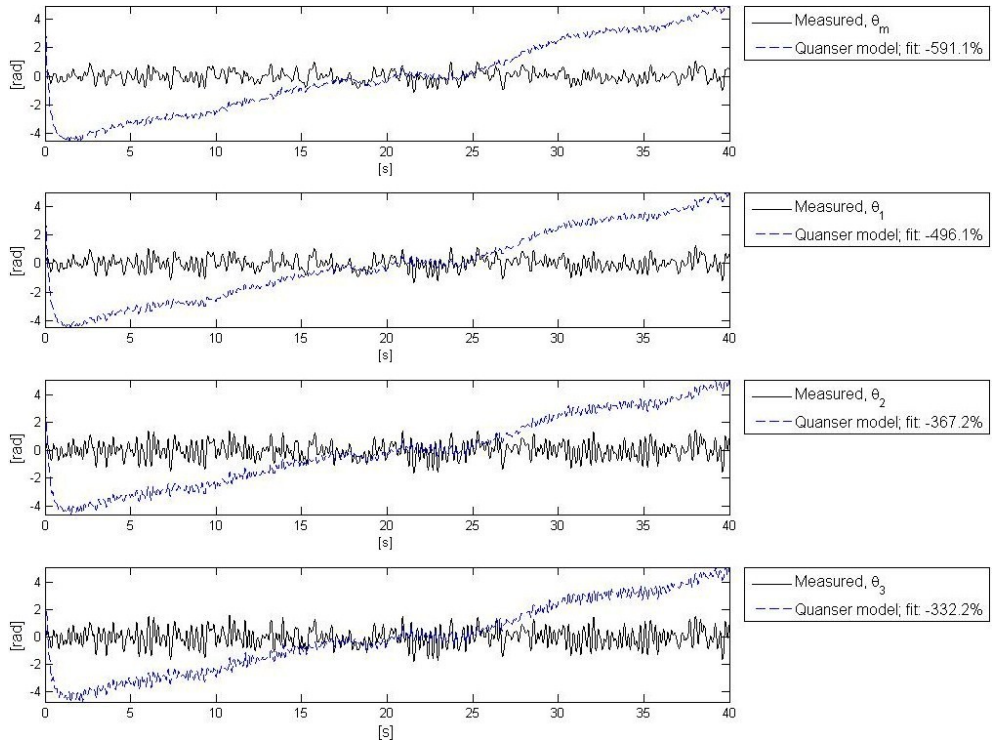


Figure 1.3. The model fit of the Quanser physical model when the system is fed with a random input signal. θ_m through θ_3 are the measured angles. For a perfect fit the simulated curve should match the measured angle and the fit value should be 100%. The negative fit obtained here indicates that the model is a worse approximation than the average value.

identification, there has been some work done for the specific torsion system used in this thesis. The torsion system is supplied by Quanser who also has made a linear physical state-space model of the system. The supplied model is however far from satisfactory. As can be seen in figure 1.3, the Quanser model has a very poor fit to measured data. θ_m through θ_3 are the system outputs and are illustrated in figure 2.2. An explanation of the fit values is found in section 4.4.1. In short, the higher fit value the better and 100% is a perfect fit.

Furthermore, the torsion system hardware and software interface have already been setup together with an LQ and a PID controller. However, the PID controller seems to be the only controller capable of keeping the system stable.

At last, much of the practical programming has made extensive use of MATLAB and its Systems Identification Toolbox, which comes with plenty of useful system identification functions. For documentation concerning the toolbox it is referred to MathWorks homepage, [7].

1.3 Outline

In chapter 2 the torsion system is introduced together with the Quanser physical model and observed nonlinearities. Followed by chapter 3 that presents the different identification algorithms that have been used in this thesis. In addition their advantages and drawbacks are discussed. Exactly how to apply these algorithms and how to determine all necessary parameters are outlined in chapter 4 and the results are presented in chapter 5. The results are discussed and summarized in chapter 6 and the thesis is ended with some possible ways to continue on the work done in this thesis.

Chapter 2

Torsion System

In this chapter the torsion system from Quanser will be presented, what parts it consist of, how it is controlled and the signals that can be measured. Then the model of the torsion system will be described, starting with the linear model given by Quanser and continuing with discussion of the nonlinearities in the system, some of which have been dealt with and some of which have only been observed.

2.1 Physical System

The lab equipment will here be briefly explained, for further reading see the Quanser Manual [6]. Also see appendix A for table of system constants.

The torsion system consists of one rotary DC motor and three rotary torsion modules as can be seen in figure 2.1. The DC motor (SRV03) is the actuator of the system and its angle can also be measured by an optical sensor. The motor has two slipping encoders, however they are not in use during these experiments. The motor is connected to the first rotational load via a flexible coupling. Subsequently the first rotational load is flexibly coupled to the second load that in turn is flexibly coupled to the third load. The actual load in a rotary torsion module is created by putting two masses on a support bar attached to the rotary axle. The masses can be placed at two different anchor points on the support bar and thereby increase or decrease the inertia of the torsion module. Throughout this thesis the modules will be set to the smallest inertia, unless otherwise stated. The position (angle) of all three loads can be measured using optical sensors.

2.1.1 Hardware Interface

An ordinary PC is connected to a Q8 Hardware-In-the-Loop (HIL) board that receives angular measurements from the torsion system and forwards the control signal to the Quanser PWM Current Amplifier Package (AMPAQ). The AMPAQ converts the control signal into a current, which is fed into the SRV03. The applied control signal is also measured and send back to the Q8 HIL. All necessary AD/DA

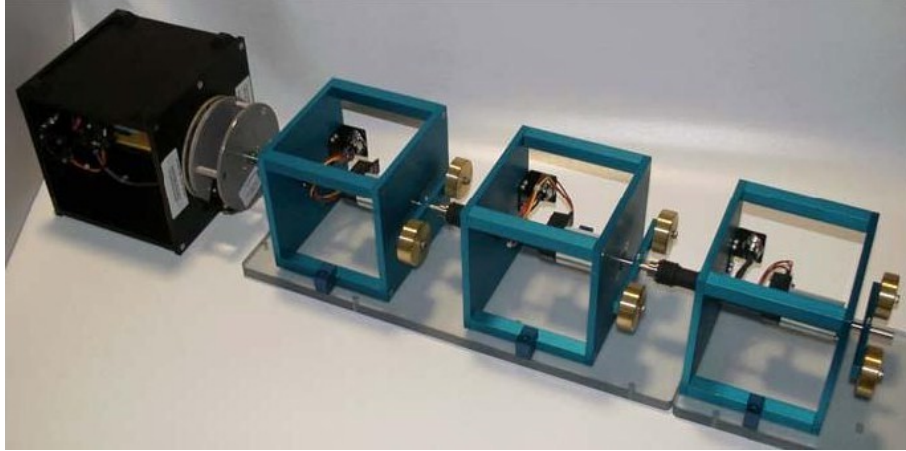


Figure 2.1. The torsion system without cabling. To the left the DC motor is housed inside the black box. Outside the black box are the slipring encoders seen, followed by the three torsion modules. The blank metal nobs on the torsion modules are the weights that can be set to two different positions. Here in the wide position.

conversions between sensors, actuators and computer are taken care of by the Q8 HIL.

2.1.2 Software Interface

The controller PC has to be equipped with MATLAB/Simulink, Real-Time Workshop and the Quanser WinCon software. The communication is then simply handled by special Simulink blocks in the Quanser toolbox. Constants in the Simulink block diagram and MATLAB workspace can be modified in real time.

2.2 Quanser Physical Model

The model given by Quanser is a linear state-space model, which has been derived from the first principals. It is originally presented in the Quanser Manual [6] and is also brought forward in this section. The model is based upon Newton's second law with inertia and viscous friction (proportional to speed) for the motor as well as for all the loads. Moreover springs are used to model the effect of the flexible couplings. This is all very similar to the torsion system (driveline) model presented in [3]. The motor is assumed to produce a torque proportional to the current through it. A sketch of the system can be seen in figure 2.2 and nomenclature can be found in table 2.1. The just mentioned modeling approach

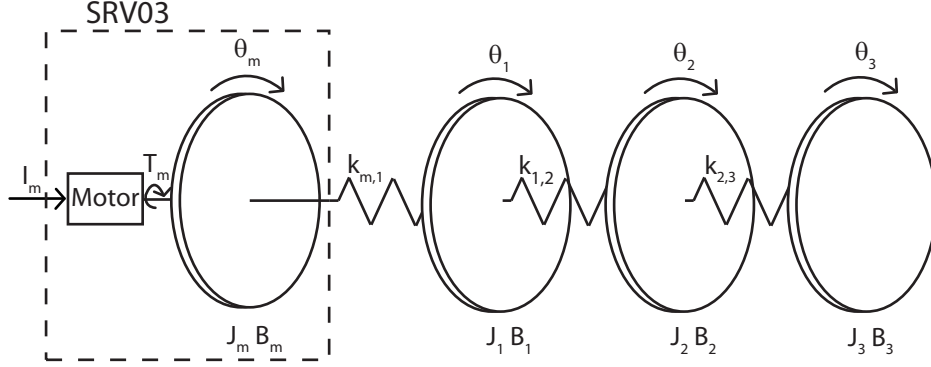


Figure 2.2. A sketch of the principle of the system.

can be expressed in equations as,

$$J\ddot{\theta} = \sum T \quad (2.1a)$$

$$T_{\text{friction}} = -B\dot{\theta} \quad (2.1b)$$

$$T_{\text{coupling}} = -k \Delta\theta \quad (2.1c)$$

$$T_m = k_t I_m \quad (2.1d)$$

Let index m denote the motor and indexes 1,2 and 3 the first, second and third rotational load counted from the motor. Choose the states to angular positions and velocities, then the state vector can be written as,

$$x = [\theta_m \quad \theta_1 \quad \theta_2 \quad \theta_3 \quad \dot{\theta}_m \quad \dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T \quad (2.2)$$

and with the input $u = I_m$ the state-space equations follow.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (2.3)$$

where,

$$\begin{aligned}
 A &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k_{m,1}}{J_m} & \frac{k_{m,1}}{J_m} & 0 & 0 & -\frac{B_m}{J_m} & 0 & 0 & 0 \\ \frac{k_{m,1}}{J_1} & -\frac{k_{m,1}+k_{1,2}}{J_1} & \frac{k_{1,2}}{J_1} & 0 & 0 & -\frac{B_1}{J_1} & 0 & 0 \\ 0 & \frac{k_{1,2}}{J_2} & -\frac{k_{1,2}+k_{2,3}}{J_2} & \frac{k_{2,3}}{J_2} & 0 & 0 & -\frac{B_2}{J_2} & 0 \\ 0 & 0 & \frac{k_{2,3}}{J_3} & -\frac{k_{2,3}}{J_3} & 0 & 0 & 0 & -\frac{B_3}{J_3} \end{pmatrix} \\
 B &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{K_t}{J_m} \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
 C &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}
 \end{aligned} \tag{2.4}$$

2.3 Nonlinearities

As with all physical systems also this torsion system contains nonlinearities that make life hard for engineers. In this section the observed nonlinearities will be described. Most nonlinearities and their effects have only been observed, not modeled in any way, and are brought up just to bring more understanding of the system and why it can be hard to fit a linear model to it.

2.3.1 Simulink Control Signal vs Real Input

The control signal that Simulink feeds into the AMPAQ controls the current. However it is not equal to the current fed to the motor. An ampere meter has been connected in between the SRV03 motor and the power cord in order to measure the motor current. The result indicates that the current can be expressed as an linear function of the control signal. This has been done using the least-square method, which can be seen in figure 2.3. The points of measurement have been concentrated around small control signals to reduce wear on the system, hence using a large control signal, for a time period long enough to get a reading, will result in very high rotational speed and vibrations. However a few measurements have been made with larger control signals to confirm the linearity.

