



# LUND UNIVERSITY

## Processidentifiering - Projektarbeten våren 1994

Johansson, Rolf; Andersson, Lennart; Jönsson, Ulf

1994

*Document Version:*  
Förlagets slutgiltiga version

[Link to publication](#)

*Citation for published version (APA):*  
Johansson, R. (Red.), Andersson, L., & Jönsson, U. (1994). *Processidentifiering - Projektarbeten våren 1994*. (Technical Reports TFRT-7522). Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*  
3

### General rights

Unless other specific re-use rights are stated the following general rights apply:  
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

ISSN 0280-5316  
ISRN LUTFD2/TFRT--7522--SE

# Processidentifiering Projektarbeten våren 1994

Rolf Johansson (Red.)  
Lennart Andersson  
Ulf Jönsson

Institutionen för Reglerteknik  
Lunds Tekniska Högskola  
Maj 1994

<b>Department of Automatic Control</b> <b>Lund Institute of Technology</b> P.O. Box 118 S-221 00 Lund Sweden		<b>Document name</b> Internal Report	
		<b>Date of issue</b> Maj 1994	
		<b>Document Number</b> ISRN LUTFD2/TFRT--7499--SE	
<b>Author(s)</b> R. Johansson, L. Andersson, U. Jönsson (Red.)		<b>Supervisor</b> Rolf Johansson	
		<b>Sponsoring organisation</b>	
<b>Title and subtitle</b> Processidentifiering — Projektarbeten våren 1994 (Term papers in Process identification—Spring Semester 1994)			
<b>Abstract</b> <p>This volume contains some term papers of the students taking the graduate course in process identification. The following identification subjects have been studied: A ball-and-beam process; A ball-and-houp process; An inverted-pendulum process; A double fan and plate process using subspace methods; Human postural dynamics; A Fan Process; Neural Networks for identification and control.</p>			
<b>Key words</b>			
<b>Classification system and/or index terms (if any)</b>			
<b>Supplementary bibliographical information</b>			
<b>ISSN and key title</b> 0280-5316			<b>ISBN</b>
<b>Language</b> Swedish, English	<b>Number of pages</b> 185	<b>Recipient's notes</b>	
<b>Security classification</b>			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Fax +46 46 110019, Telex: 33248 lubbis lund.

## FÖRORD

Föreliggande arbete innehåller några rapporter från projektarbeten inom en kurs (vt 94) i processidentifiering vid Institutionen för Reglerteknik, LTH. Kursen *Processidentifiering* gavs första gången under höstterminen 1987 såsom reguljär fortsättningskurs och forskarutbildningskurs vid Reglerteknik, LTH. Kursdeltagarna förutsättes ha goda förkunskaper i reglerteori och stokastiska processer svarande mot institutionens kurs *FRT020 Digital Reglering*. Under kursens gång har kursdeltagarna förberetts för sina uppgifter genom fyra laborationer: Frekvensanalys, Interaktiv identifiering, Realtidsidentifiering, Synthesorienterad identifiering. Vidare har kursen innehållit övningar med Matlab-uppgifter i Processidentifiering.

Uppgifterna har alla omfattat identifiering med jämförelse mellan några olika metoder. Följande identifieringsproblem har studerats:

- o Identification of a Double Fan and Plate Process Using Subspace Methods (Karl Henrik Johansson)
- o Identification of the ball-and-beam process (Jeroen Borgers, Martin Nyquist, Ola Wall)
- o "Ekorrhjulet" (Fredrik Bistedt, Björn Pettersson, Per-Ola Wictor)
- o Mänsklig balans (Henrik Amilon, Johan Kleberg)
- o Identifiering av bom- och kulprocessen (Johan Gustafsson, Jerker Linder)
- o Identifiering och reglering av inverterad pendel (Martin Nilsson, Anders Lyngsjö)
- o Identifiering och Reglering av Bom och Kula (Mats Ahnelöv, Michael Rosenberg, Carl-Johan Weiderstrand)
- o Inverterad pendel (Anders Abrahamsson, Håkan Ivarsson, Mats Jerlebo)
- o Identifiering av en fläktprocess (Roger Granath, Fahed Saouan)
- o Identifiering av kroppens balanssystem (Charlotta Johnsson, Hans Nilsson, Jerker Sjögren)
- o Neurala nätverk för identifiering och reglering (Jesper Bergstrand, Tord Björsne)

Det är vår förhoppning, att dessa projektarbeten kan tjäna som exempel på tillämpningar av identifieringsmetodik.

Rolf Johansson, Lennart Andersson, Ulf Jönsson (Red.)



# Identification of a Double Fan and Plate Process Using Subspace Methods

Karl Henrik Johansson

Department of Automatic Control  
Lund Institute of Technology  
March 1994

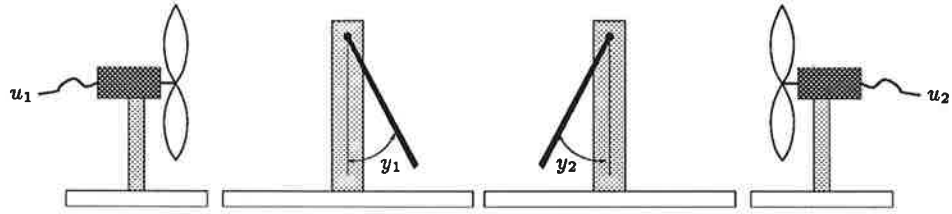


Figure 1. The process.

## 1. Introduction

Multi-inputs multi-outputs (MIMO) processes are very common in control systems. Still there exist no common tool for identification of such processes. Single-input single-output (SISO) identification methods are often used also in the MIMO case, see [Ljung, 1987] or [Johansson, 1993]. This approach has several drawbacks, *e. g.* it is then harder to see if exactly the same dynamics appears in various parts of the MIMO process, more computational effort is needed, and probably various data sequences have to be collected.

In this report we use a fairly new method for MIMO identification called subspace system identification, *e. g.* see the survey [van der Veen *et al.*, 1993]. This method makes an extensive use of the singular value decomposition (SVD), while finding a linear operator that spans a linear space including the measured outputs. For the calculation we use a numerical algorithm called N4SID [van Overschie and de Moor, 1994], which has been implemented in Matlab [Ljung *et al.*, 1994]. The process considered is constructed out of two sets of the fan and plate lab process described in [Gustafsson and Bernhards-son, 1990]. Hence, we call it *the double fan and plate process*.

In Section 2 we describe the process followed by the data collection in Section 3. The subspace system identification method is briefly discussed in Section 4 followed by the identification of the process in Section 5. Finally, in Section 6 we conclude our study.

## 2. The Process

In Figure 1 the double fan process is shown. It is a two-input two-output process, which consists of two fans and two rectangular plates each hanging from one of its edges. The plates are mounted such that they can swing back and forth. The air streams from the fans will affect the positions of the plates. The control signals  $u_1$  and  $u_2$  are the voltages to the fan motors, and the outputs  $y_1$  and  $y_2$  are the angles between the plates and the vertical plane.

By simple visual observation some characteristics of the process can be stated.

- The period time of the lightly damped oscillation of the plates are about 1 and 1.5 seconds, respectively.
- The fan motor time constants are approximately equal to 0.5 seconds.
- The dynamics describing how a fan affects the furthest plate is more involved than the dynamics between the fan and the nearest plate.

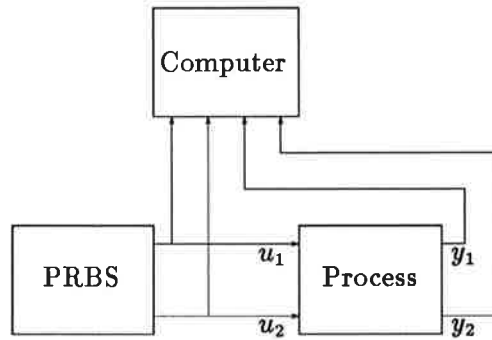


Figure 2. The experiment setup.

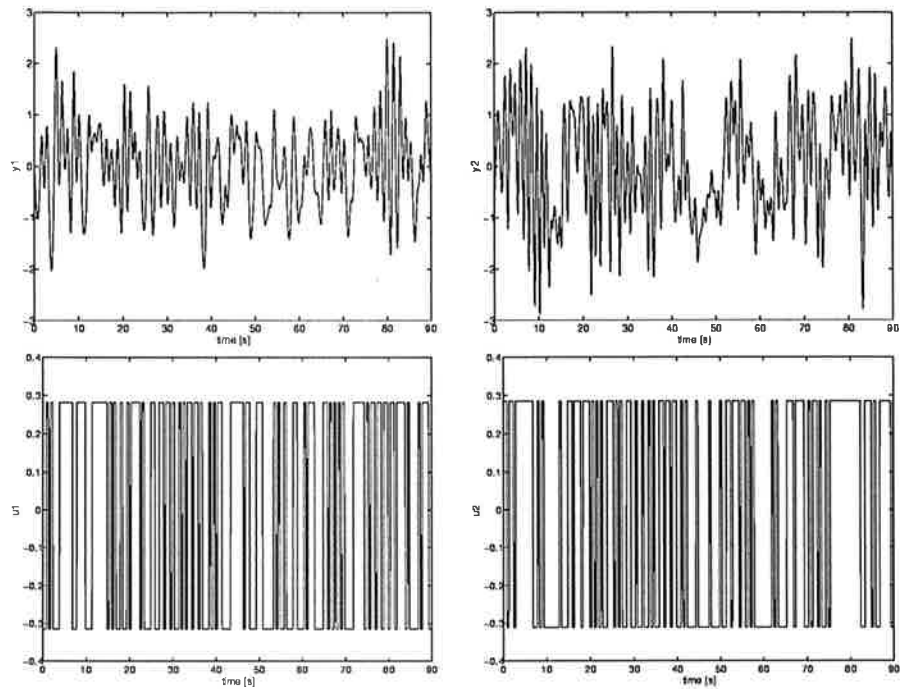


Figure 3. Data used for the identification.

### 3. The Data Collection

We now describe how the experiment was carried out.

The data were collected using an extended version of the program Logger [Gustafsson, 1990]. The new version gives two independent PRBS (pseudo random binary sequence) signals as outputs and is able to collect two input signals. The experiment setup is shown in Figure 2. Some data about the experiment are summarized in the following tabular.

Sample period	50 ms
Number of samples	3600
PRBS mean	0.7 V
PRBS amplitude	0.3 V
PRBS period	10 samples

(Here the *PRBS period* is the lowest number of samples between two switches in the PRBS signal. By choosing it higher than one, it is possible to concentrate

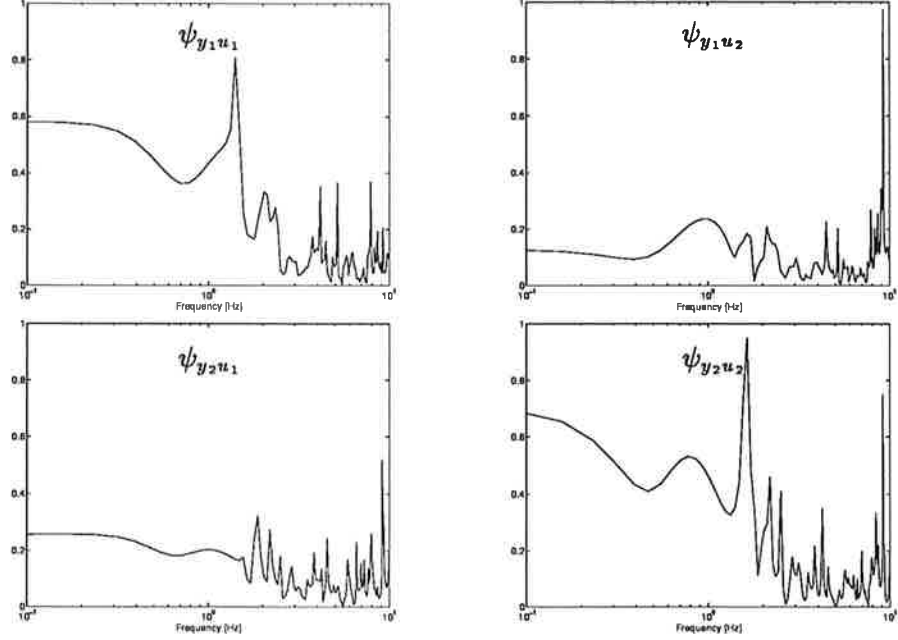


Figure 4. Coherence spectra.

the input spectrum to the most interesting frequency domain but still having a high sampling rate.)

We divide the collected data in two halves; the first one will be used for identification while the second will be used for verification. In Figure 3 the first half is shown.

To be able to tell in what frequency interval we can expect to estimate a reliable model, it is useful to study the coherence spectra. The coherence spectrum  $\psi_{yu}$  between two signals  $y$  and  $u$  is defined as

$$\psi_{yu}(\omega) = \frac{|\phi_{yu}(\omega)|}{\sqrt{\phi_u(\omega)\phi_y(\omega)}}$$

In a frequency interval where  $y$  can be expressed as a linear filtering of  $u$ ,  $\psi_{yu} \approx 1$ . In Figure 4 the coherence spectra for the process data are shown. Note that especially for the cross-couplings ( $u_1 \rightarrow y_2$  and  $u_2 \rightarrow y_1$ ) the coherence spectra are low for all frequencies. Further, all spectra are smooth up to 1.5 Hz, which thus is a kind of identifiability limit.

## 4. The Ideas Behind the Subspace Methods

In this section we will briefly discuss the ideas behind the subspace identification method. Only simple deterministic setups are considered. The discussion follows the framework of [van der Veen *et al.*, 1993], but for the last part that uses the algorithm in [Verhaegen and Dewilde, 1992]. For simplicity we will only consider SISO-systems.

### An Idealized Set Up

Assume that two sequences of data are given: the input vector

$$\bar{u} = \left( \dots \ u_{-1} \ u_0 \ u_1 \ \dots \right)^T$$

and the output vector

$$\bar{y} = \begin{pmatrix} \dots & y_{-1} & y_0 & y_1 & \dots \end{pmatrix}^T$$

They are connected by a causal linear time-invariant stable operator  $T$  such that  $\bar{y} = T\bar{u}$ . To start with, we assume that  $T$  is somehow known (we will drop this requirement later). Associated with  $T$  is its impulse response

$$h = \begin{pmatrix} \dots & 0 & 0 & h_0 & h_1 & h_2 & \dots \end{pmatrix}^T = T \begin{pmatrix} \dots & 0 & 0 & 1 & 0 & 0 & \dots \end{pmatrix}^T \quad (1)$$

In matrix representation we can write

$$\begin{pmatrix} \vdots \\ y_{-1} \\ y_0 \\ y_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \ddots & & & & \\ \dots & h_0 & & & \\ & h_1 & h_0 & & \\ \dots & h_2 & h_1 & h_0 & \\ & \vdots & & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ u_{-1} \\ u_0 \\ u_1 \\ \vdots \end{pmatrix}$$

Our system identification problem is now: given  $\bar{u}$  and  $\bar{y}$  find a state-space realization

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \quad (2)$$

such that its impulse response equals the impulse response of  $T$ , *i. e.*

$$\begin{aligned} &\begin{pmatrix} \dots & 0 & D & CB & CAB & CA^2B & \dots \end{pmatrix} \\ &= \begin{pmatrix} \dots & 0 & h_0 & h_1 & h_2 & h_3 & \dots \end{pmatrix} \end{aligned}$$

Above  $A$  is a  $n \times n$  matrix,  $B$  and  $C^T$  are  $n \times 1$  vectors, and  $D$  is a scalar. Note that we only consider minimal state-space models, but even then the representation (2) is not unique.

To solve the identification problem we divide  $u$  and  $y$  into a past input sequence

$$\bar{u}_- = \begin{pmatrix} u_{-1} & u_{-2} & \dots \end{pmatrix}^T$$

and a future output sequence

$$\bar{y}_+ = \begin{pmatrix} y_0 & y_1 & \dots \end{pmatrix}^T$$

and assume that  $u_k = 0, k \geq 0$  (which is no restriction, as will be shown in next subsection). Then we can write  $\bar{y}_+ = H\bar{u}_-$  where

$$H = \begin{pmatrix} h_1 & h_2 & h_3 & \dots \\ h_2 & h_3 & & \\ h_3 & & \ddots & \\ \vdots & & & \end{pmatrix} \quad (3)$$

is a Hankel matrix.

$H$  given from the state-space structure (2) has four important properties that is useful for our purpose. Before we list them let us introduce the notation  $H^\uparrow$ , which is the operator  $H$  with its top row deleted, and  $H^\leftarrow$ , which is the operator  $H$  with its first column deleted.  $H$  fulfills the following statements.

1.  $\text{rank } H = n$  the number of states in (2).
2.  $H = \mathcal{O}\mathcal{C}$  where

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \end{pmatrix} \quad \mathcal{C} = \begin{pmatrix} B & AB & A^2B & \dots \end{pmatrix}$$

are the observability and controllability operator, respectively.

3.  $H^\dagger = \mathcal{O}^\dagger \mathcal{C} = \mathcal{O}A \cdot \mathcal{C}$
4.  $H^\leftarrow = \mathcal{O}\mathcal{C}^\leftarrow = \mathcal{O} \cdot AC$

The first property follows directly from the second, since  $\text{rank } \mathcal{O} = \text{rank } \mathcal{C} = n$  by definition for a minimal realization. The third and fourth properties are called *shift-invariance structure*.

We will now describe an algorithm for finding a realization  $\{A, B, C, D\}$  given the transfer operator  $T$ .

1. Given  $T$ , determine  $H$  out of (1) and (3). Derive  $n = \text{rank } H$  and factorize  $H = \mathcal{O}\mathcal{C}$ .
2. Since  $\mathcal{O}A = \mathcal{O}^\dagger$  and  $\mathcal{O}$  has full row rank, we derive

$$A = \mathcal{O}^+ \mathcal{O}^\dagger$$

where  $^+$  denotes the pseudo-inverse. Further,

$$B = \mathcal{C}_{(:,1)} \quad \mathcal{C} = \mathcal{O}_{(1,:)} \quad D = h_0$$

The subscripts denotes the first column and the first row, respectively.

### A Less Idealized Set Up

In the description above we assumed that impulse response measurements were somehow available. This is seldom the case in practice. Instead we know only a segment of the system response, and often it is the response to some non-impulsive input signal. Because we want persistent excitation, see [Ljung, 1987], it is preferable to choose the input to be a white noise sequence or something similar (a PRBS is often used). Further,  $\bar{u}$  and  $\bar{y}$  are in all practical situations of finite length. We will now show how it is possible in this less idealized case to estimate  $\{A, B, C, D\}$ . We skip the case when there is additive white Gaussian noise on the input and output signals in the state-space system (2). This problem is more involved.

Assume one input sequence

$$\begin{pmatrix} u_0 & u_1 & \dots & u_{N+L-1} \end{pmatrix}$$

and one output sequence

$$\begin{pmatrix} y_0 & y_1 & \dots & y_{N+L-1} \end{pmatrix}$$

are given. Out of these sequences construct two matrices  $U$  and  $Y$ , where

$$U = \begin{pmatrix} \bar{u}_0 & \bar{u}_1 & \dots & \bar{u}_{N-1} \end{pmatrix} = \begin{pmatrix} u_0 & u_1 & \dots & u_{N-1} \\ u_1 & u_2 & & u_N \\ \vdots & & \ddots & \vdots \\ u_L & u_{L+1} & \dots & u_{N+L-1} \end{pmatrix} \quad (4)$$

and  $Y$  is defined in a similar way. We assume  $N \gg L \geq n$  and the input sequence to be persistently exciting. This leads to  $U$  and  $Y$  having full column rank.

Because the input sequence is not zero from a given time, the approach in the previous subsection can not be directly applied. But since the rank of  $U$  is  $L + 1$ , it is possible to express the last  $N - L - 1$  columns of  $U$  by a linear combination of  $L + 1$  new input vectors  $\bar{u}'_0, \dots, \bar{u}'_L$ . Further, (the unknown)  $T$  is assumed to be a linear operator so we can derive a new output sequence  $\bar{y}'_0, \dots, \bar{y}'_{N-1}$  by applying the same linear combination to the original output sequence. We may express this with an  $N \times N$  transformation matrix  $\tilde{Q}$  in the formula

$$\begin{pmatrix} \bar{u}_0 & \bar{u}_1 & \dots & \bar{u}_{N-1} \\ \bar{y}_0 & \bar{y}_1 & \dots & \bar{y}_{N-1} \end{pmatrix} \tilde{Q} = \begin{pmatrix} \bar{u}'_0 & \dots & \bar{u}'_L & 0 & \dots & 0 \\ \bar{y}'_0 & \dots & \bar{y}'_L & \bar{y}'_{L+1} & \dots & \bar{y}'_{N-1} \end{pmatrix}$$

Thus, we can view the output vectors  $\bar{y}'_{L+1}, \dots, \bar{y}'_{N-1}$  as given by

$$\begin{pmatrix} \bar{y}'_{L+1} & \dots & \bar{y}'_{N-1} \end{pmatrix} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^L \end{pmatrix} \cdot X = \mathcal{O}_{L+1} X \quad (5)$$

where  $X$  is an unknown matrix, which can be viewed as an initial-state matrix. Clearly  $\mathcal{O}_{L+1}$  is a finite part of the operator  $\mathcal{O}$  (and, of course, it also has a shift-invariance property).

Do a QR (or rather RQ) factorization as

$$\begin{pmatrix} U \\ Y \end{pmatrix} = \begin{pmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{pmatrix} \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$$

where  $R_{11}$  and  $R_{22}$  are lower triangular  $(L + 1) \times (L + 1)$  matrices.  $Q_1$  and  $Q_2$  are  $2(L + 1) \times N$  matrices, whose columns are orthogonal. Further, we derive the SVD

$$R_{22} = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

where  $U_1$  has dimension  $(L + 1) \times n$ . We need the following lemma [Verhaegen and Dewilde, 1992].

**LEMMA 1**

$U_1$  has shift-invariance structure.

*Proof:* There exists a non-singular  $n \times n$  matrix  $T$  such that

$$\mathcal{O}_{L+1} T = U_1$$

since  $\text{Im } \mathcal{O}_{L+1} = \text{Im } U_1$ . Thus,

$$\begin{pmatrix} CT \\ CAT \\ \vdots \\ CA^L T \end{pmatrix} = \begin{pmatrix} (CT) \\ (CT)(T^{-1}AT) \\ \vdots \\ (CT)(T^{-1}A^L T) \end{pmatrix} = \begin{pmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^L \end{pmatrix} = U_1$$

Hence,  $U_1$  has shift-invariance structure. □

Now the estimates  $\{A, B, C, D\}$  are given by the following equations

$$U_1^\dagger A = U_1^\top$$

$$C = U_{1(1,:)}$$

$$\text{vec}(U_2^T R_{21} R_{11}^{-1}) = \begin{pmatrix} U_{2(1,:)}^T & U_{2(2,:)}^T & \cdots & U_{2(L+1,:)}^T \\ U_{2(2,:)}^T & U_{2(3,:)}^T & & 0 \\ \vdots & & & \\ U_{2(L+1,:)}^T & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & U_1^\dagger \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix}$$

where  $U_1^\dagger$  and  $U_1^\top$  are defined in a similar way to the operator case in the previous subsection.  $\text{vec } E$  is the column vector consisting of the column vectors of  $E$  stacked on top of each other.

We illustrate the procedure above with a simple example.

### A Simple Example

Consider the first-order difference equation

$$y_{k+1} = -0.7y_k + u_{k+1}$$

It has a state-space representation

$$x_{k+1} = -0.7x_k - 0.7u_k$$

$$y_k = x_k + u_k$$

If we apply the sequence

$$\bar{u} = (\dots \ 0 \ 0 \ 1 \ 2 \ 1 \ 1 \ \dots)$$

it will result in the output sequence

$$\bar{y} = (\dots \ 0 \ 0 \ 1 \ 1.3 \ 0.09 \ 0.937 \ \dots)$$

The choices  $L = 1$  and  $N = 3$  gives

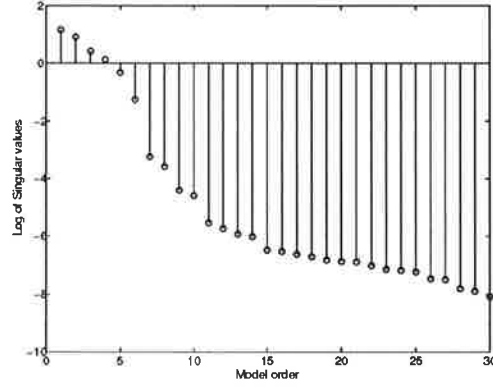
$$\begin{pmatrix} U \\ Y \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1.3 & 0.09 \\ 1.3 & 0.09 & 0.937 \end{pmatrix}$$

$$= \left( \begin{array}{cc|cc} -2.4490 & 0.0000 & 0.0000 & 0.0000 \\ -2.0410 & 1.3540 & 0.0000 & 0.0000 \\ -1.5060 & 0.2326 & -0.6121 & 0.0000 \\ -0.9867 & 1.1910 & 0.4284 & 0.0000 \end{array} \right) \times \left( \begin{array}{ccc} -0.4082 & -0.8165 & -0.4082 \\ 0.8616 & -0.4924 & 0.1231 \\ -0.3015 & -0.3015 & 0.9045 \\ 0.0000 & 0.0000 & 0.0000 \end{array} \right)$$

From this we get

$$\begin{aligned} R_{22} &= \begin{pmatrix} -0.6121 & 0 \\ 0.4284 & 0 \end{pmatrix} = \begin{pmatrix} -0.8192 & 0.5735 \\ 0.5735 & 0.8192 \end{pmatrix} \begin{pmatrix} 0.7471 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \end{aligned}$$





**Figure 5.** Singular values of the data decomposition.

Finally, using the equations in the previous subsection,

$$A = -0.7 \quad B = 0.8546 \quad C = -0.8192 \quad D = 1$$

A simple check shows this is a correct realization.

## 5. Identification of a Subspace Model

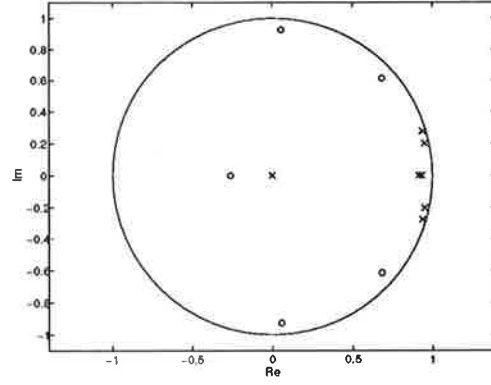
In this section the identification of a MIMO subspace model is described. We will estimate a state-space model for the double fan and plate process by using a Matlab program for subspace system identification [Ljung *et al.*, 1994]. The program is an implementation of the N4SID algorithm (numerical algorithm for state-space subspace identification) in [van Overschee and de Moor, 1994]. Only the basic ideas of the algorithm are the same as in the previous section, *e.g.* the N4SID algorithm includes a noise model.

The estimated two-inputs two-outputs model is given in the innovation form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ke_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned}$$

where  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $K$  are the matrices to be identified, and  $e$  is a white Gaussian noise sequence. We check what state-space dimension is necessary by a singular value decomposition of the data structured in a certain way. In Figure 5 these singular values are presented. We see a considerable jump between the sixth and seventh singular value. Thus it is reasonable to choose model order six. Higher and lower model orders have been examined, but they gave pole-zero cancellations and increased prediction errors, respectively.

The identification algorithm requires a specification of how the data sequences should be partitioned, *i.e.* to choose  $L$  in Equation (4), the previous section.  $L = 7$  gives the smallest prediction errors for the model of order six (auxord= 8 in the Matlab notation [Ljung *et al.*, 1994]). Inspired by the fact that the SISO fan and plate process has a time delay of about 90 ms, see [Gustafsson and Bernhardsson, 1990], we tried models with a delay equal to a couple of samples. (Note that a  $2 \times 2$  process is a special case where it is possible to include different time delays in all signal paths of the process.) It



**Figure 6.** Pole-zero plot for the tenth order model. (Poles are marked with crosses and zeros with rings.)

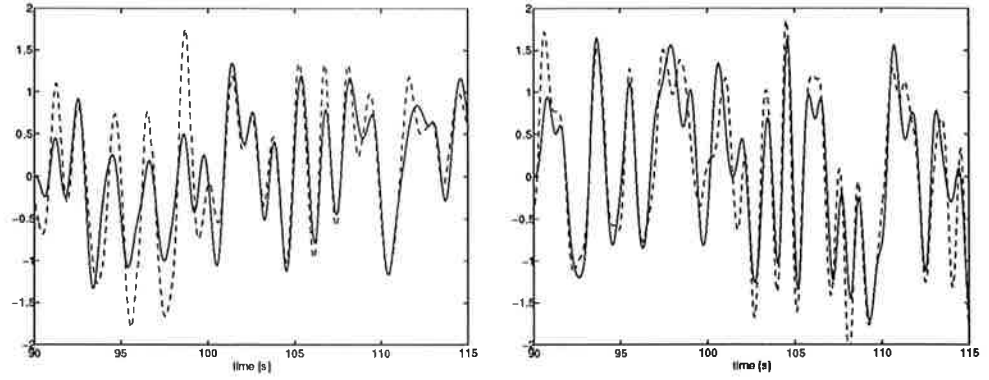
turns out that a delay of two samples, *i. e.* 100 ms, in each path gives a model with smallest prediction errors. This delay was applied by shifting the output data sequences two samples. The state-space representation for the obtained model is given by

$$\begin{aligned}
 \bar{x}_{k+1} &= \begin{pmatrix} 0.9658 & -0.1687 & -0.3530 & -0.0759 & 0.1276 & 0.0091 \\ 0.0579 & 0.9624 & -0.0644 & 0.3917 & -0.2561 & 0.2212 \\ 0.0674 & 0.0323 & 0.9230 & -0.0793 & -0.8099 & 0.2309 \\ 0.0022 & -0.0593 & 0.0464 & 0.9345 & 0.2894 & 0.8673 \\ -0.0019 & 0.0023 & 0.0343 & -0.0298 & 0.9400 & 0.1081 \\ 0.0004 & -0.0020 & -0.0017 & -0.0200 & -0.0115 & 0.9023 \end{pmatrix} \bar{x}_k \\
 &+ \begin{pmatrix} -0.2210 & 0.3754 \\ 0.2158 & 0.1103 \\ 0.0863 & -0.3010 \\ 0.1301 & 0.0799 \\ -0.0182 & 0.0671 \\ 0.0360 & 0.0055 \end{pmatrix} u_k + \begin{pmatrix} -2.8915 & 2.9318 \\ 3.7815 & 1.6690 \\ -0.5088 & -1.0606 \\ 1.1636 & 0.0473 \\ -0.0308 & -0.0258 \\ 0.0520 & -0.0312 \end{pmatrix} e_k \\
 y_{k+2} &= \begin{pmatrix} -0.2052 & 0.1779 & -0.3332 & -0.4676 & -0.0782 & 0.5694 \\ 0.2619 & 0.3373 & 0.3919 & -0.2745 & 0.5465 & 0.0437 \end{pmatrix} \bar{x}_k \\
 &+ \begin{pmatrix} 0.0059 & 0.0002 \\ -0.0005 & 0.0026 \end{pmatrix} u_k + e_k
 \end{aligned}$$

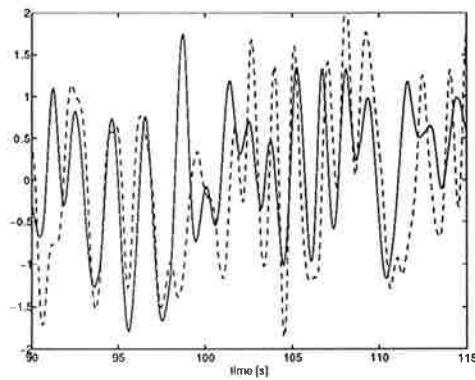
where the covariance matrix of the innovations  $e$  is equal to the identity matrix. We rewrite the model into standard form without the time delay in the subscript of  $y$ . This gives the model

$$\begin{aligned}
 x_{k+1} &= \begin{pmatrix} \bar{A} & 0 & 0 \\ \bar{C} & 0 & 0 \\ 0 & I & 0 \end{pmatrix} x_k + \begin{pmatrix} \bar{B} \\ \bar{D} \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \bar{K} \\ I \\ 0 \end{pmatrix} e_k \\
 y_k &= \begin{pmatrix} 0 & 0 & I \end{pmatrix} x_k
 \end{aligned}$$

where  $\bar{A}$ ,  $\bar{B}$ ,  $\bar{C}$ ,  $\bar{D}$ , and  $\bar{K}$  are the matrices in the previous state-space model. The model has ten poles (two complex conjugated pairs, four in origin, and two other real poles) and six transmission zeros. They are plotted in Figure 6 (except for a transmission zero in  $-10.8661$ ). If we convert the non-zero poles to continuous time, the complex conjugated pairs corresponds to the eigen-



**Figure 7.** Simulated (solid) and measured (dashed) outputs for the tenth order model.  $y_1$  is in the left graph and  $y_2$  in the right.



**Figure 8.** The measured outputs  $y_1$  (solid) and  $-y_2$  (dashed).

frequencies and the dampings

$$\omega_1 = 5.7825 \quad \zeta_1 = 0.0812$$

and

$$\omega_2 = 4.2398 \quad \zeta_2 = 0.1253$$

respectively. The real poles corresponds to the time constants

$$T_1 = 0.5673 \quad T_2 = 0.7396$$

These values seem reasonable compared to the visual estimations of the process parameters.

A comparison between noise free simulated outputs (solid lines) and measured outputs (dashed lines) are shown in Figure 7. The signals agree quite good. We notice that during the time interval 95 – 100 s the simulated  $y_1$  signal is poor. In Figure 8  $y_1$  and  $-y_2$  are plotted. It is interesting to see that during the poor interval the output oscillations are almost in phase. The same correlation appears for other time interval as well. Thus, it seems like the model does not cover all possible “modes” of the process.

To get further physical insight to the model we rewrite it into a continuous transfer function matrix (TFM) form. The TFM from  $\begin{pmatrix} u_1 & u_2 \end{pmatrix}^T$  to

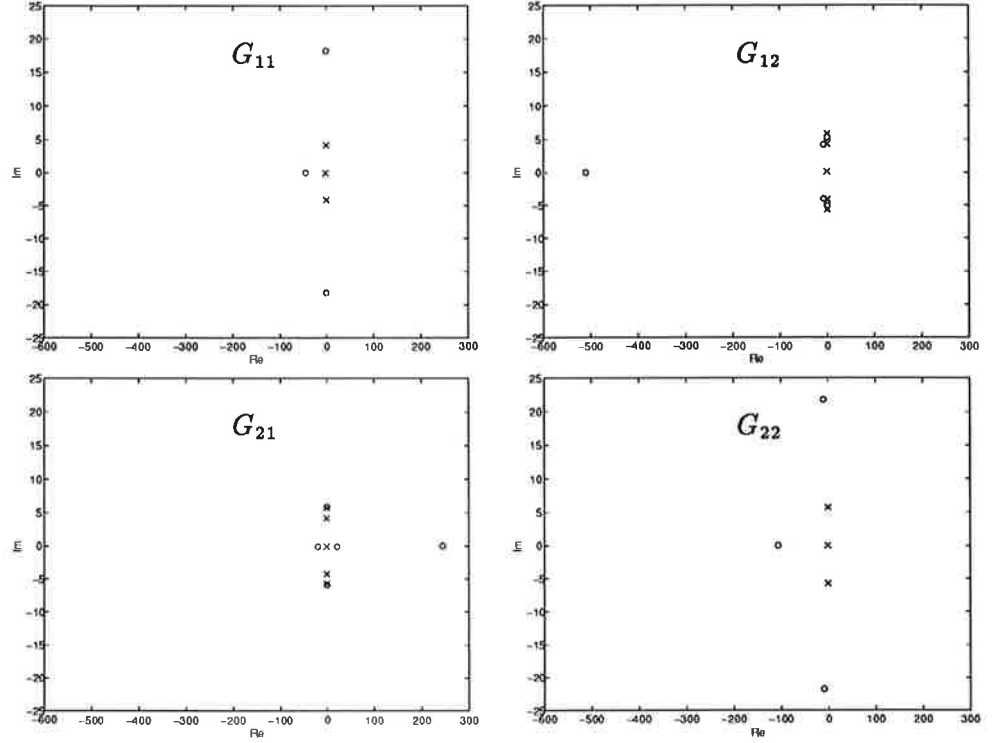


Figure 9. Pole-zero plots for the elements in the transfer function matrix  $G$ .