Symbol	Description	Units
I_m	SRV03 Motor Current	A
K_t	SRV03 Torque Constant	Nm/A
T_m	Torque Produced by SRV03	Nm
θ_m	SRV03 Angular Position	rad
$\dot{\theta}_m$	SRV03 Angular Velocity	rad/s
J_m	SRV03 Moment of Inertia	kgm ²
B_m	SRV03 Viscous Damping Coefficient	Nms/rad
$k_{m,1}$	Stiffness For the Flexible Coupling Between SRV03 and First Rotational Load	Nm/rad
θ_1	First Rotational Load Angular Position	rad
$\dot{\theta}_1$	First Rotational Load Angular Velocity	rad/s
J_1	First Rotational Load Moment of Inertia	kgm ²
B_1	First Rotational Load Viscous Damping Coefficient	Nms/rad
$k_{1,2}$	Stiffness For the Flexible Coupling Between First and Second Rotational Load	Nm/rad
θ_2	Second Rotational Load Angular Position	rad
$\dot{\theta}_2$	Second Rotational Load Angular Velocity	rad/s
J_2	Second Rotational Load Moment of Inertia	kgm ²
B_2	Second Rotational Load Viscous Damping Coefficient	Nms/rad
$k_{2,3}$	Stiffness For the Flexible Coupling Between Second and Third Rotational Load	Nm/rad
θ_3	Third Rotational Load Angular Position	rad
$\dot{\theta}_3$	Third Rotational Load Angular Velocity	rad/s
J_3	Third Rotational Load Moment of Inertia	kgm ²
B_3	Third Rotational Load Viscous Damping Coefficient	Nms/rad

Table 2.1. Nomenclature

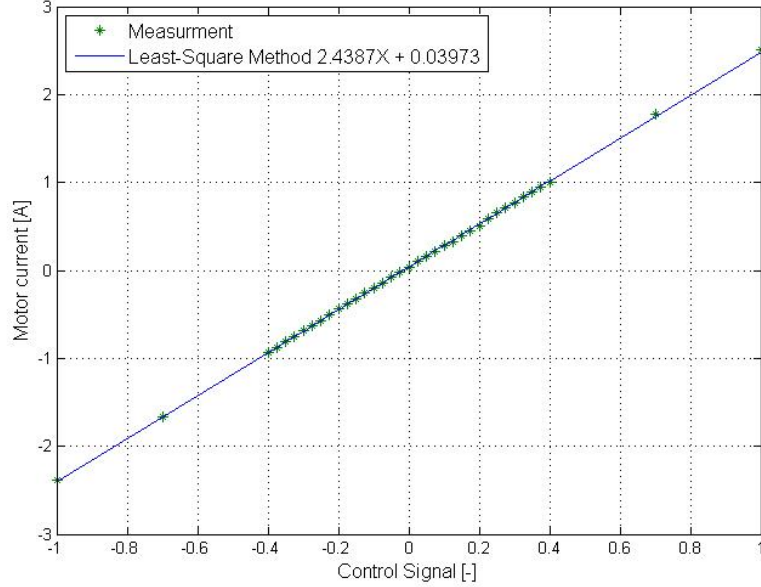


Figure 2.3. Measurement and model of the relation between control signal and motor current

Furthermore it is worth mentioning that the current measurement, available from the AMPAQ, matches the Simulink control signal, not the motor current measured with the ampere meter. In addition the AMPAQ measurement seems noisy in comparison with both the Simulink control signal and the measurement from the Amperemeter.

2.3.2 Coulomb Friction

As common with friction, the static coefficient of friction is higher than the kinetic coefficient of friction. This property of the torsion system has been observed but not modeled. This can be observed through feeding the system with an input (torque) too small to get the system in motion. Then the system is put in motion by applying an external force (poking) on one of the loads. Instead of stopping due to friction the system starts to accelerate since the dynamic friction coefficient is lower than the static coefficient.

2.3.3 Asymmetric Sliprings

On the output shaft of the SRV03 motor, two slipring encoders are attached. These are both attached on the same side of the slipring, creating an asymmetry. It has been observed that the system prefers to stop with the encoder in a downward position, see figure 2.4. It has also been observed that greater magnitudes of torque

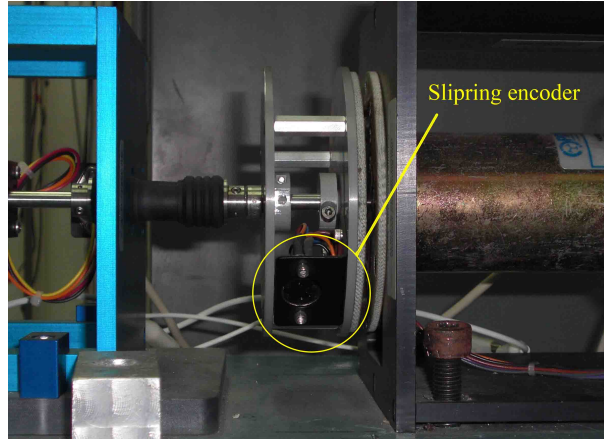


Figure 2.4. The slipring encoders in downward position

are required to get the motor moving if started with the encoders in downward position compared to the upwards position. The angle of the downward position depends on the start position since the start position always is angle zero.

There has been no modeling attempt of this nonlinearity.

2.3.4 Motor/load slip

At large steps in torque there is a slip between the motor and the first torsional load that results in an angular difference. This difference will only be changed if another slip occurs. No modeling attempt of this nonlinearity has been made as those large torque steps do not occur during normal mode of operation.

2.3.5 Saturations

Voltage Saturation

There is a voltage saturation in the motor that occurs at ± 23 V. The limit has been obtained experimentally by repeatedly reaching the limit with various control signals of various amplitudes. When the voltage is saturated it affects the current, it can no longer follow the control signal as described in section 2.3.1. However the voltage does not reach the saturation without great rotational speed of the loads. This can be realized through recalling that voltage and current in the electrical domain roughly converts into rotational velocity and torque, respectively, in the mechanical domain. Thus if the voltage is high the rotational speed has to be high too.

In a system meant for position control the loads go from stand still to stand still and do not have time to reach very high speeds. Therefore there is no need to model the voltage saturation as it will never be reached.

Other Saturations

There are limits for the current and power in both the amplifier and motor, see appendix A.

Chapter 3

Identification Algorithms

To identify the torsion system a set of different algorithms have been used. They are all introduced in this chapter but, firstly there will be a brief discussion about some system properties and how they affect the choice of algorithm. The application and evaluation of the algorithms are left to the following chapters.

3.1 Stability and Closed Loop

The first thing that has to be realized about the system, from input current to output angle, is that it is only marginally stable. This can be realized through recalling that the control signal is current, which corresponds to the motor output torque, while the output of the torsion system is position. A limited constant input (torque) will, when Newton's second equation is in equilibrium, result in a terminal velocity e.g. an ever growing output (angular position) and hence the system is not stable. This shows up as the pure integrations in the state-space model presented in section 2.2.

Since the system is unstable, care has to be taken while collecting open-loop identification data. There is no guarantee that the system will stay stable for a random signal. Therefore identification from a closed-loop system is of high interest, in addition this automatically keeps the control signal within the operating range of interest. However there are a few problems with system identification of closed-loop systems (see [5] or [8]). On one hand, direct identification, treating the data as open-loop data, might not give accurate results, while on the other hand, indirect identification is more complicated.

3.2 Prediction Error Method

The Prediction Error Method (PEM) is the classical method within system identification. It has the advantage of being able to use the structure of an existing model and only estimate the values of user decided parameters. This way the physical understanding of the model is kept while the parameters are tuned to

correct values or to compensate for non modeled effects. The drawbacks are that it needs a good initial model to converge to a "correct" model. PEM is an iterative optimization algorithm and can get stuck on local minimum, [1]. Furthermore its predictors will become unstable if the system is unstable and sometimes even if the system is just close to unstable, [4]. Moreover, to adopt PEM to closed-loop identification, it requires a complete noise model, [5]. As a consequence PEM will only be used for direct identification.

One known way to use PEM is to start by making an initial model with an other identification algorithm and then enhance it with PEM. The upside with this is that PEM for certain gets a good initial model and compared to using only the other algorithm the result is improved. This other algorithm can be any of the other algorithms presented in this chapter. Moreover those algorithms produce black-box models and therefore it will be open for PEM to refine all parameters, in contrast to when PEM is used to reparameterize the Quanser model. Then all parameters will not be open to change. The parameters, in equation 2.4, with a value of either 0 or 1 will keep their value as they define the model structure, with exception for the noise model that is estimated as a black-box model.

The PEM implementation used in this thesis is the one available in the System Identification Toolbox in MATLAB, [7], through the command `pem`.

3.3 Numerical Algorithms for Subspace State Space System Identification

Numerical Algorithms for Subspace State Space System Identification (N4SID¹) begins with estimating the state vector and from there the state-space matrices are calculated using least-square methods. This makes N4SID convergent at all times and numerically efficient, especially for MIMO systems, which is good as the torsion system is of SIMO type. On the other hand N4SID does not solve an optimization problem and therefore the obtained model is not necessarily an optimal solution. Moreover the major drawback with N4SID is that it requires the input signal to be uncorrelated with the noise, so it might not work when the system is subjected to feedback as the noise is fed back to the control signal through the controller. Nevertheless, N4SID does not need an initial model and will result in a black-box state-space model. The order of the model is a trade off between using all non-zero singular values and keeping the order of the model low. For further reading see, for example, [5] and [10].

The implementation in use in this thesis is the `n4sid` command available in the System Identification Toolbox in MATLAB .

3.4 closed-loop Identification Algorithm

There are several methods available for closed-loop indirect identification. As mentioned above, PEM will only be used for direct identification and therefore a

¹Read; Enforce it.

subspace method reworked for closed-loop identification will be used. A summary of earlier work that result in a general framework for subspace closed-loop identification can be found in [12]. This Van Overschee and De Moor Algorithm will be referred to as VODMA throughout this report. It is general from the point of view that if the controller is set to zero the indirect algorithm reduces to the direct N4SID. In addition VODMA has very few limitations.

Actually three different algorithms are presented in [12] and the first has been used in this thesis, because it was evaluated as the best in [12].

There will now follow a brief description of the algorithm, merely enough to implement it in MATLAB, see [12] for more detail.

3.4.1 Notation

The input is denoted by $u_k \in \mathbb{R}^m$, the output by $y_k \in \mathbb{R}^l$ and the reference signal by $r_k \in \mathbb{R}^m$. In the case examined in this thesis $m = 1$ and $l = 4$. The plant to be identified is characterized by the state-space equations,

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned} \quad (3.1)$$

where A is an $n \times n$ -matrix while w_k and v_k are white noise disturbances. The matrices B , C and D all have dimensions that are consistent with the other variables. In a similar fashion the controller equations are,

$$\begin{aligned} x_{k+1}^c &= A_c x_k^c + B_c y_k \\ u_k &= r_k - C_c x_k^c - D_c y_k \end{aligned} \quad (3.2)$$

where A_c is an $n_c \times n_c$ -matrix and the other matrices have dimensions that are consistent with the other variables.

The constant j , which should have a large value, is calculated from,

$$j + 2i - 1 = \text{number of data points} \quad (3.3)$$

where i is user defined and should be larger than the number of states in the plant. The gathered data used in the algorithm is represented in a block Hankel matrix as shown here,

$$U_{0|i-1} = \begin{pmatrix} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \end{pmatrix} \in \mathbb{R}^{mi \times j} \quad (3.4)$$

the matrix $Y_{0|i-1}$ is defined in the same way. The controller information is stacked in a lower triangular block-Toeplitz matrix,

$$H_i^c = \begin{pmatrix} D_c & 0 & \cdots & 0 \\ C_c B_c & D_c & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_c A_c^{i-2} B_c & C_c A_c^{i-3} B_c & \cdots & D_c \end{pmatrix} \in \mathbb{R}^{mi \times li} \quad (3.5)$$

It is also practical to use the following data matrix,

$$M_{p|q} = U_{p|q} + H_{q-p+1}^c Y_{p|q} \quad (3.6)$$

where $0 \leq p \leq q \leq 2i - 1$.

The algorithm is based upon two projections. In both cases \mathcal{A}, \mathcal{B} and \mathcal{C} are matrices with equal number of columns. The orthogonal projection,

$$\mathcal{A}/\mathcal{B} = \mathcal{A}\mathcal{B}^T (\mathcal{B}\mathcal{B}^T)^\dagger \mathcal{B}/j^2 \quad (3.7)$$

and the oblique projection,

$$\mathcal{A}/_{\mathcal{B}}\mathcal{C} = \mathcal{A} \begin{pmatrix} \mathcal{C}^T & \mathcal{B}^T \end{pmatrix} \begin{pmatrix} \mathcal{C}\mathcal{C}^T & \mathcal{C}\mathcal{B}^T \\ \mathcal{B}\mathcal{C}^T & \mathcal{B}\mathcal{B}^T \end{pmatrix}^\dagger \begin{pmatrix} I_c \\ 0 \end{pmatrix} \mathcal{C}/j^2 \quad (3.8)$$

where I_c is an identity matrix with the same number of rows as the \mathcal{C} matrix.