$\begin{pmatrix} y_1 & y_2 \end{pmatrix}^T$  will be

$$\frac{e^{-0.1s} \cdot 10^{-4}}{d(s)} \begin{pmatrix} n_{11}(s) & n_{12}(s) \\ n_{21}(s) & n_{22}(s) \end{pmatrix}$$

where

$$\begin{aligned} n_{11}(s) &= 59 \cdot (s^2 + 3.5s + 370)(s^2 + 0.93s + 34)(s + 1.3)(s + 40) \\ n_{12}(s) &= 2.4 \cdot (s^2 + 17s + 90)(s^2 + 2.2s + 28)(s + 1.9)(s + 501) \\ n_{21}(s) &= -5.0 \cdot (s^2 + 0.11s + 35)(s + 1.7)(s + 21)(s - 21)(s - 236) \\ n_{22}(s) &= 26 \cdot (s^2 + 11s + 339)(s^2 + 1.2s + 18)(s + 1.8)(s + 177) \\ d(s) &= (s^2 + 0.94s + 33)(s^2 + 1.1s + 18)(s + 1.8)(s + 1.4) \end{aligned}$$

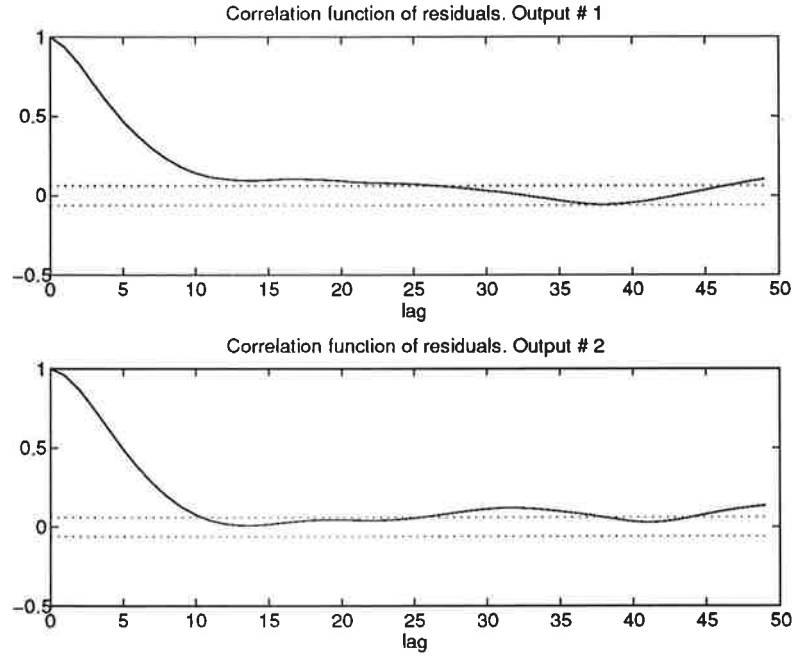
A balanced realization of each element of the TFM above gives

$$G(s) := e^{-0.1s} \cdot 10^{-4}$$

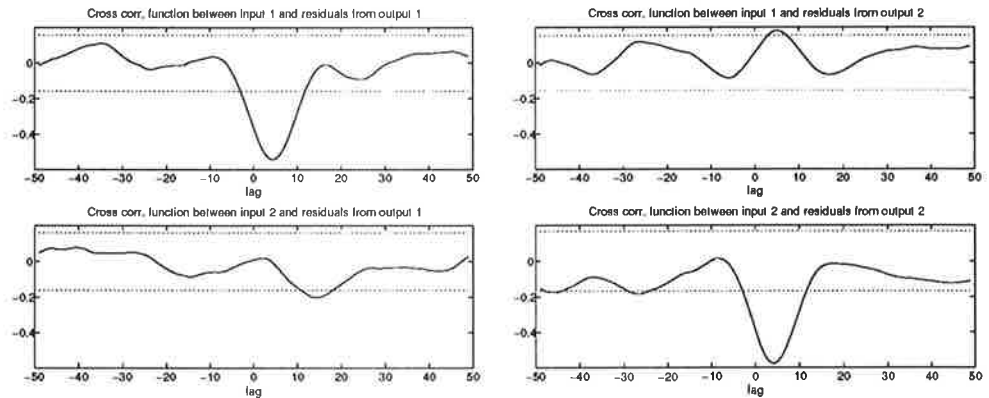
$$\times \begin{pmatrix} \frac{59 \cdot (s^2 + 2.8s + 335)(s + 45)}{(s^2 + 1.1s + 18)(s + 1.8)} & \frac{2.4 \cdot (s^2 + 17s + 89)(s^2 + 2.1s + 28)(s + 510)}{(s^2 + 0.93s + 33)(s^2 + 1.1s + 18)(s + 1.3)} \\ \frac{-5.0 \cdot (s^2 + 0.11s + 35)(s + 20)(s - 21)(s - 243)}{(s^2 + 0.94s + 33)(s^2 + 1.1s + 18)(s + 1.4)} & \frac{26 \cdot (s^2 + 19s + 564)(s + 106)}{(s^2 + 0.93s + 33)(s + 1.4)} \end{pmatrix}$$

Figure 5 shows the pole-zero plots for the transfer functions  $G_{ij}$  in  $G$ .

The TFM above reveals some further insight to the process. The diagonal elements have dominating gains. We notice that the diagonal elements also capture the dynamics of the SISO fan and plate process [Gustafsson and Bernhardsson, 1990]; they have each one pair of oscillative poles and one motor



**Figure 10.** The correlation functions for the residuals of  $y_1$  and  $y_2$ .



**Figure 11.** The cross correlation functions between the residuals and  $u_1$  and  $u_2$ .

time constant. We see that their zeros are high-frequent, and thus can be neglected. Further the anti-diagonal elements both include the oscillative modes of the diagonal elements. A puzzling thing is that the steady-state gain of  $G_{21}$  is negative, whereas ditto of  $G_{12}$  is positive. By considering the mechanics of the process (see Figure 1), one realize that they should both be negative. We recall that the coherence spectra told us that the  $u_2 \rightarrow y_1$  coupling was the weakest. This indicates that the considered sign might not be reliable. Further, the steady-state gain of  $G_{12}$  is very small, *e. g.*  $G_{12}(0) \approx 0.17G_{21}(0)$ .

The correlation functions for the residuals (the measured outputs compared to the one-step ahead prediction) are plotted in Figure 10, and in Figure 11 are the cross correlation functions between the residuals and the inputs shown. The 99% confidence intervals are displayed as dotted lines. We notice that especially some of the cross correlation curves are outside the confidence interval a considerable amount of time. We think that the oscillations of the

correlation functions come from that the subspace model does not take care of all the innovation dynamics, cf. ARMAX versus Box-Jenkins models.

## 6. Conclusions

In this report we have described the identification of a double fan and plate process with two inputs and two outputs. A subspace identification algorithm called N4SID that estimates a linear state-space model was used. The identification gave a physically realistic model with ten states. Outputs from the simulated model agreed rather good to measured outputs.

One drawback with the subspace identification method emphasized in our application was that the dynamics of the noise process must be the same as the process dynamics.

An interesting problem to examine concerning subspace identification is to compare ordinary model reduction (by balanced realization) to the model reduction that is implicitly done by truncating singular values in the subspace method. How these are related is (as far as the author knows) an unsolved problem.

## 7. Acknowledgements

The author wants to thank Ulf Jönsson and Bo Bernhardsson for valuable comments and discussions.

## 8. References

- GUSTAFSSON, K. (1990): "logger—a program for data logging." Technical Report TFRT-7457, Dept. of Automatic Control, Lund Institute of Technology, Sweden.
- GUSTAFSSON, K. and B. BERNHARDSSON (1990): "Control design for two lab-processes: the flexible serov, the fan and plate." Technical Report TFRT-7456, Dept. of Automatic Control, Lund Institute of Technology, Sweden.
- JOHANSSON, R. (1993): *System modeling and identification*. Prentice-Hall, Englewood Cliffs, NJ.
- LJUNG, L. (1987): *System identification — theory for the user*. Prentice-Hall, Englewood Cliffs, NJ.
- LJUNG, L., M. VIBERG, and T. MCKELVEY (1994): *System identification toolbox*. the Mathworks Inc. Unreleased version including N4SID.
- VAN DER VEEN, A.-J., E. F. DEPRETTERE, and A. L. SWINDLEHURST (1993): "Subspace-based signal analysis using singular value decomposition." *Proceedings of the IEEE*, **81:9**, pp. 1277–1308.
- VAN OVERSCHEE, P. and B. DE MOOR (1994): "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems." *Automatica*, **30:1**, pp. 75–93.
- VERHAEGEN, M. and P. DEWILDE (1992): "Subspace model identification — Part 1. the output-error state-space model identification class of algorithms." *Int. J. Control*, **56:5**, pp. 1187–1210.

# Identification of the *ball-and-beam* process

A project in System Identification of Automatic Control

Jeroen Borgers, Martin Nyquist and Ola Wall

April 26, 1994

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>A model of the process</b>	<b>3</b>
2.1	Angle of the beam . . . . .	3
2.2	Position of the ball . . . . .	3
<b>3</b>	<b>The experiment</b>	<b>5</b>
3.1	Setup . . . . .	5
3.2	Scaling of the differential equation . . . . .	5
3.3	Problems . . . . .	5
<b>4</b>	<b>Identification of the angle of the beam</b>	<b>7</b>
4.1	Using a frequency analyser . . . . .	7
4.2	Using an ARMAX-model . . . . .	8
4.3	Conclusions . . . . .	9
<b>5</b>	<b>Identification of the position of the ball</b>	<b>10</b>
5.1	Identification the open position process using an ARMAX model	11
5.1.1	Concluding remarks . . . . .	12
5.2	Identification of the closed loop . . . . .	14
5.2.1	Concluding remarks . . . . .	16
5.3	Identification by filtering . . . . .	16
5.3.1	Concluding remarks . . . . .	17
5.4	Conclusions . . . . .	18
<b>6</b>	<b>Final conclusions</b>	<b>19</b>
<b>A</b>	<b>Least squares method in <i>Matlab</i> for the angle process</b>	<b>20</b>
A.1	Continuous time, using a frequency analyser . . . . .	20
A.2	Discrete time, using an ARMAX model . . . . .	20
<b>B</b>	<b>The <i>Simnon</i> code for the regulators</b>	<b>22</b>



# 1 Introduction

In this project of System Identification, the *ball-and-beam* process is modelled and identified.

The process consists of a tilting beam driven by a motor, and a ball which is rolling over a metal rail on the beam.

The input signal of the beam is a voltage ( $u$ ) which results in an angular velocity proportional to the voltage. The output signals are two voltages, the angle of the beam with the horizontal plane ( $\phi$ ) and the position of the ball on the beam ( $x$ ).

At first, a physical model of the process is developed in the next section. Secondly, the two processes, position of the ball and angle of the beam are identified. To be able to identify these, the system needs to be controlled. This is done with a P-controller for the inner-loop with the *beam* and a PID-controller for the outer-loop with the *ball*. After identification, the chosen model is validated in different ways. The final section contains the conclusions on this project.

## 2 A model of the process

A model of the process can be divided in a model of the angle-process,  $G_\phi(s)$ , and a model of the position of the ball on the beam,  $G_x(s)$ , which has the angle as input. The complete transfer function from input voltage ( $u$ ) to output voltage, which indicates position ( $x$ ), then becomes  $G_\phi(s)G_x(s)$ . A block diagram of this system is shown in figure 1.

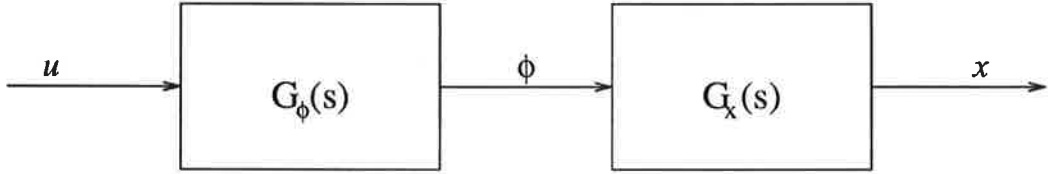


Figure 1: Block diagram of the system composed of the angle and position process.

### 2.1 Angle of the beam

In the process, a motor is used to move the beam. This motor is internally controlled, which results in an angular velocity proportional to the input voltage,

$$\dot{\phi}(t) = k_u u(t). \quad (1)$$

So we get

$$G_\phi(s) = \frac{k_u}{s}. \quad (2)$$

We need to identify the constant  $k_u$ .

### 2.2 Position of the ball

The dynamics for the position of the ball can be determined with the use of classical mechanics. In the figure below, one can see the used expressions and coordinates.

The acting force is the gravity force, which yields  $F_g = mg$ . A reaction force is the normal force ( $F_N$ ). The resultant force ( $F_R$ ) causes acceleration. The crucial point is that there are two kinds of acceleration of the ball: linear and angular. One part of the  $F_R$  together with the  $F_\theta$ , which is a reaction force because of 'no slip' of the ball, causes angular acceleration and the rest of  $F_R$  causes linear acceleration. So if we denote the forces concerning respectively linear and angular acceleration by  $F_a$  and  $F_\theta$ , we get

$$F_a + F_\theta = F_R = mg \sin \phi \quad (3)$$

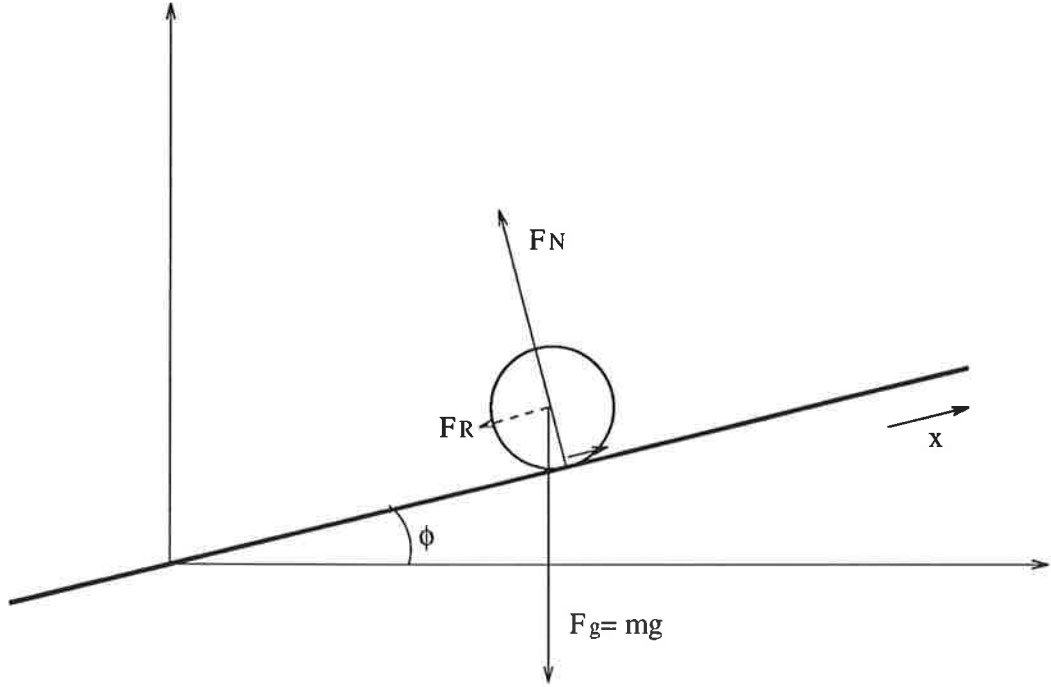


Figure 2: Schematic view of the physical model

The relations for the two forces with their accelerations yield

$$F_a = -m\ddot{x} \quad (4)$$

$$F_\theta = -J\ddot{\theta} \quad (5)$$

in which  $\theta$  is the angle of the ball and  $J$  is the moment of inertia. For a solid ball  $J = \frac{2}{5}mr^2$  and together with the use of the 'no slip' constraint  $r\theta = x$ , we get

$$F_\theta = -\frac{2}{5}m\ddot{x}. \quad (6)$$

This combined with eq.(3) and eq.(4) gives

$$mg \sin \phi = -m\ddot{x} - \frac{2}{5}m\ddot{x} \quad (7)$$

If we consider small angles with  $\sin \phi \approx \phi$ , we get

$$G_x(s) = \frac{x}{\phi} \approx -\frac{5g}{7s^2} \approx -\frac{7}{s^2}. \quad (8)$$

### 3 The experiment

#### 3.1 Setup

The experimental setup consists of the beam and motor object, on a table on wheels, which is to be connected to a computer to control the system. The input signal of the beam is ( $u$ ) which results in an angular velocity proportional to the voltage. The output signals of the beam represent the angle of the beam ( $\phi$ ) and the position of the ball ( $x$ ). Both  $\phi$  and  $x$  can be fed into the computer to compute a control signal  $u$ , which is the input for the angle process.

For identification of the closed loop system, another computer is used to measure the response of the controlled system.

For determination of the continuous time response of the angle process, a frequency analyser is used.

The controllers are implemented with the software package *Simnon*. Creating a PRBS-signal and sampling of the process output are handled by *Logger* and in a later phase also *Simnon* was used for sampling. The computations concerning model approximation and identification are performed in *Matlab* with the identification toolbox.

#### 3.2 Scaling of the differential equation

When determining the differential equation in section 2, it is assumed that all the units are SI units. However, this is not the case. The length of the beam is 1.10m which corresponds to an output signal between  $\pm 1$  V. In the same way for the angle  $1.00V \stackrel{\wedge}{=} \frac{\pi}{4}$  rad. This means that the differential equation 8,

$$\ddot{x} = -\frac{5g}{7}\phi \quad (9)$$

transforms into

$$\ddot{x} = -\frac{5g}{7} \frac{\pi}{4} \frac{1}{0.55} \phi \approx -10.00\phi \quad (10)$$

It is also possible to multiply the position input channel with -1 which we applied in the design. This leads to the transformed transfer function

$$G_x(s) = \frac{10.0}{s^2} \quad (11)$$

which should match with experimental data, instead of equation 8.

#### 3.3 Problems

From a laboratory session one is used to that everything is nicely set up, and all you have to do is to run the experiment. However, this is not the case when

you are working with a project. You don't really know how the programs you use work, and how to connect the equipment together. Actually, this has been one of the hardest things to straight out during the project.

The first problem we ran into was an internally broken connection within the computer, which was discovered due to strange data. Secondly, there were problems with the *Simnon* software, which didn't behave the way we expected.

The stored data wasn't logged with equally spaced time steps, which meant we had to process the achieved data before we could start the identification. This was discovered due to results which were in some sense contradicting each other. The result was always correct concerning the model order, but the gain seemed to depend of the sampling time. This led us to start all over again with very simple experimental setups. These experiments are not presented in this report. The advantage of these experiments is that it is easy to predict the result, but on the other hand they yield very low precision. The third problem is described in the previous section. The real differential equation has to be transformed due to amplifiers on the process output signals which wasn't obvious when we started the experiments. Maybe one could have found consistent results, but they wouldn't agree with equation 8.

## 4 Identification of the angle of the beam

### 4.1 Using a frequency analyser

A frequency analyser is used to measure the frequency response of the angle of the angle process in continuous time. The correlation method is applied. The experiment is carried out with a frequency response analyser, *Schlumberger Solatron 1250*. The data is logged by a computer program which controls the frequency analyser, and then data is processed in *Matlab*. We measured in the frequency range from 0.5 to 40 Hz., at 59 different frequencies.

In figure 3, one can see the Bode-diagrams of the frequency response. One can see that up to  $f \approx 18$  Hz., the response is of slope -1, and at higher frequencies the slope becomes about -3. There is a small resonance about 15 Hz.

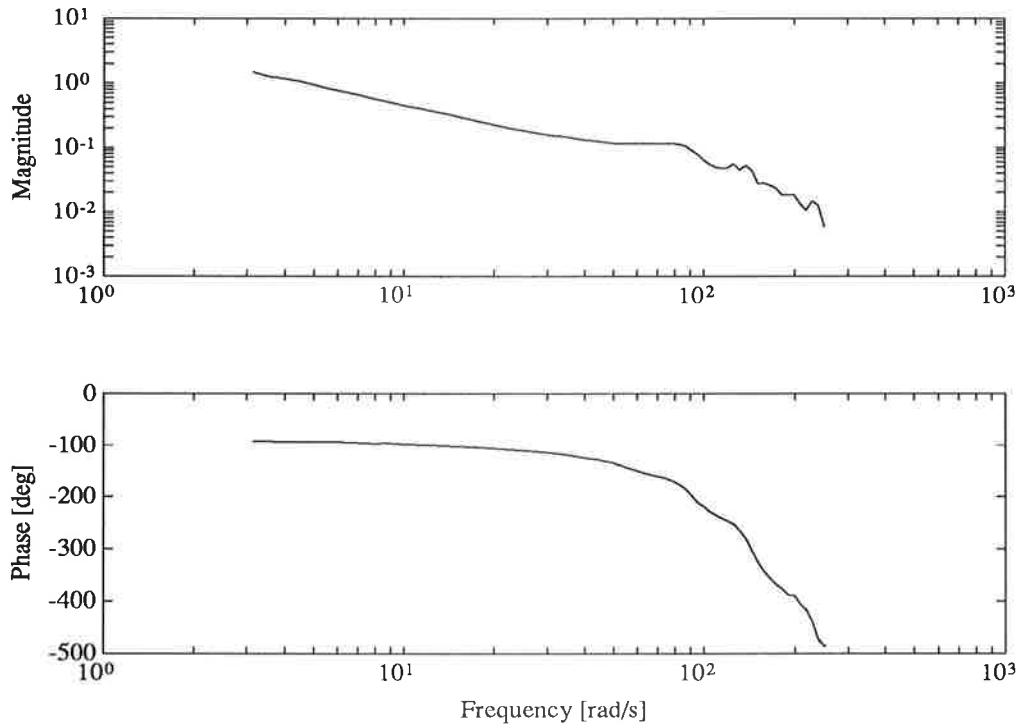


Figure 3: Bode-diagrams of the continuous time process of the angle of the beam

To get the slope and the gain of the system, the logarithm of the data is fitted to a first order polynomial,  $y = ax + {}^{10}\log K_x$ , with the use of the least squares method. The 0th-order term gives the value of the gain and  $a$  denotes the slope. The *Matlab* file to achieve this is found in appendix A.1. The gain is computed to be  $K_x = 4.60$  and the slope is found to be  $a = -0.998$ .

## 4.2 Using an ARMAX-model

To be able to identify the angle process, we used the program *Logger* to create the PRBS-signal as ( $\phi_{ref}$ )-input and to collect the ( $\phi$ )-output data of the system. These measurements are in discrete time. The parameters are set as follows:

```
tsamp    = 15 ms
amp       = 1 V
mean     = 0.034 V
PRBS per = 2 samp
N        = 2000
```

The data is imported in *Matlab* to be processed. The best approximation by an ARMAX-model, turned out to be the th3112-model. In this notation the parameters are respectively  $na$ ,  $nb$ ,  $nc$ , and time delay  $k$ . To choose this model, we took in consideration the loss function, Akaike's FPE (Final Prediction Error).

The model was validated with the correlation of the residuals, which was satisfying. The cross validation is also very good, it is shown in figure 4.

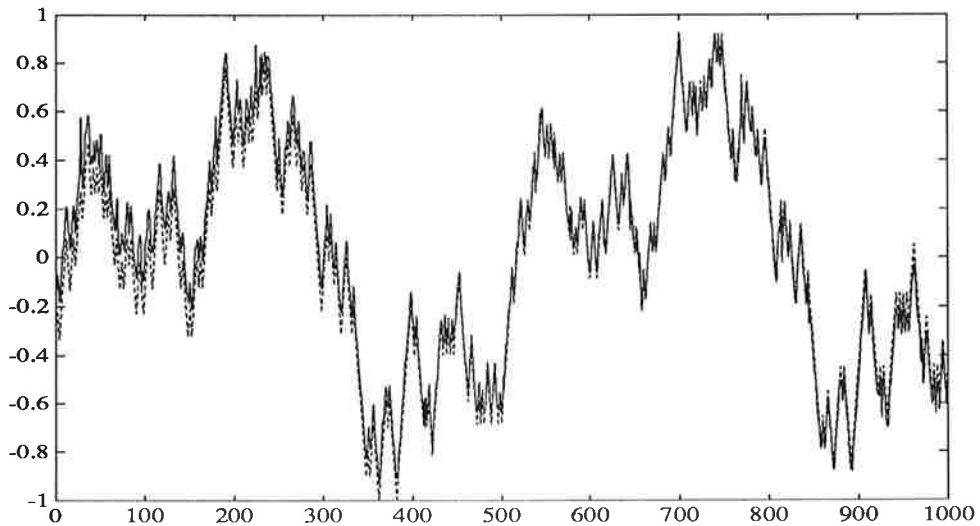


Figure 4: Cross validation of ARMAX-3112 model, 1000 data points used

The bode diagrams for this discrete time ARMAX model are shown in figure 5. One can see that the results of the ARMAX identification is consistent with the results from the correlation method in the previous section.

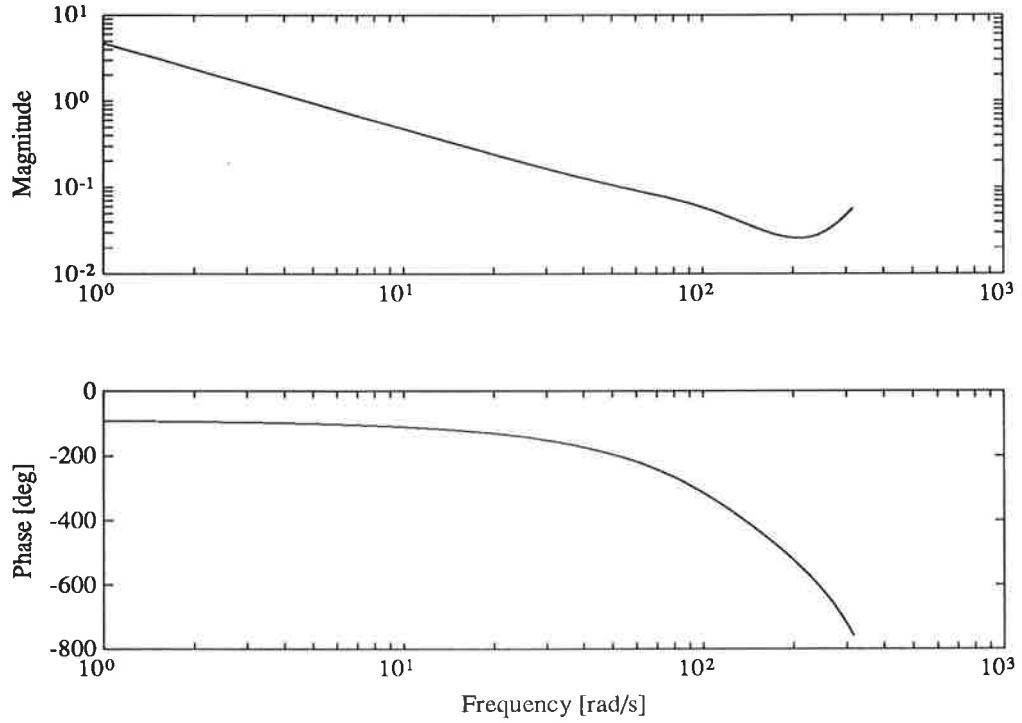


Figure 5: Bode-diagrams of the discrete time ARMAX-3112 model of the angle process,  $G_\phi$

From this data, we have also made a least-squares approximation in *Matlab*, shown in appendix A.2. The gain is computed to 4.48.

The results are also in good accordance with the results in continuous time.

### 4.3 Conclusions

Both methods described above yield the same result. The correlation method though, spans over a larger frequency interval, showing a resonance at 15 Hz. The ARMAX model fails to model the third order dynamics at high frequencies, but this is of no interest since the operation is normally carried out at much lower frequencies. One could safely conclude that the transfer function is

$$G_\phi = \frac{4.5}{s} \quad (12)$$



## 5 Identification of the position of the ball

When making an identification of the position process one has to deal with severe difficulties. The process is unstable and therefore the experiment needs to be carried out in closed loop.

The first step is to design a regulator. We used a cascade regulator strategy with a proportional regulator for the inner loop, and a PID regulator for the outer loop. The complete closed loop system is shown in figure 6.

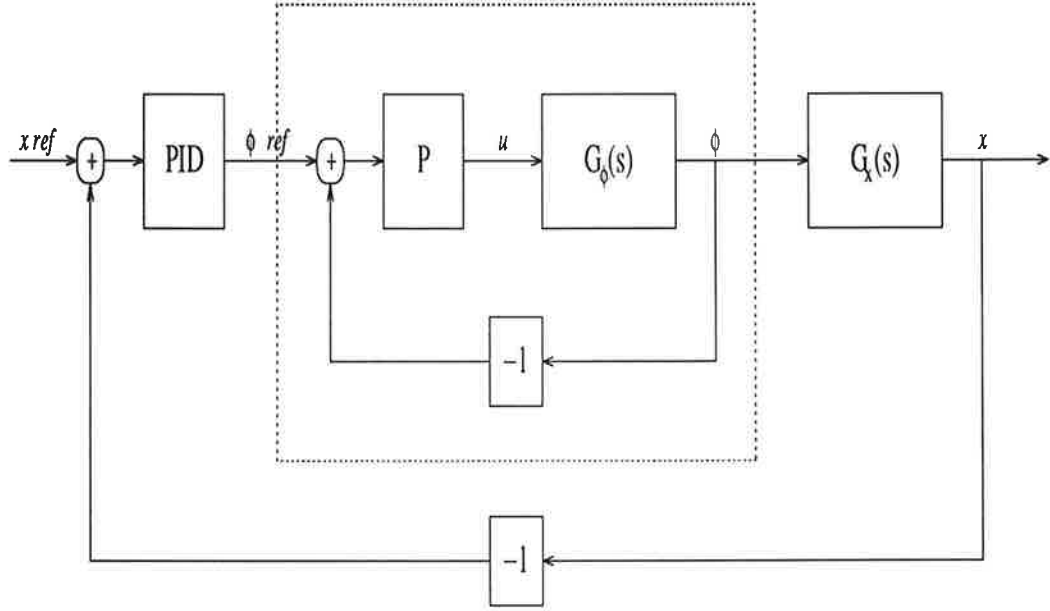


Figure 6: Block diagram of the complete closed loop system.

Parameters used are  $K_p = 1.7$ , with sampling period of  $0.05s$ . for the inner loop, which means the inner loop will be a low pass filter with breakpoint at  $\omega = 8 \text{ rad/s}$ . The PID regulator was tuned with  $K = 0.15$ ,  $T_i = 5$ ,  $T_d = 1.1$ ,  $N = 4$  and sampled with a period of  $0.1s$ . The *Simmon* code for the regulators is found in appendix B. As position reference a PRBS signal was used with the following parameters: *Amplitude* =  $3V$ , *Mean* =  $0V$ ,  $T_{samp} = 250ms$ , *period* =  $4 \text{ samples}$ .

We have chosen three different approaches. The first was to sample the angle ( $\phi$ ) and position ( $x$ ) and apply an ARMAX identification. This gives a model of the transfer function of the position process. Secondly we made an ARMAX model of the closed loop and knowing we are looking for a double integrator, we constructed a model with everything known except for the gain of the position process. The model was simulated with different values of the gain, but no conclusions could be made from the results. Third, we applied the methods of continuous time identification described by Johansson [1].

## 5.1 Identification the open position process using an ARMAX model

When identifying a double integrator, the result is often a low pass filter with low breakpoint. This leads to difficulties in validation of the model. The main instrument of validation will therefore be bode- and pole-zero plots of the achieved transfer functions.

Data used in the experiment was sampled with a rate of  $0.05s$ . Both an ARMAX model with  $na = 2$ ,  $nb = 1$ ,  $nc = 1$  and  $k = 1$  and one with  $na = 3$ ,  $nb = 2$ ,  $nc = 1$  and  $k = 1$  gave good result, concerning Akaike FPE (resp. 0.00015, 0.000013). The bodeplot of the two models looks the same in the interesting frequency interval, therefore it is concluded that the lower order model is sufficient. Fitting the amplitude-curve for lower frequencies to a straight line yields a slope of  $-2.01$  and gain  $K_x = 10.24$ .

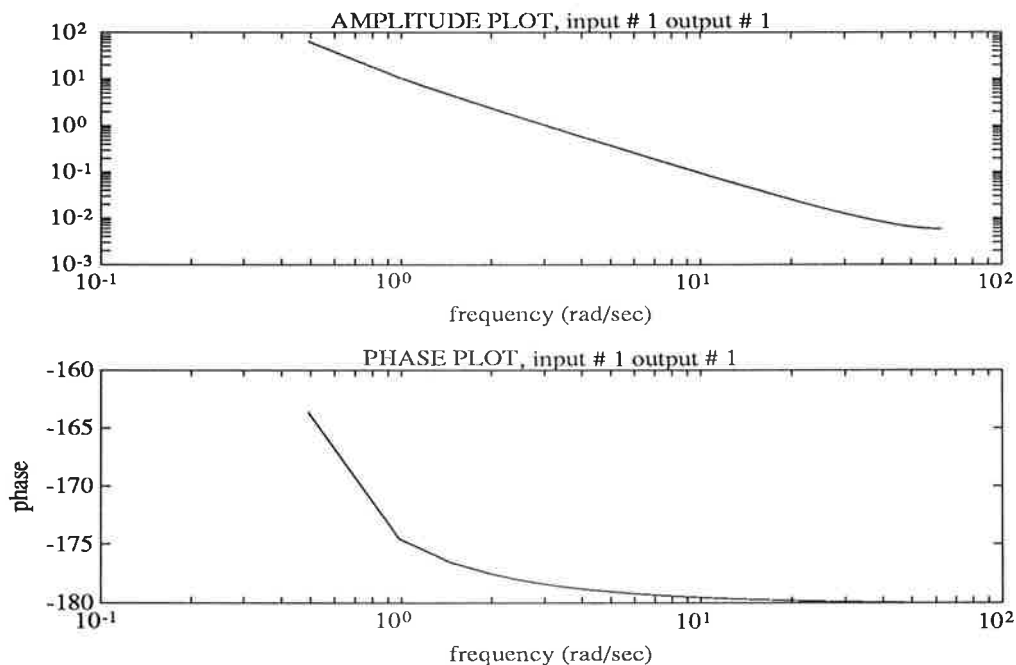


Figure 7: Bode-diagrams of position process,  $t_{samp} = 0.05s$ . ARMAX model of order  $na=2$ ,  $nb=1$ ,  $nc=1$ ,  $k=1$ .

The pole-zero plot also indicates a double integrator with two poles close to 1.

Repeating the same experiment with sampling rate of  $0.25s$ . yields the same model order. In the 3211 model there appear three poles at 1 but one is almost canceled with a zero. The result is shown in figure 9. However, using this sampling rate, the gain is 10.1.

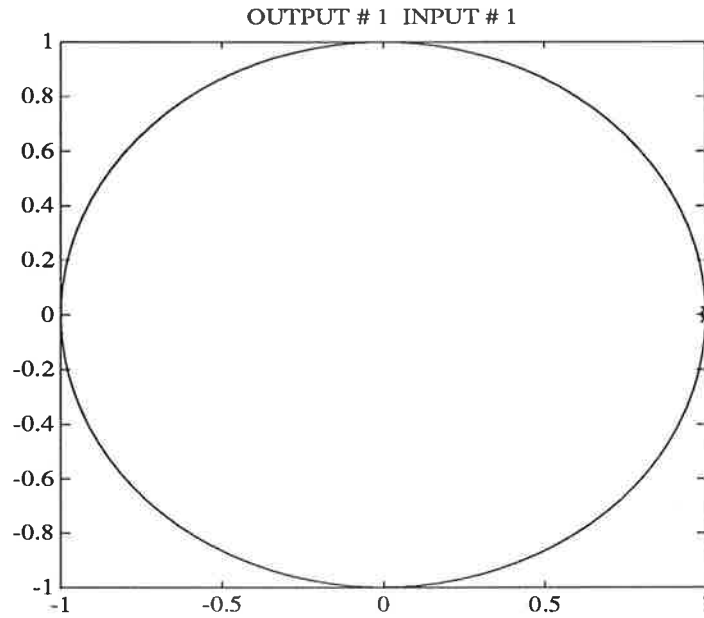


Figure 8: Pole-zero plot of position process,  $t_{sample} = 0.05s$ . ARMAX model of order  $na=2$ ,  $nb=1$ ,  $nc=1$ ,  $k=1$ .

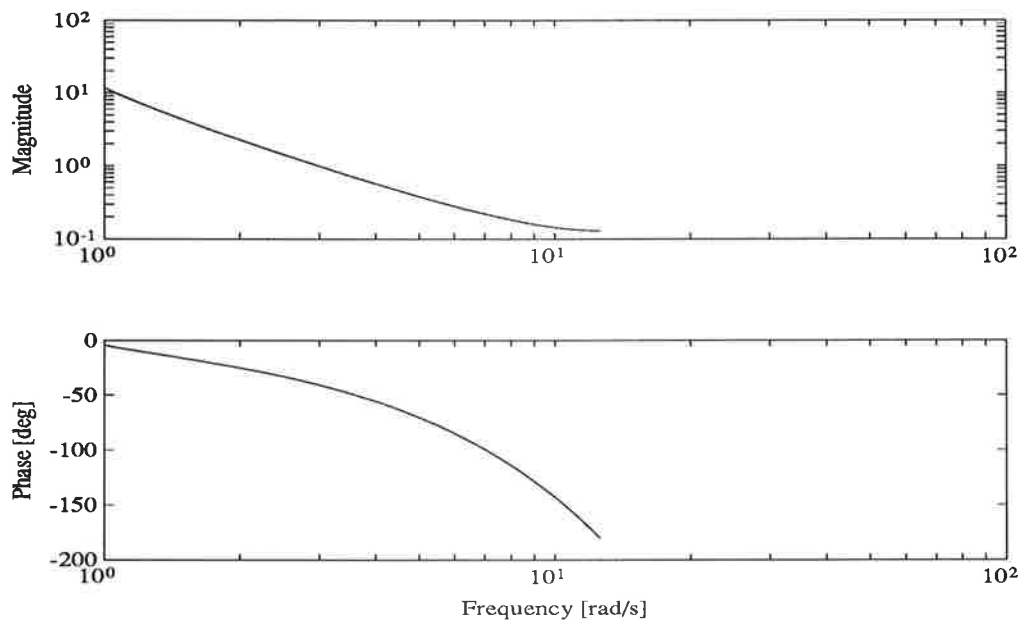


Figure 9: Bode-diagrams of position process,  $t_{sample} = 0.25s$ .

### 5.1.1 Concluding remarks

The assumption of zero order hold sampling is not correct, since the angle isn't stepwise constant. However, when the sampling rate is fast enough one

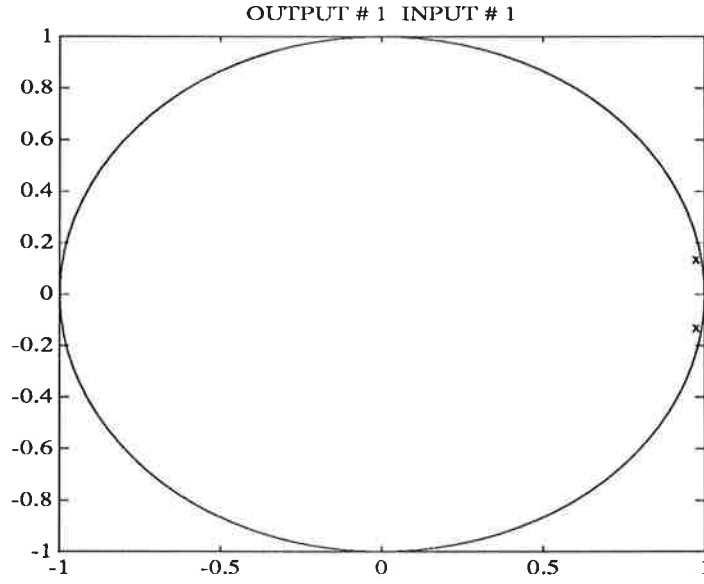


Figure 10: Pole-zero plot of position process,  $t_{samp} = 0.25s$ . The model is an ARMAX model of order  $na=2$ ,  $nb=1$ ,  $nc=1$  and  $k=1$ . The result is a non-ideal double integrator.

could expect an accurate result. Two sampling rates was used in two different experiments, due to the problems describes in section 3.3. After dealing with these problems there is no big difference in the result. The conclusion of the experiment is that the process is a double integrator, and the gain is  $10.2 \pm 0.1$ , which is a very good match with the theoretical model.

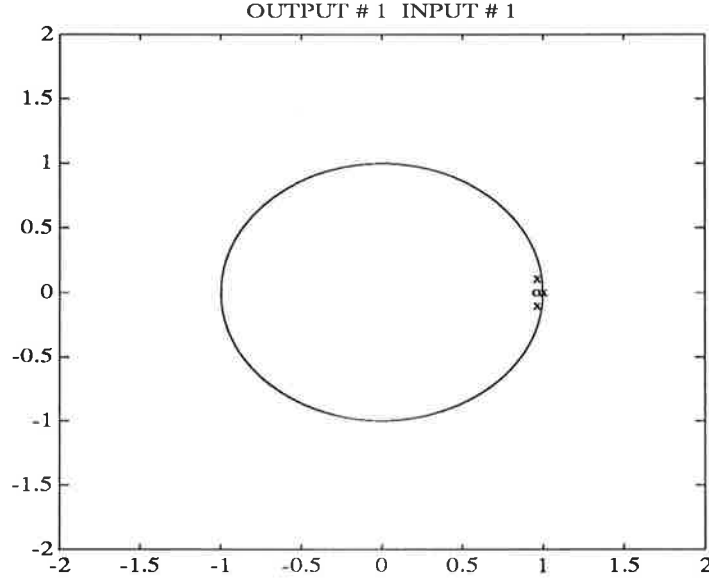


Figure 11: Pole-zero plot of position process,  $t_{samp} = 0.25s$ . This time an ARMAX of order  $na=3$ ,  $nb=2$ ,  $nc=1$  and  $k=1$  was used. One of the poles is cancelled by a zero.

## 5.2 Identification of the closed loop

In the experiment described above, the reference signal and the process output were logged. This data describes the closed loop. Applying the identification toolbox to this data an ARMAX model with  $na = 3$ ,  $nb = 2$ ,  $nc = 2$  and  $k = 1$  yields the best result. The Akaike FPE equals 0.011. A Box-Jenkins model shows no improvement of the statistical properties of the process.

The cross correlation from reference input to output residuals is almost independent, at least for reasonably low delay. However the model could be improved with a much higher model order, but this high order dynamics is of no interest. A bode plot of the closed loop is shown in figure 12.

Assuming the position process is a double integrator a continuous time model of the closed loop is easily achieved. Some algebra will show that the closed loop transfer function is

$$G_{cl} = \frac{G}{1 + G} = \frac{KK_x((T_i s + 1)(\frac{T_d}{N}s + 1) + T_i T_d s^2)}{s^3 T_i (\frac{T_d}{N}s + 1) + KK_x((T_i s + 1)(\frac{T_d}{N}s + 1) + T_i T_d s^2)} \quad (13)$$

In this transfer function  $K = 1$ , i.e. the inner loop is approximated to be a constant, but in reality it is a low pass filter. The breakpoint is  $\omega_s = 8\text{rad/s}$  and therefore, it doesn't effect the amplification in the considered range. This

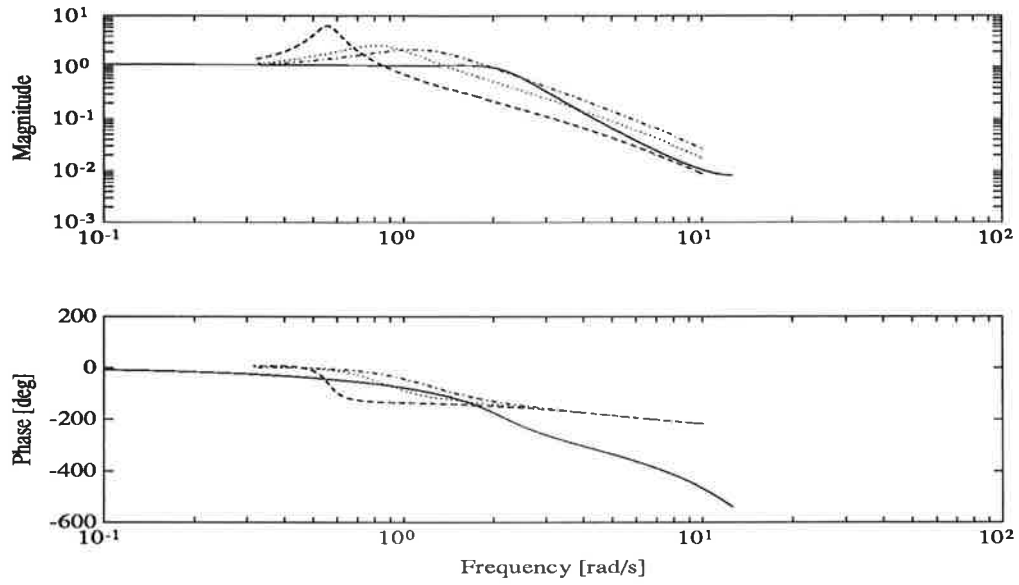


Figure 12: Bode plot of the closed loop transfer function. The dashed, dotted and dash-dotted lines show the bodeplots of the simulated closed loop for  $K_x = \{2 \ 4 \ 8\}$ .

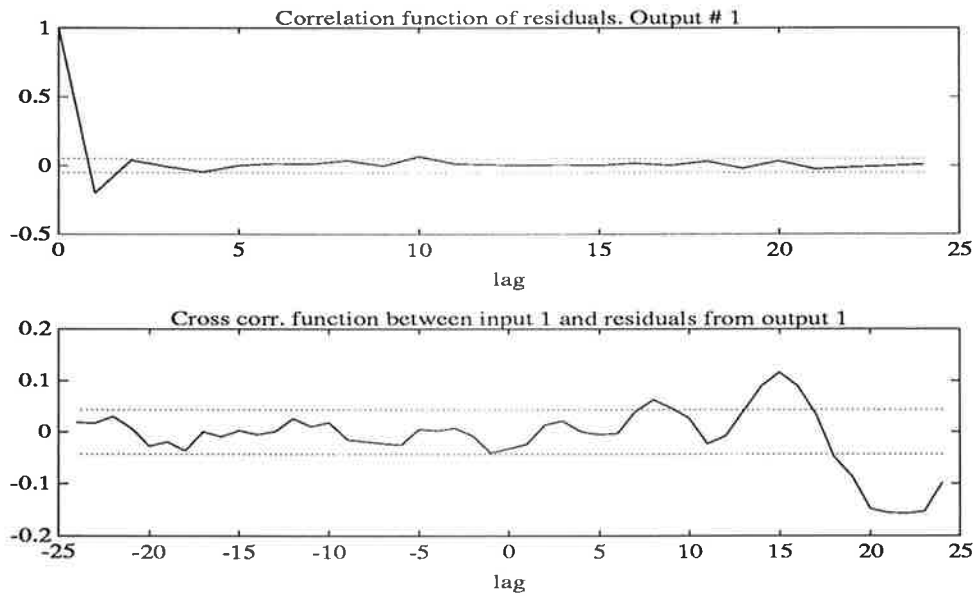


Figure 13: The residuals and the correlation of the model output.

transfer function could be evaluated for different values of the gain  $K_x$ , and is also plotted in the bode diagram for  $K_x = \{2 \ 4 \ 8\}$ .

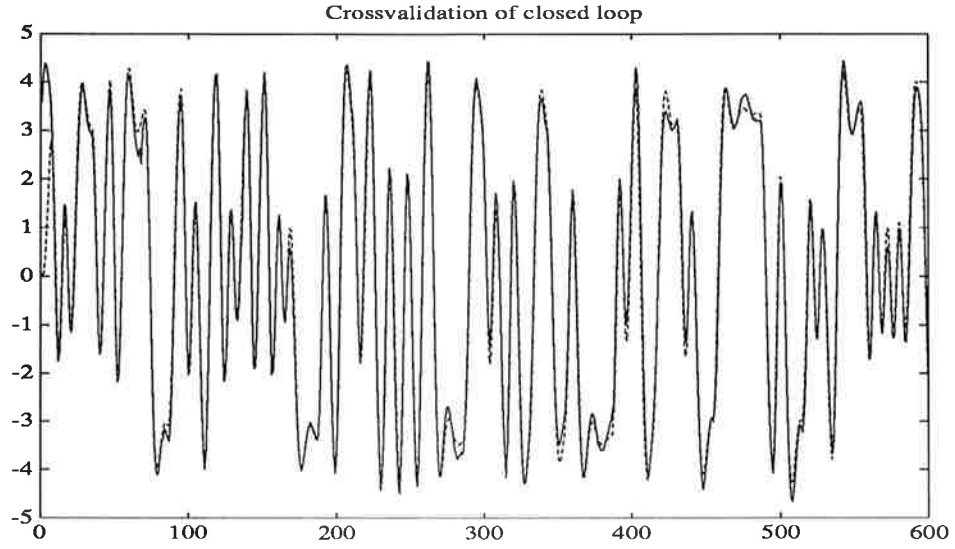


Figure 14: Cross validation between the process output and the model output.

### 5.2.1 Concluding remarks

As seen, no conclusions can be drawn from this experiment. Maybe one could argue that the frequency breakpoint is reasonable, and this result doesn't contradict the result of the other experiments.

## 5.3 Identification by filtering

Our third approach is based on continuous time models described in [1], chapter 12. According to previous experiments we assume the process from angle to position to be a double integrator with unknown gain,  $K_x$ . In order to determine  $K_x$  we introduce a low-pass filter, which in continuous time can be written as

$$\lambda = \frac{1}{1 + s\tau} \quad (14)$$

The relation between the signals will then be

$$x_f = K_x \tau^2 \phi_f \quad (15)$$

where

$$x_f = (1 - \lambda)^2 [x] \quad (16)$$

$$\phi_f = \lambda^2 [\phi] \quad (17)$$

The normal procedure is to use externally connected analog filters, and then sample the filtered signals to get the necessary data for the least squares algorithm. Instead we stored the angle and the position in *Simnon* and then filtered them in *Matlab*. Concerning the value of  $\tau$  we wanted a low pass filter with a not too low breakpoint, because we wanted the magnitude of  $\phi_f$  not too small. By looking at plots of the filtered signals we found that  $\tau=0.5$  gave the best results. Figure 15 shows that the filtered signals are proportional, which is to be expected according to equation 15.

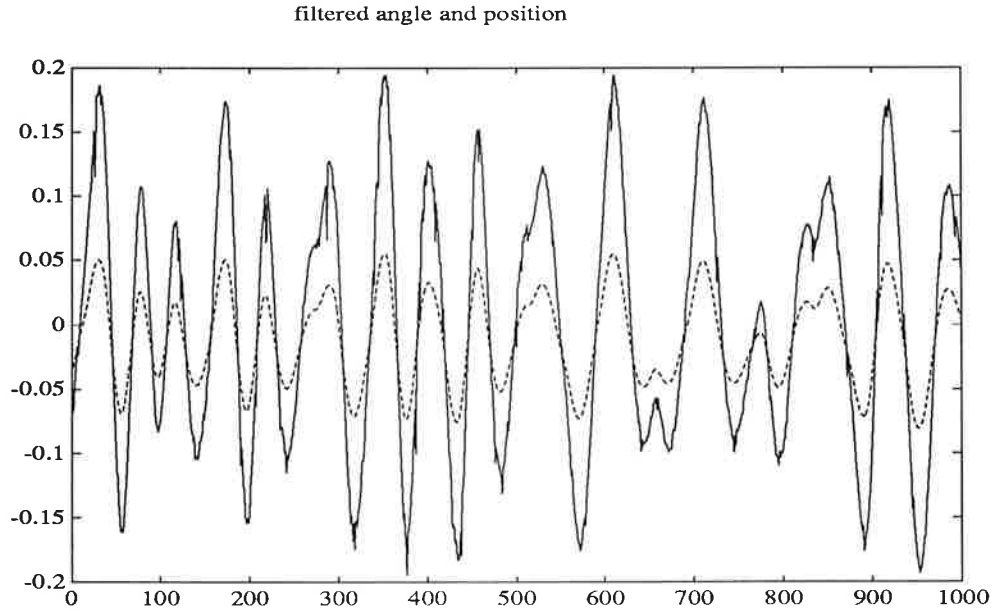


Figure 15: Plot of  $x_f$  and  $\phi_f$ . The dotted curve is the filtered angle signal ( $\phi_f$ ) and the solid curve is the filtered position signal ( $x_f$ ).  $\tau = 0.5$

The sampling interval 0.05s and  $\tau=0.5$  gave the gain  $K_x = 9.79$  using the least squares method. We also tried other values of  $\tau$ , for example  $\tau=0.25$  yielded  $K_x = 11.49$ . Larger values of  $\tau$  resulted in small magnitudes of  $\phi_f$ .

### 5.3.1 Concluding remarks

This method applied as above is sensitive for different choices of  $\tau$  and therefore the accuracy isn't very good. From earlier experiments we also know that the model used above is good at lower frequencies, but since we have used the high pass filtered position in this identification, the result may not be totally correct. Our conclusion is that the gain could be somewhere around 10. Thus this does not contradict the identification by an ARMAX model, in the previous subsection.



## 5.4 Conclusions

We used three different methods. One of them failed to give any result, but on the other hand, it didn't contradict the other results. Both the ARMAX model and the continuous time method yield the same result concerning model order. The last method was sensitive concerning the filter time constant and therefore the conclusion is that the ARMAX model gives the best result. This method was also independent of sampling rate. The transfer function of the ball process is

$$G_x = \frac{10.1}{s^2} \quad (18)$$

## 6 Final conclusions

To get to the final result, many difficulties have been dealt with. Most of them involved practical aspects in the experimental procedure. The gain of the angle process was unknown when we started, and the theoretical model was supplied by our instructor. The two different experiments described in section 4 yielded the same result both concerning gain and model order. The result can not be verified against any expected (theoretical) values.

The position process was theoretically modeled and then the experiments were carried out. This time we had results which were consistent, and they verified the theoretical model, only differing with a scaling factor. This scaling is due to amplifiers in the chain of signals, and the impact of these amplifiers is discussed in section 3. This means an almost perfect match between the results of the theoretical and practical model.

The result is repeated:

$$G_\phi = \frac{4.5}{s} \quad (19)$$

$$G_x = \frac{-7.1}{s^2} \quad (20)$$

this time with the  $G_x$  rescaled to SI units.

## A Least squares method in *Matlab* for the angle process

### A.1 Continuous time, using a frequency analyser

```
% Matlab-file "temp.m"

x = Gtemp(:,1);
y = Gtemp(:,2);
fi = [ones(1,20)',x];
th = inv (fi'*fi)*fi'*y

th =
    +0.6624
   -0.9980
    % slope should be -1

exp(th(1)*log(10))

ans =
    4.5963

    % statistical determined gain
```

### A.2 Discrete time, using an ARMAX model

```
fr = frd(b,a,h,0,2,200);
x = log10(fr(:,1));
y = log10(abs(fr(:,2)));
fi = [ones(200,1),x];
th = fi \ y

th =

    +0.6513
   -0.9598

    % slope should be -1
```

```
exp(th(1)*log(10))
```

```
ans =  
4.4805
```

```
% gain in ARMAX model
```

## B The *Simnon* code for the regulators

For controlling the position of the ball a regulator was implemented in *Simnon*.

```
DISCRETE SYSTEM preg
INPUT fi firef
OUTPUT u
TIME t
TSAMP ts
u=-Kp*(fi-firef)
ts = t+h
"Parameters
Kp:1.7
h :0.05
end
```

```
DISCRETE SYSTEM pidreg
TIME t
TSAMP ts
INPUT posref pos
OUTPUT firef
STATE i D posold
NEW ni nD nposold

e=posref-pos
nposold=pos
nD=Td/(Td+N*h)*D - Kd*Td/(Td+N*h)*N*(pos-posold)
firef=Kd*e+nD+i
ni = i+Kd*h/Ti*e
ts=t+h

h:0.1
Kd:0.15
Td:1.1
N:4
Ti : 5

END
```

```
CONNECTING SYSTEM beamco
TIME t
posref[pidreg]=ADIN(3,t)
pos[pidreg]=-ADIN(1,t)
firef[preg]=firef[pidreg]
fi[preg]=ADIN(2,t)
AOO=DAOUT(0,u[preg])
a:0.3
w:2
END
```

## References

- [1] R. Johansson, *System modeling & identification* Prentice-Hall, New Jersey, 1993
- [2] K.J. Åström and B. Wittenmark, *Computer Controlled Systems* 2nd ed. Prentice-Hall, New Jersey, 1990

# **Projekt i Processidentifiering VT 1994 ”Ekorrhjulet”**

Fredrik Bistedt E-89  
Björn Pettersson E-89  
Per-Ola Wictor E-89

Handledare: Ulf Jönsson

28 april 1994



## Innehåll

<b>1. Inledning . . . . .</b>	<b>3</b>
<b>2. Modellering . . . . .</b>	<b>4</b>
<b>3. Inledande experiment . . . . .</b>	<b>6</b>
Impulssvarsmetoden . . . . .	6
Stegsvarsmetoden . . . . .	7
Frekvenssvarsanalys . . . . .	7
<b>4. Identifiering . . . . .</b>	<b>9</b>
Datainsamling . . . . .	9
Box-Jenkins modeller . . . . .	10
ARMAX-modeller . . . . .	12
IV-modeller . . . . .	14
<b>5. Kontinuerliga systemet . . . . .</b>	<b>15</b>
<b>6. Sammanfattning . . . . .</b>	<b>16</b>

## 1. Inledning

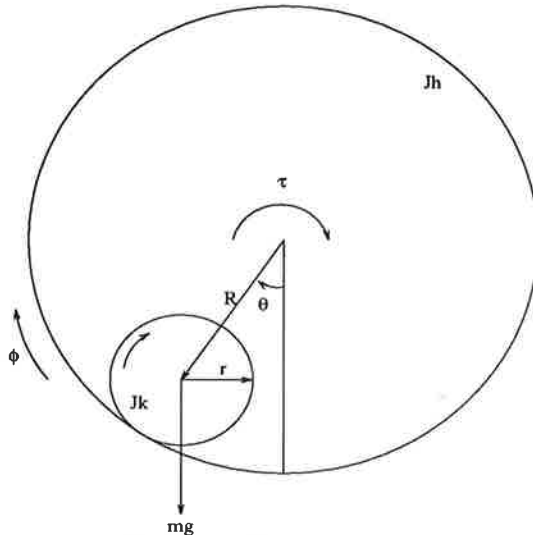
Som ett avslutande moment i kursen Processidentifiering utföres ett mindre projekt. Vi har valt processen som kallas "ekorrhjulet" eller mer internationellt "Ball and hoop". Processen ser ut precis som den kallas, ett ekorrhjul. En likströmsmotor driver ett hjul, ekorrhjulet. Inuti detta hjul finns en lös kula som roterar med hjulet. Systemet har endast en insignal, nämligen spänningen till motorn samt två utsignaler hjulets vinkel relativt nolläget samt kulans läge i hjulet eller rättare: kulans vinkel i förhållande till lodlinjen.

Uppläggning av projektet följer i stort huvudrubrikerna i denna rapport. För att få en liten uppfattning om processens ordning gav vi oss raskt på att försöka sätta upp systemekvationerna. Efter detta följde vi upp med enkel identifiering, steg- och impulssvarsanalys samt frekvensanalys. I identifieringsmomentet har vi ängnat oss åt tre olika metoder, ARMAX, Box-Jenkins och Instrumental Variable (IV). I kapitlet om respektive metod redovisas resultaten från identifieringen. I kapitlet 'Kontinuerliga systemet' försöker vi översätta de diskreta modellerna till kontinuerlig tid. Anledningen till detta är flera; parametrarna har en fysikalisk tolkning, man har en större känsla för poler och nollställen etc.

Identifieringen har utförts i MATLAB där System Identification Toolbox har använts flitigt.

## 2. Modellering

För att få en uppfattning om ordning och utseende av processen matematiskt, härleds systemets differentialekvationer enligt Lagrange's metod. Figur 1 visar den fysikaliska modellen samt beteckningar som kommer att användas i härledningen. Enligt Lagrange's metod gäller det att sätta upp systemets energi-



Figur 1. Processen - fysikalisk modell

ekvationer; den kinetiska energiekvationen  $T$  samt potentiella energiekvationen  $U$ . Lagrangianen  $L$  bildas som

$$L = T - U \quad (1)$$

Euler-Lagrange ekvationerna fås som

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \tau \quad (2)$$

där  $q$  är en generaliserad koordinat och  $\tau$  betecknar externa krafter som verkar på systemet. I fallet med "Ekorrhjulet" är det lämpligt att införa koordinaterna  $\theta$  vilket betecknar armens vinkel relativt lodlinjen samt  $\phi$  som betecknar hjulets position/vinkel i förhållande till ett givet nolläge. För att modelleringen inte ska bli allt för komplicerad bortser vi tillsvidare från all form av friktion. Ytterligare ett antagande görs, nämligen att det råder fullständig koppling mellan hjul och kula, dvs kulan slirar inte.

Kulans rotationshastighet kan uttryckas i koordinaterna  $\phi$  och  $\theta$  enligt

$$\omega_k = \frac{v_{\text{periferi}}}{r} = \frac{(R + r)(\dot{\phi} - \dot{\theta})}{r} \quad (3)$$

där  $R$  är armens längd från centrum till kulans mittpunkt och  $r$  är kulans radie. Energiekvationerna blir, då kulans massa är  $m$ ,

$$U = mg(1 - \cos \theta)R \quad (4)$$

$$T = \frac{1}{2} (mR^2 \dot{\theta}^2 + \frac{J_k (R + r)^2 (\dot{\phi} - \dot{\theta})^2}{r^2} + J_h \dot{\phi}^2) \quad (5)$$

Lagrangianen  $L$  bildas nu enligt (1),

$$L = T - U = \frac{1}{2}(mR^2\dot{\theta}^2 + \frac{J_k(R+r)^2(\dot{\phi} - \dot{\theta})^2}{r^2} + J_h\dot{\phi}^2) - mg(1 - \cos \theta)R \quad (6)$$

$J_k$  och  $J_h$  är tröghetsmomenten för kulan respektive för hjulet. Systemekvationerna kan nu deriveras fram via (6), men innan vi gör detta måste de externa krafterna  $\tau$  utredas. Den enda externa kraft eller snarare moment som verkar på systemet, är motorns alstrade moment. Detta moment kommer in då  $L$  deriveras med avseende på  $\phi$ -koordinaten. Momentet  $\tau$  är proportionellt mot derivatan av pålagd spänning, dvs

$$\tau = K\dot{U} \quad (7)$$

ty motorn är av likspänningstyp. Parametern  $K$  är en motorkonstant. Det är förutsatt att motorns tidskonstant är liten. Systemekvationerna blir (2) enligt följande:

$$mR^2\ddot{\theta} - \frac{J_k(R+r)^2(\ddot{\phi} - \ddot{\theta})}{r^2} - mg\theta = 0 \quad (8)$$

$$\frac{J_k(R+r)^2(\ddot{\phi} - \ddot{\theta})}{r^2} + J_h\ddot{\phi} = K\dot{U} \quad (9)$$

I ekvation (8) har följande approximation gjorts för att ekvationen ska vara linjär,  $\sin \theta \approx \theta$  för små värden på vinkeln  $\theta$ . Genom att lösa detta ekvationsystem, (8) och (9), kan man lösa ut  $\ddot{\theta}$ ,  $\ddot{\phi}$ ,

$$mR^2\ddot{\theta} - mg\theta = K\dot{U} \quad (10)$$

Här har ytterligare en approximation gjorts nämligen att hjulets tröghetsmoment är litet då det är ett aluminiumhjul, dvs termen  $J_h\ddot{\phi}$  kan försummas. Överföringsfunktionen  $G(s)$  från  $U$  till  $\theta$  har följande utseende:

$$G(s) = \frac{bs}{s^2 + a_2} \quad (11)$$

Som synes erhålles en överföringsfunktion av ordning 2 men utan några  $s$ -termer. Detta beror på att vi antog att friktionen var noll vid modelleringen. Ansätter man att det finns friktion som viskös dämpning kommer det in en  $s$ -term i nämnaren i  $G(s)$ , dvs

$$G(s) = \frac{bs}{s^2 + a_1s + a_2} \quad (12)$$

eller enligt gängse modell med dämpning,  $\zeta$ , och resonansfrekvens,  $\omega$ .

$$G(s) = \frac{bs}{s^2 + 2\zeta\omega_0s + \omega_0^2} \quad (13)$$

Systemet som ska identifieras är av ordning 2 med nollställe i origo, dvs stationär förstärkning lika med noll.

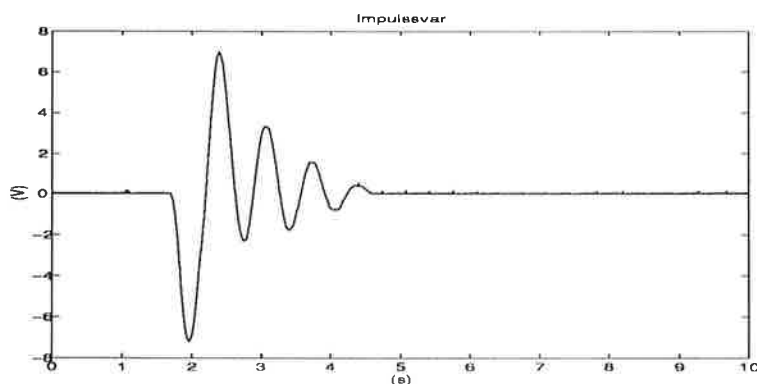
### 3. Inledande experiment

När vi väl hade modellerat färdigt, och således fått en ganska klar uppfattning om hur pendeln skulle uppföra sig, var det så dags för att med första stapplande steg göra initiala experiment för att i viss mån skönja samstämmigheten mellan teoretiskt system och detsamma i praktiken. Det vi ville se var alltså ett system som hade likspänningsförstärkningen noll - ty systemets nollställe hamnade i origo, ett impulssvar som klingar av svängande runt sin ännu okända egenfrekvens, ett stegsvar som gav översläng enligt andra ordningens system med en dämpning som fortfarande var okänd. Till vår hjälp tog vi impulssvarsmetoden, stegvarsmetoden och frekvensvarsanalys - alla med sin speciella uppgift.

#### Impulssvarsmetoden

I de två inledande experimenten fick vi använda oss utav programmet Logger för att samla mätdata. För att i bästa möjliga mån kunna efterlikna ett impulssvar satte vi själva igång ekorrhjulet med en spikartad snärt. (Detta gick ej att få fram med Logger, vilket gjorde att vi ej kunde skatta tidsfördröjningen i systemet). Vi lyckades efter ett par försök få till ett riktigt bra impulssvar mycket tack vare en hyfsat snäll process. De problem som man kan få vid impulssvarsanalys med synkronisering mellan impuls och sampling undvek vi genom att sampla snabbt. Känslighet för brus är en annan men vi hade lite brus enligt vad vi kunde se på svaren. I normala fall har man problem med mättnig av systemet men eftersom vi inte hade någon dirac som insignal och använde impulssvaret för att skatta dämpning och resonansfrekvensen så blev inte det något problem.

Vilka parametrar kan man utläsa ur experimentet? Jo, dämpningen  $\zeta$  och pendelns resonansfrekvens,  $f_0$ .



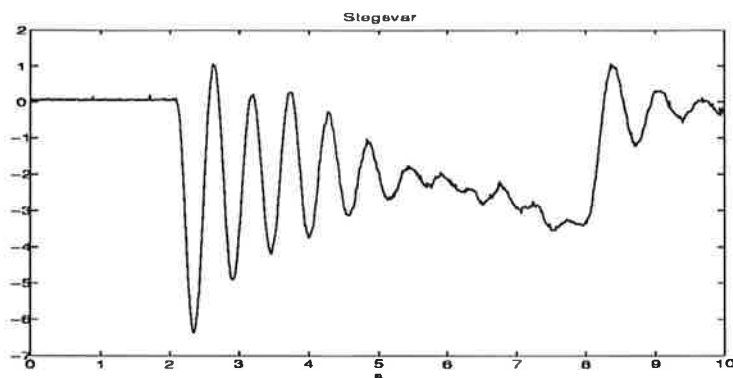
Figur 2. Impulssvar

För envelopen hos impulssvaret gäller formeln:

$$y(t) = ke^{-\zeta\omega t} \quad (14)$$

För att eliminera konstanten togs ur diagrammet två funktionsvärden för tidpunkterna  $t_1$  och  $t_2$ . Dessa dividerades enligt:

$$\frac{y(t_1)}{y(t_2)} = \frac{ke^{-\zeta\omega t_1}}{ke^{-\zeta\omega t_2}} \quad (15)$$



Figur 3. Stegsvär

vilket gav med  $t_1 = 1.9$  och  $t_2 = 2.8$  att systemets dämpning  $\zeta = 0.14$ . Ytterligare en parameter erhöles mha impulssvarsmetoden, nämligen resonansfrekvensen,  $f_0$ , vilken vi fick genom att uppskatta de två första periodernas negativa toppvärden. Dessa ligger mellan 0.4 och 1.7 s. Med en periodlängd,  $T$  på 0.65 s blir därför  $f_0 = 1.57$  Hz. Processens impulssvar återfinns i figur 2.

### Stegvarsmetoden

Med stegvarsmetoden kan man både få fram resonansfrekvens och stationär förstärkning. Vad vi intresserade oss för var stationära förstärkningen,  $G(0)$ . Låt

$$\lim_{s \rightarrow 0} G(s) = \int_0^{\infty} g(\tau) e^{-s\tau} d\tau, \quad (16)$$

vilket erhålles om systemet utsätts för en konstant signal, dvs ett steg. I vårt fall, då friktion och tröghetsmoment övervunnits och kulans periferihastighet uppnått ekorrhjulets, lade sig åter pendeln i lodrät läge. Processens stegsvar återfinns i figur 3. Vi konstaterade att det behövs en acceleration av hjulet för att kulan ska förflytta sig. Anledningen till att det ser konstigt ut är att kulans rörelse hindras av pendeln. Vi tror att pendelbygeln klämmer kulan i vissa lägen.

### Frekvensvarsanalys

Frekvensvarsanalys är en metod för skattning av systemets frekvensfunktion  $G(i\omega)$ .

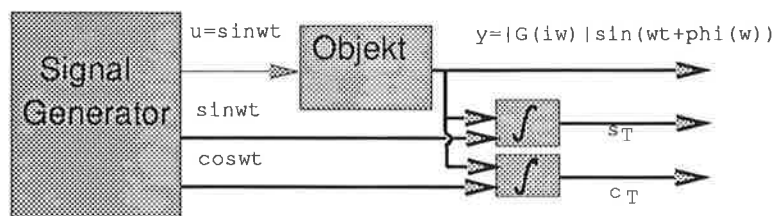
$$u(t) = u_1 \sin \omega t \quad (17)$$

Utsignalen från objektet, då det matats med en sinussignal  $y(t)$  karaktäriseras av amplituden  $|G(i\omega)|$  samt fasen  $\phi(\omega)$  enligt

$$y(t) = |G(i\omega)| u_1 \sin(\omega t + \phi(\omega)) \quad (18)$$

För att eliminera brusets inverkan vid mätningen använder man ofta en korrelationsmetod, se fig 4. Objektets utsignal multipliceras med en sinussignal och en cosinussignal. De multiplicerade signalerna integreras över tiden  $T$ , vilket ger en brusreduktion. Tiden  $T$  väljs som ett helt antal perioder  $k$  hos testsignalen (17).

Sinuskanalen ger



Figur 4. Blockschemata för frekvensvarsanalys

$$s_T(\omega) = \int_0^T y(t) \sin \omega t dt = \frac{1}{2} T |G(i\omega)| u_1 \cos \phi(\omega) \quad (19)$$

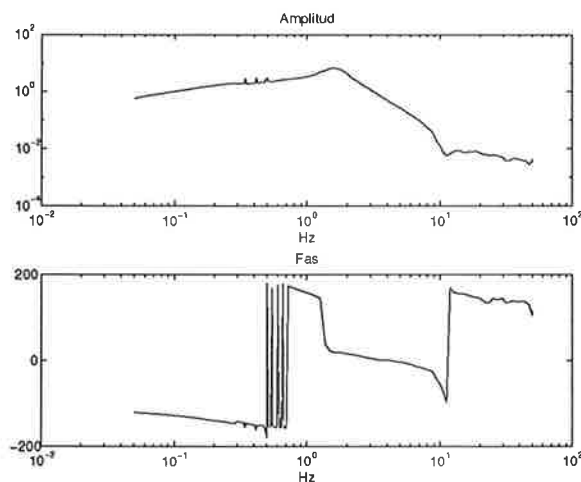
medan cosinuskanalen ger

$$c_T(\omega) = \int_0^T y(t) \cos \omega t dt = \frac{1}{2} T |G(i\omega)| u_1 \sin \phi(\omega) \quad (20)$$

Estimaten av amplituden och fasen kan nu lätt bildas

$$|\hat{G}(i\omega)| = \frac{2}{T u_1} \sqrt{s_T^2(\omega) + c_T^2(\omega)} \text{ och } \hat{\phi}(\omega) = \arctan \frac{c_T(\omega)}{s_T(\omega)} + k\pi \quad (21)$$

I vårt fall användes en Schlumberger frekvensvarsanalysator tillsammans med en PC. Ur figur 5 kan man med lätthet avläsa resonansfrekvensen  $f_0$  till ca 1.57 Hz, vilket överensstämmer bra med värdet ifrån impulssvaranalysen. Man kan också se att det finns ett nollställe i origo ty lutningen fram till resonansfrekvensen är positiv. Lutningen är dock mindre än 1, vilket vi tror beror på att insignalen inte har varit tillräckligt exciterande vid låga frekvenser. Efter resonansfrekvensen är lutningen inte  $-1$  utan snarare  $-3$ . Detta kan bero på att det finns olinjäriteter. Dessa framträder tydligast vid högre frekvenser, troligen orsakad av glappet mellan kulan och pendelarmen.



Figur 5. Bodediagram från frekvensanalys

## 4. Identifiering

När själva identifieringen kunde börja, hade vi mycket information från modelleringen och de inledande experimenten. Vi visste att det borde vara ett andra ordningens system med ett nollställe i origo. Dvs vår modell såg ut som

$$G(s) = \frac{ks}{s^2 + a_1s + a_2} \quad (22)$$

eller om systemet samplas

$$H(q) = \frac{b_1q + b_2}{q^2 + a_1q + a_2} \quad (23)$$

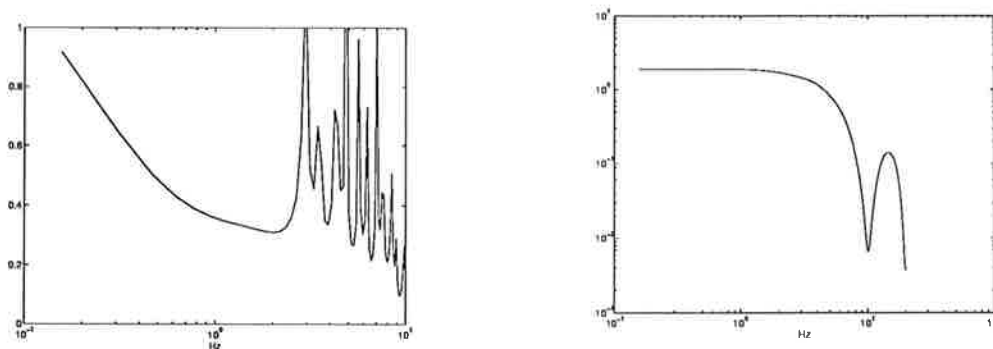
I detta kapitel redovias resultaten från de olika identifieringsmetoderna vi använt oss av. ARMAX och IV lite mer kortfattat än Box-Jenkins ty den senare är den som ger bäst resultat. Men allra först kommer en diskussion angående insamlandet av data som sedermera användes vid identifieringen.

### Datainsamling

Datamängden som ska användas för skattning av parametrar kan inte se ut hur som helst, man måste ställa vissa krav på mängden så att man kan skatta det man önskar. För datainsamlandet används än en gång programmet Logger. Programmet genererar en PRBS-signal, Pseudo Random Binary Sequence, där vissa parametrar kan ställas så önskade krav uppfylles. I vårt fall är det mycket önskvärt att få en bandbredd på PRBS-signalen så att resonansfrekvensen täcks väl in, se figur 6. Vidare bör koherensen vara så nära ett som möjligt. Naturligtvis önskar man så lite brus som möjligt och om det går, inget alls i det intressanta spektrumet. Med följande val av parametrar uppfyller vi uppställda krav någorlunda:

PRBS Period	4
Amplitud	0.7 V
Medelvärde	1.7 V
Tsamp	25 ms

Anledningen till att vi använder en PRBS-signal med bias eller medelvärde är att vi vill att hjulet ska snurra åt samma håll hela tiden och därmed övervinna colombfriktionen.



Figur 6. Koherens och autospektrum för insignalen



Tillgänglig datamängd för identifiering var 2048 sampel. Denna datamängd justerades för att precisionen skulle bli så hög som möjligt vid identifieringen. Till att börja med togs trender bort ifrån mängden via MATLAB-kommandot `dtrend`, därefter delades datamängden i två lika delar dvs om 1024 sampel, för att använda ena hälften vid identifieringen och den andra vid validering.