3.4.2 LQ-decomposition

As recommended in [12] the data matrix,

$$\mathcal{H} = \begin{pmatrix} U_{0|2i-1} \\ Y_{0|2i-1} \end{pmatrix} \quad (3.9)$$

should be LQ-decomposed and not used directly in order to obtain a data reduction. This is not only wise but also necessary for MATLAB to handle the calculations. LQ-decomposition means the decomposition of matrix A into lower triangular matrix L and unitary matrix Q (i.e. $QQ^T = I$ and $Q^T Q = I$) so that $L * Q = A$. The LQ-decomposition can be obtained from QR-decomposition (R is an upper triangular matrix) of A^T , which is easily calculated in MATLAB using the `qr` command. Consider the following,

$$L * Q_{\text{LQ}} = A = (A^T)^T = (Q_{\text{QR}} * R)^T = R^T * Q_{\text{QR}}^T \quad (3.10)$$

and hence the transpose of an upper triangular matrix becomes a lower triangular matrix the LQ-decomposition is simply obtained via,

$$\begin{aligned} L &= R^T \\ Q_{\text{LQ}} &= Q_{\text{QR}}^T \end{aligned} \quad (3.11)$$

The \mathcal{H} matrix, which contains all the data and is of the size $2(m+l)i \times j$, is subjected to the LQ-decomposition. The resulting Q matrix will not be needed in the final expressions while the L matrix will have the mere size of $2(m+l)i \times 2(m+l)i$. This is a remarkable data reduction since $2(m+l)i \approx 100$ in comparison with the large, at least 10^4 , value of j . To easier derive the new equations that

take use of the LQ-decomposition, the following notation will be used,

$$\begin{array}{c} mi \\ m \\ m(i-1) \\ li \\ l \\ l(i-1) \end{array} \underbrace{\begin{pmatrix} j \\ U_{0|i-1} \\ U_{i|i} \\ U_{i+1|2i-1} \\ Y_{0|i-1} \\ Y_{i|i} \\ Y_{i+1|2i-1} \end{pmatrix}}_{\mathcal{H}} = \begin{pmatrix} mi & m & m(i-1) & li & l & l(i-1) \\ L_{11} & 0 & 0 & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 & 0 & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} & 0 & 0 \\ L_{51} & L_{52} & L_{53} & L_{54} & L_{55} & 0 \\ L_{61} & L_{62} & L_{63} & L_{64} & L_{65} & L_{66} \end{pmatrix} \begin{pmatrix} j \\ Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{pmatrix} \quad (3.12)$$

and in addition, let $L_{32:44}$ denotes the submatrix $\begin{pmatrix} L_{32} & L_{33} & 0 \\ L_{42} & L_{43} & L_{44} \end{pmatrix}$ of L . With this notation $M_{i|2i-1}$ can be expressed as,

$$M_{i|2i-1} = \begin{pmatrix} 0 & I & 0 & 0 & H_i^c \\ 0 & 0 & I & 0 & 0 \end{pmatrix} * LQ \quad (3.13)$$

Before proceeding with the derivation of new equations in the next section, please note that the unitary property of Q also holds for Q_x since,

$$\begin{aligned} Q * Q^T &= \begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{pmatrix} * (Q_1^T & Q_2^T & Q_3^T & Q_4^T & Q_5^T & Q_6^T) = \\ &= \begin{pmatrix} Q_1 * Q_1^T & Q_1 * Q_2^T & Q_1 * Q_3^T & Q_1 * Q_4^T & Q_1 * Q_5^T & Q_1 * Q_6^T \\ Q_2 * Q_1^T & Q_2 * Q_2^T & Q_2 * Q_3^T & Q_2 * Q_4^T & Q_2 * Q_5^T & Q_2 * Q_6^T \\ Q_3 * Q_1^T & Q_3 * Q_2^T & Q_3 * Q_3^T & Q_3 * Q_4^T & Q_3 * Q_5^T & Q_3 * Q_6^T \\ Q_4 * Q_1^T & Q_4 * Q_2^T & Q_4 * Q_3^T & Q_4 * Q_4^T & Q_4 * Q_5^T & Q_4 * Q_6^T \\ Q_5 * Q_1^T & Q_5 * Q_2^T & Q_5 * Q_3^T & Q_5 * Q_4^T & Q_5 * Q_5^T & Q_5 * Q_6^T \\ Q_6 * Q_1^T & Q_6 * Q_2^T & Q_6 * Q_3^T & Q_6 * Q_4^T & Q_6 * Q_5^T & Q_6 * Q_6^T \end{pmatrix} = \\ &= \begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{pmatrix} \end{aligned} \quad (3.14)$$

where the last equality comes from the unitary property of Q . From this it can be concluded that,

$$Q_p * Q_q = \begin{cases} I & \text{if } p = q \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

3.4.3 Algorithm

Next after the LQ-decomposition the first step of the algorithm is to calculate the orthogonal projections, Z_i and Z_{i+1} , together with the oblique projection, \mathcal{O}_i .

$$\begin{aligned}
Z_i &\equiv Y_{i|2i-1} / \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \\ M_{i|2i-1} \end{pmatrix} = L_{51:66} Q / \left(\overbrace{\begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & H_i^c \\ 0 & 0 & I & 0 & 0 & 0 \end{pmatrix}}^{F_1} LQ \right) = \\
&= L_{51:66} Q (F_1 LQ)^T [F_1 LQ (F_1 LQ)^T]^\dagger F_1 LQ / j^2 = \\
&= L_{51:66} Q Q^T L^T F_1^T [F_1 LQ Q^T L^T F_1^T]^\dagger F_1 LQ / j^2 = \\
&= L_{51:66} (F_1 L)^T [F_1 L (F_1 L)^T]^\dagger F_1 LQ / j^2 \quad (3.16)
\end{aligned}$$

$$\begin{aligned}
Z_{i+1} &\equiv Y_{i+1|2i-1} / \begin{pmatrix} U_{0|i} \\ Y_{0|i} \\ M_{i+1|2i-1} \end{pmatrix} = \\
&= L_{61:66} Q / \left(\overbrace{\begin{pmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & I & 0 & 0 & 0 & H_{i-1}^c \end{pmatrix}}^{F_2} LQ \right) = \\
&= / \text{ In the same manner as equation 3.16 } / = \\
&= L_{61:66} (F_2 L)^T [F_2 L (F_2 L)^T]^\dagger F_2 LQ / j^2 \quad (3.17)
\end{aligned}$$

$$\begin{aligned}
\mathcal{O}_i &\equiv Y_{i|2i-1}/M_{i|2i-1} \begin{pmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{pmatrix} = \\
&= L_{51:66}Q / \underbrace{\begin{pmatrix} 0 & I & 0 & 0 & H_i^c \\ 0 & 0 & I & 0 & \end{pmatrix}}_{F_4} LQ \left(\overbrace{\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}}^{F_3} L_{11:44}Q_{1:4} \right) = \\
&= L_{51:66}Q \begin{bmatrix} (F_3 L_{11:44}Q_{1:4})^T & (F_4 LQ)^T \end{bmatrix} \\
&\begin{bmatrix} F_3 L_{11:44}(F_3 L_{11:44})^T & F_3 L_{11:44}(F_4 L_{11:64})^T \\ F_4 L_{11:64}(F_3 L_{11:44})^T & F_4 L(F_4 L)^T \end{bmatrix}^\dagger \begin{bmatrix} F_3 L_{11:44}Q_{1:4} \\ 0 \end{bmatrix} / j^2 = \\
&= \begin{bmatrix} L_{51:64}(F_3 L_{11:44})^T & L_{51:66}(F_4 L)^T \end{bmatrix} \\
&\begin{bmatrix} F_3 L_{11:44}(F_3 L_{11:44})^T & F_3 L_{11:44}(F_4 L_{11:64})^T \\ F_4 L_{11:64}(F_3 L_{11:44})^T & F_4 L(F_4 L)^T \end{bmatrix}^\dagger \begin{bmatrix} F_3 L_{11:44}Q_{1:4} \\ 0 \end{bmatrix} / j^2 \quad (3.18)
\end{aligned}$$

The second step is to calculate the Singular Value Decomposition (SVD) of the oblique projection, $\mathcal{O}_i = USV^T$. As stated in [10], the SVD can be calculated without the trailing $Q_{1:4}$ in expression 3.18. It will not affect the U and S matrices that are the ones of interest. After the SVD has been carried out the following partition of the matrices is obtained.

$$\mathcal{O}_i = (U_1 \ U_2) \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \quad (3.19)$$

where S_1 in theory contains all the non-zero singular values along the diagonal. However in practise it only contains the relatively large singular values, hence the length of the diagonal decides the number of states, which is desired to be kept low. This is a point at which the user has to make a trade off. The next step is to calculate $\mathcal{G} = U_1 S_1^{1/2}$ and then \mathcal{K} by solving the equation,

$$U_2^T Z_i = U_2^T \mathcal{K} M_{i|2i-1} \quad (3.20)$$

The (scalar) equation system can be multiplied by Q^T from the right in order to get rid of the Q matrices in 3.13 and 3.16. This leads to a significant decrease in the number of equations. Moreover this is where MATLAB might get stuck because of the otherwise too large equation system.

In order to solve the equation system, without the need of calculating $M_{i|2i-1}^\dagger$ that might be ill-conditioned, it is suggested in [12] to use the following theorem.

If \otimes denotes the Kronecker product and $\text{vec}(A)$ the vectorization of A , i.e. stacking all of A 's columns on top of each other, the following holds true.

$$\text{vec}(AXB) = (B^T \otimes A) \text{vec}(X) \quad (3.21)$$

Applied to equation 3.20 together with the multiplication of Q^T the resulting equation system is,

$$\text{vec}(U_2^T Z_i Q^T) = ((M_{i|2i-1} Q^T)^T \otimes U_2^T) \text{vec}(\mathcal{K}) \quad (3.22)$$

In the next step it is time to determine the states and this is done as follows

$$\begin{aligned}\hat{X}_i &= \mathcal{G}^\dagger [Z_i - \mathcal{K}M_{i|2i-1}] \\ \hat{X}_{i+1} &= \underline{\mathcal{G}}^\dagger [Z_{i+1} - \underline{\mathcal{K}}M_{i+1|2i-1}]\end{aligned}\tag{3.23}$$

where $\underline{\mathcal{G}}$ means \mathcal{G} without the last block row and $\underline{\mathcal{K}}$ means \mathcal{K} without the last block row and block column. It is wise to calculate $\hat{X}_i Q^T$ and $\hat{X}_{i+1} Q^T$ instead of \hat{X}_i and \hat{X}_{i+1} , as the matrices are kept smaller and not needed explicitly if the next equation also is multiplied by Q^T .

Go on by solving for \mathcal{S} (and the residuals \mathcal{T}) from the subsequent set of equations.

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{21} & \mathcal{S}_{22} \end{pmatrix} \begin{pmatrix} \hat{X}_i \\ M_{i|i} \end{pmatrix} + \mathcal{T}\tag{3.24}$$

where $M_{i|i} = U_{i|i} + D_c Y_{i|i}$. This equation might as well be multiplied by Q^T from the right, to reduce the sizes of the matrices.

Finally, to obtain the state-space matrices,

$$\begin{aligned}B &= \mathcal{S}_{12}(I_m - D_c \mathcal{S}_{22})^{-1} \\ D &= \mathcal{S}_{22}(I_m - D_c \mathcal{S}_{22})^{-1} \\ A &= \mathcal{S}_{11} + B D_c \mathcal{S}_{21} \\ C &= (I_l + D D_c)^{-1} \mathcal{S}_{21}\end{aligned}\tag{3.25}$$

Chapter 4

Experiment Setup and Identification Process

In order to identify the system a number of experiments have to be performed for the data acquisition. For each experiment an input signal has to be prepared. If the system needs feedback a controller has to be chosen too. The available measurements are controller input and output, system input and output, in other words, reference signal, control signal, measured control signal and all four angles. The control signal and the measured control signal can differ because of saturations and noise.

After acquiring the data it has to be checked for errors, an identification algorithm chosen as well as the number of states/initial model and sample time. In addition for VODMA there is a design variable i that has to be chosen.

It is quite common that the measurements are out of bounds in the beginning, probably because sensors and computer are not perfectly timed and it therefore takes some time to synchronize. There is a similar phenomenon when the system is shutting down at the end of a measurement series. This has not been investigated as the problem is easily solved by omitting the first and last second of measurement.

The sample time during measurement is always 0.001 s, well below the time constants of the system. However when the data is used for identification it should be down sampled to heavily reduce the computational time. The effects of this can be read about in section 5.6.

In this chapter, only a few results will be presented. The focus is instead at the identification methodology and data acquisition. How have the experiments been carried out and why? Which are the different settings that have been tested? The actual result will be presented in its own designated chapter, see chapter 5.