### Box-Jenkins modeller

Box-Jenkins modell har separat överföringsfunktion för brus och insignal (24). Fem olika parametrar måste initieras vid identifieringen; täljarpolynomens och nämnarpolynomens gradtal och fördröjningen.

$$y_k = \frac{b_1 q^{-k} + \dots + b_{nb} q^{-k-nb+1}}{1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf}} u_k + \frac{1 + \dots + c_{nc} q^{-nc}}{1 + \dots + d_{nd} q^{-nd}} w_k \quad (24)$$

Genom att testa på Akaikes FPE så kan man få en uppfattning om modellordningen och se om den stämmer med modellering.

nb	nc	nd	nf	k	Akaike
1	1	2	2	1	0.02214
2	1	2	2	1	0.01942
3	1	2	2	1	0.01940
4	1	2	2	1	0.01938
5	1	2	2	1	0.01944

Tabell 1. Inverkan av  $nb$

Enligt tabell 1 framgår det att modell 41221 är bäst, men man ser att från och med modell 21221 blir förbättringen obetydlig. Det tyder på att 21221 är en bra modell. Om man istället ändrar på  $nf$  så kan man utläsa av tabell 2 att efter 21221 blir förbättringen liten när ordningen ökas.

nb	nc	nd	nf	k	Akaike
2	1	2	1	1	0.03287
2	1	2	2	1	0.01942
2	1	2	3	1	0.01939
2	1	2	4	1	0.01931

Tabell 2. Inverkan av  $nf$

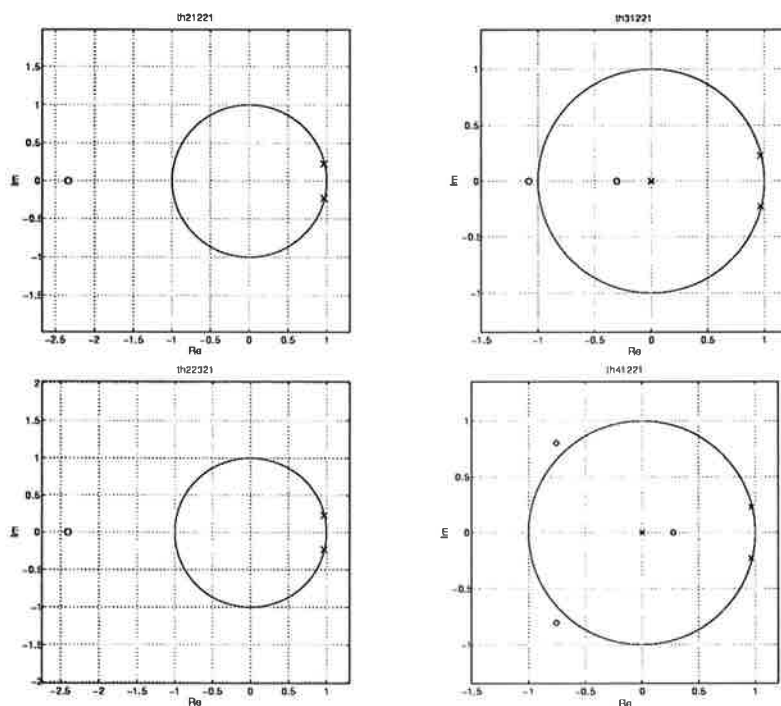
Vidare önskar man veta om modellen för bruset inverkar på estimeringen. Det framgår av tabell 3 att modell 21221 ger ett bra resultat.

Ett annat sätt att se vad som händer vid olika modellordningar, är att studera pol-nollställediagram. I diagrammen, figur 7, finns två poler vid  $ca 0.9 \pm i0.2$  vilket motsvarar resonansfrekvensen. Om vi lägger till ett pol-nollställepar (övre högra bilden), hamnar polen i origo, dvs en fördröjning, och nollstället precis till vänster om denna. I nedre högra bilden tillkommer ett nollställe och då bildas ett komplexkonjugerat par som endast ger bidrag vid höga frekvenser. Där exciteras systemet dåligt, så de är troligen falska. Nedre vänstra bilden visar vad som händer när vi lägger till ett pol-nollställepar i brusmodellen  $C(q)/D(q)$ , dvs mycket lite. Polerna och nollstället ligger kvar på samma

nb	nc	nd	nf	k	Akaike
2	1	1	2	1	0.03393
2	1	2	2	1	0.01942
2	2	3	2	1	0.01950

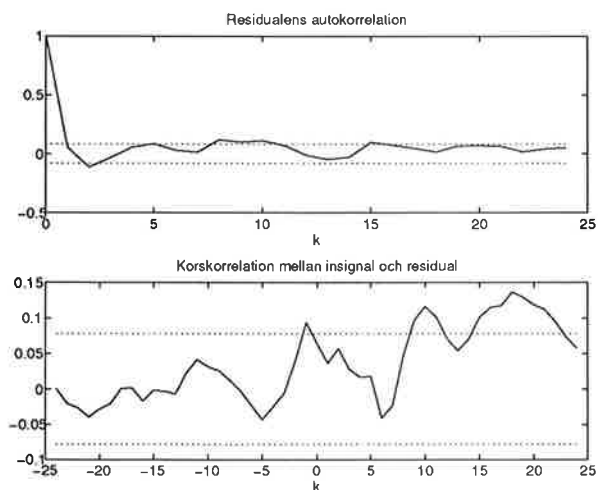
Tabell 3. Brusets inverkan.

ställe. Det tyder på att en brusmodell med ett nollställe och två poler duger. I korrelationstesterna för modell 21221, se figur 8, ses att korrelationen för



Figur 7. Pol-nollställe

residualen i stort sett är vitt brus, vilket den skall vara. I korskorrelationen går kurvan upp vid  $k = 20$  ( $t = 0.5$  s) men den lyckades vi inte få bort oavsett vilken metod vi använde. Slutligen kan vi i figur 9 se resultatet av modelleringen.



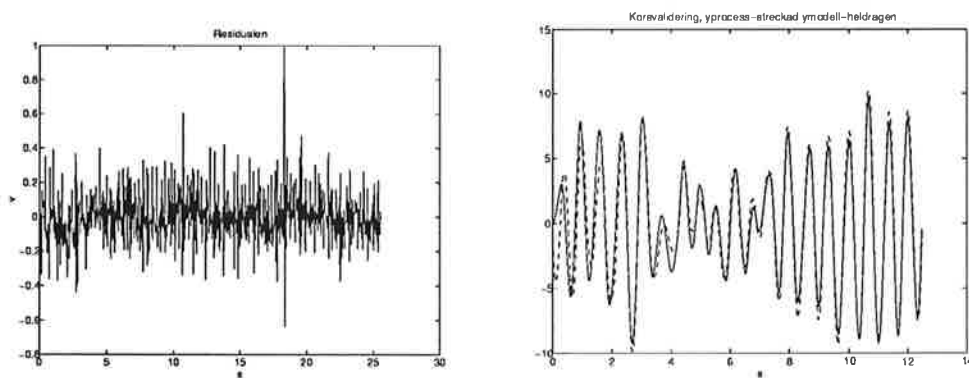
Figur 8. Korrelationstester

Det är den simulerade signalen som väl följer processens utsignal. Därför kan vi nu presentera vår modell i diskret form se (23)

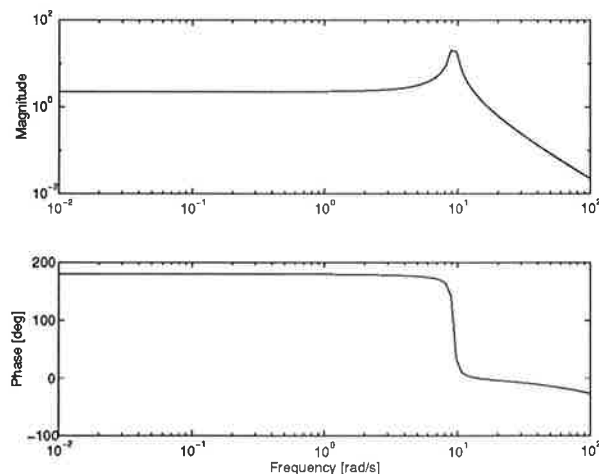
$$H(q) = \frac{-0.0389q - 0.0909}{q^2 - 1.9306q + 0.9844} \quad (25)$$

och i kontinuerlig form (22) där vi kan utläsa  $\omega$  och  $\zeta$  till 9.3 rad/s respektive 0.03 1/s. I våra inledande experiment fick vi fram  $\omega = 9.8$  rad/s och  $\zeta = 0.14$  1/s. Man kan slutligen säga att resonansfrekvensen stämde bra men att  $\zeta$  var något litet och att vi fick ett nollställe som vi inte hade väntat oss. Orsaken till att detta nollställe erhålles kan vara att vid insamlandet av data används en signal med stor excitation i högfrequensområdet och lite excitation i lågfrekvensområdet. Efter översättning av (25 till kontinuerlig tid erhålles 26.

$$G(s) = \frac{s - 197}{s^2 + 0.628s + 87.2} \quad (26)$$



Figur 9. Residualen och korsvalideringen



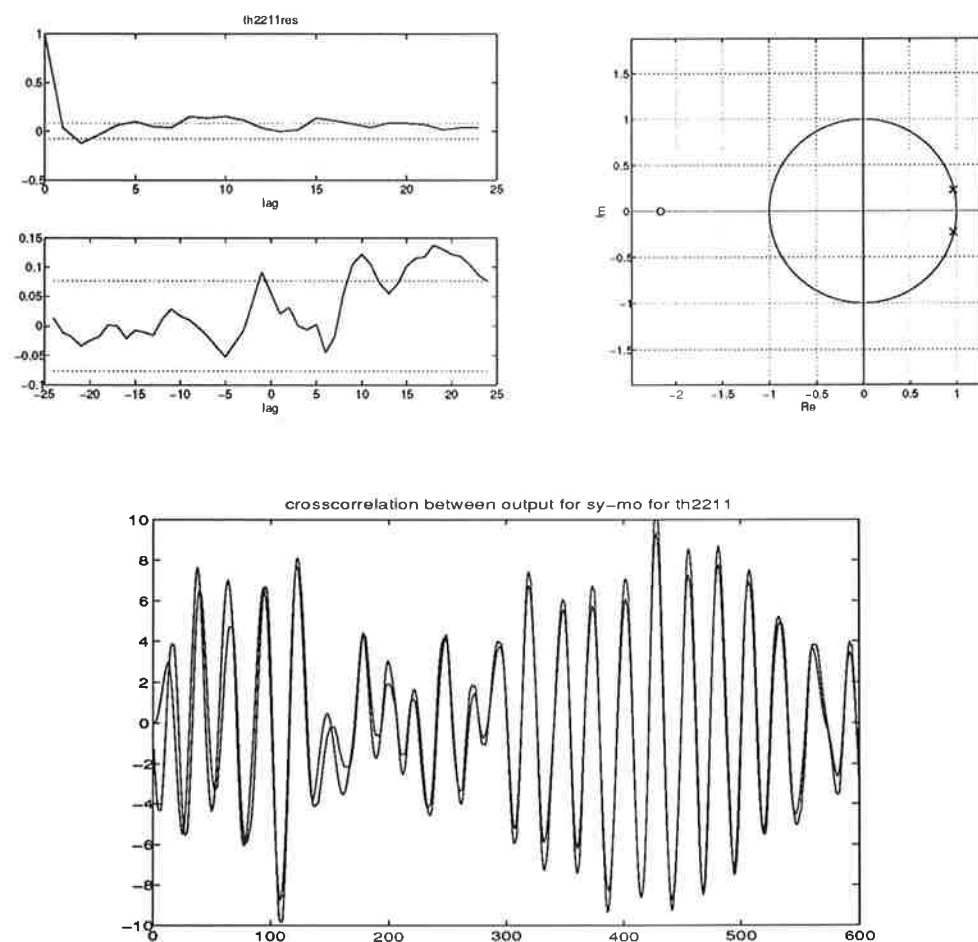
Figur 10. Modellens bodeplot

## ARMAX-modeller

ARMAX-modellen, AutoRegressiv Moving Average with eXogenous input är en generell klass av differensekvationer, och ger därför noggrannare modeller än sina syskon AR, MA och ARMA. AR-modellen utnyttjas till att modellera system med harmoniska svängningar. För design av FIR filter använder man MA-modeller, och ARMA-modeller till modellbaserad spektrumanalys. ARMAX genererar en modell enligt

$$y_k = \frac{b_0 q^{-d} + \dots + b_{n_b} q^{-d-n_b}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} u_k + \frac{1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c}}{1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}} w_k \quad (27)$$

Resultaten av skattningen blev att ett andra ordningens system var tillräcklig och gav i stort sett lika god "Loss function" och "Akaikes FPE" som betydligt högre ordningars modeller. Av de andra ordningarnas modeller gav den med  $n_a = 2, n_b = 2, n_c = 1$  och  $k = 1$  dvs th2211-modellen, bäst resultat. Detta eftersom ett nollställe försvinner vid omvandling till kontinuerliga systemet, och vi då erhåller vårt förväntade system enligt den teoretiska beräkningen. Vad vi vidare kan konstatera är att Box-Jenkins- och ARMAXmodellbygge producerar klart likartade modeller. Detta såg vi i korrelationen mellan insignal och residual, i pol- nollställediagram och i korsvalideringen, se figur 11.



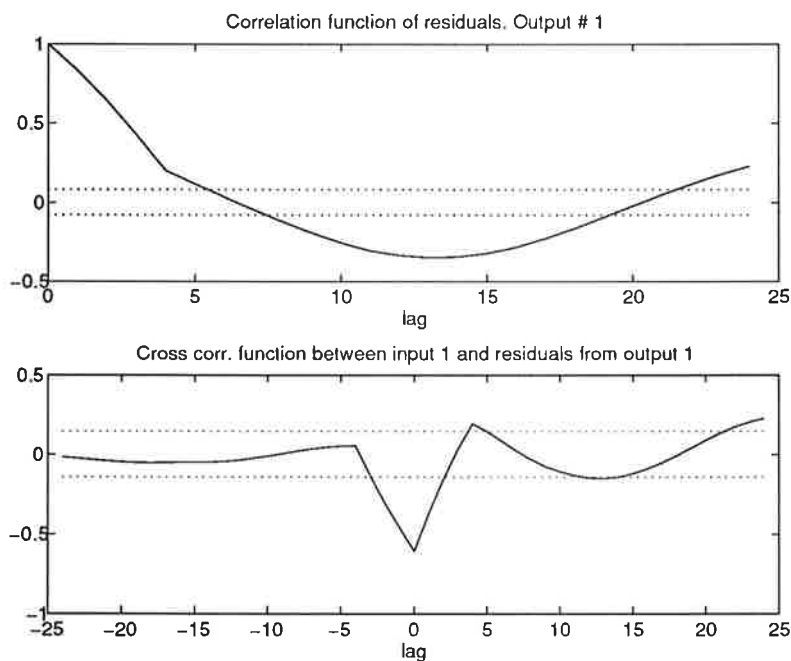
**Figur 11.** ARMAX-modellens korskorrelation insignal-residual (övre till vänster), pol/nollställe-diagram (övre till höger) och korsvalidering (nedre)

Det är värt att notera att modellreducering av ett högre ordningens system ger ett sämre resultat. Detta kan normalt vara bra för att utnyttja en högre ordnings modell för att ta hand om systemets hela karakteristika och sedan reducera ner det till en lägre nivå enligt vad gramianen tillåter. Varje diagonalelement i gramianen talar om den relativa betydelsen av respektive tillstånd i modellen för in- utkaraktistikan. Om detta är mycket mindre än det största värdet i gramianen, betyder detta att styrbarheten och observerbarheten ej påverkas så mycket av dessa matrisparametrar och de kan därför strykas.

Vad matlabfunktionerna gjorde var att göra en balanserad realisering av vår högre ordnings modell, dvs göra en tillståndstransformation så att tillstånden blir lika styrbara som observerbara. Denna görs mha kommandot `dbalreal`, som även räknar ut gramianen. Därefter görs modellreduktionen av systemet utan brusmodell med kommandot `dmodred`.

$$y_k = \frac{b_0 q^{-d} + \dots + b_{nb} q^{-d-nb}}{1 + a_1 q^{-1} + \dots + a_{na} q^{-na}} u_k \quad (28)$$

Modellreduceringen gav ganska dåliga resultat vad avser autokorrelationen för residualen samt korskorrelationen mellan insignal och residual, se figur 12



Figur 12. Autokorrelation residual (övre), korskorrelation insignal-residual (undre)

## IV-modeller

När man identifierar fram parametrar  $\Theta$  via linjär regression enligt

$$Y = \Phi\Theta + v \quad (29)$$

där  $v$  är en brusvektor,  $\Phi$  betecknar regressorerna, leder det ofta till att  $\hat{\Theta}$  kommer att bli ett estimat med bias. IV-metoden förebygger detta genom att man ansätter någon variabel  $Z$ , instrument variabel, som regressor. Parameterestimatet erhålles som

$$\Theta = (Z^T \Phi)^{-1} Z^T Y \quad (30)$$

För att estimatet ska bli konsistent krävs att instrumentvariabeln  $Z$  är okorrelerad med bruset, samt att matrisen  $Z^T \Phi$  är inverterbar. Svårigheten med denna metod är att man ska hitta lämpliga variabler så att de är okorrelerade med bruset. I MATLAB finns det rutiner som automatiskt väljer instrumentvariabler åt användaren. Inparameter till rutinen är modellordning  $na, nb, d$  enligt

$$y_k = \frac{b_0 q^{-d} + \dots + b_{nb} q^{-d-nb}}{1 + a_1 q^{-1} + \dots + a_{na} q^{-na}} u_k \quad (31)$$

i bakåtskiftoperatorn  $q^{-1}$ . Estimatet blir således efter en ARX-modell.

För att göra ett första val av modellordning användes rutinerna `ivstruc` och `selstruc` som beräknar lämpliga modellparametrar. Rutinen gav att modell 212 är ett lämpligt val. Ändras modellen till 222 erhålles ett bättre resultat. Ökas modellordningen ytterligare sker i stort sett ingen förändring av tillförlitlighetsmått "Loss function" och "Akaike's FPE". Överföringsfunktionen till modellordning 222 får följande utseende

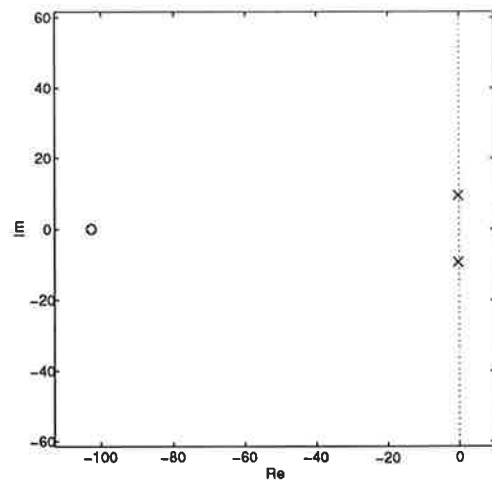
$$H(q) = \frac{-0.123q + 0.010}{q^2 - 1.931q + 0.985} \quad (32)$$

Transformerar (32) till kontinuerlig form erhålles

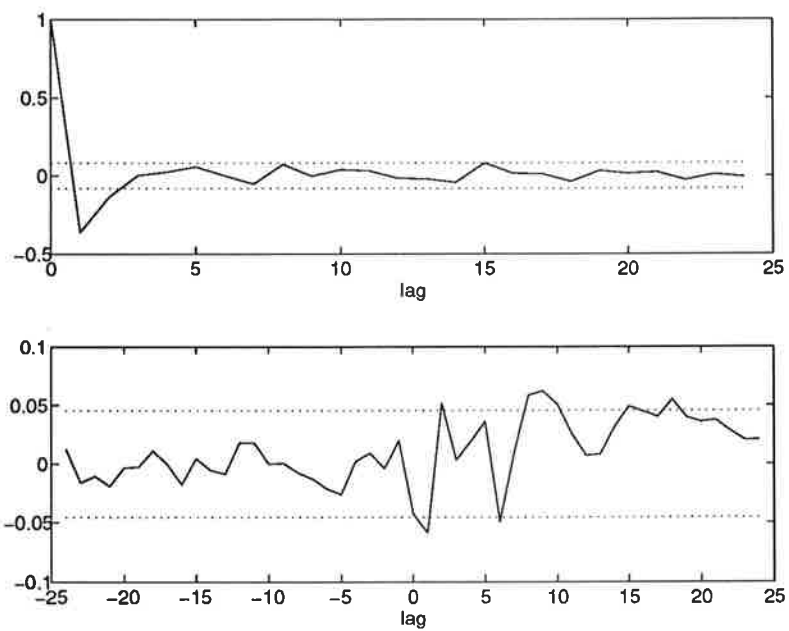
$$G(s) = -\frac{1.918s + 197.065}{s^2 + 0.592s + 87.764} \quad (33)$$

Resultatet är mycket likt resultaten i de andra metoderna. Nämnarpolynomet identifieras bra men täljarpolynomet är inte det önskade. Problemet är det samma som i de andra identifieringarna, dvs använd datamängd innehåller för mycket högfrekvenssegenskaper.

Nedan i figur 13 visas poler och nollställe för det kontinuerliga systemet. Figur 14 visas residualkorrelationer.



**Figur 13.** Pol- nollställediagram för det kontinuerliga systemet



**Figur 14.** Autokorrelationresidual (övre) och Korskorrelation mellan insignal och residual

## 5. Kontinuerliga systemet

För att skatta de kontinuerliga parametrarna kan man göra på olika sätt. Ett är att införa ett filter innan signalerna samplas i datorn. Ett annat sätt är att använda rutiner i MATLAB. Den senare metoden har använts, ty denna är den enklaste och ger samtidigt bra resultat. Man bestämmer utseendet på matriserna  $A, B, C$ , dvs man bestämmer vilka element som ska identifieras fram. Själva identifieringen görs enligt prediktionsfelsmetoden. För att få bra parameterkonvergens initieras parametrarna till lämpliga värden.

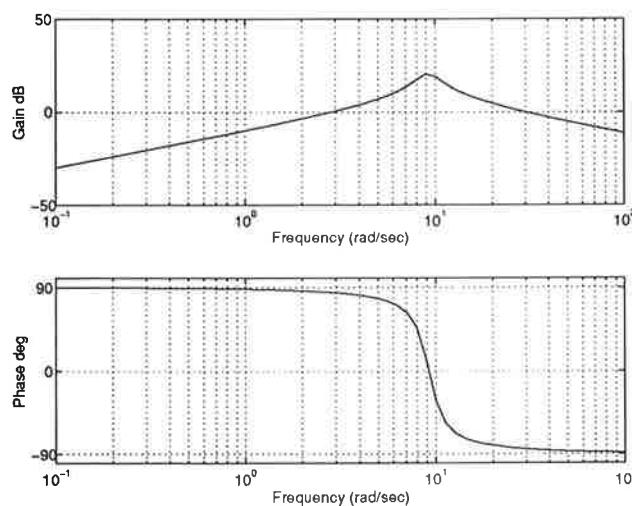
Med utgångspunkt från Box-Jenkinsidentifieringen (BJ) och där den kontinuerliga översättningen, ansattes matriserna på styrbar kanonisk form. De parametrar som identifierats fram enligt BJ sattes som initialvärden på de okända matriselementen. Följande kommandosekvens användes:

```
A=[NaN,NaN;1,0];  
B=[1,0]';  
C=[NaN 0];  
ms=modstruc(A,B,C,[0],[0,0]');  
th0=ms2th(ms,'c',[-0.628,-87.2,10]);  
th=pem(Zcont,th0);  
[num,den]=th2tf(th);
```

Sekvensen ger täljar- och nämnarpolynom i den kontinuerliga överföringsfunktionen.

$$G(s) = \frac{27.02s}{s^2 + 2.52s + 85.80} \quad (34)$$

Vi får alltså  $\omega = 9.3$  rad/s och  $\zeta = 0.14$  1/s, vilket stämmer väldigt bra överens med de initiala experimenten. Där fick vi ju  $\omega = 9.8$  rad/s oeg  $\zeta = 0.14$  1/s. Därmed kan vi slå fast att den kontinuerliga skattningen gav det bästa resultatet av de metoder vi använt. Det kan delvis bero på att en ny datamängd samlades in och användes för denna skattningen. Vi trodde nämligen att vår insignal hade för mycket energi vid höga frekvenser och därmed gav dålig excitation vid låga frekvenser. Det resulterande bodediagrammet syns i figur 15



Figur 15. Kontinuerliga skattningens bodeplot



## 6. Sammanfattning

Till projekt i kursen processidentifiering fick vi till uppgift att modellera teoretiskt och experimentera fram den kontinuerliga skattningen för processen *Ball and Hope*, eller *ekorrhjulet* som det svenska epitetet lyder. Den teoretiska beräkningen gjordes mha Maple och de inledande experimenten gav dämpning och vinkelhastighet för detta andra ordningens system. Dessa gjordes med impulssvar, stegsvar och frekvenssvar.

För själva identifieringen användes tre metoder, nämligen Box-Jenkins, ARMAX och Instrumental Variable. Dessa gav tämligen likartade resultat, men med visst övertag för Box-Jenkins modell enligt tabell 4, eftersom denna bättre brusreducerar ty den har en specialdesignad nämnare för bruset. Det visade sig mycket riktigt att ett andra ordningens system gav ungefär lika goda resultat som vilket högre ordningens system som helst. Dock blev modellens noll-

Metod	Loss Function	Akaike
Box-Jenkins	0.01915	0.01942
ARMAX	0.01967	0.01987
IV	0.02133	0.02149

**Tabell 4.** Tabell över Loss Function och Akaike FPE för de olika metodernas modeller

ställeskattning felaktig pga en för dåligt exciterande signal vid låga frekvenser. Därför gjordes till slut en kontinuerlig skattning mha Identification i MATLAB för att verifiera den teoretiskt framtagna modellen, vilket gav god överensstämmelse. Vår slutgiltiga kontinuerliga modell har följande utseende

$$G(s) = \frac{ks}{s^2 + 2\zeta\omega s + \omega^2} = \frac{27.02s}{s^2 + 2.52s + 85.80} \quad (35)$$

med  $\omega = 9.3$  rad/s och  $\zeta = 0.14$  1/s, vilket överensstämmer med de initiala experimenten. Det bär dock påpekas att skattning av konstanten  $k$  inte är helt verifierad.

# Projekt i processidentifiering

Henrik Amilon, f89  
Johan Kleberg, f89  
Handledare: Rolf Johansson

20 maj 1994

## **Innehåll**

<b>1 Inledning</b>	<b>3</b>
<b>2 Modell</b>	<b>3</b>
<b>3 Experimentet</b>	<b>3</b>
<b>4 Modeller</b>	<b>4</b>
<b>5 Alkoholtest</b>	<b>9</b>
<b>A Appendix</b>	<b>11</b>

## 1 Inledning

Projektet utfördes delvis i samarbete med Mikael Åkesson och P A Fransson på Balanslaboratoriet vid Lunds lasarett. Vi undersökte kroppens balansreglering genom att approximera denna som en inverterad pendel och försöka identifiera processparametrarna i modellen. Uppläggningsen av projektet följer till stor del Appendix E i System Modeling & Identification av Rolf Johansson.

För parameterskattningarna användes två olika metoder: Identifiering med hjälp av ARMAX-rutinen i System Identification Toolbox respektive egenhändigt implementerad Pseudolinjär regression.

Vi undersökte dessutom hur alkohol påverkade parametrarna.

## 2 Modell

Härledning av PID-reglering av en inverterad pendel är gjord i Appendix E, där det framgår att regulatorparametrarna ges av  $k$ ,  $\eta$  och  $\rho$ . Parametern  $k$  kan i en mekanisk modell tolkas som en fjäderkonstant,  $\eta$  som en dämpningsfaktor och  $\rho$  som en faktor som svarar mot integraldelens tidskonstant.  $\rho$  är alltså en parameter som kompenserar för eventuell bias i vinkelregleringen. Vi nöjer oss här med att presentera överföringsfunktionen från stimuli till det stabiliserande vrismomentet.

$$T_{bal}(s) = \frac{(b_1 + b_2)(s^3 - \frac{g}{l}s)}{s^3 + \eta s^2 + ks + \rho} V(s)$$

Ett sätt att normalisera överföringsfunktionen är:

$$T_{bal}(s) = \frac{(b_1 + b_2)((\frac{s}{\omega_0})^3 - \frac{g}{l\omega_0^2}(\frac{s}{\omega_0}))}{(\frac{s}{\omega_0})^3 + \frac{\eta}{\omega_0}(\frac{s}{\omega_0})^2 + \frac{k}{\omega_0^2}(\frac{s}{\omega_0}) + 1} V(s), \quad \omega_0 = \rho^{1/3}$$

Ur denna representation kan följande egenskaper utläsas:

<i>Snabbhet</i>	$\omega_0 = \rho^{1/3}$
<i>Styvhet</i>	$k/\omega_0^2$
<i>Dämpning</i>	$\eta/\omega_0$

Snabbheten ger information om systemets bandbredd. Ett högt värde på denna innebär snabbt svar på störningar. Ett högt värde på styvheten innebär små avvikelser från jämviktsläget. Dämpningsfaktorn är kopplad till kroppens svajhastighet.

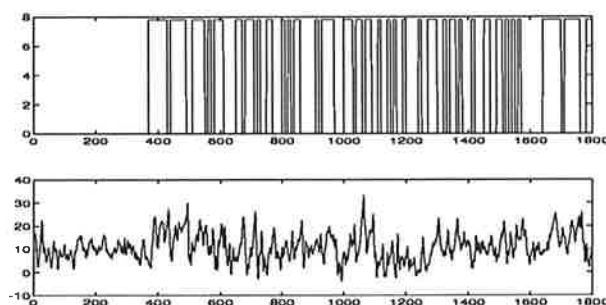
## 3 Experimentet

Försökspersonen, man i 25-årsåldern, ställde sig på en s.k kraftplattform, med vilken moment och normalkrafter uppmättes. Därefter applicerades vibratorer på vadmuskulaturen. Försökspersonen stod upprätt med slutna ögon och som stimuli användes en PRBS-signal. Två mätningar genomfördes: en vid frekvensen 60 Hz och en vid 80 Hz.

En naturlig begränsning vid försöket är kroppens självreglering kring det instabila jämviktsläget. För att kvantifiera denna inleddes varje försök med 30 sekunders mätning utan insignal. Med hjälp av tumregeln som säger att  $\omega h \approx 0.2 - 0.6$ , uppskattade vi utifrån pendelresonemang  $\omega \approx \sqrt{g/l} \approx 3$  rad/s och valde sampelfrekvensen 10 Hz (63 rad/s).

Med hjälp av balanslaboratoriets matlabfunktion Transfer beräknades de olika kraft- och momentkomponenterna i x-, y- och z-led. Vid beräkningarna togs även hänsyn till det faktum att kraft- och masscentrum vanligtvis inte sammanfaller.

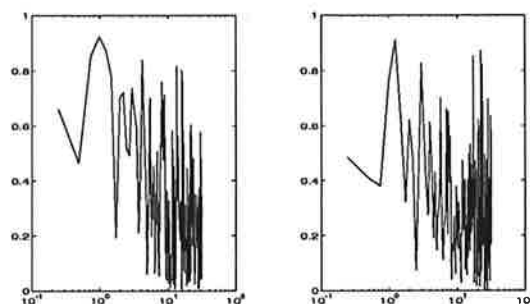
För vårt vidkommande fann vi det naturligast att arbeta med momentkomponenten i x-led, dvs i näsans längdriktning.



Figur 1: Insignal och utsignal vid 60 Hz.

## 4 Modellering

För att undersöka rimligheten att ansätta en linjär modell till våra data studerades koherensen mellan in- och utsignal för de båda mätningarna. Dessa får anses vara tillfredställande.



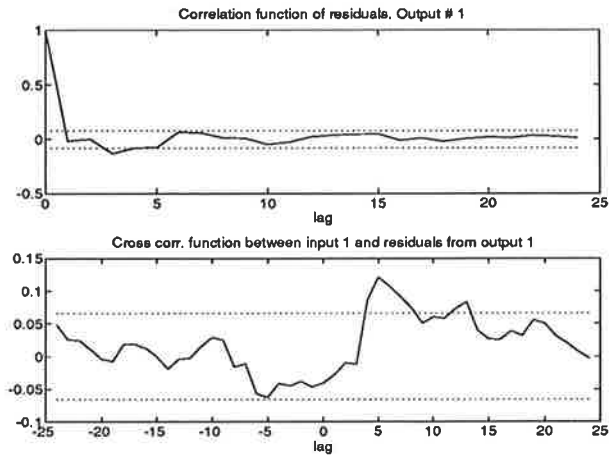
Figur 2: Koherensspektrum (rad/s) för 60 Hz resp. 80 Hz.

För den vidare identifieringen och valideringen, delade vi upp våra mätdata i två serier. En användes till modellering och en till validering. Vårt mål var att kontrollera den kontinuerliga modellens giltighet genom att undersöka om en tredje ordningens ARMAX-modell kunde anpassas till våra data och utifrån denna identifiera  $k$ ,  $\eta$  och  $\rho$ .

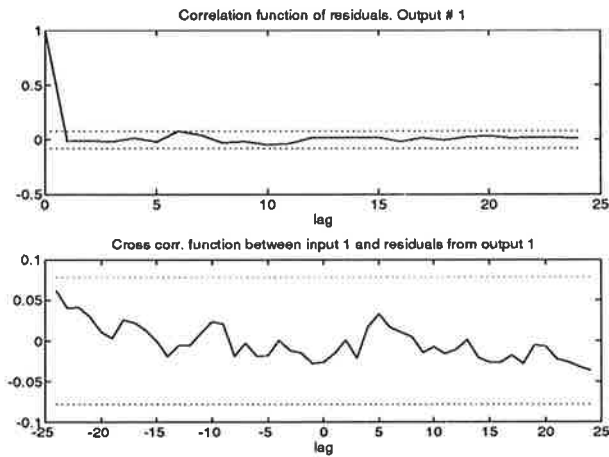
Med hjälp av funktionen ARMAX i System Identification Toolbox anpassades modeller av ordningar 1-4. Genomgående visade körningarna att tidsfördröjningen ungefär kunde fastställas till två sampel, dvs 0.2 sekunder. Detta beror troligtvis på den naturliga fördröjningen i nervbanorna.

Akaike's FPE-test visade en stor skillnad mellan modeller av ordning 2 och 3, medan däremot modeller av ordning 3 och 4 hade liknande resultat. Detta indikerar att en modell av ordning 3 borde vara nödvändig och kanske tillräcklig.

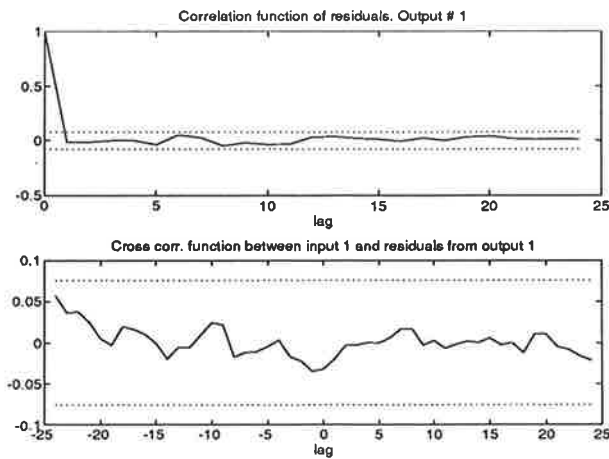
Ordning	FPE(60 Hz)	Ordning	FPE(80 Hz)
1	2.037	1	1.875
2	1.694	2	1.678
3	1.562	3	1.559
4	1.541	4	1.543



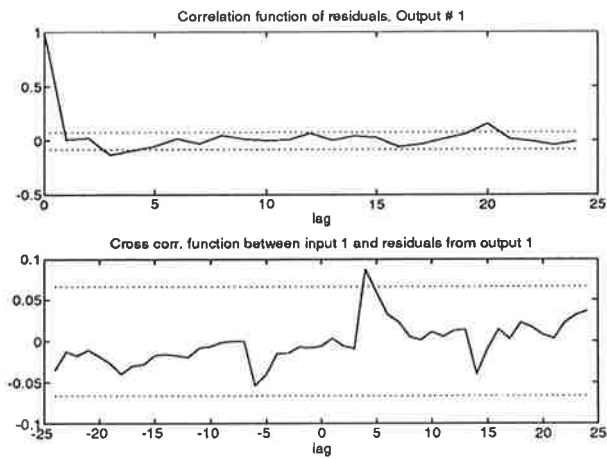
Figur 3: Ordning 2 (60 Hz). I figuren är 99%-iga konfidensintervall markerade.



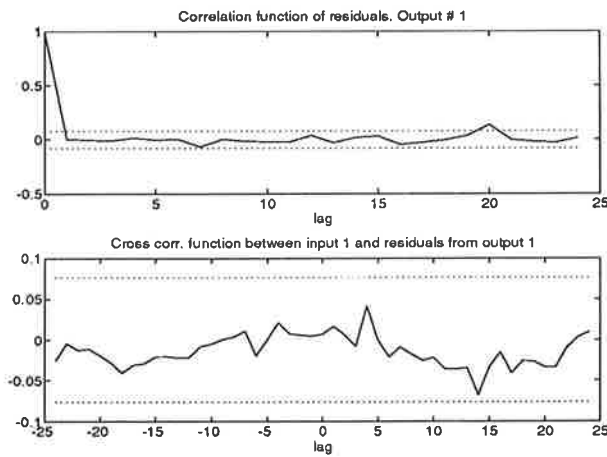
Figur 4: Ordning 3 (60 Hz). I figuren är 99%-iga konfidensintervall markerade.



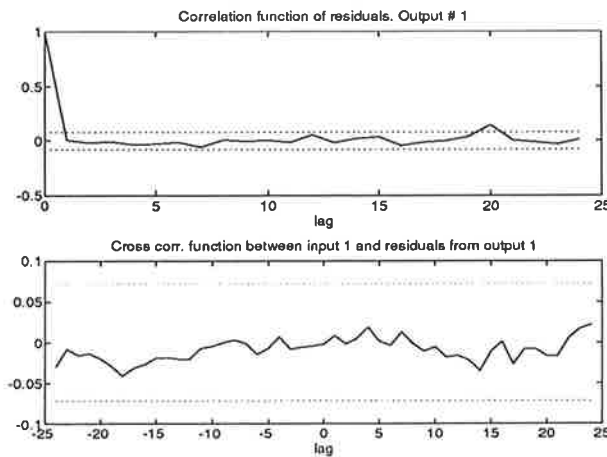
Figur 5: Ordning 4 (60 Hz). I figuren är 99%-iga konfidensintervall markerade.