4.1 Input Signal

For successful system identification a well made input signal is of high importance. The algorithms in use in this thesis are made for random signals that are ranging

over the entire frequency band of interest. The input signals should be random to lessen the impact of the noise and cover all the frequencies so all aspects of the system are excited. To create the random signals the `idinput` MATLAB command has been used. It has been used to create both Random Gaussian Signals (RGS) and Pseudo-Random Binary Signals (PRBS).

With some reasoning around the system and its properties a couple of benchmark signals have been made. These benchmark signals have later on been adjusted in both frequency and magnitude in order to obtain the best model. However as benchmark signals they will still be used to describe the new signals. For instance, "The new signal has 80% magnitude." where it is implied that it means "...80% of the magnitude of the benchmark signal."

As benchmark the reference signals are made to contain frequencies between 0-10 Hz and values alternating between $\pm 36^\circ$, for the PRBS, or have two standard deviations in between $\pm 60^\circ$, for the RGS. When conducting experiments on the open-loop system the maximum frequency can be doubled as the control signal varies faster than the reference signal. The magnitude should be divided by 60° (input now without unit) to end up on a more reasonable level. The frequency range has been experimentally obtained as (around) the highest one the system is able to respond to. The reference signal magnitude range is chosen as the largest range that does not saturate the control signal, see section 4.3.2 for explanation of the saturation. Despite the approach for the closed-loop case, the control-signal range for open loop is kept fairly small compared to the saturation limit to avoid instability. This is due to the fact that a directly applied RGS or PRBS has no sharp peaks and therefore causes trouble at high magnitudes. On the contrary, when the system is subjected to feedback, the control signal can peak and give high values for very short periods.

The signals are made 80 s in length for the 0-10 Hz range and the length is inversely proportional against the frequency scaling. One signal can be used for identification and a completely different signal for validation. Although it is possible to split the signal in two and only use half for identification and half for validation, without any noticeable loss in accuracy. The latter approach has mostly been used throughout the thesis.

4.2 Nonlinear Compensation

There is one of the nonlinearities that is known, namely the offset in input signal. Moreover it can be modeled with great accuracy as described in section 2.3.1. Since it can be modeled it also can be compensated for. This is done by applying the inverse model before the control signal is fed into the system and by doing so the nonlinear compensation together with the real system becomes (more) linear, hence the control signal can be seen as the actual motor current. Experiments have been conducted both with and without this linearization.

If u denotes the control signal fed into the real system and \tilde{u} the control signal

before linearization, the inverse of the linear function found in figure 2.3 becomes,

$$u = \frac{\tilde{u} - 0.0397}{2.44} \quad (4.1)$$

4.3 Controller and Saturation

Some experiments have been carried out under feedback. The controller that has been used is a PID controller, made by earlier users. It was simply an already existing well working controller. For identification purposes there are no requirements on reference tracking, robustness, etc. Just keep the system stable. Therefore no time has been spent controller design.

4.3.1 PID controller

The PID controller implemented for use with the torsion system has a modified D-part. It only depends on the system output, not the error that is formed as the difference between reference and output signal. The controller equation can be seen in equation 4.2 and the used parameter values are found in table 4.1. Experiments have been carried out where various parts of the controller have been removed. The only alternative that resulted in a sufficient controller was the PD controller.

$$u = K_p(r - y) + K_i \int (r - y) dt - K_d \frac{dy}{dt} \quad (4.2)$$

Parameter	Value
K_p	8.3867
K_i	0.3255
K_d	0.8

Table 4.1. Controller parameters. The parameters are calculated from a MATLAB script supplied with the system and have been rounded in the table

4.3.2 Control Signal Saturation

There is a saturation of ± 2.95 for the controller in order to avoid damaging the system. The saturation limits can also be found in the Quanser manual. This saturation has not been tampered with since it exist for safety reasons.

4.4 Identification Process

For the identification purpose there are mainly three algorithms to disposal, PEM, N4SID, VODMA. As stated in section 3.2, PEM will be used with different initial models, the Quanser model as well as the resulting models from the subspace

methods. However sometimes VODMA result in an unstable model which can not be enhanced with PEM, in this case no enhancement will be made, naturally. Furthermore VODMA will be tried out through direct identification, i.e. setting the controller equations equal to zero inside the algorithm. This was from the beginning done to verify the function of VODMA but proved interesting in other ways too.

To summarize there are seven different ways of applying the algorithms, VODMA and PEM, only VODMA, VODMA simplified to Direct Identification (DI) and enhanced with PEM, only VODMA DI, N4SID and PEM, N4SID and lastly PEM using the Quanser model as initial model.

4.4.1 Model Fit

To compare different identification experiments with each other and being able to rank the resulting models, some kind of quality measure is required. Extensive use has been made of the Systems Identification toolbox in MATLAB that in turn uses the so called fit value. The fit value is calculated by

$$\text{FIT} = 1 - \frac{\|\hat{y} - y\|_2}{\|y - \bar{y}\|_2} \quad (4.3)$$

where y is a vector containing all the samples from one measured output (e.g. θ_m), \hat{y} the simulated output and \bar{y} the average of y . With this definition 100% fit means a perfect fit and 0% fit corresponds to the fit obtained by simply using the average value as model, $\hat{y} = \bar{y}$. Note that every measured output get a separate fit value. When needed to weight them together the average has been used.

4.4.2 Identification Algorithm Verification

To verify that the identification algorithms are working properly they have been tested with data created from simulation. It is the Quanser model that has been simulated when subjected to the same input as in some of the real-life experiments. A bandwidth limited white noise has been added to all the outputs. The amplitudes range from $\pm 4^\circ$, a considerably amount of noise. There have been simulations of as well open-loop as closed-loop system, where the closed loop has been constructed with the same PD controller used in real life.

No results from the algorithm verification will be presented in the results chapter (5), instead they are found in Appendix B. In summary, all the algorithms pass the test with fit values around 90%. This is sufficient considering that the noise can cause the fit for the true model to drop some 10%. Also worth mentioning is that a significant drop in fit occurs if the input is saturated. Saturation should be avoided, even in small doses.

4.4.3 Input Determination

Without good input the identification is useless. Due to this fact different inputs are tested before parameters in the algorithms are fine tuned. However, the parameters need to be assigned some values for it to be possible to run the algorithms.

A few quick experiments together with some reasoning can sort this matter out. The physical model indicates that the system would have eight states, adding one state for some unmodeled dynamics and nine states it is. Experiments can also confirm, via the singular values, that it is plausible with nine states. Letting the sample time be 0.01 s has proved good results and i is set to 28, clearly higher than the number of states.

The first thing to answer is whether open-loop identification is possible to use, start with the simplest thing first and perhaps the next step is not necessary. Secondly the linearization is validated, it should be tested early since it has good precision and removing a nonlinearity is a huge advantage when working with linear methods. Thirdly the type of signal should be determined, starting with RGS or PRBS, hence it is the most characteristic signal property.

Now the major choices have been made and fine tuning is up next for the input. It is wise to start by tuning the frequency since it is the easiest to check. It should be as high as possible to excite every frequency response of the system but not contain frequencies too high for the system to respond to. The benchmark signal is already set in such way that an increase is bad, although a decrease might help. Therefore the frequency is lowered until worse results are obtained. Continue the evaluation by changing the input magnitude and lastly, the controller (if closed loop).

4.4.4 Parameter Determination

As the input has been tuned, the algorithms should too. A good beginning is to set the sample time hence it is a property that depends on the system, data and algorithm. When the sample time has been found the number of states and value of i can be varied. These three algorithm parameters are harder to assign a best value to than the signal properties, since they are a trade off between on one hand, model complexity and computational time, and on the other hand, precision.

Chapter 5

Experimental Results

This chapter will follow the identification process outlined in section 4.4. Since the identification process has already been outlined no further discussion about the process will follow, merely establish which result is proven best and continue thereafter. It is recommended that the reader is familiar with the chosen identification process before proceeding with this chapter.

The results are presented in tables throughout the chapter. In these tables a cross, 'x', represents a poor fit value, less than 0%. A bar, '-', indicates that no data is available for that certain post. Moreover some charts will be presented in order to demonstrate different fit values. They consist of four graphs, each showing the data and simulated value for one output, with the fit values right next to them.

All experiments have been conducted with one data set for identification and one separate set for validation. If not otherwise stated, the identification set and validation set comes from different parts of the same input signal. Throughout the chapter all the presented fit values are from validation, hence it is validation that indicates how good the model will work when simulating with a new input.

In the underlying experiments to this chapter the masses in the torsion modules are all set to the narrow distance. The masses have later on, in additional experiments, been set to the wide distance but those results are found in appendix C

5.1 Open Loop or Closed Loop

An RGS is applied to the open-loop system and as a result it drifts away to large outputs, see figure 5.1, compared to those of interest that are clearly within $\pm 90^\circ$ (≈ 1.5 rad). To change the input to a better signal is not trivial. Lowering the frequency makes the drift worse or even unstable while increasing the frequency makes the torsion system shudder. Open-loop identification has to be dismissed due to the instability of the system.

When the RGS instead is applied to the closed-loop system it indicates a potential of identifying the system even though it has to be processed further. In figure 5.2 the result from direct identification with VODMA enhanced with PEM is shown and in figure 5.3 it can be confirmed that the control signal is not saturated.

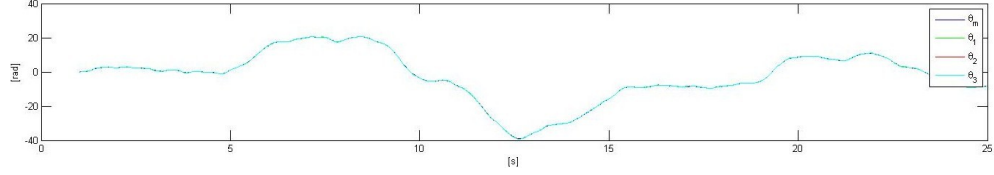


Figure 5.1. Open-loop experiment with RGS input. The four angles have been plotted on top of each other. They are too close to each other to tell apart. Nonetheless, the important observation is in the magnitudes. The angles are pending between ± 20 rad, approximately 1200° or three complete rotations, far from the operating range. This demonstrates the problem with open-loop identification.

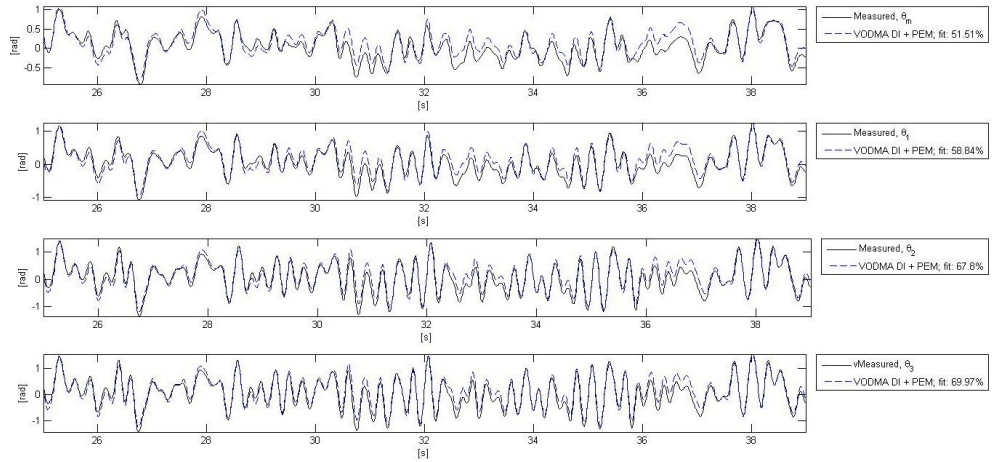


Figure 5.2. Fit obtained from closed-loop experiment with RGS input and model estimated with direct identification VODMA enhanced with PEM. The fit is descent which suggests that it is feasible to identify a satisfactory model when the system subjected to feedback.