Figur 6: Ordning 2 (80 Hz). I figuren är 99%-iga konfidensintervall markerade.



Figur 7: Ordning 3 (80 Hz). I figuren är 99%-iga konfidensintervall markerade.

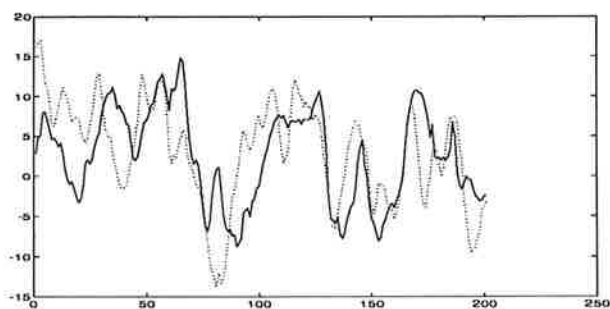


Figur 8: Ordning 4 (80 Hz). I figuren är 99%-iga konfidensintervall markerade.

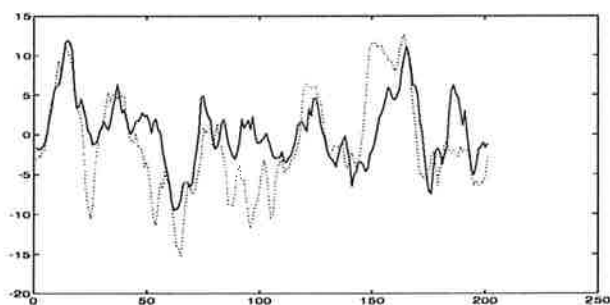
För att avgöra om residualerna kan approximeras med vitt brus gjordes residualtest. Dels studerades autokorrelationen och dels korskorrelationen mellan residualer och insignal.

Även residualtesten visar ett oberoende mellan residualerna och mellan residualerna och insignalen för en modell av ordning 3 och större.

Som ett ytterligare test gjorde vi en korsvalidering dvs vi jämförde simulerade ut signaler med verkliga ut signaler som ej använts vid modelleringen.



Figur 9: Utsignal (-) och simulerad utsignal (60 Hz).



Figur 10: Utsignal (-) och simulerad utsignal (80 Hz).

## Resultat

Den modell vi slutligen valde för de båda mätningarna var två ARMAX(3,2,3)-modeller.

60 Hz

Loss fcn: 1.539 Akaike's FPE: 1.562 Sampling interval 1

The polynomial coefficients and their standard deviations are

B =

0	0	-0.1120	0.1594
0	0	0.0203	0.0228

A =

1.0000	-2.1729	1.6748	-0.4739
0	0.0642	0.1121	0.0526

C =

1.0000	-0.7061	0.1292	-0.0686
0	0.0697	0.0444	0.0390



80 Hz

Loss fcn: 1.537 Akaike's FPE: 1.559 Sampling interval 1

The polynomial coefficients and their standard deviations are

B =

0	0	-0.0989	0.1319
0	0	0.0151	0.0161

A =

1.0000	-2.0156	1.4526	-0.4196
0	0.0715	0.1257	0.0606

C =

1.0000	-0.6754	0.2529	-0.1093
0	0.0766	0.0527	0.0379

Eftersom vårt mål var att identifiera återkopplingsparametrarna  $k, \eta$  och  $\rho$  gavs tillräcklig information för detta i A-polynomen. Med kännedom om hur kontinuerliga poler mappas vid sampling kan de kontinuerliga A-polynomen rekonstrueras (se Appendix).

Dessa blir:

$$A_{60}(s) = s^3 + 7.47s^2 + 41.92s + 41.15$$

$$A_{80}(s) = s^3 + 8.68s^2 + 62.06s + 27.41$$

Identifiering av koefficienterna i en ARMAX(3,2,3)-modell gjordes också genom Pseudolinjär regression (se Appendix).

$$A_{60}(s) = s^3 + 8.43s^2 + 43.83s + 41.29$$

$$A_{80}(s) = s^3 + 11.20s^2 + 78.37s + 35.29$$

Ett försök till jämförelse mellan dessa metoder är en beräkning av förlustfunktionen  $V = \frac{1}{2} \sum_{i=1}^n \varepsilon_i^2$ .

$$\begin{aligned} 60 \text{ Hz: } \quad V &= 835 \\ V_{plr} &= 842 \end{aligned}$$

$$\begin{aligned} 80 \text{ Hz: } \quad V &= 816 \\ V_{plr} &= 813 \end{aligned}$$

Förlustfunktionerna är tämligen lika för de båda identifieringsmetoderna. I 60 Hz-fallet är dock avvikelserna i parametervärdena små, medan 80 Hz-mätningen uppvisar anmärkningsvärt stora skillnader, trots lägre förlustfunktioner. Detta är ett exempel på att en lägre förlustfunktion inte alltid behöver betyda en bättre parameterkonvergens. Sambanden dem emellan är ej triviala.

## Summering

Test	$\eta$	k	$\rho$	Snabbhet	Styvhet	Dämpning
60 Hz	7.47	41.92	41.15	3.45	3.52	2.16
60 Hz (PLR)	8.43	43.83	41.29	3.45	3.67	2.44
80 Hz	8.68	62.06	27.41	3.02	6.83	2.88
80 Hz (PLR)	11.20	78.37	35.29	3.28	7.28	3.41

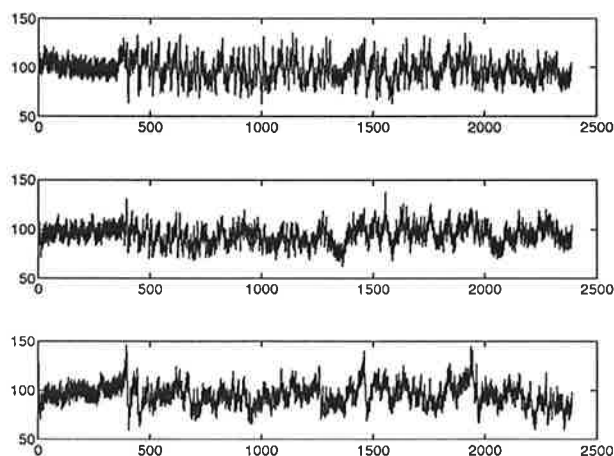
Den approximativa modellen ovan med en PID-reglering kring det instabila jämviktsläget (Vinkeln  $\theta=0$ ) via vristmomentet  $T_{bal}$  visade sig vara tillräcklig för att förklara mätresultaten.

Vi har inte gått in närmare på huruvida modeller av större ordning än 3 skulle vara bättre utifrån statistiska kriterier. De lägre FPE-talen för dessa modeller talar till viss del för detta även om osäkerheten, dvs standardavvikelserna, i parameterskattningarna är ett problem.

En fysiologisk tolkning av resultaten överlåter vi med varm hand till läkarskrået.

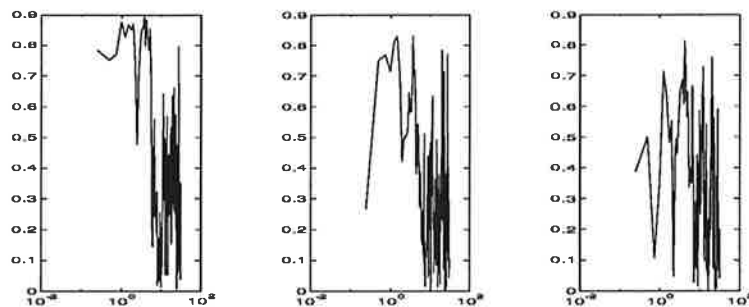
## 5 Alkoholtest

För att studera alkoholens inverkan gjordes ytterligare tre mätningar. Ett par kraftigare vibratorer (1 mm amplitud mot tidigare 0.4 mm) användes denna gång. Vibratorfrekvensen var 60 Hz och sampelfrekvensen som tidigare, 10 Hz. Testen utfördes på följande sätt: Som referens gjordes först ett nolltest, dvs utan alkoholförtäring. Därefter fick försökspersonen, man i 25-årsåldern, hålla i sig 5 cl 40%-ig sprit (Absolut Vodka med citrussmak). Efter 30 minuter togs blodprov och omedelbart därefter gjordes en mätning. Med viss möda (försöket utfördes onsdag em) förtärdes ytterligare 15 cl sprit och proceduren upprepades.



Figur 11: Utsignaler för 0, 5 respektive 20 cl.

Som synes av figuren verkar känsligheten för stimuli vara större för nolltestet. Detta borde ge en bättre koherens för nolltestet än för de övriga. Notera också skillnaderna under de 30 första sekunderna som beskriver självsvajet, dvs ingen insignal.



Figur 12: Koherensspektrum (rad/s) för 0, 5 respektive 20 cl.

En möjlig tolkning av detta är att man vid alkoholpåverkan fäster mindre vikt vid pålagda störningar i balanssystemet. Hur påverkas då parametrarna?

## Summering

Identifiering med ARMAX-rutin och omräkning enligt appendix till sökta parametrar gav resultat enligt tabellen nedan. Notera dock att för 20 cl-provet gav en tidsfördröjning på 0.3 sekunder ett bättre resultat än den tidigare använda fördröjningen som var 0.2 sekunder. De stora skillnaderna i parametrarna jämfört med tidigare mätningar får skyllas på de kraftigare vibratorerna.

Test	Promille	$\eta$	k	$\rho$	Snabbhet	Styvhet	Dämpning
0 cl	0.0	2.89	161.02	124.25	4.99	6.47	0.58
5 cl	ej påvisat	0.67	158.08	95.25	4.57	7.58	0.15
20 cl	0.82	1.07	160.91	67.30	4.07	9.73	0.26

Snabbheten, dvs förmågan att svara på jämviktsstörningar, minskar, kanske ej helt överraskande, med ökande alkoholintag. Ett högt värde på styvheten innebär små avvikelser från jämviktsläget. Att denna faktor ökar kan bero den tidigare nämnda ignoreringen av insignalen. Man kan konstatera att dämpningsfaktorn, som är kopplad till svajhastigheten, minskar kraftigt vid alkoholintag. Något förvånande är att dämpningsfaktorn hörande till 20 cl-provet är högre än den hörande till 5 cl-provet.

Några långtgående slutsatser är det osäkert att dra med tanke på att endast en person ingår i undersökningen. Vid upprepade tester vänjer sig dessutom försökspersonen vid störningen, vilket försvårar precisionen i tolkningen av resultatet.

## A Appendix

Matlabkod för Pseudolinjär regression:

```
clear phi Y r
n1=19;           %19:e ordningens modell
y=m6(:,1);
u=m6(:,2);
kol=1;
for i = n1:-1:1
    for rad = 1:(size(y)-n1)
        phi(rad,kol)=-y(i+rad-1);
        phi(rad,kol+n1)=u(i+rad-1);
    end
    kol=kol+1;
end
Y=y(n1+1:size(y));
thhat=inv(phi'*phi)*(phi'*Y);
yhat=phi*thhat;
r=Y-yhat;

clear phi2 Y2 yhat2 r2
n2=3;           %vår modellordning
kol=1;
for i = n2:-1:1
    for rad = 1:(size(r)-n2)
        phi2(rad,kol)=-y(i+n1+rad-1);
        phi2(rad,kol+n2+2)=r(i+rad-1);
        if i < n2
            phi2(rad,kol+n2-1)=u(i+n1+rad-1);
        end
    end
    kol=kol+1;
end

Y2=y(n2+n1+1:size(y));
thhat2=inv(phi2'*phi2)*(phi2'*Y2)
yhat2=phi2*thhat2;
r2=Y2-yhat2;

%Förlustfunktionerna
V2=0.5*r2'*r2
size(r2)
e=e(1:size(r2));
V1=0.5*e'*e
```

Matlabkod för beräkning av kontinuerliga koefficienter:

```
h=0.1;
A=[1.0000 a1 a2 a3];
r=roots(A);
z1=[1 -log(r(1))/h];
z2=[1 -log(r(2))/h];
z3=[1 -log(r(3))/h];
Ac=conv(z1,conv(z2,z3))
```

# Identifiering av bom- kulaprocessen

## Projekt i kursen Processidentifiering

Johan Gustafsson, E-90

och

Jerker Linder, E-90

Handledare: Lennart Andersson, Inst. för reglerteknik

25 april 1994

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
<b>2</b>	<b>Modellering</b>	<b>1</b>
2.1	Modellering av överföringsfunktion mellan motor och bom . . . . .	1
2.2	Modellering av överföringsfunktion mellan bom och kula . . . . .	1
<b>3</b>	<b>Identifiering</b>	<b>2</b>
3.1	Uppdelning i vinkel- och lägesprocess . . . . .	2
3.2	Identifiering av vinkelprocessen . . . . .	3
3.3	Identifiering av lägesprocessen . . . . .	4
3.3.1	Identifiering av lägesprocessen med tidskontinuerlig metod	6
<b>4</b>	<b>Validering</b>	<b>7</b>
4.1	Inledning . . . . .	7
4.2	Validering av vinkelmodellen . . . . .	7
4.2.1	Slutlig modell för vinkelprocessen . . . . .	9
4.3	Validering av lägesmodellen . . . . .	10
4.3.1	Slutlig modell för lägesprocessen . . . . .	11
<b>5</b>	<b>Reglering</b>	<b>12</b>
<b>6</b>	<b>Slutsatser</b>	<b>13</b>

## Sammanfattning

Projektet bestod av att identifiera en process där en kula rullades fram och tillbaka på en bom vars lutning styrdes av en motor. Då processen identifierats skulle modellen valideras och en regulator konstrueras. Processen delade vi upp i två delar, en som beskrev överföringsfunktionen mellan motorns insignal och bommens lutning och en som beskrev överföringsfunktionen mellan bommens lutning och kulans läge. Delarna identifierade vi var för sig genom att försöka anpassa olika modeller, t.ex. ARX-, ARMAX- och Box-Jenkins-modeller till insamlade mätdata. Modellerna validerades genom att titta på residualernas autokorrelation, korrelationen mellan residualer och insignal, pol-nollställediagram, bodediagram samt genom att simulera modellernas utsignaler och jämföra dem med de verkliga.

För att reglera processen använde vi en P-regulator för att reglera bommens vinkel och en RST-regulator för hela systemet. Vi fick på detta sätt en regulator som fungerade bra.

Våra slutliga modeller blev: Överföringsfunktionen mellan motorns insignal och bommens lutning modellerades med en Box-Jenkins-modell av tredje ordningen:

$$H(q) = \frac{0.0050q}{q^3 - 2,1811q^2 + 1,5629q - 0,3818}$$

Samplingsintervall 5 ms

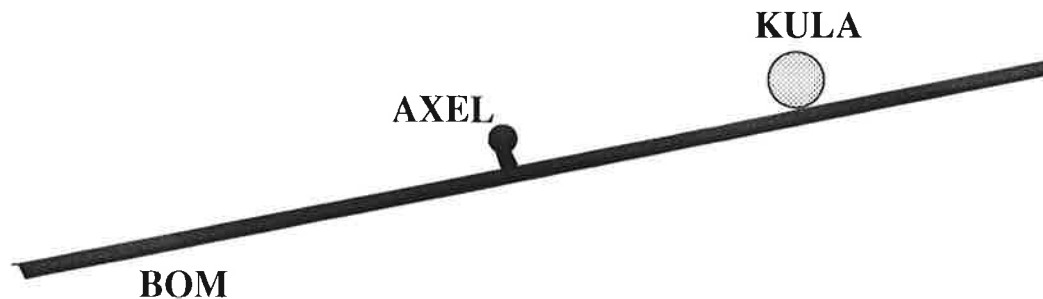
Överföringsfunktionen mellan bommens lutning och kulans läge modellerades med en Box-Jenkins-modell av andra ordningen:

$$H(q) = \frac{-0.0228q}{q^2 - 1,9968q + 0,9968}$$

Samplingsintervall 50 ms

## 1 Inledning

I kursen Processidentifiering vid LTH ingår ett projekt. I projektarbetet skall studenterna visa att de behärskar kunskaperna från kursen och kan tillämpa dem. Denna rapport beskriver ett projekt där man skulle undersöka en motordriven bom med en rullande kula. I systemet finns två mätbara storheter, bommens vinkel och kulans läge på bommen enligt figur 1.



Figur 1: Bom- och kula-processen

## 2 Modellering

### 2.1 Modellering av överföringsfunktion mellan motor och bom

Genom att undersöka överföringsfunktionen mellan motor och bom med hjälp av en frekvensvarsanalysator fås en approximativ bodeplot. Detta gjordes under en laboration tidigare i kursen och processen visade sig då ha följande egenskaper: -1 lutning fram till ca 10-20 Hz och därefter en -3 lutning. Det fanns även en resonanstopp vid brytpunkten. Den förväntade modellen blir därför:

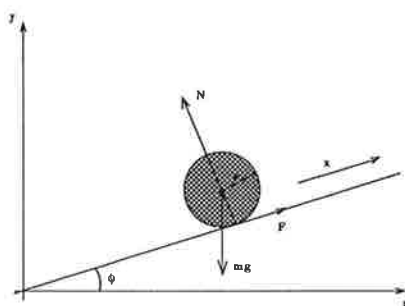
$$G_\phi = \frac{b_0}{s(s^2 + 2\xi\omega s + \omega^2)}$$

### 2.2 Modellering av överföringsfunktion mellan bom och kula

Dynamiken för kulans läge på bommen kan bestämmas med klassisk mekanik. Antag att kulans läge är  $x$ , bommens vinkel är  $\phi$  och att bommen går genom centrum för vridningen  $\phi$ . Vidare inför vi koordinater och krafter enligt figur 2. Kraftekvationen ger:

$$m\ddot{y} = -mg + N \cos \phi + F \sin \phi$$





Figur 2: Krafter och koordinatriktningar som används för att beräkna dynamiken för kulans läge på bommen

$$m\ddot{z} = -N \sin \phi + F \cos \phi$$

Om vi multiplicerar med  $\sin \phi$  respektive  $\cos \phi$  och adderar fås

$$m(\ddot{y} \sin \phi + \ddot{z} \cos \phi) = -mg \sin \phi + F. \quad (1)$$

Villkor för friktionsfritt rullande kula blir  $Fr = J\dot{\omega} = -J\ddot{x}/r$ . För ett klot är  $J = 2mr^2/5$ , alltså  $F = -2m\ddot{x}/5$ . Vidare gäller  $z = x \cos \phi$  och  $y = x \sin \phi$ . Deriveras dessa samband två gånger och man multiplicerar med  $\cos \phi$  respektive  $\sin \phi$  samt adderar så erhålls

$$\ddot{y} \sin \phi + \ddot{z} \cos \phi = \ddot{x} - x\dot{\phi}^2.$$

Sätt in detta uttryck och uttrycket för F i ekvation 1. Härvid erhålles

$$m(\ddot{x} - x\dot{\phi}^2) = -mg \sin \phi - \frac{2}{5}m\ddot{x}.$$

Om vi nu antar att  $\phi$  och  $\dot{\phi}$  är små erhålls slutligen

$$G_x(s) = -\frac{5g}{7s^2} \approx -\frac{7}{s^2}.$$

### 3 Identifiering

#### 3.1 Uppdelning i vinkel- och lägesprocess

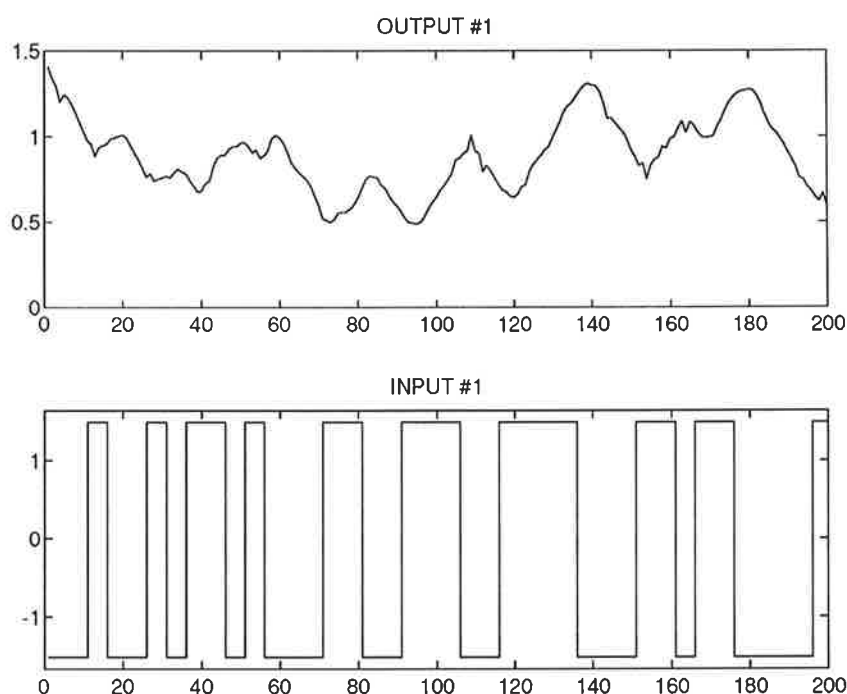
Eftersom kulan påverkar bommen försumbart lite delas processen in i två delar, vinkel och kula. Vinkelprocessen är den process som finns mellan bommens vinkel och bommens motor. Kulprocessen är den som finns mellan kulans läge och bommen. Många vinster kan göras om experimentet läggs upp på detta sätt. Snabba förlopp hos kulprocessen dränks inte i den långsamma vinkelprocessen. Flera fördelar fås även vid regulatordesignen senare.

### 3.2 Identifiering av vinkelprocessen

Experiment för identifiering av  $G_\phi$  gjordes med en PRBS-signal som insignal och vinkeln  $\phi$  mättes. Alla mätdata samlades upp av programmet Logger. Genom en del provande och genom att titta på koherencefunktionen mellan insignal och utsignal kom vi fram till att följande inställningar av PRBS-signalen exciterade processen bra:

- Samplingstid 5 ms
- PRBS-period 5 sampel
- Amplitud 1,5 V
- Offset -0.015 V

Med dessa värden blev högsta frekvensen på PRBS-signalen ungefär 40 Hz. Vid högre frekvens gjorde bommen inget utslag. 6000 mätdata sparades vilka delades upp i två delar, en för identifiering och en för validering. De 200 första mätvar-dena framgår av figur 3. Efter att ha dragit bort medelvärdet och "outliners" från



Figur 3: In- och utsignal som användes vid identifiering av vinkelprocessen

mätdata med kommandot `dtrend` i MATLAB började vi själva identifieringen genom att försöka anpassa olika modeller, främst ARMAX och Box-Jenkins, till mätdata. Som ett första urvalskriterium använde vi "Akaike's final prediction error", men de flesta modellerna jämförde vi även genom att titta på korrelationen

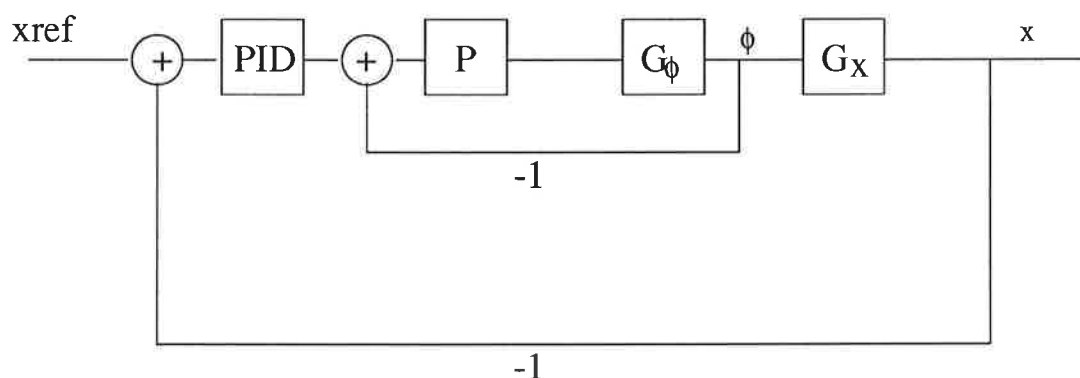
mellan residualerna, pol-nollställediagram och genom att simulera modellens utsignal med de mätdata som inte användes vid identifieringen och jämföra med den verkliga utsignalen. Vi kom slutligen fram till att följande Box-Jenkins modell beskrev systemet bäst:

$$y(t) = \frac{B(q)}{F(q)}u(t - k) + \frac{C(q)}{D(q)}e(t)$$

$$[\deg B \deg C \deg D \deg F k] = [1 \ 4 \ 4 \ 3 \ 2]$$

### 3.3 Identifiering av lägesprocessen

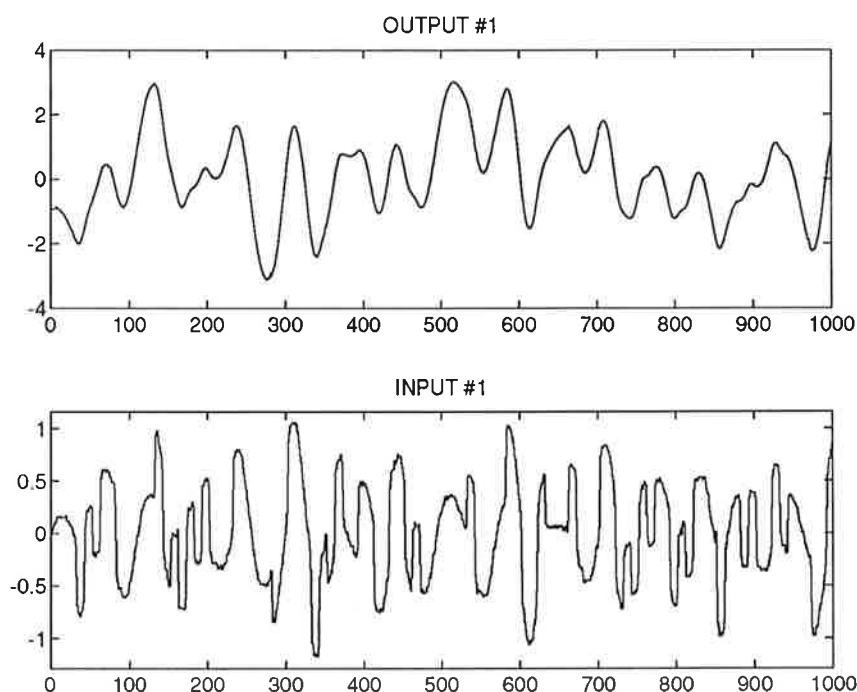
För att kunna utföra identifieringen gjorde vi en uppkoppling av processen där den inre vinkelprocessen reglerades med en P-regulator, och den yttre kulprocessen, reglerades med en PID-regulator 4. Detta förfaringssätt är ett måste eftersom processen är instabil.



Figur 4: Uppkoppling för att identifiera lägesprocessen

Både P- och PID-regulatorn implementerades i RealtidsSimnon och efter ganska mycket experimenterande med regulatorparametrarna lyckades vi få ett system som klarade att kontrollera kulan även då vi anslöt en PRBS-signal som insignal. Med uppkopplingen enligt figur 4 är det nu möjligt att identifiering överföringsfunktionen mellan  $\phi$  och  $x$ . Ett fel finns dock inbyggt i denna typ av identifiering. Identifieringsalgoritmerna bygger på ZOH-sampling vilket ger upphov till ett fel eftersom våra insignalerna är kontinuerliga vilket kan ses i figur 5. Detta skulle man kunna motverka genom att t.ex. använda metoder baserade på first-order-hold-sampling men vi valde att bortse från felet och använda de vanliga metoderna. PRBS-signalen som användes som insignal hade följande värden:

- Samplingstid 50 ms
- PRBS-period 10 sampel



Figur 5: Del av insignal- och utsignaldata som användes vid identifieringen av lägesprocessen

- Amplitud 2 V
- Offset -0.015 V

Den långa PRBS-perioden var nödvändig eftersom vi ville få en bra excitation även vid låga frekvenser beroende på att lägesprocessen är betydligt långsammare än vinkelprocessen. Vi sparade även nu 6000 mätvärden och använde 3000 för identifiering och 3000 för validering. Vid identifieringen använde vi "Akai-kes final prediction error", residualernas korrelation, pol-nollställediagram samt genom att simulera modellens utsignal med de mätdata som inte användes vid identifieringen och jämföra utsignalen med den verkliga. Vi provade att anpassa ARX-modeller, ARMAX-modeller och Box-Jenkins-modeller till våra data. Efter åtskilliga modelljämförelser hade vi två modeller som vi tyckte beskrev systemet ungefär lika bra, en ARMAX-modell och en Box-Jenkins-modell. Eftersom Box-Jenkins-modellens simulerade utsignal följde den verkliga lite bättre och båda modellerna var av samma ordning så valde vi Box-Jenkins-modellen:

$$y(t) = \frac{B(q)}{F(q)}u(t-k) + \frac{C(q)}{D(q)}e(t)$$

$$[\text{degB degC degD degF } k] = [1 \ 2 \ 3 \ 2 \ 1]$$

### 3.3.1 Identifiering av lägesprocessen med tidskontinuerlig metod

Då vi bestämde den teoretiska överföringsfunktionen mellan bommens vinkel och kulans läge i kapitel 2 fick vi den till

$$G_x(s) = \frac{c}{s^2} \quad (2)$$

med  $c \approx -5g/7$ . Genom tidskontinuerlig identifiering kan man bestämma faktorn  $c$ . För att göra detta inför man operatoren  $\lambda$  definierad som

$$\lambda = \frac{a}{s+a} \rightarrow s = \frac{a(1-\lambda)}{\lambda}$$

Sätter man in detta i ekvation 2 fås

$$\begin{aligned} G(\lambda) &= \frac{c}{\frac{a^2(1-\lambda)^2}{\lambda^2}} = \frac{c\lambda^2}{a^2(1-2\lambda+\lambda^2)} \\ \Rightarrow x(t) - 2[\lambda x(t)] + [\lambda^2 x(t)] &= \frac{c}{a^2}[\lambda^2 \phi(t)] \\ \Rightarrow x_f(t) &= \frac{1}{a^2}[\lambda^2 \phi(t)]c = \varphi^T \theta \\ \hat{c} &= [\text{Min.kvadrat}] = (\Phi^T \Phi)^{-1}(\Phi^T X_f) \end{aligned}$$

med

$$X_f = \begin{pmatrix} x_f(1) \\ \vdots \\ x_f(N) \end{pmatrix}$$

$$\Phi = \frac{1}{a^2} \begin{pmatrix} \lambda^2 \phi(1) \\ \vdots \\ \lambda^2 \phi(N) \end{pmatrix}$$

Filtreringen gjordes i diskret tid med snabb sampling och vi valde filterkonstanten  $a$  till 1. Valet av  $a$  baserade vi på den empiriska bodeplotten (överföringsfunktionen beräknad av MATLAB baserad direkt på in- och utdata) som var störd av brus för frekvenser högre än 1 Hz. Vi fick på detta sätt  $\hat{c} = -8,6$ . För att kunna jämföra värdet på konstanten med det teoretiska var vi tvungna att räkna om in- och utsignalerna innan identifieringen för att ta hänsyn till skalfaktorerna i uppkopplingen (vinkeln räknas om från Volt till rad och läget från Volt till meter). Då vi gjorde detta blev  $c$  ungefär lika med -5.3 vilket kan jämföras med det teoretiska värdet -7,0. Anledningar till skillnaden kan vara felaktiga skalfaktorer, påverkan vid samplingen av signalen eller för stora approximationer i den teoretiska modellen.

## 4 Validering

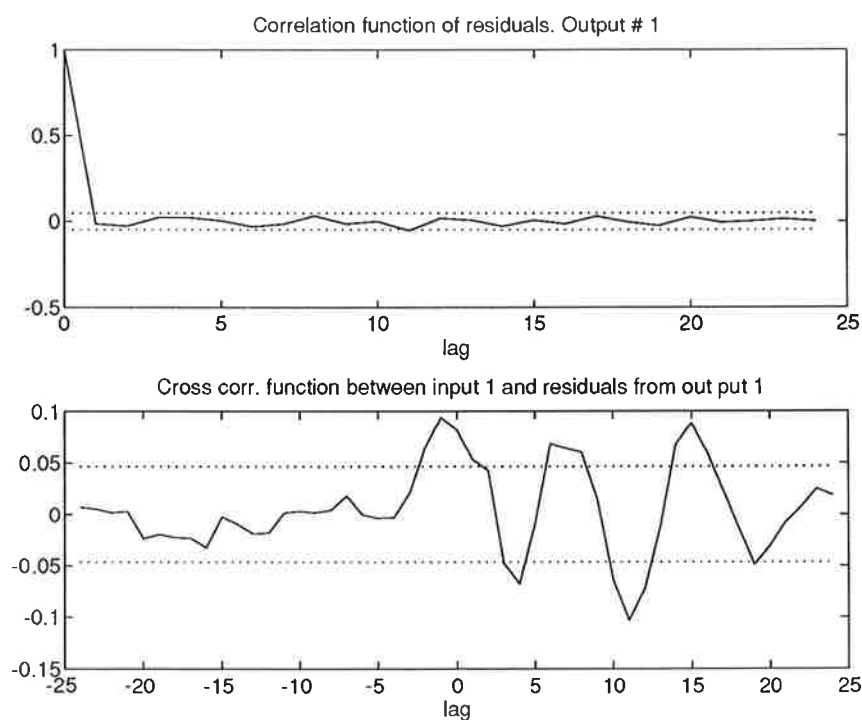
### 4.1 Inledning

För att validera våra modeller kontrollerade vi följande:

- För en perfekt modell bör residualernas autokorrelationsfunktion vara ett för "lag" noll och noll för övrigt.
- Korrelationsfunktionen mellan insignal och residualer bör även den vara 0
- Pol-nollställe-diagram
- Bodediagrammets utseende
- Överensstämmelsen mellan simulerad och verklig utsignal

### 4.2 Validering av vinkelmodellen

Residualernas autokorrelation och korrelationsfunktionen mellan residualer och insignal framgår av figur 6. Som framgår av figur 6 är residualernas autokor-

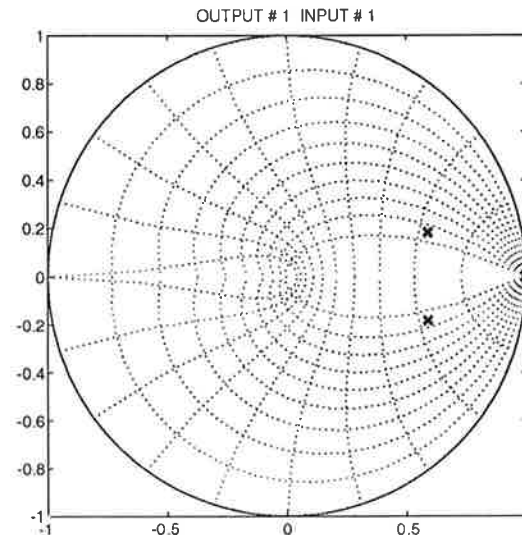


Figur 6: Residualernas autokorrelation och korrelationsfunktionen mellan residualer och insignal

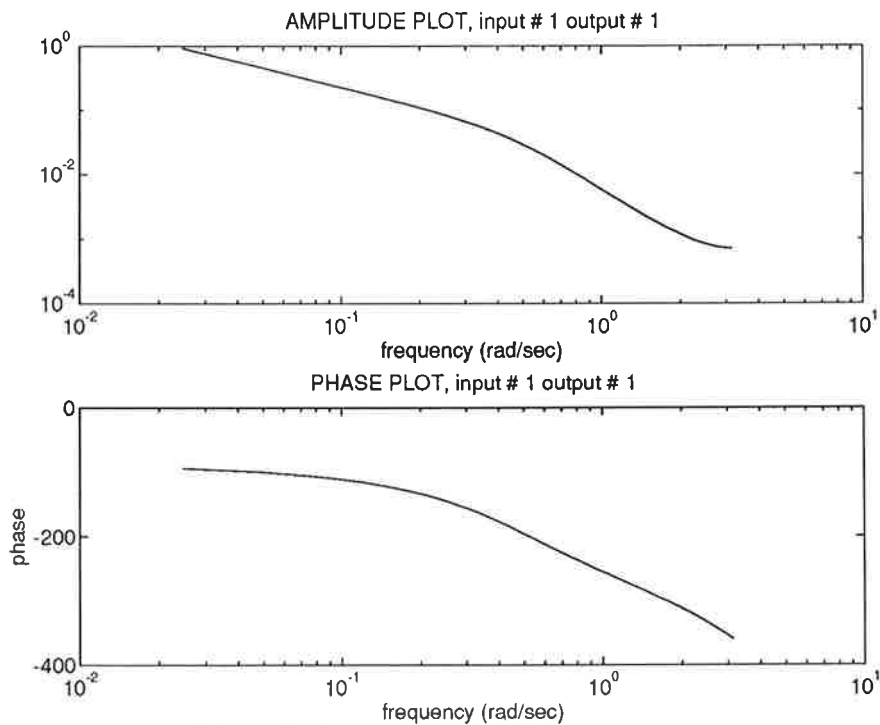
relation bra medan korrelationsfunktionen mellan residualer och insignal inte är

så bra för "lag" större än noll. Eftersom kurvan inte fick bättre utseende för modeller av högre ordning och eftersom våra resultat i övrigt är bra valde vi att bortse från detta.

Modellens pol-nollställe-diagram och bodediagram finns i figur 7 och 8. Pol-



Figur 7: Pol-nollställediagram för vinkelprocessen

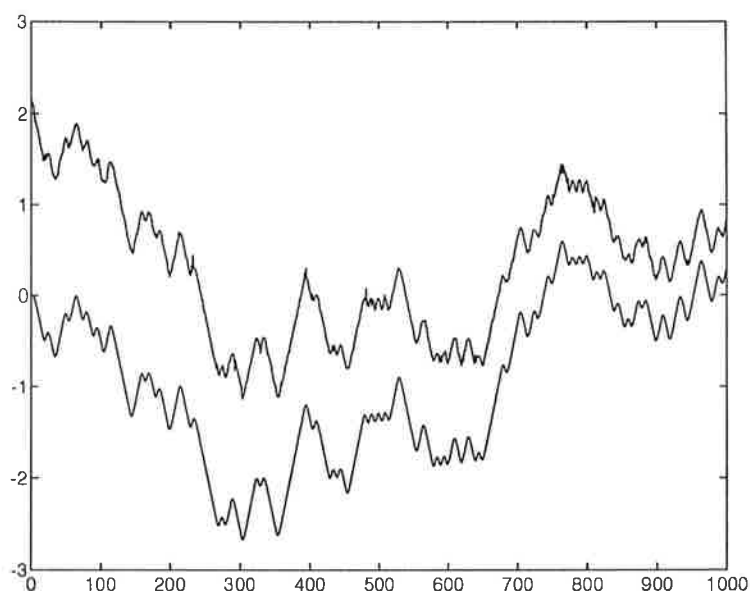


Figur 8: Bodediagram för vinkelprocessen

nollställediagrammet har det utseende som vi väntade oss efter att ha räknat

fram den teoretiska modellen. Vi har en pol på enhetscirkeln vilken svarar mot integratorn som gav oss lutningen -1 vid frekvensanalysen. De två övriga polerna ligger symmetriskt kring reella axeln. De ligger dock ganska långt från enhetscirkeln. Dessa båda iakttagelser tyder på ett bodediagram med lutning -1 för låga frekvenser och -3 för höga och en liten resonanstopp vid brytfrekvensen. Vi hade väntat oss en något mera markant resonanstopp än den vi har i figur 8 eftersom en sådan framkom vid frekvensanalysen, men en sådan uteblev i alla modeller vi provade.

Den valideringsmetod som vi fäste störst uppmärksamhet vid var jämförelsen mellan utsignalen som vi fick då vi simulerade modellens utsignal med de mätdata som inte användes vid identifieringen och jämförde den simulerade utsignalen med den verkliga. Resultatet av jämförelsen syns i figur 9. Som framgår av figur 9



Figur 9: Jämförelse mellan simulerad (undre kurvan) och verklig (övre kurvan) utsignal

följer den simulerade utsignalen den verkliga mycket bra. Den enda skillnaden är en förskjutning i höjdlid vilken dock lätt tas bort vid en reglering genom att införa integralverkan.

#### 4.2.1 Slutlig modell för vinkelprocessen

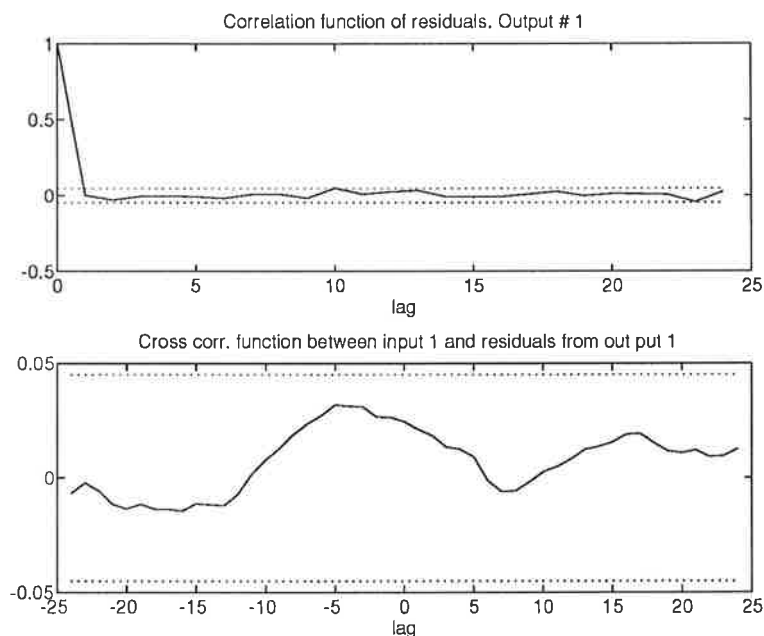
Den modell vi valde var alltså en Box-Jenkins-modell av tredje ordningen med överföringsfunktionen (mellan motorns insignal och bommens vinkel):

$$H(q) = \frac{0.0050q}{q^3 - 2.1811q^2 + 1.5629q - 0.3818}, \quad tsamp = 5ms$$

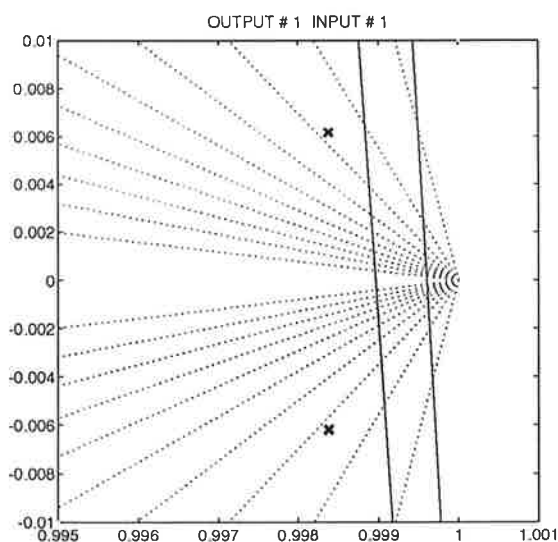


### 4.3 Validering av lägesmodellen

Den modell vi fått fram för överföringsfunktionen mellan bommens vinkel och kulans läge validerade vi på samma sätt som vinkelmodellen. Som framgår av



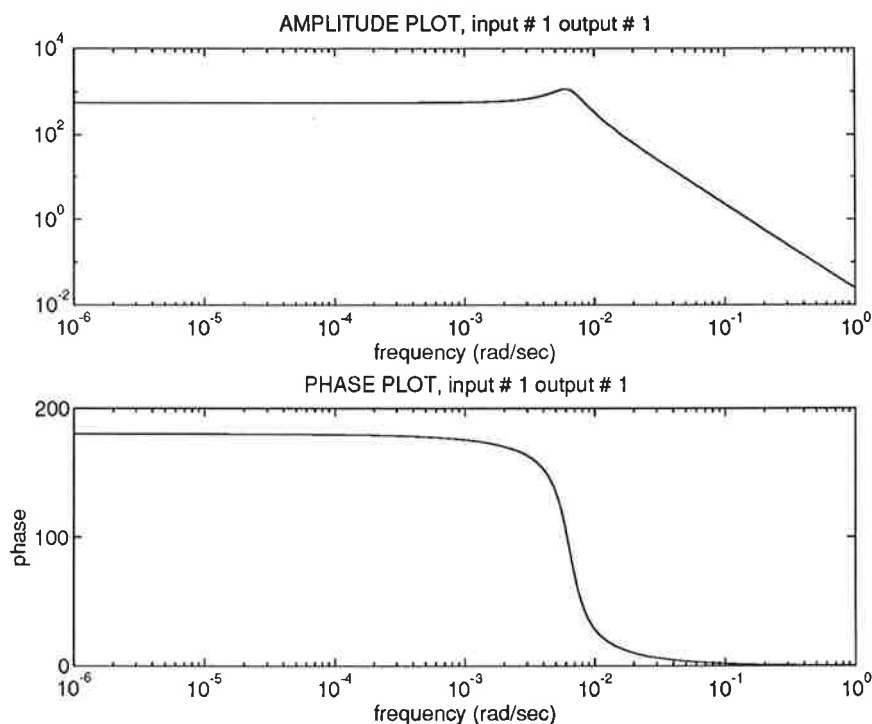
Figur 10: Residualernas autokorrelation och korrelationen mellan residualer och insignal



Figur 11: Pol-nollställediagram för lägesmodellen (De två vertikala strecken beror på att MATLAB ej klarar av att rita enhetscirkeln rätt)

figur 10 är både residualernas autokorrelation och korrelationen mellan residua-

ler och insignal mycket bra. Tittar man på modellens pol-nollställediagram ser det vid en första anblick ut som om båda polerna ligger på enhetscirkeln men vid en närmare granskning framgår det att de ligger komplexkonjugerade precis innanför enhetscirkeln enligt figur 11. Utseendet på pol-nollställediagrammet ger bodediagrammet enligt figur 12. Modellen är alltså en ren dubbelintegrator



Figur 12: Bodediagram för lägesmodellen

förutom för mycket låga frekvenser. Anledningen till att den inte är en dubbelintegrator för alla frekvenser tror vi kan bero på att PRBS-signalen inte exciterade processen tillräckligt för dessa frekvenser eller på att avståndet mellan motorns axel och bommen spelar in.

Resultatet av jämförelsen mellan simulerad och verklig utsignal framgår av figur 13. Figuren visar att modellen klarar av att följa alla snabba förlopp mycket bra. Signalen driver dock lite vilket tyder på att vår modell inte klarar av att följa väldigt långsamma förändringar. Detta kan dock lätt regleras bort genom att införa integralverkan i regulatorn.

#### 4.3.1 Slutlig modell för lägesprocessen

Den modell vi valde var alltså en Box-Jenkins-modell av andra ordningen med överföringsfunktionen (mellan bommens vinkel och kulans läge):

$$H(q) = \frac{-0.0228q}{q^2 - 1.9968q + 0.9968}, \quad tsamp = 50ms$$

Identifiering & reglering  
av  
inverterad pendel  
Projekt i Processidentifiering

Martin Nilsson, D90  
och  
Anders Lyngsjö, E90  
Institutionen för Reglerteknik

2 maj 1994

**Sammanfattning**

Vi har identifierat en inverterad pendel och skrivit en regulator i simuleringsspråket SIMNON. Regulatorn ser till så att pendeln står upp utan att den arm som pendeln är monterad på snurrar runt för mycket. I programmet finns även kod för att kasta upp pendeln då programmet startas.

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>3</b>
1.1	Syfte . . . . .	3
1.2	Avgränsningar . . . . .	3
1.3	Metod . . . . .	3
<b>2</b>	<b>Modellering</b>	<b>4</b>
2.1	Pendel nedåt . . . . .	4
2.2	Pendel uppåt . . . . .	5
<b>3</b>	<b>Identifiering</b>	<b>5</b>
3.1	Pendel . . . . .	5
3.1.1	Sinussignal (frekvensanalys) . . . . .	5
3.1.2	PRBS-signal . . . . .	6
3.2	Arm . . . . .	6
<b>4</b>	<b>Validering</b>	<b>7</b>
<b>5</b>	<b>Reglering</b>	<b>7</b>
5.1	Pendel . . . . .	7
5.2	Arm . . . . .	8
<b>6</b>	<b>Hur pendeln kastas upp</b>	<b>8</b>
<b>7</b>	<b>Resultat och slutsatser</b>	<b>8</b>
<b>A</b>	<b>SIMNON-kod</b>	<b>10</b>

# 1 Inledning

Vårt projekt gick ut på att identifiera och reglera en inverterad pendel. Då en inverterad pendel är en instabil process går det inte att bara identifiera den sådär utan vidare. Ett sätt är att hålla den i uppåtläget med en dålig regulator och identifiera den där. Anledningen till att regulatorn inte får vara för bra är att den i så fall kompenserar direkt så fort någonting händer. På så vis kommer inte pendelns dynamik att märkas, utan det blir i stället regulatorn som identifieras! Problemet är att hitta en "tillräckligt dålig" regulator som ändå lyckas hålla pendeln uppåt.

En bättre metod är att identifiera pendeln i nedåtläge och sedan räkna ut vad som skiljer mellan uppåtläget och nedåtläget. Vi valde denna metod tack vare att identifieringen på det här sättet blir betydligt enklare.

När vi sedan lyckats hitta en regulator som kunde balansera pendeln rakt upp, visade det sig att armen väldigt lätt började snurra runt. Därför valde vi att även mäta armens vinkel för att sedan kunna reglera denna. Regleringen av armvinkeln fixade vi med hjälp av en yttre långsammare reglerloop.

Slutligen behövde vi ett sätt att kunna kasta upp pendeln på. Detta för att slippa att hålla i den när man skall starta regulatorn. Anledningen till att man inte kan starta regulatorn direkt, är att den bara fungerar för små vinklar på pendeln och dessutom bara i det övre halvplanet.

## 1.1 Syfte

Vårt syfte med projektet är att få en praktisk inblick i konstruktionsarbete och projektering i den annars så teoretiska skola vi går i och samtidigt lära oss vad man kan använda reglertekniken till.

## 1.2 Avgränsningar

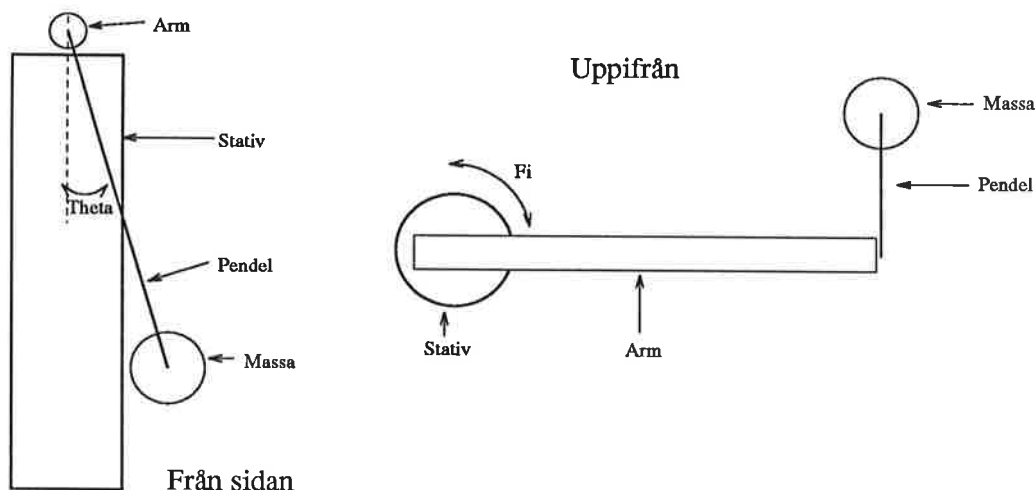
För att få ett mer väldefinierat problem har vi infört två tydliga begränsningar. För det första kan pendelns vinkel bara regleras inom ett väldigt snävt område kring uppåtläget. Dessutom fungerar bara uppkastningsmekanismen på en stillastående pendel.

## 1.3 Metod

Vi använde oss av de metoder som vi lärt oss i kursen Processidentifiering. I grova drag kan arbetet beskrivas som modellering, identifiering, validering och reglering. Flera av stegen är inte entydiga utan kräver att man provar sig fram till den lösning man tycker är bäst.

## 2 Modellering

Innan man börjar med själva identifieringen är det oftast en god idé att studera hur processen ser ut i verkligheten. På så vis kan man skapa sig en matematisk modell för sin process. I vårt fall bestod processen av en inverterad pendel som satt fast i en roterande arm. Den schematiska uppbyggnaden finns uppritad i figur 1.



Figur 1: Pendelns utseende i vy från sidan och uppifrån.

För att ta fram en matematisk modell använde vi oss av *Lagrange metod*. Denna går ut på att man sätter upp uttryck för både den kinetiska energin ( $T$ ) och den potentiella energin ( $U$ ). Därefter bildar man skillnaden

$$L = T - U$$

och utför ett antal derivationer med avseende på tiden och vinklarna  $\varphi$  och  $\theta$ .

Till att börja med gjorde vi en modell för hur pendeln fungerar hängande nedåt. Detta på grund av att pendeln då är stabil och därmed lätt att identifiera. Därefter modellerade vi pendeln i läge uppåt, för att se om det fanns ett direkt samband mellan dessa två lägen.

### 2.1 Pendel nedåt

Med beteckningar enligt figur 1 fick vi fram nedanstående två olinjära ekvationer då pendeln hänger ned. All friktion har försumrats.

$$\begin{cases} \frac{L^2 M}{4} \ddot{\varphi} + L^2 m \ddot{\varphi} - mLl\dot{\theta}^2 \sin \theta + mLl\ddot{\theta} \cos \theta = \tau \\ mLl\ddot{\varphi} \cos \theta + ml^2 \ddot{\theta} + mgl \sin \theta = 0 \end{cases}$$

$M$  anger här armens massa och  $m$  pendelns massa. Variabeln  $\tau$  är det yttre pålagda momentet i  $\varphi$ -led, det vill säga det moment som vill vrida armen runt sin axel.

För att praktiskt kunna använda formlerna linjariserade vi kring vinkeln  $\theta = 0$ . Vid denna vinkel gäller det att  $\sin \theta \approx \theta$  och  $\cos \theta \approx 1$ . Dessutom antog vi att  $\dot{\theta}^2 \approx 0$ . På detta sätt fick vi två ekvationer som är linjära i  $\ddot{\varphi}$ ,  $\ddot{\theta}$  och  $\theta$ .

$$\begin{cases} (\frac{L^2 M}{4} + L^2 m) \ddot{\varphi} + mLl\ddot{\theta} = \tau \\ mLl\ddot{\varphi} + ml^2 \ddot{\theta} + mgl\theta = 0 \end{cases}$$

Med hjälp av de linjariserade ekvationerna kunde vi nu ta fram överföringsfunktionen för systemet. Utsignalen var given som pendelns vinkel  $\theta$ . Det var denna vinkel som sedan skulle regleras. Som insignal till processen använde vi momentet  $\tau$ . Anledningen till detta är att en likströmsmotor ger ett moment som är direkt proportionellt mot den pålagda spänningen.

Genom att eliminera  $\ddot{\varphi}$  i ekvationen ovan fick vi fram sambandet mellan  $\tau$  och  $\theta$ . Uttryckt i den komplexa frekvensen  $s$  blev sambandet

$$\theta(s) = -\frac{b}{s^2 + a}\tau(s)$$

där  $a$  och  $b$  är konstanter. Polerna är rent imaginära, vilket motsvarar en odämpad svängningsrörelse.

## 2.2 Pendel uppåt

Under denna rubrik följer resultatet från Lagrange metod då pendeln är i läge uppåt. Skillnaden mot föregående beräkningar är endast tecknet hos några av termerna. De olinjära ekvationerna blir

$$\begin{cases} \frac{L^2 M}{4}\ddot{\varphi} + L^2 m\ddot{\varphi} + mLl\dot{\theta}^2 \sin \theta - mLl\ddot{\theta} \cos \theta = \tau \\ -mLl\ddot{\varphi} \cos \theta + ml^2\ddot{\theta} - mgl \sin \theta = 0 \end{cases}$$

Efter linjarisering kring vinkeln  $\theta = 0$  fick vi fram ekvationerna

$$\begin{cases} (\frac{L^2 M}{4} + L^2 m)\ddot{\varphi} - mLl\ddot{\theta} = \tau \\ -mLl\ddot{\varphi} + ml^2\ddot{\theta} - mgl\theta = 0 \end{cases}$$

Överföringsfunktionen från  $\tau$  till  $\theta$  blev nu

$$\theta(s) = \frac{b}{s^2 - a}\tau(s)$$

där  $a$  och  $b$  är samma konstanter som för pendeln i läge nedåt. Polerna är här rent reella. Den ena av polerna är positiv, vilket tyder på att processen är instabil.

Skillnaden mellan de båda överföringsfunktionerna är således tecknet och polplaceringen. De ingående konstanterna är dock samma i båda fallen. Därmed går det att omvandla en skattad modell i nedåtläget till en modell i uppåtläget. Den omvandlade modellen kan sedan användas för att räkna ut en regulator som kan balansera pendeln rakt upp.

## 3 Identifiering

### 3.1 Pendel

Identifieringen av pendeln utfördes som det tidigare antytts i inledningen i nedåtläge. Vi använde två olika metoder för identifieringen, dels excitering med en sinussignal (frekvensanalys) och dels med en PRBS-signal (Pseudo Random Binary Sequence).

#### 3.1.1 Sinussignal (frekvensanalys)

Identifieringen utfördes här med en frekvensanalysator, som mäter amplitud och fasförskjutning hos processen för olika frekvenser. Det uppmätta bodediagrammet visade tydliga tecken på olinjariteter hos vår process. Troligen beror detta på att friktionen inte kan försummas.

Trots detta kunde vi utläsa vissa egenskaper hos vår process. Till exempel fanns en resonanstopp vid 1 Hz, vilket motsvarar pendelns egenfrekvens. Dessutom var fasförskjutningen maximalt  $-180^\circ$ , vilket tyder på att det rör sig om ett andra ordningens system.

### 3.1.2 PRBS-signal

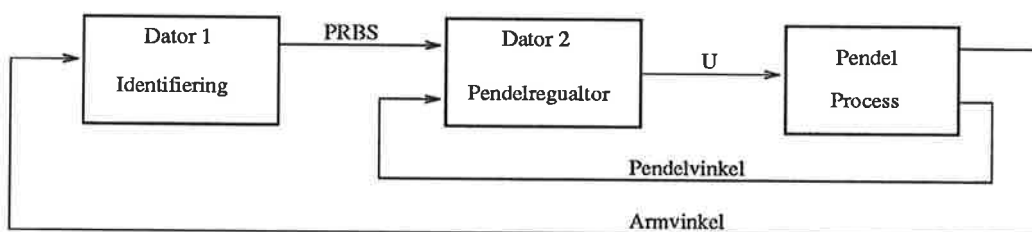
Vi skickade in en PRBS-signal till motorn samtidigt som vi loggade värdena på pendelvinkeln  $\theta$ . Det var viktigt att skicka in en lagom stark signal till motorn. Vid för stora utslag på  $\theta$  blir nämligen processen kraftigt olinjär, medan friktionen tar överhanden vid för små utslag. Dessutom ville vi reglera vinkeln kring noll och inte i något större intervall.

Mätvärdena laddades in i beräkningsprogrammet Matlab, där vi gjorde en rad ARMAX-skattningar (AutoRegressive Moving Average eXternal input model) av processen. Vi jämförde sedan modeller av olika ordning och studerade skattningarnas FPE (Final Prediction Error). Skillnaden mellan en tredje ordningens modell och en andra ordningens modell var inte särskilt stor. Tredje ordningens modell var visserligen något bättre, men skillnaden var inte tillräckligt stor. Dessutom var den matematiska modellen av andra ordningen, vilket gav oss en tydlig vink om vad vi borde välja. Därför valde vi slutligen nedanstående andra ordningens modell efter omräkning till uppåtläget.

$$\begin{aligned} A &= \begin{bmatrix} 1 & -2.1007 & 0.934 \end{bmatrix} \\ B &= \begin{bmatrix} -0.0112 & -0.0040 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & -0.0144 \end{bmatrix} \end{aligned}$$

## 3.2 Arm

Efter att ha bestämt oss för att använda kaskadreglering till att reglera armens vinkel  $\varphi$  kopplade vi ihop två datorer enligt figur 2. Vi körde regleringen av pendelns vinkel på den ena och identifieringen av sambandet mellan armens vinkel  $\varphi$  och styrsignalen till pendelvinkeln  $\theta_r$  på den andra. Datorerna var hopkopplade så att från DA-omvandlaren i identifieringsdatorn gick en sladd med styrsignalen till AD-omvandlaren i reglerdatorn. När vi utförde denna identifiering hade vi alltså redan hittat vilken regulator som var bäst för att reglera pendeln.



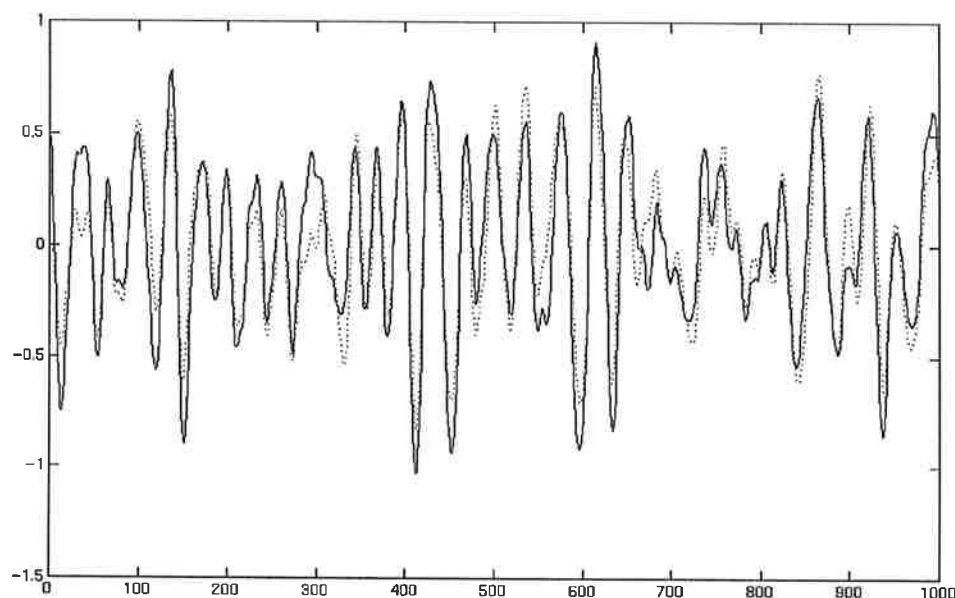
Figur 2: Blockschemat över identifieringen av armen.

Identifieringen utfördes även här med en PRBS-signal, som fick verka på systemet samtidigt som data samlades in. Vi använde sedan Matlab för att göra några ARMAX-skattningar på samma sätt som för pendeln. Arbetet försvårades av att givaren för armvinkeln inte fungerade som den skulle. Givaren verkade slira mot armen, vilket resulterade i att mätvärdet inte följde armens position. Den skattning vi trots detta tyckte var bäst var en andra ordningens modell med två poler och ett nollställe.



## 4 Validering

När en viss modell är vald är det dags för *validering*. Det innebär att man kontrollerar hur väl modellen stämmer överens med verkligheten. För detta ändamål använde vi data som inte använts vid identifieringen. På så vis testas modellen under mer verklighetstroga förhållanden.



Figur 3: Jämförelse mellan modell och verklighet. (Modellen prickad linje)

Resultatet av valideringen för pendeln visas i figur 3. Där finns två kurvor uppritade som beskriver vinkelutslaget hos pendeln. Den heldragna visar den verkliga, samplade signalen från pendeln, medan den prickade är en simulering. Vid simuleringen använde vi samma PRBS-signal som vi använt vid identifieringen. Överensstämmelsen är förvånansvärt bra!

## 5 Reglering

Regleringen av både pendel och arm görs med digitala regulatorer. Beräkningarna av dessa har gjorts med Matlab och är av typen *R-S-T regulatorer*. Dessa byggs upp av de tre polynomen  $R$ ,  $S$  och  $T$  enligt nedanstående differensekvation.

$$R(q^{-1})u(k) = T(q^{-1})u_c(k) - S(q^{-1})y(k)$$

Variabeln  $q^{-1}$  står för en tidsfördröjning av ett sampelintervall. Signalen  $u(k)$  är styrsignal,  $u_c(k)$  är börvärde och  $y(k)$  är systemets utsignal.

För den som är väl förtrogen med simuleringsspråket SIMNON, finns koden för våra regulatorer i appendix A.

### 5.1 Pendel

Pendeln regleras med en integrerande tredje ordningens R-S-T regulator, där polerna placerats enligt nedan.

$$\begin{cases} \omega_m = 18 \\ \varphi = 45^\circ \end{cases}$$

Variabeln  $\omega_m$  är polernas avstånd till origo och  $\varphi$  är halva öppningsvinkeln i ett Butterworthfilter.

## 5.2 Arm

Armens regulator är av andra ordningen och utan integrator. Polerna har placerats enligt nedanstående data.

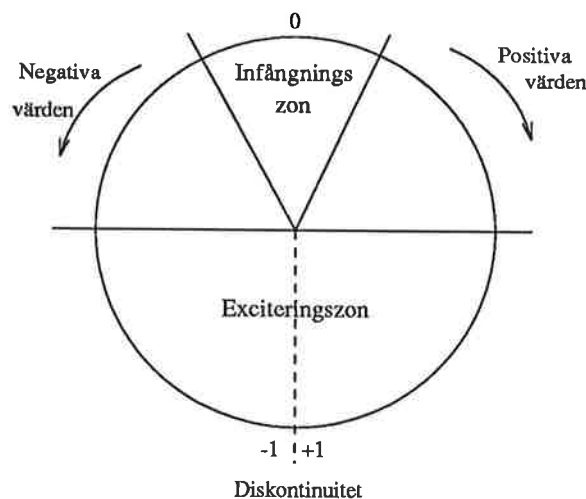
$$\begin{cases} \omega_m = 2.5 \\ \varphi = 10^\circ \end{cases}$$

Notera att armens regulator är betydligt långsammare än pendelns. Detta är normalt vid kaskadreglering.

## 6 Hur pendeln kastas upp

Experiment med processen avslöjade att det var omöjligt att kasta upp pendeln genom att ge armen maximal fart i ena riktningen. För att få upp pendeln i övre halvplanet krävdes det att man gungade den fram och tillbaka några gånger. Att bara excitera pendeln så pass att den kom upp var inte så svårt. Däremot att excitera den lagom så att den hade låg hastighet när den passerade uppåtläget var betydligt svårare.

En metod för att excitera pendeln är att skicka in en sinussignal till motorn med samma frekvens som pendelns egenfrekvens, och sedan koppla in regulatorn när pendeln är nästan rakt upp. Denna metod fungerar säkert men vi valde istället att använda en lite mer adaptiv metod.



Figur 4: Pendelns vinkel indelad i de olika zoner som uppkastningsalgoritmen använder. I infångningszonen är regulatorn inkopplad.

Vår idé fungerar på samma sätt som när barn gungar på lekplatser. Barnet (och gungan) gungar av sig själva medan en vuxen tillför kraft till gungan genom att

putta på den. Man puttar på åt samma håll som gungan rör sig och man puttar med olika kraft beroende på var gungan är.

Vi har delat in den cirkel som vinkeln  $\theta$  genomlöper i olika zoner enligt figur 4. ( $\theta$  antar värdena -1 till +1 med 0 rakt upp och -1 och +1 rakt ner). Längst upp finns en infångningszon där regulatören är inkopplad. Denna zon sträcker sig från -0.2 till +0.2. När pendeln är i denna zon försöker alltså regulatören att få den att stå rakt upp. Det nedre halvplanet används till att excitera pendeln tills dess att den får energi nog att komma in i infångningszonen.

Vi räknar ut åt vilket håll pendeln är på väg och ger den mer kraft i den riktningen med motorn. Spänningen till motorn bestäms av var pendeln är någons. Är den rakt ner är spänningen maximal och är den nära övre halvplanet är spänningen minimal. Spänningen räknas ut som  $u = \sin((|\theta| - 0.5) \cdot \pi)$ . När pendeln är i övre halvplanet men utanför infångningszonen så tillförs ingen kraft. Detta för att bromsa den så att den inte skall ha för hög fart när den kommer in i reglerområdet.

## 7 Resultat och slutsatser

Avslutningsvis kan vi konstatera att regleringen av pendeln i uppåtläge fungerar bra. Däremot står armen inte helt stilla utan vrider sig lite fram och tillbaka. Detta beror troligen på att vinkelgivaren för armen inte verkar fungera riktigt som den skall.

Uppkastningen av pendeln fungerar för det mesta när pendeln är stilla. Om pendeln rör sig för mycket spårar uppkastningsrutinen ur och armen roterar okontrollerat.

Regleringen av pendeln klarar till och med av ganska stora laststörningar utan att tappa pendeln.

## Referenser

- [1] Rolf Johansson. *System Modeling and Identification*. Prentice Hall, 1993.
- [2] Karl Johan Åström & Björn Wittenmark. *Computer Controlled Systems 2ed*. Prentice Hall, 1990.

## A SIMNON-kod

Discrete System rstreg1

```
input  y fi fir
output u
state  u1 u2 y1 y2 y3 yr1 yr2 fi1 fir1
new    nu1 nu2 ny1 ny2 ny3 nyr1 nyr2 nf11 nfir1

time  t
tsamp ts

" Yttre regulator för armens vinkel
yr = -(r21*yr1)-(s20*fi+s21*fi1)+(t20*fir+t21*fir1)
yrk = IF yr<-0.03 THEN -0.03 ELSE IF yr>0.03 THEN 0.03 ELSE yr

" Inre regulator för pendelns vinkel
v = -(r1*u1+r2*u2)-(s0*y+s1*y1+s2*y2)+(t0*yrk+t1*yr1+t2*yr2)
ureg=IF v<-1 THEN -1 ELSE IF v>1 THEN 1 ELSE v

" Rutin för att kasta upp pendeln
q = y - y3
a = IF (y>0 AND y3>0) OR (y<0 AND y3<0) THEN q ELSE -q
b = sin((abs(y)-0.5)*3.1415)
w = IF a>0 THEN b ELSE -b
ugung = IF abs( y ) > 0.55 THEN w ELSE 0

" Val av styrsignal (reglering för infångningsområdet -0.2<y<0.2)
u = IF abs(y) < 0.2 THEN ureg ELSE ugung

" Uppdatering av tillstånd
nu1 = u
nu2 = u1

ny1 = y
ny2 = y1
ny3 = y2

nyr1 = yrk
nyr2 = yr1

nf11 = fi
nfir1 = fir

ts = t+h

h : 0.02

" Parametrar för pendelregulatorn
r1 : -1.105
r2 : 0.1051

s0 : -40.6
s1 : 68.18
```

```
s2 : -29.63

t0 : -6.591
t1 : 6.918
t2 : -2.381

" Parametrar för armregulatorn
r21 : -0.9611

s20 : -1.7704
s21 : 1.6977

t20 : -0.8868
t21 : 0.8024

END
```

# Identifiering och Reglering av Bom och Kula

Mats Ahnelöv D-90  
Michael Rosenberg D-90  
Carl-Johan Weiderstrand D-90

26 april 1994

# Innehåll

<b>1 Inledning</b>	<b>3</b>
1.1 Bakgrund och Problembeskrivning . . . . .	3
1.2 Uppläggning av Projektet . . . . .	3
<b>2 Modellbygge</b>	<b>3</b>
2.1 Inledande Modellbygge . . . . .	3
2.2 Matematiskt Modellbygge . . . . .	4
<b>3 Identifiering av Bomprocessen</b>	<b>4</b>
3.1 Frekvenssvarsmetoden . . . . .	4
3.2 Tidsserieanalys . . . . .	6
3.3 Validering av Bommodellen . . . . .	8
<b>4 Identifiering av Kulprocessen</b>	<b>8</b>
4.1 Tidsserieanalys . . . . .	8
4.2 Validering av Kulmodellen . . . . .	10
4.3 Skattning av Tidkontinuerlig Parameter $J$ . . . . .	10
<b>5 Reglering</b>	<b>12</b>
5.1 Inledning . . . . .	12
5.2 Design av $RST$ -regulatorn . . . . .	12
5.3 Det Slutna Systemets Beteende . . . . .	13

# 1 Inledning

## 1.1 Bakgrund och Problembeskrivning

I kursen *Processidentifiering* som ges av Institutionen för Reglerteknik vid Lunds Tekniska Högskola ingår ett projekt, vars syfte är att ge praktisk träning på de teorier som lärts ut på föreläsningarna. Projektet utförs i grupper om tre teknologer, och vår process bestod av en DC-motor som driver en bom. På bommen ligger en kula vars läge kan mätas. Uppgiften var att identifiera dynamiken dels mellan inspänningen till DC-motorn och vinkeln hos bommen, dels att identifiera dynamiken mellan vinkeln på bommen och kulans läge. Med den gjorda identifieringen skulle sedan en regulator konstrueras som reglerade kulans läge.

## 1.2 Uppläggning av Projektet

Vi valde att lägga upp arbetet med vårt projekt enligt

- Modellering
- Identifiering av bomprocessen
  1. Frekvenssvarsanalys
  2. Undersökning av koherens
  3. Tidsserieanalys
  4. Validering av den erhållna modellen
- Identifiering av kulprocessen
  1. Undersökning av koherens
  2. Tidsserieanalys
  3. Validering av den erhållna modellen
  4. Ta fram den tidskontinuerliga parametern  $J$
- Konstruera regulator för hela systemet
- Utvärdering och rapportskrivning

# 2 Modellbygge

## 2.1 Inledande Modellbygge

Tidigare under kursen hade vi utfört en laboration på processen, så vi kände till en del av dynamiken hos bommen och DC-motorn. Själva motorn kan modelleras som

$$G_m(s) = \frac{1}{s(s+T)}$$

och resonansen hos bommen modelleras som ett andra ordningens system så överföringsfunktionen från inspänning  $u(t)$  till  $\varphi(t)$  ges av



$$G_{u\varphi}(s) = \frac{1}{s(s+T)(s^2 + 2\zeta\omega_0 s + \omega_0^2)}$$

där  $\zeta$  är dämpningen och  $\omega_0$  är resonansfrekvensen. Det finns förmodligen även resonanser vid högre frekvenser, men dessa försummas eftersom vi anser att vi inte behöver information om systemet vid så höga frekvenser.

Överföringsfunktionen från bommens vinkel  $\varphi(t)$  till läget  $x(t)$  hos kulan ges i stort av en dubbelintegrator

$$G_{\varphi x}(s) = \frac{K}{s^2}$$

och vi kan alltså förvänta oss ett totalt system av ordning sex.

Observera att processen i sig är olinjär och att uttrycken ovan endast gäller för små vinklar  $\varphi$  hos bommen.

## 2.2 Matematiskt Modellbygge

Efter det inledande modellbygget försökte vi skaffa oss en mer exakt modell. I [2] finns en härledning av den olinjära differentialekvation som beskriver sambandet mellan bommens vinkel  $\varphi$  och kulans läge  $x$ . Sambandet ges av

$$(1 + \alpha)\ddot{x} - x\dot{\varphi}^2 + d\ddot{\varphi} = -g \sin \varphi \quad (1)$$

där  $\alpha$  är en konstant och  $d$  är avståndet från motorns axel till bommen.

Under härledningen har det antagits att kulans massa är så liten att den inte påverkar läget på bommen. Det har också antagits att kulan hela tiden ligger mot bommen och att den aldrig släpper när den rullar.