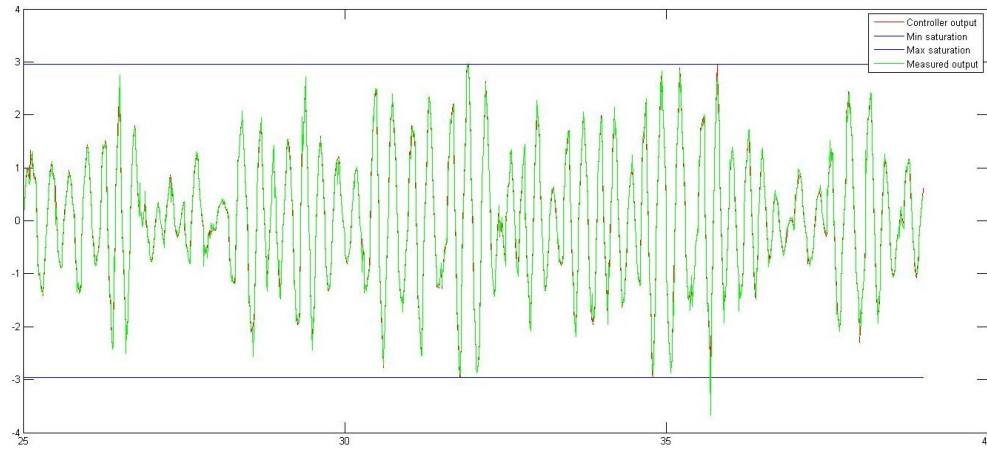


Figure 5.3. The Simulink control signal and measured control signal from the closed-loop experiment. There are no saturations in the control signal. The negative peak that can be seen after 35 s is in measured control signal, which is measured after the saturation.

5.2 Nonlinear Compensation

The nonlinear compensation has strong roots in theory and, according to section 2.3.1, the nonlinearity is well known. It is quite expected that it improves the fit as a more linear system is subject to the linear system identification algorithms. The fit values of the RGS, both with and without linearization, can be found in table 5.1.

5.3 Signal Type and Frequency

A PRBS is set against the RGS in various tests. All the results can be found in table 5.1. Both signals (unmodified) have approximately the same frequency spectra and are very close to amplitude saturation, see figure 5.4, consequently the results are practically the same. Nevertheless, if the frequency of both signals are decreased in steps, the PRBS with 80% frequency stands out as the winner. Notice that the low frequency PRBSs have been scaled to 90% magnitude. This is due to saturations in the control signal that occurs if there is no scaling. Even lower frequencies are not of interest as the 67% frequency signal already reaches steady state during the longer steps, see figure 5.5.

5.4 Magnitude of Input Signal

The magnitude has been altered in order to obtain the best fit, accordingly with equation 4.3. It has already been decreased to 90% to avoid saturation but it might give even better results if it is further decreased. Firstly it was decreased

Sig.	Amp. [%]	Freq. [%]	Lin.	Ctrl	VODMA+PEM	VODMA	VODMA	DI+PEM	VODMA	DI	N4SID+PEM	N4SID	PEM
RGS	100	100	no	PID	71.54	33.23		62.03		x	68.91	56.85	49.43
RGS	100	100	yes	PID	73.53	73.40		76.04		74.66	74.40	73.79	64.96
RGS	100	80	yes	PID	65.27	60.07		56.21		56.21	58.29	56.59	55.80
RGS	100	67	yes	PID	79.59	76.36		79.59		79.01	77.46	77.30	43.24
PRBS	100	100	yes	PID	64.36	64.36		71.12		71.12	79.20	70.85	59.06
PRBS	90	80	yes	PID	87.02	85.69		88.06		88.06	86.50	85.95	81.41
PRBS	90	67	yes	PID	85.73	84.79		87.17		85.94	85.29	83.06	68.63
PRBS	83	80	yes	PID	85.65	85.11		84.82		82.45	85.62	79.20	61.02
PRBS	20	80	yes	PID	65.46	35.50		66.04		26.85	75.59	57.64	5.62
PRBS	90	80	yes	PD	72.83	72.83		x		x	30.31	48.02	30.04

Table 5.1. Average fit values in percent for different identification algorithms, input signals and controllers.

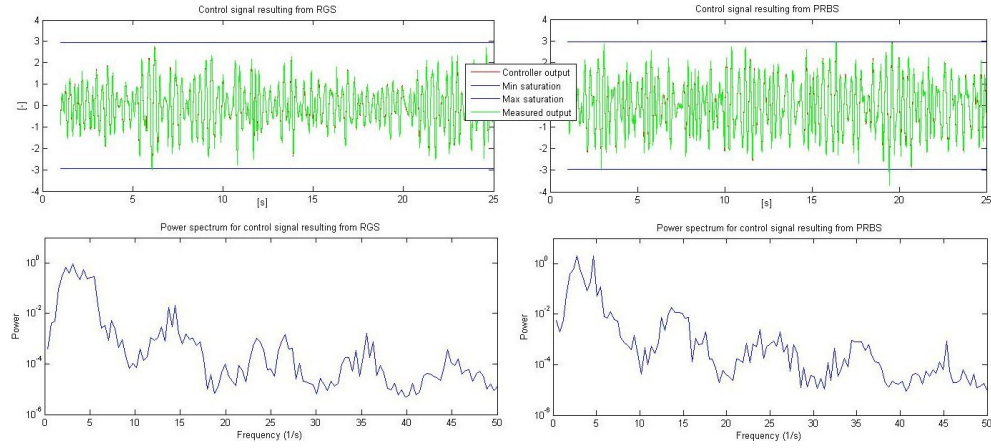


Figure 5.4. The actual signal and its power spectrum for both the RGS and the PRBS. Neither of the signals saturate but they almost do. The frequency power spectrums are highly similar with most of the power below 10 Hz.

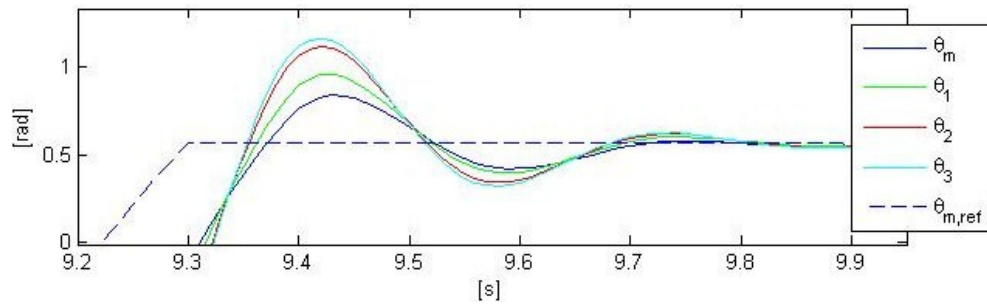


Figure 5.5. During one of the longer pulses of the 67% frequency PRBS the system reaches steady state.

to 83% with slightly worse result. Secondly it was lowered to 20%, it is common the systems behave more linearly at small magnitudes, but still with worse result. Therefore the 90% magnitude was kept for future experiments. Results can be found in table 5.1.

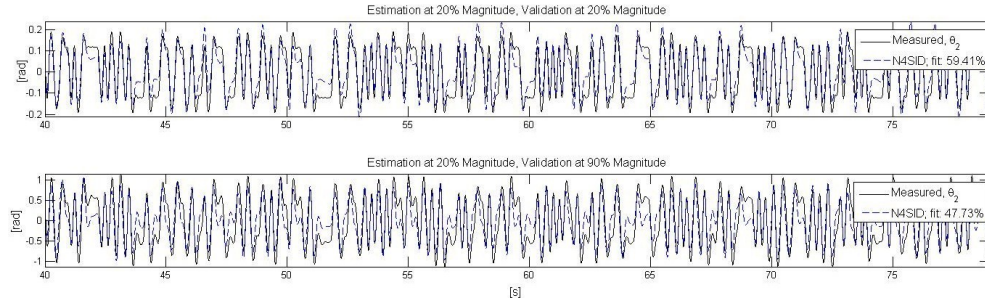


Figure 5.6. The identification has been performed with a 20% magnitude, 80% frequency PRBS. Validation has been performed with both 20% and 90% magnitude. The different validation plots for θ_2 are shown above. The result is significantly better when the amplitudes are equal during both identification and validation.

For the results in the table the validation and identification data come from different parts of the same input. When the experiments with the large and small magnitudes instead were validated against data from the other signal, results were poor. The case when identification has been performed at small magnitudes can be seen in figure 5.6. This could be due to the larger signal-to-noise ratio for smaller signals or simply different system behavior at different magnitudes. A discussion favoring the latter can be found in section 6.2.1. No matter what is the reason for this phenomenon, the identification signal should be chosen with the controllers desired operating range in mind.

5.5 Controller

Only two controller configurations have been tried out as removing the D part of the controller makes the system oscillate too much and the control signal gets saturated. The comparison between the PD and PID controller can be found in table 5.1. The result favored the PID controller.

5.6 Sample Time

Various sample rates have been tried out. The nominal sample time of 0.01 s can be increased to cut the computational time without significant losses in fit. The fit values start to slide when the sample time is increased up to and beyond 0.06 s. The other way around, if the sample time is decreased to 0.05 s an almost zero improvement is obtained whereas the computational time is massively increased. Results are found in figure 5.7 and 5.8.

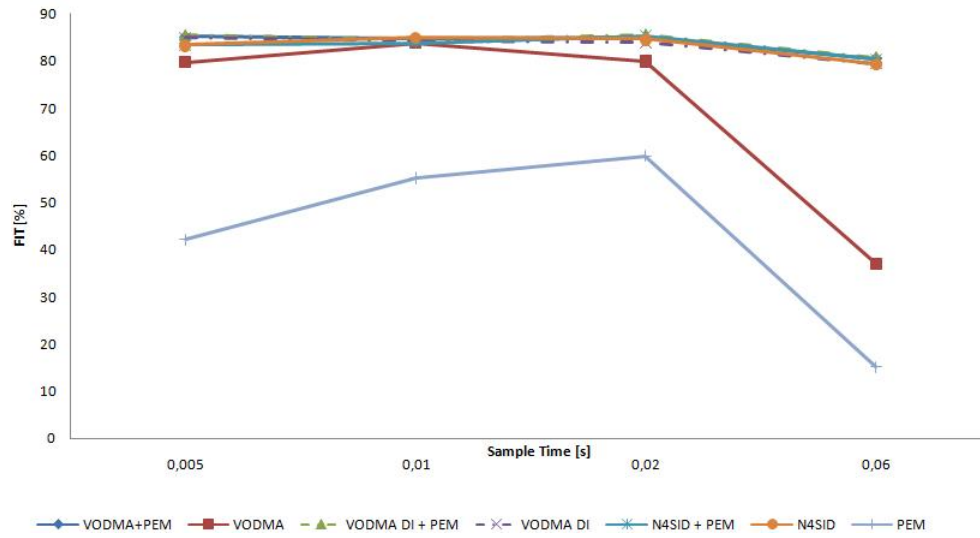


Figure 5.7. The average of the fit values over different sample times for different algorithms. The algorithms start to drop in fit for sample times of 0.06 s or more.

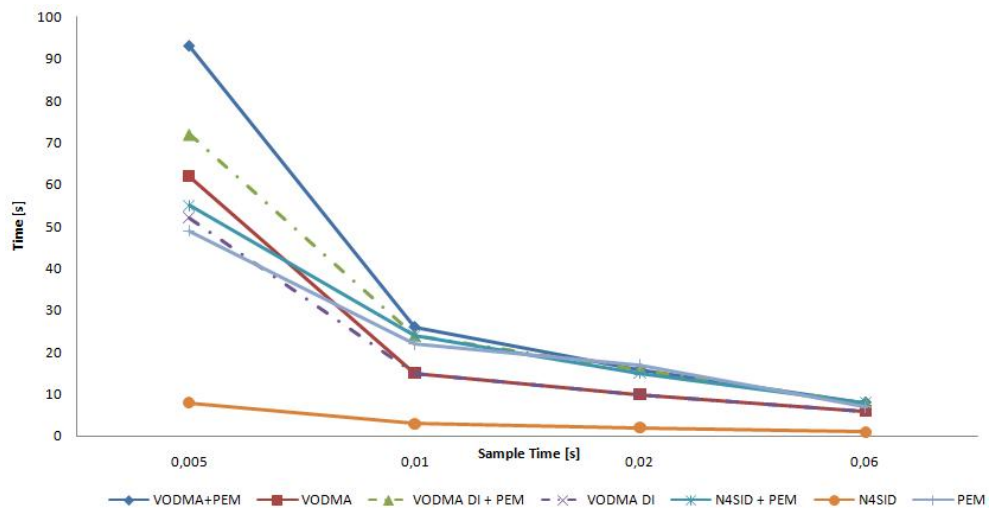


Figure 5.8. The computational time over different sample times for different algorithms. There is a clear correlation between low sample time and high computational time.

Even though the sample time safely can be reduced it has not in continued tests. This is in order to make new tests comparable with old tests. Since the computational time is acceptable for the old sample time, there exists no real drawback with keeping it.

5.7 Number of States

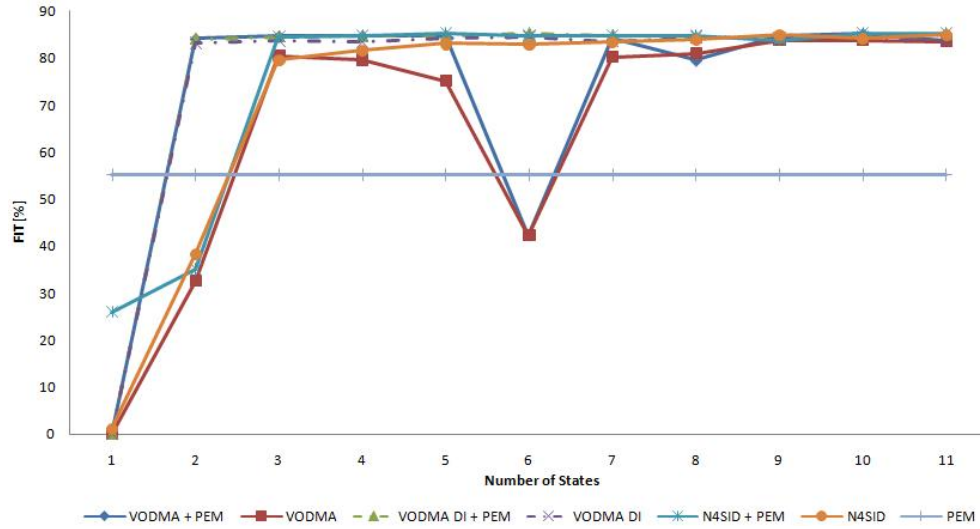


Figure 5.9. The average of the fit values over different number of states for different algorithms. With exception for the irregularity at six states for VODMA, the fits level out above five states. Only VODMA DI can handle two states. The PEM line (with Quanser initial model) is constant as the number of states can not be set for the Quanser model.

In theory the number of states can be determined from the singular values of the system. In the ideal case the number of nonzero singular values equals the number of states, but in practise this would make the number of states way too high. Therefore some singular values have to be approximated to zero. Which values to neglect can be chosen by examining a plot that shows the singular values sorted in decreasing order.

The singular-value plot for the torsion system, figure 5.10, indicate that the number of states is around five, hence the curve levels out after that. Exactly where the curve levels out is a bit subjective, the plot could be interpreted differently. In addition the plot looks slightly different for the different subspace methods. Moreover it has been proven in appendix B that a singular-value plot does not need to be correct. Furthermore the physical model indicates eight states, but it clearly has some errors. Consequently the physical model is a bad indicator for the number of states. Due to all the these reasons the algorithms have been swept

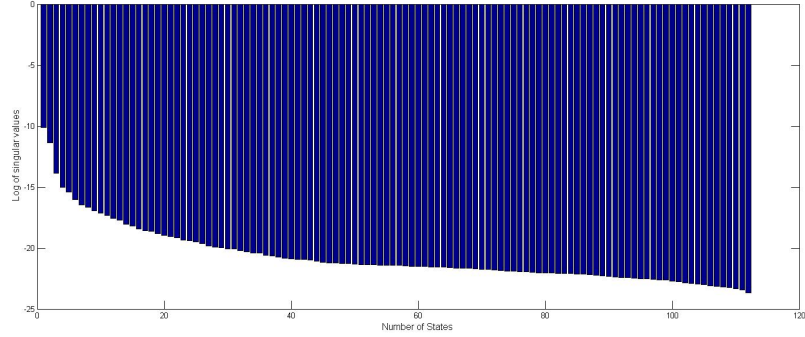


Figure 5.10. Singular-value plot from direct identification VODMA. The experiment setup is PRBS signal with 80% frequency and 90% magnitude. The singular values can be neglected for more than five states hence the curve stabilizes.

from one state up to eleven states. The resulting average fits are presented in figure 5.9.

As can be seen not much happens to the fit in the interval 5-11 states, in agreement with the singular values. For some unknown reason the indirect identification with VODMA does not work very well for six states. The PEM line is constant as it uses the physical model where the number of states is fixed to eight. PEM is merely included in the graph to get all identification algorithms together in one graph. Yet another remark is that more states do not necessary increase the fit values. With less than five states the effect of decreasing the number of states is still limited. With three states all algorithms work fine while with two states no algorithms are sufficient besides direct identification with VODMA and one state simply will not do.

5.8 VODMA Parameter i

Not much has been said about how to choose the integer parameter i in the VODMA. In [12] it is only stated that i should be greater than the number of states. Different values of i have been tested at a high number of states, 9, and a low number of states, 3. The parameter's effect on the fit values can be found in figure 5.11 and the effect on the computational time can be found in figure 5.12. The fit values are quite steady when i is above six, at lower values the identification collapses for nine states. Remarkably the collapse does not occur until i is 3, well under nine.

As it comes down to the computational time, it steadily increases, as i increases. This is owing to the fact that many matrices in VODMA are sized with i .

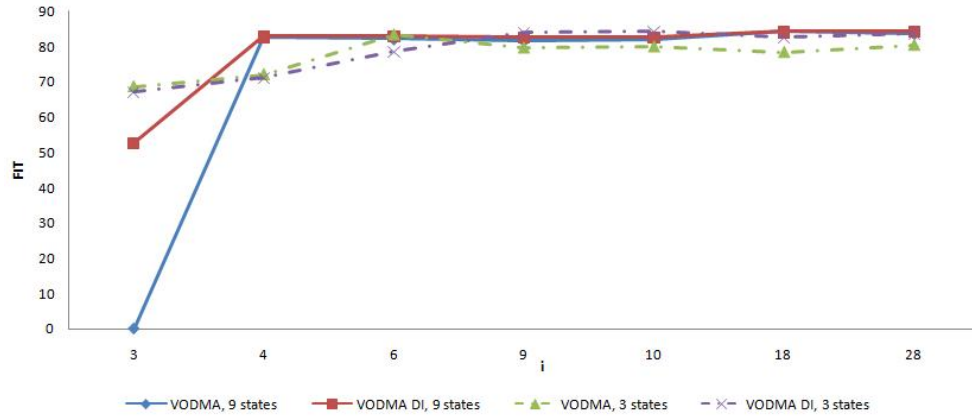


Figure 5.11. The average of the fit values over different values of design parameter i . The fits are stable above six and below they slowly decline for the threes states models whereas for the six states models they do not plunge until threes states.

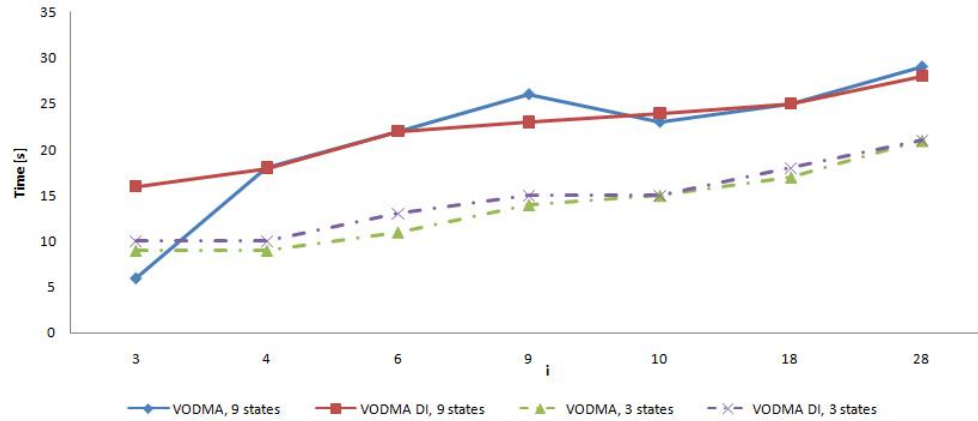


Figure 5.12. The computational time over different values of design parameter i . The computational time steadily increases together with i .

	VODMA + PEM	VODMA	VODMA DI + PEM
Average Fit [%]	77.10	68.13	75.76
Best No. Of Times	8	1	10

VODMA DI	N4SID + PEM	N4SID	PEM
71.01	75.91	73.66	51.05
1	7	1	0

Table 5.2. The average of all the average fit values for each method together with the number of times the method proved to be the best

5.9 Algorithm Comparison

In order to compare the different identification algorithms with each other the averages of the average validation fits, throughout all the tests conducted, have been studied. In addition it has been counted how many times an algorithm has come out with the best validation from each experiment. The results have been summarized in table 5.2.

It is a close call between the different algorithms but it can be said that enhancement with PEM gives an improvement. Moreover to use PEM with the physical model is a major improvement compared to the physical model. Nevertheless the performance of PEM is significantly worse when the physical model is used as initial model. When enhancing with PEM indirect identification VODMA has a slightly higher average fit while direct identification gave the best fit more times. Without PEM it is the other way around and N4SID is the most successful. However none of the algorithms are outstanding.

Chapter 6

Discussion and Conclusions

In this chapter conclusions will be drawn from the results in the preceding chapter. Which algorithm to use and why, how about the magnitudes, what is the resulting model, how many states, use linearization or not and which experiment setup? The chapter will be rounded off with some ideas about future work.

6.1 Identified Model

It has already been concluded in chapter 5 that closed-loop identification has to be used due to that the plant is marginally stable. Furthermore there is no reason to change the supplied PID controller as it gives adequate control over the system without preventing identification. In addition the sample time, T_s , of 0.01 s and a value of 28 for the VODMA parameter i are kept since they give good results without a significant increase in computational time. Moreover the best fit is obtained if the input is a PRBS with $\pm 32.4^\circ$ magnitude and a frequency range of 0-8 Hz. Whether this magnitude is the best for identification will be discussed later and below it will follow a more thorough reasoning concerning some other possibilities that are worth more attention than the just mentioned ones.

6.1.1 Linearization

It is of great importance to apply the nonlinear compensation to the system as it makes the system more linear by removing one of the nonlinearities. By considering the linearization as a part of the system a linear model can be obtained for the system. In further calculations on the system it can also be considered linear as long as the linearization is kept, this is a huge advantage since most mathematical tools are made for linear systems.

6.1.2 Number of States

To avoid unnecessary high computational times and give simpler results it is desired to have the lowest possible system order that does not compromise the model

fit too much. As shown in figure 5.9, there is hardly any reduction in fit down until five states. Five is a quite low number and hence the recommended number of states to use. That is for simulation, for controller design the model is not required to be as accurate since a feedback controller usually has at least some robustness against model errors. Therefore the recommendation is to try if the two-state model is accurate enough for controller design.

6.1.3 Model

For the experiment described in section 6.1 N4SID with PEM yields the best fit. The resulting equations follow beneath while graphs showing the fits can be found in figure 6.1.

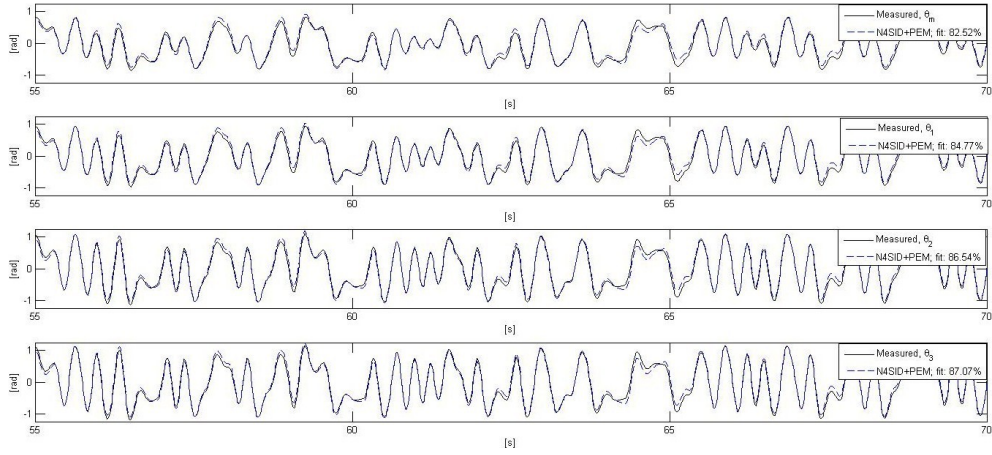


Figure 6.1. The fit values for the model of choice. A five state model estimated with N4SID and enhanced with PEM.