För att linjärisera ekvation (1) antas att vinkeln  $\varphi$  och bommens acceleration är liten. Då fås följande linjära samband mellan bommens vinkel och kulans läge

$$\ddot{x} = -\frac{g}{1 + \alpha}\varphi \quad (2)$$

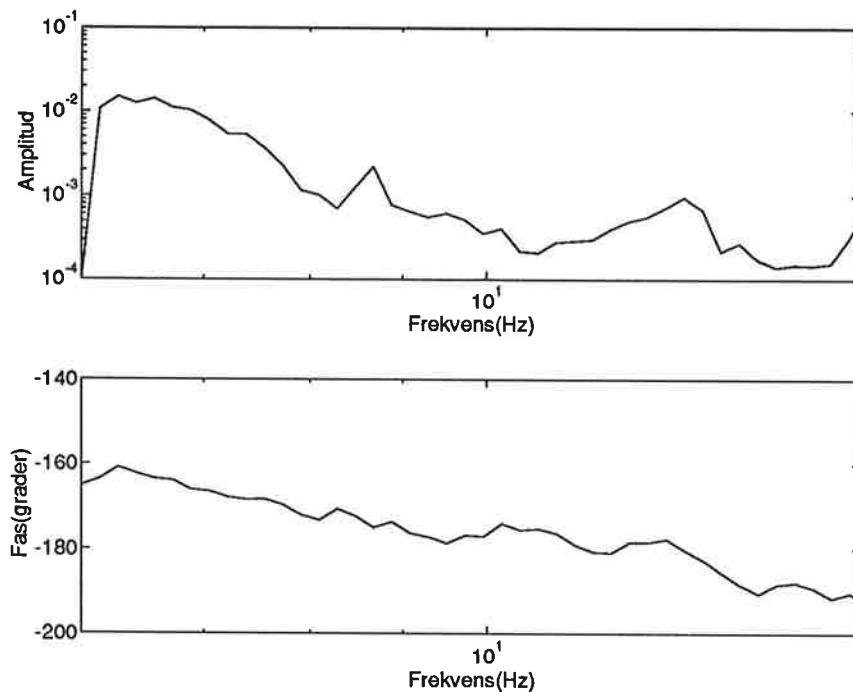
vilket är en dubbelintegrator, vilket antogs under det inledande modellbygget.

## 3 Identifiering av Bomprocessen

### 3.1 Frekvenssvarsmetoden

För att få en första uppskattning av dynamiken hos bommen, dvs överföringsfunktionen från inspänningen  $u(t)$  till vinkeln hos bommen  $\varphi(t)$ , användes *frekvenssvarsmetoden*. Den går ut på att man skickar in en sinusformad signal på systemet som skall undersökas. Då transienterna har dött ut mäts utsignalen  $y(t)$  och följande storheter beräknas

$$s_T(\omega) = \int_0^T y(t) \sin \omega t dt = \frac{T}{2} |G(j\omega)| u_1 \cos \phi(\omega)$$



Figur 1: Bodediagram för experiment 1.

$$c_T(\omega) = \int_0^T y(t) \cos \omega t dt = \frac{T}{2} |G(j\omega)| u_1 \sin \phi(\omega)$$

där mättiden  $T$  väljs som en heltalsmultipel av periodtiden hos insignalen  $u(t)$ , dvs  $T = \frac{\omega}{2\pi} k$  där  $k$  är ett heltal.

Skattningarna av amplitud och fas ges sedan av

$$|\hat{G}(j\omega)| = \frac{2}{Tu_1} \sqrt{s_T^2(\omega) + c_T^2(\omega)}$$

$$\hat{\phi}(\omega) = \arctan \frac{c_T(\omega)}{s_T(\omega)} + k\pi$$

Fördelarna med metoden är att den är okänslig mot brus (tack vare integrationen över det hela antalet perioder) och att man snabbt får en uppskattning av systemets dynamik. Metoden är dock icke-parametrisk, vilket innebär att informationen om koefficienterna i överföringsfunktionen är svårtillgänglig.

I det första experimentet svepte vi över ett ganska stort frekvensområde för att få en uppfattning om var den intressanta dynamiken fanns. Nästa steg blev sedan att svepa frekvensområdet 6-16 Hz noggrannare, och resultatet visas i Figur 1. I figuren syns svaga toppar vid ca 8 Hz och 12 Hz. Dessa toppar indikerar resonansfrekvenser hos systemet.

### 3.2 Tidsserieanalys

Som nästa identifieringsmetod användes *tidsserieanalys*. Metoden går ut på att man skickar in en stokastisk process  $u_k$  på systemet man vill identifiera. Genom att mäta utsignalen  $y_k$  kan man sedan bestämma (tidsdiskreta) modeller för systemet. Först testade vi olika typer av ARX- och ARMAX-modeller. Det visade sig dock att framför allt ARX-modellen inte räckte till för att skatta vårt system på ett bra sätt. Detta berodde på att bruset i vårt system förmodligen inte var vitt. ARMAX-modellen klarade sig betydligt bättre men vi valde ändå en Box-Jenkin modell, vilken har formen

$$y_k = \frac{B(q^{-1})}{F(q^{-1})} u_{k-d} + \frac{C(q^{-1})}{D(q^{-1})} w_k \quad (3)$$

där  $w_k$  är vitt brus,  $d$  är en tidsfördröjning och  $B, C, D$  och  $F$  är polynom som ges av

$$\begin{aligned} B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_{n_B} q^{-n_B} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_{n_C} q^{-n_C} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \dots + d_{n_D} q^{-n_D} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \dots + f_{n_F} q^{-n_F} \end{aligned}$$

Målet med metoden är att skatta de okända parametrarna i de olika polynomen.

För att utföra mätningar på vår process fanns ett datorprogram *Logger* till vårt förfogande. Programmet kan både skicka ut signaler till processen och sampla utsignaler från processen. Användaren kan t.ex. själv välja hur länge programmet skall mäta och vilken amplitud insignalen till processen skall ha. Se [4] för en utförligare beskrivning av datorprogrammet.

Som insignal  $u_k$  har vi valt en *pseudorandom binary sequence (PRBS)* som genererats av ett skiftregister med längden 8 i programmet. Detta ger en signal med period  $(2^8 - 1)h = 255h$ , där  $h$  är samplingsperioden. Med  $h = 5$  ms fås därför en period på 1275 ms, vilket vi ansåg vara för liten. I programmet fanns det dock möjlighet att hålla utsignalen från skiftregistret ett visst antal sampel, och vi valde att hålla den 5 sampel. Detta gav därför en period på 6.375 s.

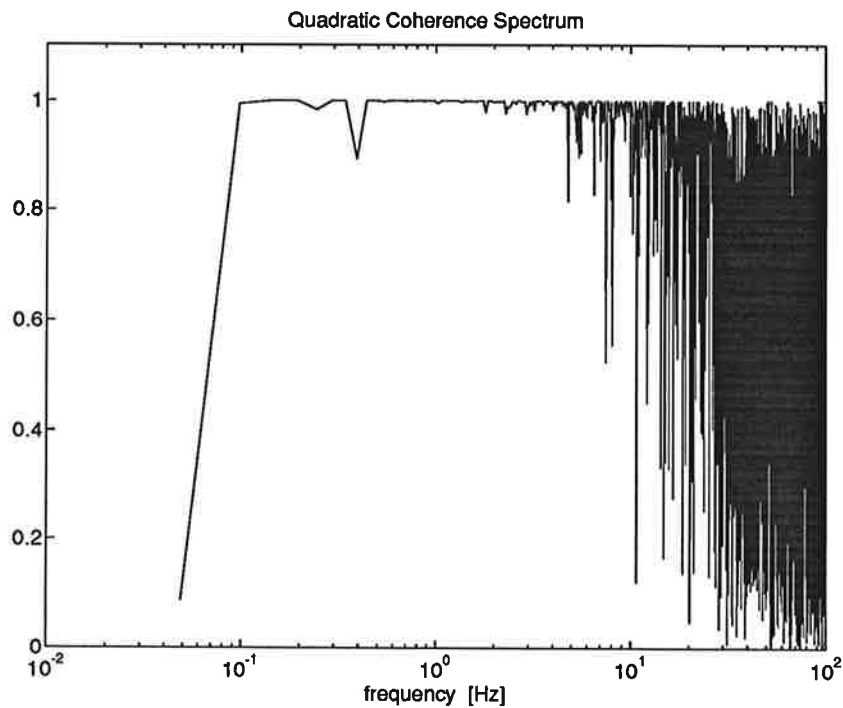
Då man håller utsignalen  $u_k$  från skiftregistret ett visst antal sampel så kommer spektraltätheten hos  $u_k$  att förändras. Från att ha varit approximativt vitt brus med en nästan konstant spektraltäthet, så får man en "sinc"-formad spektraltäthet. Med hjälp av MATLAB-funktionen *spectrum* kunde vi dock skatta  $u_k$ 's spektraltäthet och den var vit upp till ca 40 Hz, dvs i hela det intressanta frekvensområdet.

Med vårt val av  $u_k$  så har vi försäkrat oss om att vi exiterar systemet tillräckligt. Nästa steg blir då att studera *koherensen*, vilket gjordes i MATLAB med funktionen *spectrum*. Figur 2 visar vår erhållna koherens och man ser att upp till ca 20 Hz så kan vi förvänta oss en bra skattning av systemet. Det bör dock tilläggas att skattningen av koherensen verkar vara numeriskt besvärlig och att t.ex. val av fönsterlängd i *spectrum* påverkar den erhållna koherensen mycket.

För att modellera vårt system valde vi att använda en Box-Jenkin modell (3), eftersom några inledande försök med ARX- och ARMAX-modeller visade att dessa inte klarade av att beskriva bruset i systemet. Som resultat fick vi

```
This matrix was created by the command BJ      on 3/25 1994 at 11:22
Loss fcn: 0.0002175   Akaike's FPE: 0.0002182 Sampling interval 0.005
The polynomial coefficients and their standard deviations are
```

B =



Figur 2: Koherensdiagram för den identifieringen bommen.

0	0	0.0015
0	0	0.0001

F =

1.0000	-2.6121	2.9540	-2.0080	0.6661
0	0.0910	0.2169	0.1803	0.0566

C =

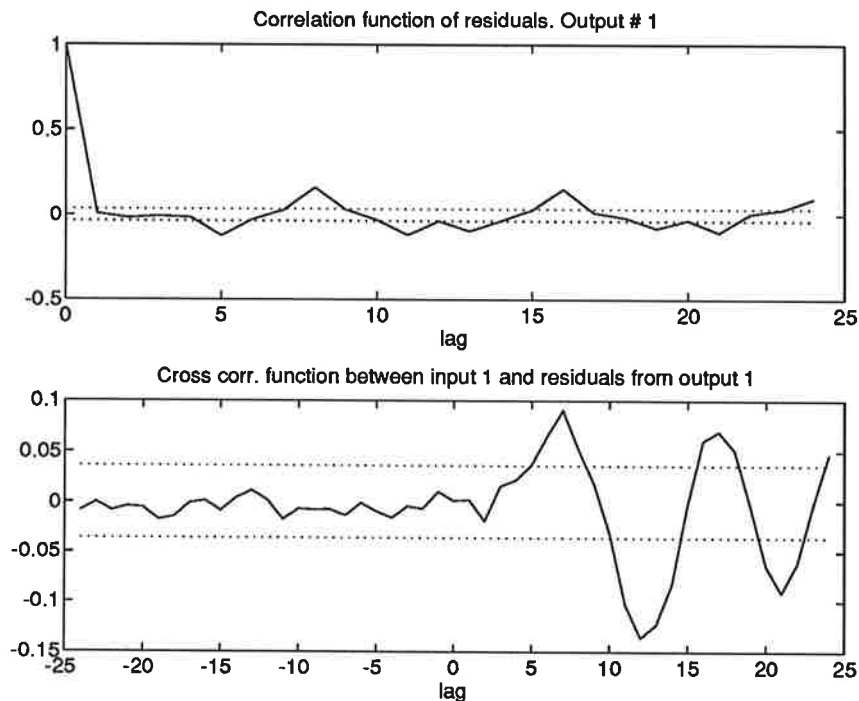
1.0000	0.1631	0.1612
0	0.0140	0.0140

D =

1.0000	-0.9998
0	0.0004

och därmed ges den tidsdiskreta överföringsfunktionen från inspänning till bommens vinkel av

$$H_{u\varphi}(q) = \frac{0.0015q^2}{q^4 - 2.6121q^3 + 2.9540q^2 - 2.0080q + 0.6661} \quad (4)$$



Figur 3: Korrelation hos residualerna.

med samplingsintervallet  $h = 5$  ms.

### 3.3 Validering av Bommodellen

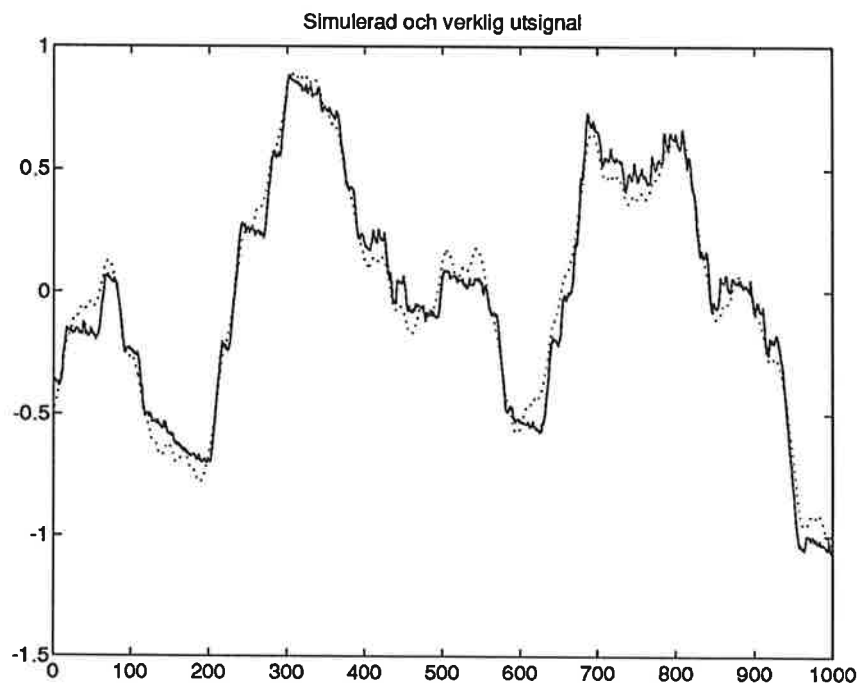
För att validera vår erhållna modell studerades först residualerna. Resultatet visas i Figur 3 och där syns att residualerna inte är helt vita. Vidare syns att de inte heller är okorrelerade med insignalen. Båda dessa resultat tyder på att det inte är den bästa modellen vi har hittat. Vi har även provat med ett antal andra modeller med liknande resultat, vilket vi tolkar som att detta är den bästa modell vi kan få fram.

Däremot visar en simulering av det skattade systemet, med tidigare oanvänd data som insignal, att det skattade systemet uppför sig som det verkliga. Simuleringen visas i Figur 4 och vi anser därför att vi funnit en god skattning av bomprocessen.

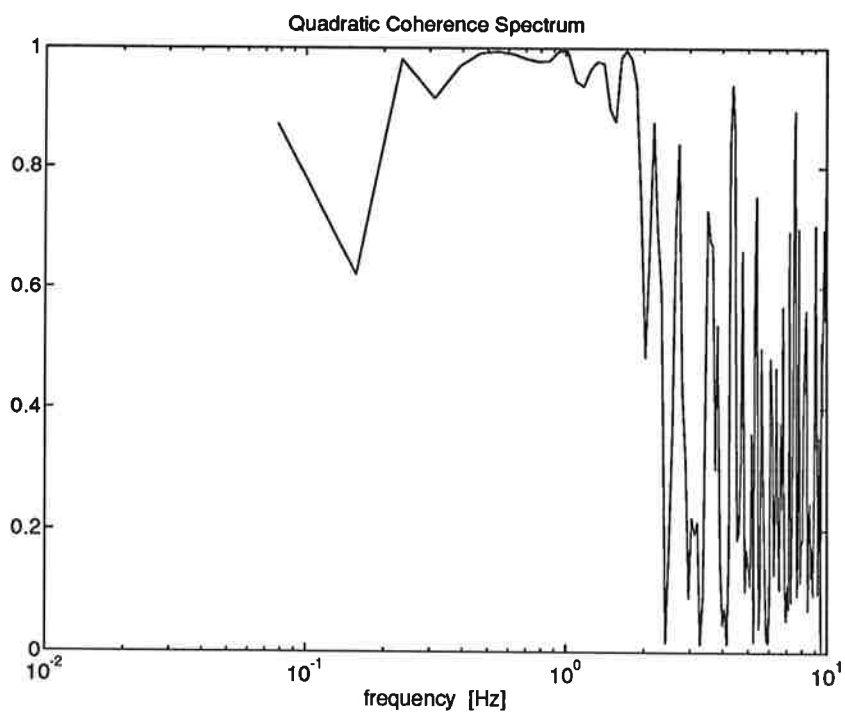
## 4 Identifiering av Kulprocessen

### 4.1 Tidsserieanalys

Samma metoder som användes vid identifieringen av bommen användes vid identifieringen av kulans läge. En viktig skillnad var dock att eftersom kulprocessen är instabil så var vi tvungna att utföra identifieringen i sluten loop. Regulatorn som vi använde valdes avsiktligt med dålig prestanda för att inte ta bort excitationen från insignalen. Koherensen visas i Figur 5 och den visar att vi kan förvänta oss god identifiering upp till ca 2 Hz, trots den slutna slingan.



Figur 4: Simulerad (heldragen) och verklig(streckad) utsignal.



Figur 5: Koherensdiagram för identifiering av kulans läge

Vårt val föll till sist på en ARMAX-modell enligt

This matrix was created by the command ARMAX on 4/20 1994 at 14:10  
Loss fcn: 0.0004819 Akaike's FPE: 0.0004867 Sampling interval 1  
The polynomial coefficients and their standard deviations are

B =

0	0	0.0227
0	0	0.0005

A =

1.0000	-1.9761	0.9761
0	0.0023	0.0023

C =

1.0000	-0.9727	0.3019
0	0.0312	0.0321

och vi får därmed den tidsdiskreta överföringsfunktionen från bommens vinkel till kulans läge som

$$H_{\varphi x}(q) = \frac{0.0227}{q^2 - 1.9761q + 0.9761} \quad (5)$$

med samplingsintervallet  $h = 50$  ms.

Vi provade även med Box-Jenkins modellen men den gav ej bättre resultat, utan istället så blev  $F(q^{-1}) \approx D(q^{-1})$  vilket ju visar att en ARMAX-modell duger lika bra.

## 4.2 Validering av Kulmodellen

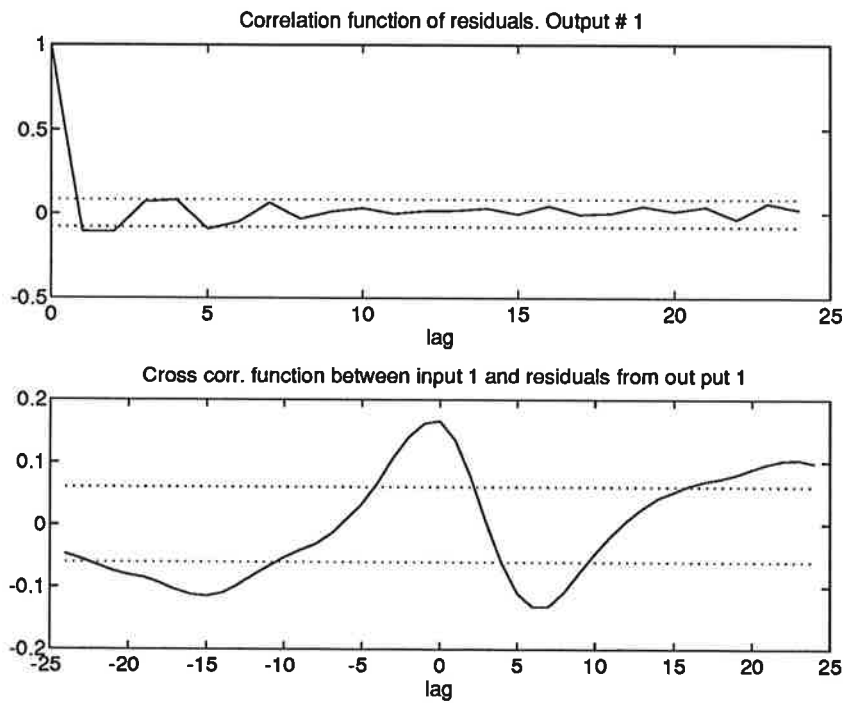
För att validera vår erhållna modell studerades först residualerna. I Figur 6 visas residualernas korrelationsfunktion, ur vilken man kan dra slutsatsen att residualerna är vita. Däremot är inte residualerna okorrelerade med insignalen. Vi har dock provat andra modeller utan att få ett nämnvärt bättre resultat.

En simulering av systemet (se Figur 7) visar på hyfsad överensstämmelse mellan modell och verklighet. Som insignal till simuleringen har vi använt tidigare oanvänd data.

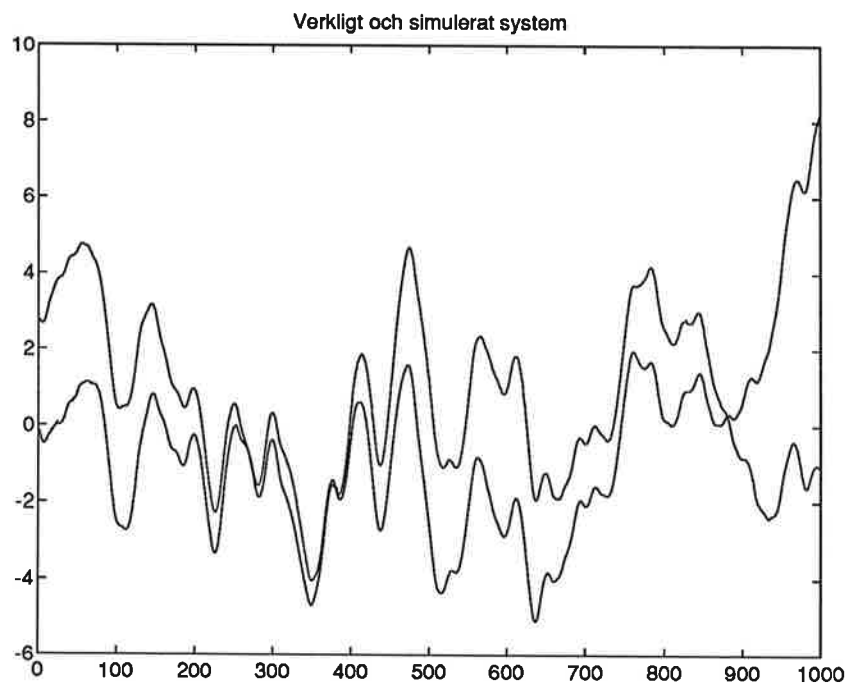
## 4.3 Skattning av Tidkontinuerlig Parameter $J$

Eftersom de flesta metoderna som vi använde för att identifiera processen endast gav de tidsdiskreta motsvarigheterna till de fysikaliska parametrarna i processen, så försökte vi även uppskatta  $J$  hos kulan. Detta kan göras genom att utgå från (2) som är en linjärisering av (1). Genom att införa differentialoperatören  $p = \frac{d}{dt}$  kan (2) skrivas som

$$p^2 x = -\frac{g}{1 + \alpha} \varphi \quad (6)$$



Figur 6: Korrelation hos residualerna.



Figur 7: Simulerad (övre) och verklig(undre) utsignal.



Sedan införs en  $\lambda$ -operator enligt

$$\lambda = \frac{1}{1 + \tau p} \quad \Rightarrow \quad p = \frac{1}{\tau} \frac{1 - \lambda}{\lambda}$$

Med  $\lambda$ -operatoren transformeras nu (6) till

$$x_f = (1 - \lambda)^2 \{x\} = -\frac{\tau^2 g}{1 + \alpha} \lambda^2 \{\varphi\} = \varphi_f^T \theta \quad (7)$$

där  $\varphi_f^T = \lambda^2 \{\varphi\}$  och  $\theta = -\frac{\tau^2 g}{1 + \alpha}$ . Nu är (7) på standard regressionsform, och vi kan direkt skatta  $\theta$  med minsta kvadrat metoden

$$\hat{\theta} = (\varphi_f^T \varphi_f)^{-1} \varphi_f^T x_f$$

Skattningen av  $\alpha$  fås sedan som  $\hat{\alpha} = -\frac{\tau^2 g}{\hat{\theta}} - 1$  och till sist fås

$$\hat{J} = \hat{\alpha} m r^2$$

där  $m$  är kulans massa och  $r$  är kulans radie. Ett problem med metoden är att välja ett lämpligt  $\tau$ . Detta problem löstes genom att beräkna  $\hat{\alpha}$  för ett antal olika  $\tau$ . Vi fann då att  $\hat{\alpha}$  konvergerade för lämpliga val av  $\tau$ , och med  $\tau = 0.5 \text{ s} \cdot \text{rad}^{-1}$  erhöles  $\hat{\alpha} = 0.9$ . Genom att mäta  $m$  och  $r$  fick vi  $\hat{J} = 1.6 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2$ .

För att testa rimligheten i ovanstående räkningar utförde vi även en inverssampling av (5) m h a MATLAB-funktionen d2c. Överslagsräkningar gav sedan att  $\hat{\alpha} = 0.8$ , vilket visar att våra räkningar är rimliga.

## 5 Reglering

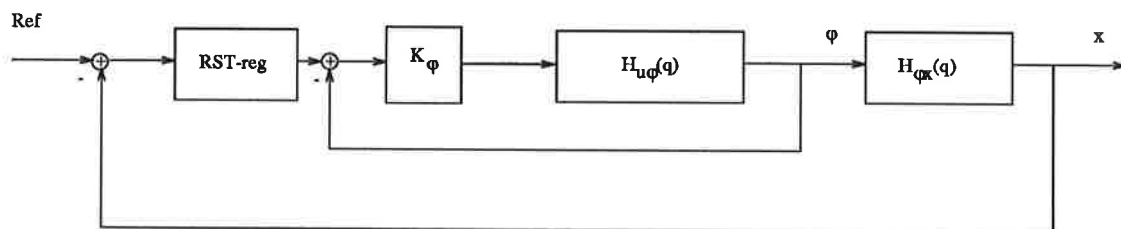
### 5.1 Inledning

Med hela systemet identifierat kunde vi nu designa en regulator för att reglera kulans läge på bommen. För att underlätta regulatorbyggandet delade vi upp systemet i två delar, en inre slinga som reglerar bommens vinkel och en yttre slinga som reglerar kulans läge. Som regulator i den inre slingan användes en  $P$ -regulator med hög förstärkning, vilket ger ett så snabbt system att det approximativt kan betraktas som en konstant förstärkning. Detta underlättar sedan designen av regulatorn i den yttre slingan. Valet av regulator i den yttre slingan föll på en  $RST$ -regulator. I Figur 8 visas uppdelningen av systemet i en inre och en yttre slinga.

### 5.2 Design av $RST$ -regulatorn

Genom valet av en snabb  $P$ -regulator i den inre slingan fås den approximativa totala överföringsfunktionen för systemet som (se (5))

$$\frac{B(q)}{A(q)} \approx H_{\varphi x}(q)$$



Figur 8: Uppdelning av systemet för reglering.

och som önskat slutet system (i kontinuerlig tid) väljs

$$\frac{Bm(s)}{Am(s)} = \frac{1}{(s+2)^3}$$

och som observerarpolynom väljs  $A_o(s) = (s+4)^3$ , vilka sedan samplas i MATLAB. För att undvika stationära fel tog vi med en integrator i regulatorn.

Med funktionen `rstd` i MATLAB löses sedan den diophantiska ekvationen

$$(q-1)A(q)R_1'(q) + B'(q)S(q) = A_o(q)A_m(q)$$

och vi fick regulatorn

$$\begin{aligned} R(q) &= q^3 - 1.7173q^2 + 0.6914q + 0.0259 \\ S(q) &= 1.2657q^3 - 2.4696q^2 + 1.2049q \\ T(q) &= 0.1096q^3 - 0.1984q^2 + 0.0897q \end{aligned}$$

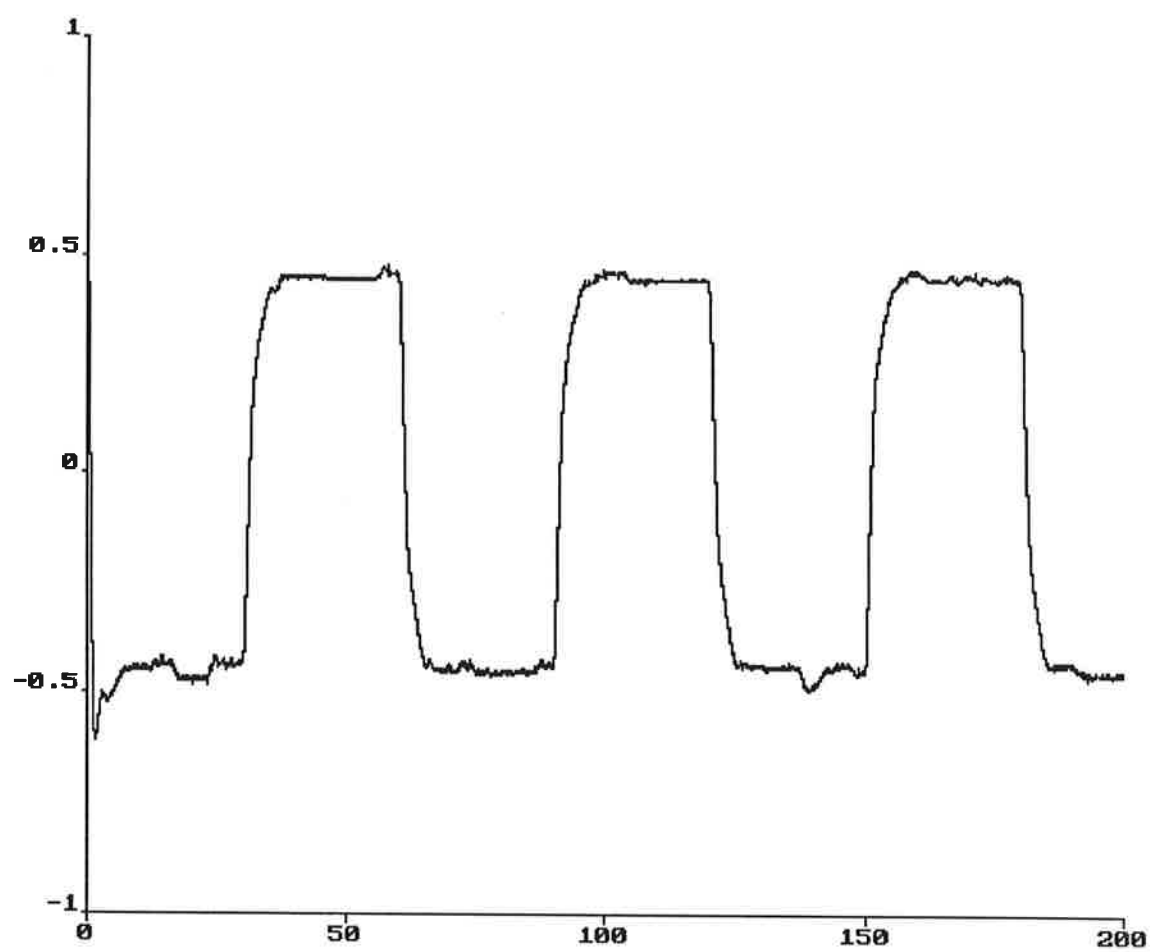
### 5.3 Det Slutna Systemets Beteende

Den erhållna regulatorn implementerades sedan i SIMNON och i Figur 9 visas en typisk körning där kulan regleras från ena sidan av bommen till den andra.

Resultatet visar att regleringen fungerar väl trots att inte mycket kraft har lagts på regulatorbygget. Vi hade givetvis kunnat bygga en bättre regulator om vi lagt ner mer tid och t.ex. inte approximerat den inre slingan så kraftigt.

## Referenser

- [1] Rolf Johansson, *System Modeling and Identification*. Prentice Hall 1993
- [2] Rolf Johansson, *System Modeling and Identification - Solutions Manual*. KF-Sigma Lund 1993
- [3] Karl J. Åström och Björn Wittenmark, *Computer Controlled Systems*. Prentice Hall 1989
- [4] Kjell Gustafsson, *Logger - A Program For Data Logging*. CODEN:LUTFD2/(TFRT-7457)1-4/1990



Figur 9: Resultatet av en körning med RST-regulatorn.

# Projekt i Processidentifiering

## Inverterad pendel

Anders Abrahamsson E-90

Håkan Ivarsson E-90

Mats Jerlebo E-90

Handledare:

Lennart Andersson

26 April 1994

## Innehåll

<b>1 Inledning</b>	<b>3</b>
<b>2 Modellering</b>	<b>3</b>
<b>3 Experimentplanering</b>	<b>4</b>
<b>4 Identifiering och Validering</b>	<b>6</b>
4.1 ARX-modell . . . . .	6
4.2 ARMAX-modell . . . . .	8
4.3 Balanserad realisering och modellreduktion . . . . .	9
4.4 Box-Jenkins modell . . . . .	10
<b>5 Reglering</b>	<b>11</b>
<b>6 Sammanfattning</b>	<b>13</b>
<b>A Lagrange-modellering</b>	<b>14</b>

# 1 Inledning

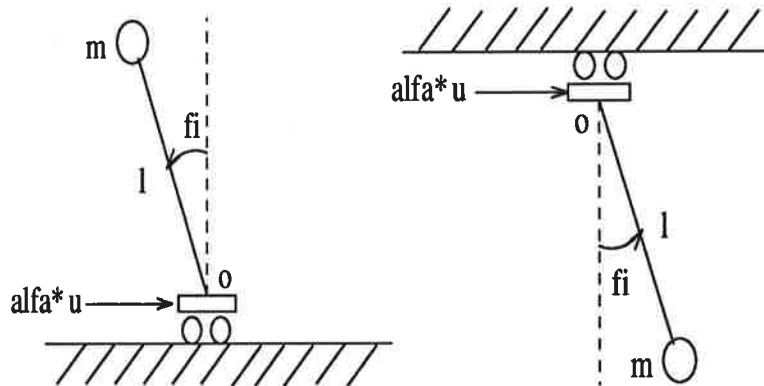
I föreliggande rapport studerar vi den inverterade pendeln. Vi modellerar och identifierar processen med olika metoder för att finna en modell som är lämplig för reglering. Vårt mål är att hitta en pulsöverföringsfunktion eller en stokastisk modell, som vi kan använda vid regulatordesign. Eftersom tyngdpunkten i projektet ligger på identifiering ägnar vi mindre tid åt regleringen. Vi föreslår dock möjliga lösningar.

Vi väljer att skatta processparametrarna i diskret tid. Ett alternativ är att skatta de kontinuerliga parametrarna enligt kapitel 12 i [1]. Vi avstår från denna metod då den kräver analoga filter, samt att rutiner som behandlar skattning av kontinuerliga system saknas i Matlab.

Pendeln kan identifieras antingen hängande eller i upprätt läge. För att slippa omräkningar av parametrar skulle vi vilja identifiera i uppläge, men detta medför vissa problem. Bland annat krävs identifiering i slutan loop som leder till förlust av identifierbarhet. Dessutom är det svårt att hitta en regulator som är tillräckligt bra, men ej för bra. Detta gör att vi väljer identifiering i nedåtläge.

# 2 Modellering

För att få en uppfattning om processens dynamik härleder vi en överföringsfunktion genom att ställa upp en momentekvation. Nedan genomförs detta med pendeln i uppläge. Pendeln betraktas som en punktmassa belägen på avståndet  $l$  från infästningspunkten. Se figur 1. Vi väljer att försumma motorarmens dynamik. Vi menar att detta inte är någon inskränkning, eftersom dess tidskonstant är mycket mindre än pendelns. Momentjämvikt kring infästningspunkten O ger:



Figur 1: Modell av pendeln i uppläge- och nedåtläge.

$$J\ddot{\varphi} = -dl\dot{\varphi} + mgl \sin(\varphi) + u\alpha l \cos(\varphi)$$

där  $\alpha$  är en omvandlingsfaktor mellan pålagd spänning  $u$  och kraft. Antagandet att  $\varphi$  är liten och därpå följande laplacetransformering ger:

$$(Js^2 + dls - mgl)\Phi(s) = \alpha l U(s)$$

$$G_{upp}(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{\alpha l}{J}}{s^2 + \frac{dl}{J}s - \frac{mgl}{J}} = \frac{\beta}{s^2 + 2\zeta\omega s - \omega^2}$$

På samma sätt härleder vi överföringsekvationen i nedåtläget.

$$G_{ned}(s) = \frac{\beta}{s^2 + 2\zeta\omega s + \omega^2}$$

Som synes är det endast tecknet framför  $\omega^2$ -termen som skiljer fallen åt. Detta kan vi utnyttja vid omräkning av parametrarna från nedåtläget till uppåtläget.