$$y = [\theta_m \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$$

$e(t)$ is white noise

u is the simulink control signal

$$x(t + T_s) = Ax(t) + B\tilde{u}(t) + Ke(t)$$

$$y(t) = Cx(t) + D\tilde{u}(t) + e(t)$$

$$\tilde{u}(t) = 2.44u(t) + 0.0397$$

$$A = \begin{pmatrix} 0.98625 & -0.088859 & 0.018178 & -0.011338 & 0.0060056 \\ 0.067406 & 0.99302 & -0.095223 & -0.069741 & 0.037538 \\ 0.18323 & 0.035865 & 0.71091 & -0.37252 & 0.15316 \\ -0.094799 & 0.16618 & 0.43111 & 0.93454 & -0.14788 \\ -0.0014655 & -0.023699 & -0.077026 & 0.16783 & 0.96445 \end{pmatrix}$$

$$B = \begin{pmatrix} -3.7319 * 10^{-5} \\ -0.00091545 \\ 0.0021908 \\ -0.015816 \\ 0.0059062 \end{pmatrix}$$

$$C = \begin{pmatrix} 21.632 & -0.53214 & 1.189 & -0.27991 & 0.21151 \\ 22.159 & -0.90058 & 0.36038 & -0.28709 & -0.31406 \\ 22.645 & -1.4813 & -0.66776 & 0.041256 & -0.11099 \\ 22.762 & -1.7469 & -0.98825 & 0.19018 & 0.032384 \end{pmatrix}$$

$$D = 0$$

$$K = \begin{pmatrix} 0.037197 & 0.0063496 & 0.010479 & 0.01299 \\ 0.0441 & -0.11378 & -0.14457 & -0.30301 \\ 0.69956 & 0.077472 & -0.069649 & -0.32686 \\ -0.058394 & -1.3051 & 0.55654 & 1.7211 \\ 3.1179 & -2.2402 & 0.07179 & 0.33551 \end{pmatrix}$$

When estimating the two state model there is a clear advantage in using direct identification VODMA with PEM enhancement. The equations become as follow,

$$y = [\theta_m \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$$

$e(t)$ is white noise

u is the simulink control signal

$$x(t + T_s) = Ax(t) + B\tilde{u}(t)$$

$$y(t) = Cx(t) + D\tilde{u}(t) + e(t)$$

$$\tilde{u}(t) = 2.44u(t) + 0.0397$$

$$A = \begin{pmatrix} 1.0278 & -0.053976 \\ 0.064834 & 0.9477 \end{pmatrix}$$

$$B = \begin{pmatrix} -110.27 \\ 18.013 \end{pmatrix}$$

$$C = \begin{pmatrix} -0.00013521 & -0.00061628 \\ -0.00013181 & -0.00063565 \\ -0.00012196 & -0.00065227 \\ -0.00011971 & -0.00065568 \end{pmatrix}$$

$$D = \begin{pmatrix} 0.053703 \\ 0.023234 \\ -0.023235 \\ -0.039438 \end{pmatrix}$$

A set of models has also been calculated for the system when the torsion module masses are set to the wide distance, these models and the accompanying results are found in appendix C.

6.2 Magnitude of Input Signal

In the chosen model an input magnitude of $\pm 32.4^\circ$ has been used, which is the largest amplitude that does not saturate the control signal. However it has been proven that the model agreement is worse at amplitudes that differ from the identification signal. Therefore the identification signal should, in magnitude, match the desired operating range. The wider spectrum of magnitudes the operating range contains, the more difficult it is to identify a model with good fit values. This is very natural since the more specialized something is, the more details it can care for. To keep the model application in mind while identifying the model is one of the rules of thumb within system identification.

6.2.1 Small Magnitudes

The identification results with the linear methods deteriorates as the magnitudes are decreased. This is interpreted as that the system behaves less linear at small magnitudes. An other possible thought is that the signal-to-noise ratio is higher when the signal is smaller. It is therefore harder to identify a good model at low magnitudes. On the other hand the system seems to have very little noise, see for example figure 5.5. However even though the system does not suffer from an easy seen high frequency output noise it might suffer from some kind of disturbance earlier in the chain. What makes the signal-to-noise explanation less plausible is the fact that a low magnitude identification with high magnitude validation gives a worse result than validation with low magnitudes. If the same model would work for both low and high magnitudes, only with different noise ratios, the model identified at low magnitudes should validate at similar levels for the different magnitudes, even a bit better for the large magnitudes as the noise ratio is lower. Hence the results are the opposite, the signal-to-noise-ratio theory is dismissed.

The different, less linear, behavior at small magnitudes is probably owing to the coulomb friction that acts nonlinearly on the system when it is leaving stand still. In addition, the effect of the asymmetric sliprings should be worse at a certain, quite small, amplitude when the system is just about to leave the downward position, or the upward.

6.3 Algorithms

The most obvious conclusion is that PEM should be used to enhance the results from the other algorithms, while on the other hand it should not be used by itself as it is heavily dependent on a good initial model. One remark is that it is not always possible to enhance with PEM, for example if the model is unstable.

VODMA is the algorithm that has support from the theory and it produces the best average fit value. Although it sometimes produces better fits through direct identification. Moreover it sometimes fail to identify the system but this fact as well as the long computational time, compared to N4SID, might be due to the implementation. The implementation of VODMA has been developed for

this thesis whereas N4SID is part of a commercial MATLAB toolbox that has had years of time for development and optimization.

When MATLAB is available the implementation of N4SID is very simple, only one command. Furthermore it has been proven the N4SID can give sufficient results even though the system is subjected to feedback. It only has slightly less fit than VODMA and has never failed completely. In the spirit of try the simplest thing first, N4SID (perhaps enhanced with PEM) is the recommended way to go. If a VODMA implementation is available one might as well try it too, but it is not worth the work of the implementation if N4SID can deliver sufficient results. It is possible that for a system that suffers from more noise and has a more advanced controller the theoretical advantage of VODMA, not needing the input and noise to be uncorrelated, will turn out in favor of VODMA.

6.4 Summary

In short, when the nonlinear compensation for the input signal is applied, a good model can be estimated for a desired, not too wide, operating range. The identification can be carried out with either VODMA or N4SID and then enhanced with PEM. Furthermore models of different complexity can be estimated for different applications.

6.5 Future Work

The most natural follow up for this work is to start developing model-based (or) robust controllers for torsion systems since this was the reason for developing the system model. In order to design robust controllers a model of uncertainties could be useful, it depends on the design method.

Of course it is also possible to develop the model further. More of the nonlinearities can be modeled and a complete physical model can be sought. Although it is sometimes too much hard work to acquire a physical model, it is always preferable as it gives more understanding about the system and how it will change when some parameters are altered.

Jumping back to the black-box research area, a model that works well for different magnitudes would be desirable. One solution to this problem is to have two models that transits into each other when the magnitudes change. Moreover several other identification algorithms could be evaluated using the torsion system.

Bibliography

- [1] David Banjerdpongchai, Manop Wongsaisuwan, Jitkomut Songsiri, Addy Wahyudie, and Shinji Hara. Development of Infrastructure of Control Systems Technology (Year II). Technical report, Control Systems Research Laboratory, Department of Electrical Engineering, Chulalongkorn University, 2005.
- [2] Alan Daniels and Beau Daniels. beaudaniels.com. "<http://www.beaudaniels.com>", January 2009.
- [3] Lars Eriksson and Lars Nielsen. *Modeling and Control of Engines and Drivelines*. 1 edition, 2008.
- [4] Urban Forssell and Lennart Ljung. Identification of Unstable Systems Using Output Error and Box-Jenkins Model Structures. *IEEE Transactions on Automatic Control*, 45(1):137, January 2000.
- [5] Lennart Ljung and Tomas McKelvey. Subspace Identification from Closed Loop Data. *Signal Processing*, 52:209, February 1996.
- [6] Quanser Consulting Inc., 80 Esna Park Drive, Unit #1-3 Markham, Ontario, L3R 2R6 Canada. *3-DOF Torsion SRV03 Reference Manual*. Document Number:608 Revision:01.
- [7] Inc. The MathWorks. System Identification Toolbox 7.2.1. "<http://www.mathworks.com/products/sysid/>", February 2009.
- [8] Jitendra K. Tugnait and Yi Zhou. On Closed-Loop System Identification Using Polyspectral Analysis. *IEEE*, (37), December 1998.
- [9] Peter Van Overschee and Bart De Moor. N4SID: Identification of Combined Deterministic-Stochastic Systems. Technical report, Departement Elektrotechniek at Katholieke Universiteit in Leuven, 1992.
- [10] Peter Van Overschee and Bart De Moor. N4SID: Identification of Combined Deterministic-Stochastic Systems. *Automatica*, 30(1):75–93, 1994.
- [11] Peter Van Overschee and Bart De Moor. Closed Loop Subspace System Identification. Technical report, Departement Elektrotechniek at Katholieke Universiteit in Leuven, September 1996.

- [12] Peter Van Overschee and Bart De Moor. Closed Loop Subspace System Identification. In IEEE, editor, *Proceedings of the 36th Conference on Decision & Control*, pages 1848–1853, December 1997. San Diego, California.

Appendix A

Torsion System Specifications

The following table is a direct reprint of the one found in the Quanser Manual [6]. It includes all given system specifications.

<i>Description</i>	<i>Value</i>	<i>Unit</i>
SRV03 DC Motor		
Motor Maximum Continuous Torque	0.219	Nm
Motor Peak Torque	1.6	Nm
Motor Maximum Continuous Current	2.95	A
Motor Torque Constant	0.0742	A
Motor Peak Current	21.4	A
Motor Armature Resistance	1.11	Ω
Motor Armature Inductance	1.81	mH
Moment of Inertia of Motor Rotor	3.11E-005	kg m ²
Motor Mechanical Time Constant	6.27	ms
PWM Current Amplifier (Each Channel):		
PWM Amplifier Maximum Continuous Current	10	A
PWM Amplifier Peak Current	20	A
PWM Amplifier Maximum Continuous Power	800	W
PWM Amplifier Peak Power	1600	W
PWM Amplifier Bandwidth (Current Mode)	3	kHz
PWM Amplifier Gain	1.0	A/V
Rotary Torsion Module (Each):		
Disc Weight Mass	0.123	kg
Disc Weight Diameter	0.038	m
Load Support Bar Length	0.044	m
Load Support Bar Mass	0.021	kg
Flexible Coupling Torsional Stiffness Constant	1.0	Nm/rad
Equivalent Moment of Inertia, as seen at the Torsion Module Load Shaft	5.46E-004	kg m ²
SRV03 Optical Encoder:		
Encoder Line Count	2048	lines/rev
Encoder Resolution (In Quadrature)	8192	counts/rev
Encoder Sensitivity (In Quadrature)	7.67E-004	rad/count
Encoder Type	TTL	
Encoder Signals	A, B, Index	
Torsion Module Optical Encoder:		
Encoder Line Count	1024	lines/rev
Encoder Resolution (In Quadrature)	4096	counts/rev
Encoder Sensitivity (In Quadrature)	15.34E-004	rad/count
Encoder Type	TTL	
Encoder Signals	A, B, Index	
Current Sense:		
Current Sense Calibration at $\pm 10\%$	0.5	V/A

Table A.1. The specifications of the torsion system parameters

Appendix B

Algorithm Verification

To verify that the identification algorithms are working properly they have been tested with data created from simulation. It is the Quanser model that has been simulated when subjected to the same RGS input as in some of the real-life experiments. A bandwidth limited white noise has been added to all the outputs. The amplitudes are ranging from $\pm 4^\circ$, a considerably amount of noise. There have been simulations of as well open-loop as closed-loop system, where the closed loop has been constructed with the same PD controller that is used in real life.

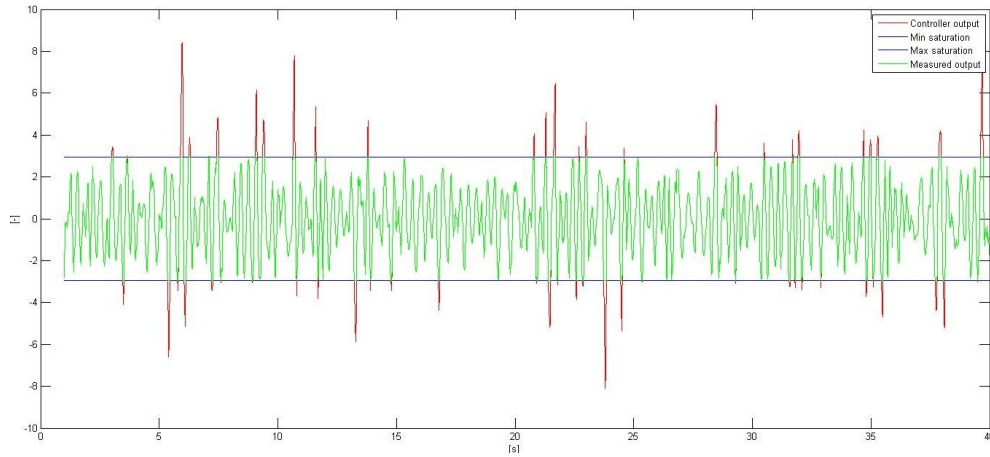


Figure B.1. The saturated control signal used for identification in the fourth experiment

B.1 Simulation and Results

Firstly the algorithms undertake an open-loop experiment where they give almost flawless results. Then they are subjected to a closed-loop experiment where the

Simulation	VODMA+PEM	VODMA	VODMA DI+PEM	VODMA DI	N4SID+PEM	N4SID	PEM	Qanser Model
Open loop	-	-	100	100	99.98	99.98	95.41	99.46
Closed loop	99.53	99.60	99.53	99.58	99.36	99.37	86.82	92.96
Closed loop w. Noise	87.30	87.30	93.10	93.10	93.24	93.15	89.14	90.36
Closed loop w. Saturation	78.82	78.82	42.50	65.50	47.93	32.70	55.18	39.12

Table B.1. Fit values in percent for different identification algorithms and simulations. Note that the Qanser model does not reach all the way to 100% fit in the top two experiments. This is due to the fact that simulation data comes from Simulink whereas the fit has been calculated from a simulation using the System Identification Toolbox in MATLAB.

fit values only have a slight dip from the open-loop experiments. In conclusion all algorithms seems plausible for closed-loop identification. However since all real systems suffer from noise, the interesting case is if the algorithms still can identify the system when a disturbance exists. While the real system gives quite stable output, or in other words, is short on noise, an added $\pm 4^\circ$ white noise disturbance will clearly do for verification. The fit values naturally decrease, but important to notice is that in an open-loop simulation, the Quanser model, the true system, falls almost equally much in fit. Much better results can not be asked for.

As a final experiment a saturation of the control signal is replacing the noise. The result was significantly worse than with the noise as the system no longer is linear. This means that it is of great importance to avoid saturation of the control signal when conducting the real-life experiments. The saturation of the control signal can be seen in figure B.1 and the model fits resulting from the simulations are shown in table B.1.

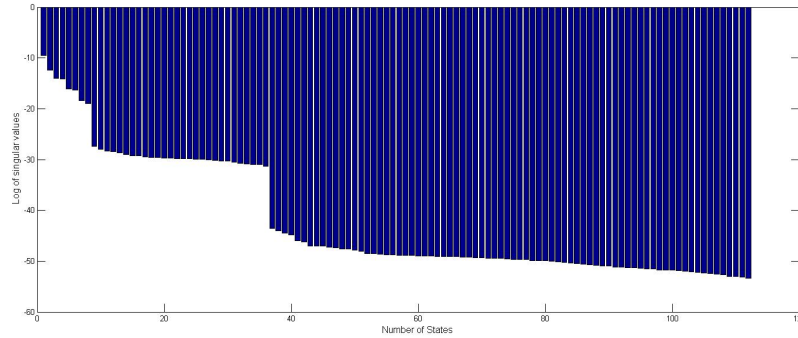


Figure B.2. The logarithms of the singular values obtained from VODMA during open-loop simulation without noise. The correct number of eight states is a sensible choice.

B.2 Singular Values

The subspace methods examine the singular values to decide the model order. The model order should equal the number of non-zero singular values but to keep the order within reasonable limits some singular values have to be approximated to zero. This done by manual inspection of a plot with the singular values sorted in descending order. In figure B.2 the singular-value plot, from VODMA¹ in the open-loop simulation, is found. The plot clearly shows that eight states is a sensible choice. A perfect match with the true model. However when noise is added in the closed-loop case the plot changes into figure B.3, where it is more difficult to read out a suitable number of states. As a consequence several different number

¹The singular-value plot generated by N4SID is highly similar and using the N4SID plots instead of the VODMA plots would in no way affect the conclusions

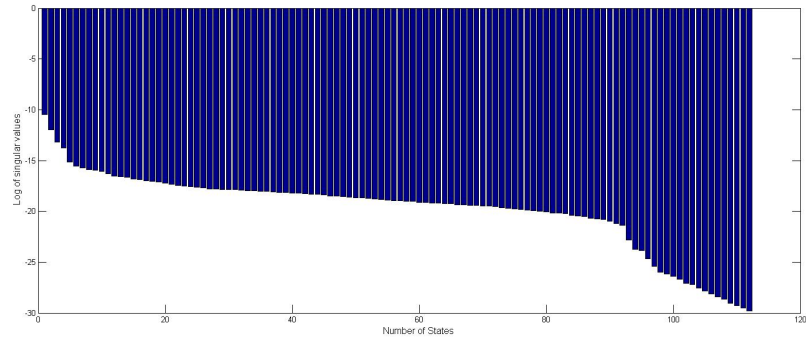


Figure B.3. The logarithms of the singular values obtained from VODMA during closed-loop simulation with noise. It is not so clear how many states the system has.

of states are tried out for the real-life experiments as the singular-value plots have proven untrustworthy.

Appendix C

Model for Torsion System with Increased Inertia

The masses on the torsion modules can be shifted to a wider distance and thereby increase the inertia of the torsion module. This is interesting hence, when the system will be used to develop robust controllers, the torsion system should be altered to test the robustness. Of course it is of interest to verify the robustness already at the simulation stage. This means that a model for the wide-spaced masses is needed.

All torsion modules can be altered separately but only the case when all modules are set to wide distance has been identified. The identification approach is built upon that the new system is highly similar to the old system and thereby can be identified in the same way. Even though increased inertia increases the sluggishness of the system and thereby, in accordance with Newton's second equation, more torque (higher input) is required to change the motion of the discs, the control signal still dodge the saturation. This is a very satisfying result hence the identification signal should not be altered too much when identifying the new system because the desired operating range still is unchanged. It is unchanged even though the system has changed hence the system change is thought of as a modeling error, not a new application.

C.1 Model

In contrast to when identifying the system with narrow mass spacing, direct identification through VODMA with PEM yields the best fit. The resulting equations

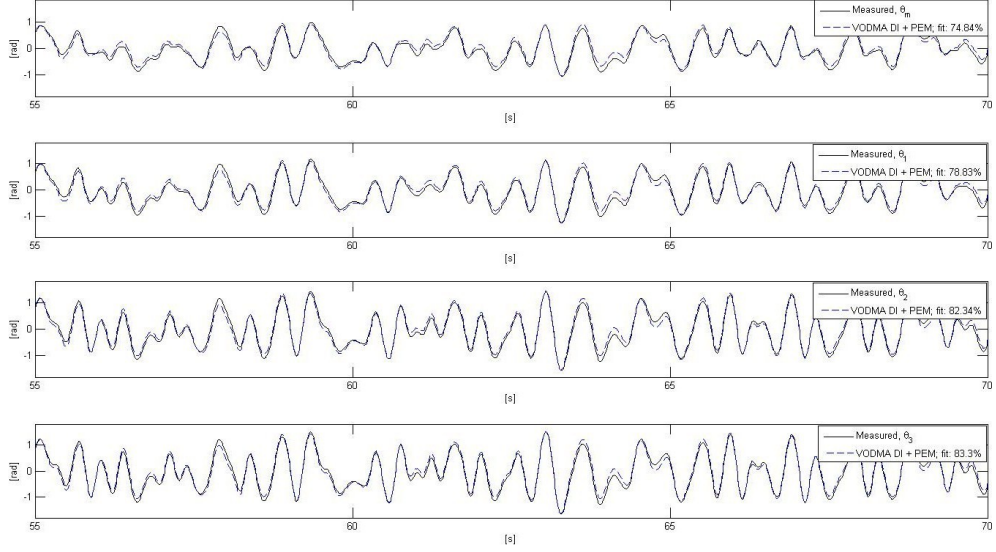


Figure C.1. The fit values for the model of choice. Five state model estimated with direct identification VODMA and enhanced with PEM.

follow beneath while graphs showing the fits can be found in figure C.1.

$$y = [\theta_m \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$$

$e(t)$ is white noise

u is the simulink control signal

$$x(t + T_s) = Ax(t) + B\tilde{u}(t)$$

$$y(t) = Cx(t) + D\tilde{u}(t) + e(t)$$

$$\tilde{u}(t) = 2.44u(t) + 0.0397$$

$$A = \begin{pmatrix} 1.0294 & -0.053416 & 0.00077334 & -0.001887 & -0.0022221 \\ 0.05224 & 0.95104 & 0.0015586 & 0.001363 & 0.02468 \\ 0.0016215 & -0.074788 & 0.82245 & -0.22456 & 0.11306 \\ 0.038584 & 0.13612 & 0.40111 & 0.88703 & 0.10991 \\ 0.010985 & -0.022959 & 0.0053411 & -0.18381 & 0.83922 \end{pmatrix}$$

$$B = \begin{pmatrix} -73.276 \\ 15.205 \\ 34.589 \\ -54.536 \\ 7.9447 \end{pmatrix}$$

$$C = \begin{pmatrix} -0.0001027 & -0.0004661 & 0.00073895 & 5.3693 * 10^{-5} & -0.00060132 \\ -0.00011601 & -0.00054262 & 0.00052594 & 5.7563 * 10^{-5} & -0.00037542 \\ -0.00013117 & -0.00070143 & 0.00012638 & -3.0326 * 10^{-5} & -0.00025135 \\ -0.00013764 & -0.00076339 & -2.2118 * 10^{-5} & -7.0494 * 10^{-5} & -0.00024277 \end{pmatrix}$$

$$D = \begin{pmatrix} -0.015261 \\ -0.028516 \\ -0.025851 \\ -0.021586 \end{pmatrix}$$

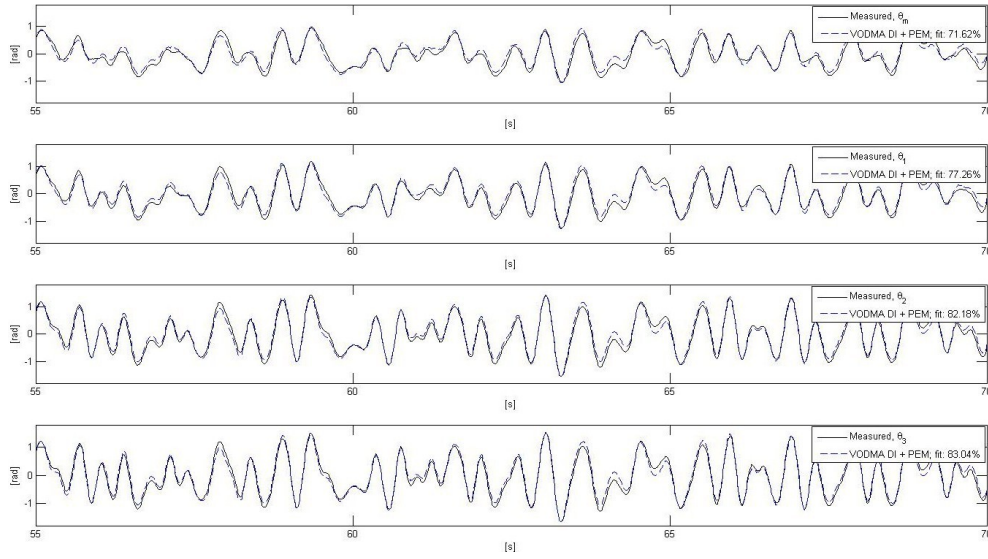


Figure C.2. The fit values for the two states model. Estimated with direct identification VODMA and enhanced with PEM

When estimating the two state model there is a clear advantage in using direct identification VODMA with PEM enhancement. The fit value graphs are found in figure C.2 and the equations becomes as follow,

$$y = [\theta_m \quad \theta_1 \quad \theta_2 \quad \theta_3]^T$$

$e(t)$ is white noise

u is the simulink control signal

$$x(t + T_s) = Ax(t) + B\tilde{u}(t)$$

$$y(t) = Cx(t) + D\tilde{u}(t) + e(t)$$

$$\tilde{u}(t) = 2.44u(t) + 0.0397$$

$$A = \begin{pmatrix} 1.0301 & -0.055162 \\ 0.052002 & 0.94723 \end{pmatrix}$$

$$B = \begin{pmatrix} -72.089 \\ 15.499 \end{pmatrix}$$
$$C = \begin{pmatrix} -0.00016506 & -0.00066832 \\ -0.00015897 & -0.00069919 \\ -0.00014327 & -0.00071396 \\ -0.00013888 & -0.00071526 \end{pmatrix}$$
$$D = \begin{pmatrix} 0.072638 \\ 0.0036687 \\ -0.023006 \\ -0.04417 \end{pmatrix}$$

Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>