Med hjälp av Maple beräknade vi pulsöverföringsfunktionen för både uppåt- och nedåtläge. ZOH-sampling förutsättes. Pulsöverföringsfunktionen för ett andra ordningens system har följande utseende:

$$H(q) = \frac{B(q)}{A(q)} = \frac{b_1q + b_2}{q^2 + a_1q + a_2}$$

$B(q)$  är samma för både uppåt och nedåtläge, medan  $A(q)$  skiljer sig i koefficienten  $a_1$ .

$$a_1 = -e^{-\frac{dh}{2}} (e^{\sqrt{d^2 \mp 4\omega^2} \frac{h}{2}} + e^{-\sqrt{d^2 \mp 4\omega^2} \frac{h}{2}})$$

$$a_2 = e^{-dh}$$

För att få  $a_1$ -parametern i uppåtläget adderar vi uttrycken för parametern i uppåt respektive nedåtläge. I detta uttryck tar vissa termer ut varandra och kvar blir följande:

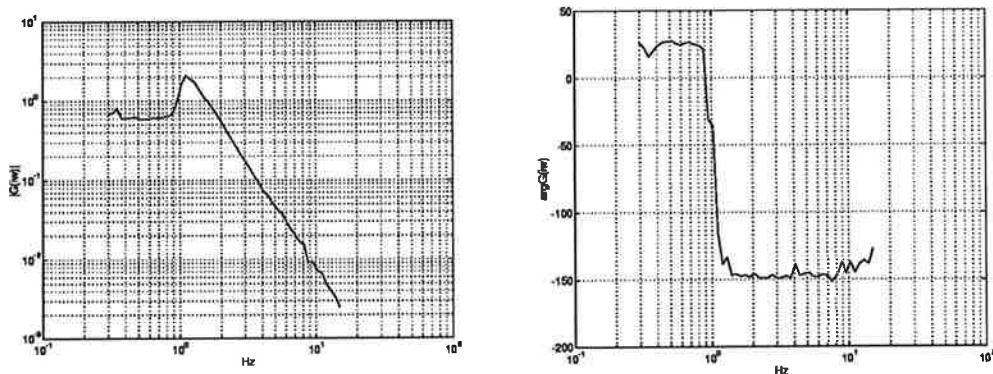
$$a_{1u} + a_{1n} = -4e^{-\frac{dh}{2}} - \frac{h^2 d^2}{2} + O(h^4)$$

Detta uttryck Taylorutvecklar vi och löser sedan ut  $a_{1u}$  i uppåtläget, uttryckt i  $a_{1n}$  i nedåtläget.

$$a_{1u} = -a_{1n} - \frac{\sqrt{a_2}}{2}(8 - (\ln a_2)^2) + O(h_4)$$

Vi har nu fått en andra ordningens modell som utgångsläge för identifiering.

### 3 Experimentplanering



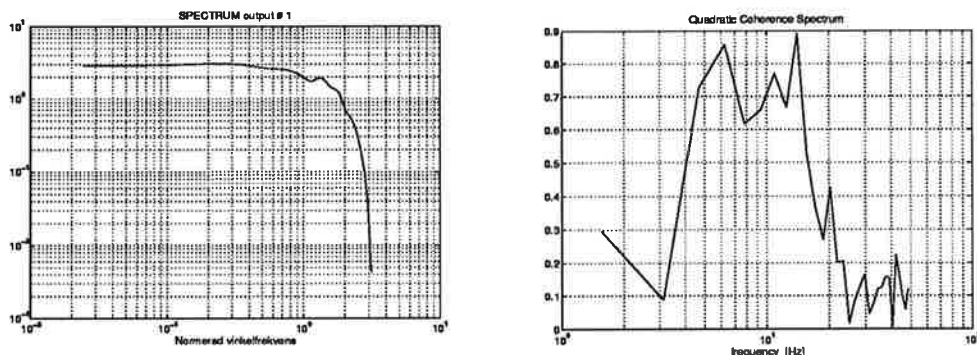
Figur 2: Bode-diagram från frekvensanalysen (Amplitud och fas).  $A_{in}=0.9$  V,  $F_{min}=0.3$  Hz,  $F_{max}=15$  Hz, 30 steg/dekad,  $T=20$  cykler och fördröjning=5 s

Som ett inledande försök att få kännedom om processen utför vi frekvensanalys. Vi använder oss av korrelationsmetoden som finns implementerad i frekvensanalysatorn

med tillhörande programvara PCFRA. Vi utför en rad experiment med varierande amplitud och undersöker att det inte förekommer några hoppresonanser, vilka är tecken på olinjäriteter. Detta resulterar i ett Bode-diagram (Se figur 2) ur vilket vi kan utläsa att systemet har en resonansfrekvens vid ca 1.1 Hz. Ur Bode-diagrammet framgår att processen troligen är av ordning två och med statiska förstärkningen 0.6. Efter att ha studerat processen i några olika frekvensområden drar vi slutsatsen att endast ett litet frekvensområde är intressant ur regleringssynpunkt. Vid låga frekvenser blir vinkelutslaget stort och pendeln befinner sig då i olinjärt område. I diagrammet syns detta i form av den lilla toppen vid början av amplitudkurvan. Dessutom uppkommer oönskade fluktuationer i fasen. Vi ser också att faskurvan börjar på 30 grader följt av en fäsvridning på 180 grader ner till -150 grader. Denna 30-gradersavvikelse från ett linjärt system beror troligtvis på någon olinjäritet i processen som avancerar fäsen.

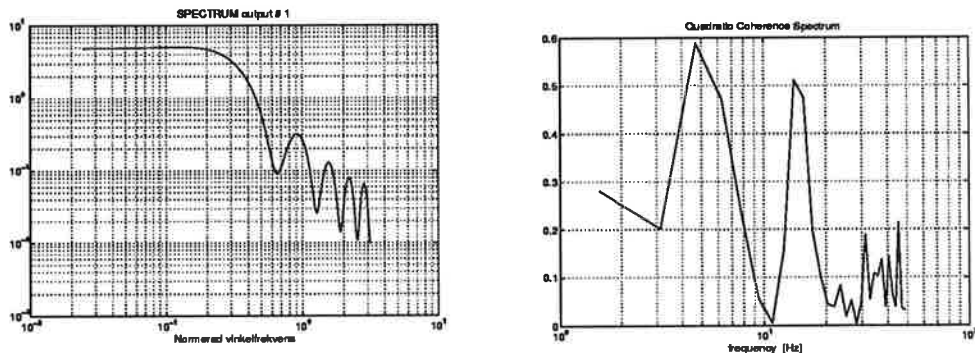
För höga frekvenser ovanför systemets bandbredd blir vinkelutslaget mycket litet och vi drar slutsatsen att det intressanta frekvensområdet ligger i storleksordningen 0.5-3 Hz.

För att kvantitativt bestämma parametrarna i modellen utför vi identifiering av processen. Första ledet i identifieringsfasen är insamling av data. Vi utför mätdatainsamlingen med programvaran Logger. Som insignal använder vi en PRBS-signal där amplitud och periodtid justeras så att pendeln exciteras i sitt linjära område. Vi samlade in flera mätserier och studerade insignalens spektrum map vitbrusegenskaper samt koherensspektrum. I figur 3 och 4 visas resultatet av två insamlingar. Av figurerna framgår att mätdata tillhörande figur 3 bör väljas, eftersom denna mätserie har bättre brusegenskaper och högre och mer utbrett koherensspektrum. För denna mätserie är insignalamplituden 1.2 V, sampelfrekvensen 100 Hz och kortaste tid för PRBS-signalen innan den byter tecken är 2 sampelintervall. Vi ser i figur 3 att koherensspektrum är i storleksordningen 0.7 i intervallet 4.5 till 15 Hz vilket täcker det frekvensområde där det är intressant att reglera (Vi väljer i avsnittet om reglering  $\omega_m = 25$ ).



Figur 3: Insignalens spektrum och koherensspektrum i fallet  $A_{in}=1.2$  V,  $F_{sample}=100$  Hz och PRBS konstant över 2 sampel.





Figur 4: Insignalens spektrum och koherensspektrum i fallet  $A_{in}=0.8$  V,  $F_{sample}=100$  Hz och PRBS konstant över 10 sampel.

## 4 Identifiering och Validering

Vi väljer att lägga momenten Identifiering och Validering i ett avsnitt då det är naturligt att utvärdera sina resultat efter hand. Regleringen som är en del av valideringen väljer vi att placera i ett eget kapitel, då det bör behandlas för sig. Experimentfasen resulterade i en 1600 punkter lång mätserie ( $F_{sample}=100$  Hz) som vi i Matlab kan utföra identifiering på. Vi delar upp mätserien i två delar där ena halvan används till validering. Innan all identifiering påbörjades kontrollerade vi att mätserien inte innehöll några outliers.

Utgående från pulsöverföringsfunktionen,  $H(q)$  ser vi att en ARX-struktur är lämplig som första modell. Vi provar även några andra modeller och ser om de ger en bättre beskrivning av det verkliga systemet. Vi undersöker ARMAX-modeller, metoden med Balancerad realisering och modellreduktion samt Box-Jenkins modell.

Vid validering undersöker vi residualer map brusegenskaper, korskorrelation mellan insignal och residual, deterministisk simulering och korsvalidering.

### 4.1 ARX-modell

$$M : \quad A^*(q^{-1})y(t) = B^*(q^{-1})u(t - k) + e(t)$$

När data skall anpassas till en ARX-modell kan olika metoder tillämpas. Ett givet förstahandsval är minstakvadratmetoden, MK, sedan kan varianter på denna tillämpas. Första delmålet är att hitta "rätt" modellordning. Vi testar Akaikes informationskriterium, AIC och Minimum Description Length, MDL, för val av modellordning. Om AIC används blir den optimala modellordningen (6,6,1). Med (6,6,1) menar vi antalet A- och B-parametrar samt polöverskottet.

Vi utför MK-identifiering i Matlab. Den estimerade modellen presenterar vi i Bode-diagrammet i figur 5. Diagrammet överensstämmer med resultatet från frekvensanalysen map resonansfrekvens, modellordning och statisk förstärkning.

När vi validerar får vi resultat enligt figur 5, högra halvan. Residualerna är vitt brus (kurvan ligger innanför konfidensintervallet), vilket tyder på att (6,6,1)-modellen lyckas utvinna all information ur data. Dessutom ser vi att insignal och residualer, korskorrelationen, inte uppvisar något större beroende, vilket är ett krav när MK-metoder används.

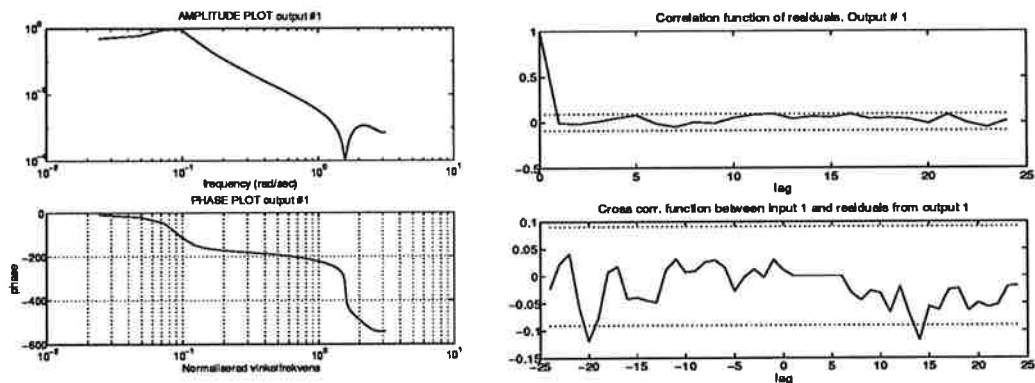


Figure 5: Bodediagram för estimerad (6,6,1)-modell samt residualer och korskorrelation.

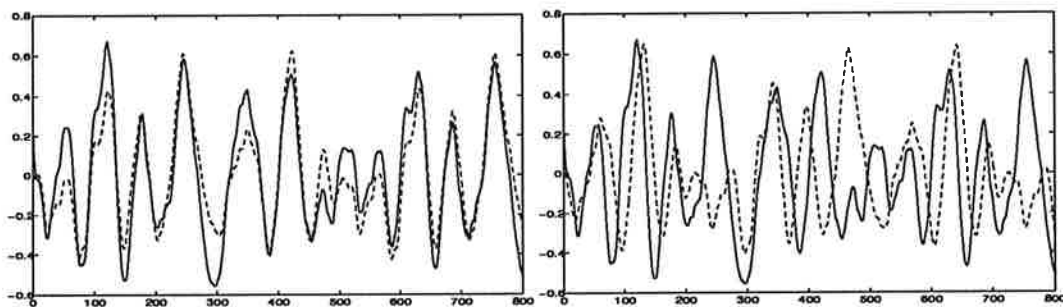
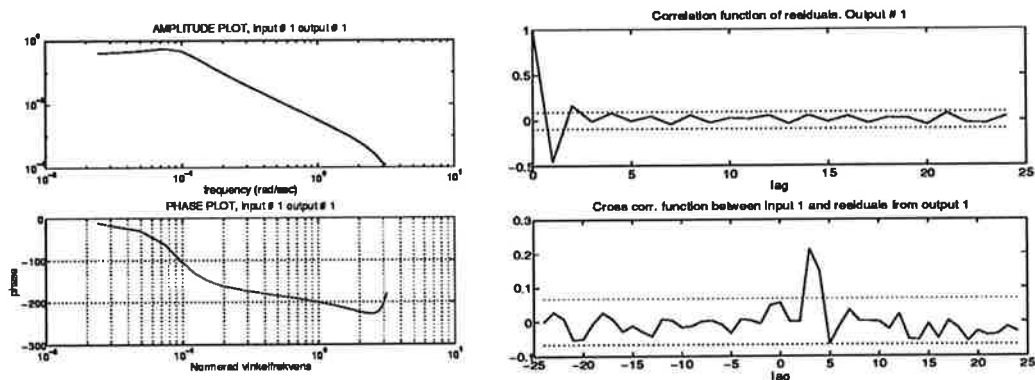


Figure 6: Deterministisk simulering och korsvalidering för (6,6,1).

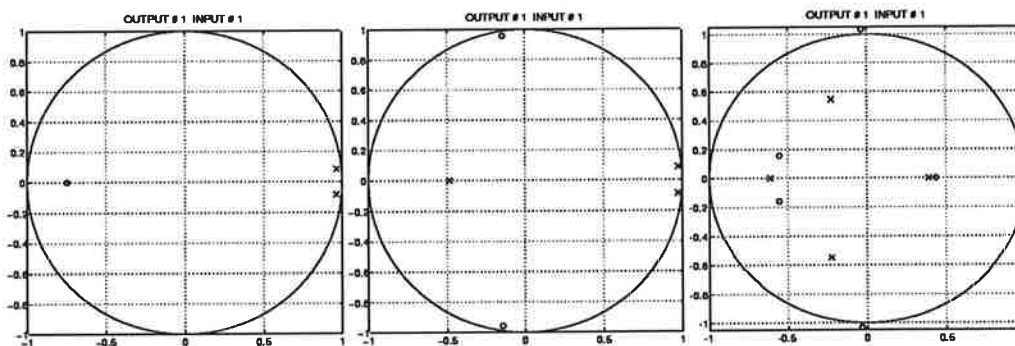
I figur 6 ser vi resultaten av den deterministiska simuleringen och korsvalideringen. Vid den deterministiska simuleringen förser vi vår estimerade modell med samma insignalsekvens som vid identifieringen. Simuleringen visar att modellutsignal följer den verkliga utsignalen och amplituden är i samma storleksordning, dvs insignalamplituden vid mätdatainsamlingen var troligtvis lämpligt vald. Korsvalideringen (andra halvan av insignaldata in till modellen) är litet svårare att analysera. Utsignalen vid simuleringen följer den verkliga signalen i viss utsträckning. Ett korrekt uppförande av modellutsignalen är orimligt att begära eftersom modellen ej är lika bra i alla frekvensområden. Vi vet tack vare koherensspektra att vår modell är bra i intervallet 4.5 till 15 Hz.

Om vi istället använder MDL-kriteriet blir (4,4,1)-modellen optimal. Bodediagrammet i detta fall ser ut precis som i fallet (6,6,1). Residualer och korskorrelation för denna modellordning överensstämmer i allt väsentligt med (6,6,1). Den deterministiska simuleringen och korssimulering uppvisar inte heller några större skillnader.

Eftersom (4,4,1)-modellen i stort sett är lika bra som (6,6,1)-modellen är det intressant att undersöka modeller av lägre ordning. Då en (3,3,1)-modell används får vi Bode-diagram av samma utseende som i de föregående fallen. Även de statistiska kriterierna är likvärdiga med resultaten ovan. Till sist undersöker vi en (2,2,1)-modell. I detta fall får vi ett resultat som skiljer sig från de ovanstående ganska likvärdiga modellerna. Bodediagrammet i figur 7 saknar resonanstopp. Dessutom ser vi att residualerna uppvisar mindre vitbruskaraktär, samt ett visst beroende mellan insignal och residualer.



Figur 7: Bode-diagram för (2,2,1) samt motsvarande residualer och korskorrelation.



Figur 8: Pol- nollställe diagram för ARX (2 2 1), (3 3 1) och (6 6 1).

Vi har tidigare konstaterat att bandbredden för det öppna systemet är ca 3 Hz. I figur 8 motsvarar detta en cirkelsektor med båginkel 22 grader. Polen som motsvarar resonanstoppet ligger för de tre olika modellordningarna på samma ställe. Övriga poler och nollställen gör sig gällande först vid högre frekvenser men bidrar i viss mån till utseendet hos amplitud- och faskurvan.

Modellordningskriterierna indikerar att vi skall välja en modell av ordning (6,6,1) eller (4,4,1). Eftersom modellerna av ordning tre och högre ger så likartade resultat bör ordning tre (eventuellt ordning två) vara tillräcklig för att beskriva systemet. Nedan listas parametrarna som kan användas för omräkning till upplägg.

Ordning (3,3,1)	$a_0 = 1$	$a_1 = -1.462$	$a_2 = 0.0135$	$a_3 = 0.4608$
	$b_1 = 0.027$	$b_2 = 0.0008$	$b_3 = 0.0025$	
Ordning (2,2,1)	$a_0 = 1$	$a_1 = -1.9271$	$a_2 = 0.9352$	
	$b_1 = 0.0018$	$b_2 = 0.0013$		

## 4.2 ARMAX-modell

För att förbättra beteendet i de deterministiska simuleringarna samt korsvalideringarna provar vi med att lägga till ett C-polynom.

$$M : \quad y(t) = \frac{B^*(q^{-1})}{A^*(q^{-1})} u(t-k) + \frac{C^*(q^{-1})}{A^*(q^{-1})} e(t)$$

På detta sätt kan man möjligen ta hänsyn till systemets brusegenskaper. Vi väljer att studera ARMAX-modeller av ordning motsvarande våra bästa ARX-modeller. I dessa varierar vi antalet C-parametrar i hopp om bättre resultat.

Vi identifierar våra ARMAX-modeller i Matlab, där Maximum-Likelihood metoden används. Först provar vi modellen (2,2,1,1) där den tredje positionen anger antal C-parametrar. Resultatet blir i princip samma som för motsvarande ARX-modell. Residualerna är dock något bättre. Två eller tre brustertermer spelar ingen roll. För modellerna av ordning fyra och sex med varierande antal brustertermer kan vi ej påvisa bättre beteende än för motsvarande ARX-modeller.

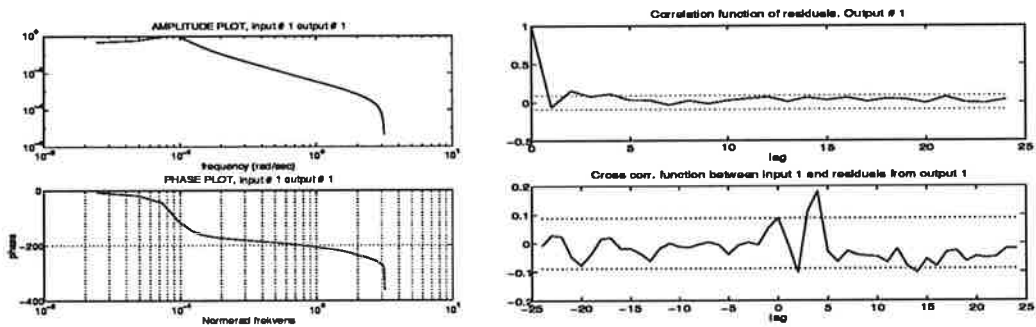


Figure 9: Bode-diagram för (2,2,1,1) samt motsvarande residualer och korskorrelation.

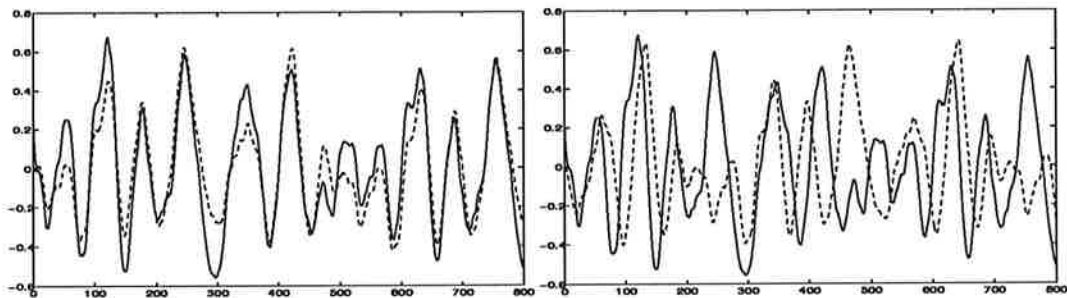


Figure 10: Deterministisk simulering och korsvalidering för (2,2,1,1).

Vår slutsats blir att det ej är motiverat att använda ARMAX-modeller, utom möjligen för ordning två, där vi ser en viss förbättring. Parametrarna för denna modell är:

$$\begin{array}{lll} \text{Ordning (2,2,1,1)} & a_0 = 1 & a_1 = -1.9569 \quad a_2 = 0.9651 \\ & b_1 = 0.0018 & b_2 = 0.0018 \\ & c_0 = 1 & c_1 = -0.4834 \end{array}$$

Vi konstaterar att  $a$ -koefficienterna är så gott som samma som i fallet med ARX-modellen.

### 4.3 Balanserad realisering och modellreduktion

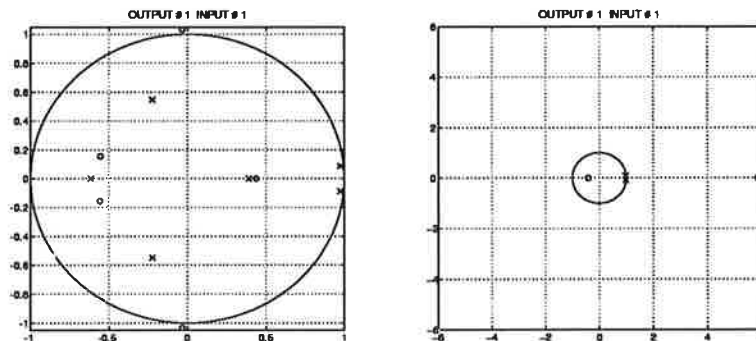
I våra tidigare metoder har vi fått bra resultat vad gäller residualer och korskorrelationer. Korsvalideringen visar möjligen brister i insignal-utsignal relationen hos modellen. Vi undersöker om vi kan få ett bättre beteende mha en balanserad realisering följt av modellreduktion. Då infinner sig frågan vilken modellordning vi skall utgå från. Det viktiga vid valet är att utgångsmodellen uppvisar ett bra beteende så vi provar olika alternativ av tillräckligt hög ordning.

Vi utgår först ifrån en ARX (6,6,1) och får en balanserad realisering med Gramianen  $\Sigma = \text{diag}(0.6679 \ 0.4377 \ 0.0012 \ 0.0010 \ 0.0003 \ 0.0000)$ . De två första singularvärdena är mycket större än de övriga och vi kan därför reducera till en

modell med två tillstånd motsvarande de två första singularvärdena. Vi får som resultat en modell (2,3,0) som vi undersöker. Vi ser dock ingen förbättring jämfört med de tidigare metoderna och drar slutsatsen att känslighetsanalysen i metoden med balancerad realisering inte har någon betydelse. Reduktion från (6,6,1) till (2,3,0) ger parametrarna:

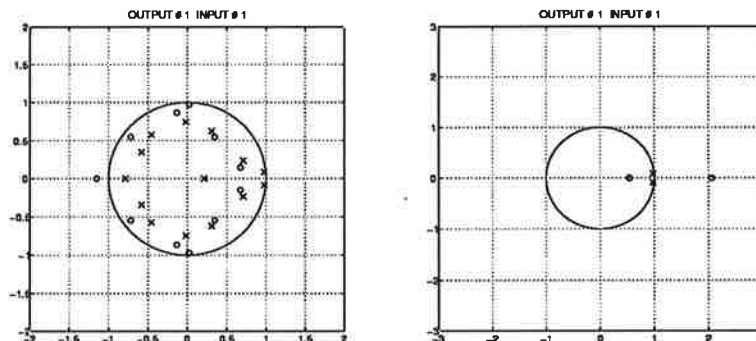
$$\begin{array}{llll} \text{Ordning (2,3,0)} & a_0 = 1 & a_1 = -1.9532 & a_2 = 0.9614 \\ & b_1 = -0.0007 & b_2 = 0.0033 & b_3 = 0.0012 \end{array}$$

Vi ser att  $a$ -koefficienterna överensstämmer mycket väl med ARX och ARMAX.



Figur 11: Pol- nollställe diagram för ARX (6 6 1) och motsvarande reducerade modell (2 3 0).

Vi provar även att starta ifrån en (14,12,1)-modell. I detta fall reduceras modellen till en andra ordningens modell. Inte heller i detta fall ser vi något bättre beteende.



Figur 12: Pol- nollställe diagram för ARX (14 12 1) och motsvarande reducerade modell (2 3 0).

I figur 11 och 12 kan man se att polen som motsvarar resonanstoppet ej förkortas. Enligt resonemanget om poler och nollställen i avsnittet om ARX-modellering bibehålls modellens karakteristiska egenskaper efter reduktionen.

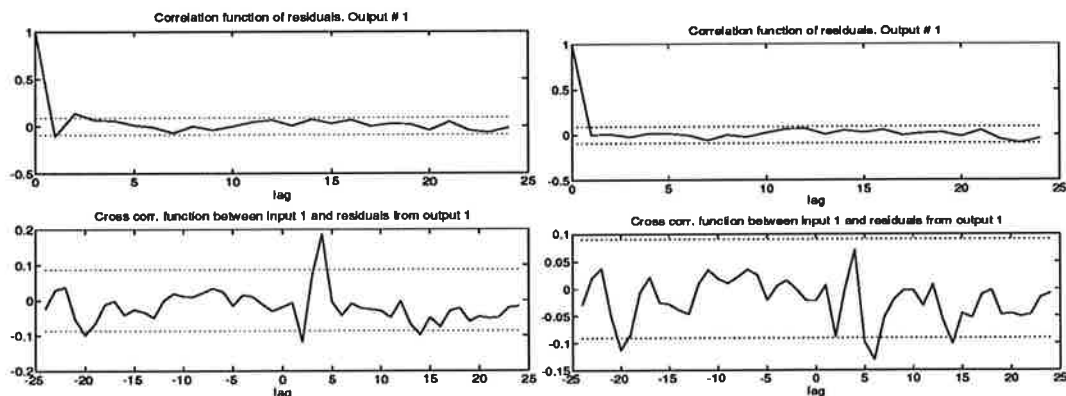
#### 4.4 Box-Jenkins modell

Vi har tidigare konstaterat att en ARMAX- modell inte ger bättre resultat än för ARX. För att undersöka möjligheten att bruset kommer in annorlunda på processen försöker vi med en Box-Jenkins modell.

$$M : \quad y(t) = \frac{B^*(q^{-1})}{F^*(q^{-1})} u(t-k) + \frac{C^*(q^{-1})}{D^*(q^{-1})} e(t)$$

Vi har anledning att misstänka att BJ's modell inte kommer att förbättra resultatet jämfört med motsvarande ARMAX-modell och ARX-modell. Detta beroende på att vi tidigare konstaterat att brusbeskrivningen spelar en mindre roll.

Vi provar med de intressanta modellordningarna, dvs två och tre. I fallet med en modell av ordning  $(nb,nc,nd,nf,k)=(3,1,2,2,0)$  ser vi ingen skillnad jämfört med motsvarande ARX- och ARMAX-modeller. Även i fallet  $(3,2,2,2,0)$  påvisas ingen förbättring. Residualer samt korskorrelation i fallet  $(3,1,2,2,0)$  visas i figur 13.



Figur 13: Residualer och korskorrelation för BJ-modeller. Bilderna till vänster är resultaten av modellordning  $(3,1,2,2,0)$  och till höger för  $(3,3,3,3,0)$

Korsvalidering och deterministisk simulering är som i fallet ARMAX. Vi studerar dessutom parametrarna och konstaterar att  $F$ - och  $D$ -polynomen är så gott som lika, vilket tyder på att en ARMAX-modell är tillräcklig.

Parametrar i fallet  $(3,1,2,2,0)$ :

$$\begin{array}{lll} f_0 = 1 & f_1 = -1.9491 & f_2 = 0.9577 \\ b_1 = -0.0007 & b_2 = 0.0009 & b_3 = 0.0023 \\ d_0 = 1 & d_1 = -1.9565 & d_2 = 0.9614 \\ c_0 = 1 & c_1 = -0.5579 & \end{array}$$

Vi provar även en tredje ordningens modell  $(3,3,3,3,0)$ . Även nu blir resultatet som i fallet med ARMAX. Resultatet visas i figur 13.

Slutsatsen blir att en Box-Jenkins modell i vårt fall inte är en bättre modell än ARMAX.

## 5 Reglering

Som resultat av identifieringen har vi fått en diskret modell för pendeln i nedåtläge. För andra ordningens system kan parametrarna räknas om till uppåtläge med hjälp av resultatet i avsnittet om modellering eller i Matlab. Den förra metoden är lämplig om man vill beräkna parametrarna i ett realtidsprogram.

För tredje ordningen och högre får vi problem vid omräkningen. Om det diskreta systemet har någon pol på negativa reella axeln existerar inte avbildningen från  $z$ -planet till  $s$ -planet och därmed misslyckas vi i försöket att räkna om parametrarna. Man kan lätt förledas att tro att Matlab klarar av denna transformation. Matlab presenterar en lösning på transformationen, men vi menar att resultatet saknar innebörd. En pol på negativa reella axeln i  $z$ -planet hamnar på negativa reella axeln i  $s$ -planet och detta är felaktigt. Polen skulle hamna enligt figur 3.5 på sidan 71 i [2].

På samma sätt som för andra ordningens kontinuerliga system försöker vi hitta ett direkt samband mellan de diskreta parametrarna i upp- och nedläge. Strategin är

samma som i modelleringsavsnittet, dvs sampla det stabila och instabila systemet och hitta ett samband mellan de diskreta parametrarna. Samplingen utfördes i Maple, men resulterade i ett 1827 rader långt uttryck. Metoden övergavs.

Ett sätt att undvika problemet med odefinierade transformationer är att direkt skatta de kontinuerliga parametrarna enligt teorier från kapitel 12 i [1].

De modeller som är intressanta att använda vid reglering är följande. Parametrarna gäller för uppåtläge omräknade enligt avsnitt 2. Samma resultat kan för ett andra ordningens system erhållas med hjälp av Matlab under förutsättning att det inte finns några poler på negativa reella axeln. I fallet med ARMAX-modellen har vi räknat om blocken  $\frac{B(q)}{A(q)}$  och  $\frac{C(q)}{A(q)}$  för sig. Vi får då samma  $A$ -polynom i de båda blocken. Det är möjligen mer riktigt att anta att bruset kommer in på samma sätt i både uppåt- och nedåtläge och i så fall behålla blocket  $\frac{C(q)}{A(q)}$ , men i så fall fås inte en ARMAX utan en Box-Jenkin modell.

ARX-modellen:

$$(q^2 - 1.9433q + 0.9352)y(k) = (0.0018q + 0.0013)u(k)$$

ARMAX-modellen:

$$(q^2 - 1.9733q + 0.9651)y(k) = (0.0018q + 0.0018)u(k) + (q^2 - 0.4878q + 0.0051)e(k)$$

Vi designar en RST-regulator för ARX-modellen. Vi väljer en polplacering med  $\omega_m = 25$  vilket stämmer överens med tumregeln  $w_m h = 0.2-0.6$  i [2]. Dämpningen väljes  $\zeta = 0.7$  och sampeltakten är 100 Hz. Observeraren väljes till deadbeat för att få mycket snabb dynamik. Då erhålles följande regulatorparametrar:

$$r_0 = 1 \quad r_1 = 0.1113 \quad s_0 = 99.8966 \quad s_1 = -80.0575 \quad t_0 = 16.9355$$

Denna regulator klarar att stabilisera pendeln i uppåtläge. Den är också mycket stabil för laststörningar i form av yttre kraftpåverkan. Vi konstaterar dock att motorarmen har en liten tendens att driva iväg. Pendelns drift är ej begränsad åt ett håll. Den driver lika gärna med- som moturs. Detta beror på att det finns en dödzon i potentiometern som mäter vinkelutslaget hos pendeln. Pendeln lutar en aning, men potentiometern visar inget utslag varvid tyngdkraften gör sig gällande och pendeln driver iväg.

För att råda bot på driften kan man införa någon form av olinjär kompensering. Ett annat alternativ kan vara att estimerar motorarmens hastighet och införa denna som ett tillstånd i modellen.

Det är också mycket viktigt att ställa in referensvärdet rätt.

När det gäller ARMAX-modellen är denna minst lika bra som ARX-modellen enligt identifieringen. Därmed kan vi förvänta oss att vi lyckas med regleringen även i detta fall. Eftersom vi redan har lyckats att stabilisera pendeln nöjer vi oss med att föreslå minimal varians alternativt LQG som en lämplig strategi.

## 6 Sammanfattning

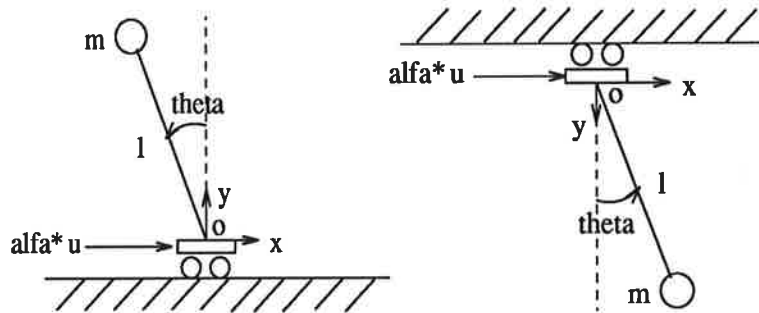
Vi hade som mål att hitta en modell av processen som kan användas för stabilisering. Via modellering och identifiering har vi succesivt arbetat fram en modell som på bästa sätt beskriver processen. Modelleringen gav resultatet att en andra ordningens modell är tillräcklig. Identifieringsfasen indikerade att modeller av ordning fyra och sex var de bästa alternativen. Valideringen gav att modeller av ordning tre och högre är så gott som lika. För praktiskt bruk är därmed ordning tre det bästa valet. Vad gäller ordning två kunde något större skillnader observeras. När vi reglerade på denna modellordning visade det sig dock att det fungerade utmärkt.

En svårighet vi har stött på under projektet är det faktum att vi var tvungna att räkna om de estimerade parametrarna till en ny modell (uppåtläget). Vi tror att motsvarande problem kan dyka upp i andra tillämpningar men om man kan undvika sådana omräkningar är det att föredra.

Ett alternativ för att slippa odefinierade transformationer vid omräkningen är att direkt skatta de kontinuerliga parametrarna enligt metoder i kap 12 i [1]. I Matlab finns en funktion `pem` för detta ändamål. Vid en närmare kontroll visade det sig dock att denna samplar systemet och identifierar i diskret tid, varför vi har samma problematik som tidigare. Lösningen på problemet är att själv implementera någon metod från kap 12 i [1], men detta faller utanför detta projekts tidsram.



## A Lagrange-modellering



Figur 14: Modell av pendeln i uppåt- och nedåtläge.

Här följer en alternativ härledning av dynamiken hos pendeln mha Lagrange-mekanik. Pendelarmens längd är  $l$ , klumpen i armens spets betraktas som en punktmassa med massan  $m$ , samt pendelarmens massa försummas. Först härleder vi ekvationerna för pendeln i uppåtläge. De kinematiska uttrycken och dess derivator är:

$$x = -l \sin \theta$$

$$y = l \cos \theta$$

$$\dot{x} = -l\dot{\theta} \cos \theta$$

$$\dot{y} = -l\dot{\theta} \sin \theta$$

Lägesenergin och kinetiska energin ges av

$$U(\theta) = mgl \cos \theta$$

$$T(\theta) = \frac{1}{2}(\dot{x}^2 + \dot{y}^2) = \frac{1}{2}ml^2\dot{\theta}^2$$

Lagrangianen blir då

$$L = T - U = \frac{1}{2}ml^2\dot{\theta}^2 - mgl \cos \theta$$

Euler-Lagranges ekvation ges av

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau$$

Det yttre momentet  $\tau$  ges av

$$\tau = \alpha ul \cos \theta - d\dot{\theta}l \cos \theta$$

där första termen är bidraget från pålagd spänning och andra termen är en friktionsterm. Då blir Euler-Lagranges ekvation

$$ml^2\ddot{\theta} - mgl \sin \theta = \alpha ul \cos \theta - d\dot{\theta}l \cos \theta$$

som för små vinklar blir

$$ml^2\ddot{\theta} + dl\dot{\theta} - mgl\theta = \alpha lu$$

Division med  $ml^2$  och Laplacetransformering ger

$$G(s) = \frac{\Theta(s)}{U(s)} = \frac{\frac{\alpha}{ml}}{s^2 + \frac{d}{ml}s - \frac{g}{l}} = \frac{\beta}{s^2 + 2\zeta\omega s - \omega^2}$$

På samma sätt kan dynamiken för pendeln i nedåtläge härledas. Resultatet av dessa räkningar är

$$G(s) = \frac{\Theta(s)}{U(s)} = \frac{\frac{\alpha}{ml}}{s^2 + \frac{d}{ml}s + \frac{g}{l}} = \frac{\beta}{s^2 + 2\zeta\omega s + \omega^2}$$

Vi konstaterar att skillnaden ligger i tecknet framför  $\omega^2$ -termen. Vi har fått fram samma resultat som i avsnitt 2 ( $J = ml'^2$ ), vilket var väntat.

## Referenser

- [1] Rolf Johansson. *System modeling and identification*. Prentice Hall, 1993.
- [2] K-J Åström & B Wittenmark. *Computer controlled systems*. Prentice Hall, 1990.

# Identifiering av en fläktprocess

Roger Granath E-89 och Fahed Saouan E-89

Handledare: Lennart Andersson

Institutionen för Reglerteknik

24 maj 1994

<i>INNEHÅLL</i>	1
-----------------	---

## Innehåll

<b>1 Sammanfattning</b>	<b>2</b>
<b>2 Fysikalisk modell av processen</b>	<b>3</b>
<b>3 Identifiering av fläktprocessen</b>	<b>4</b>
3.1 Frekvenssvarsanalys . . . . .	4
3.2 Stegsvarsanalys . . . . .	6
3.3 Undersökning av kvadrerade koherensfunktionen . . . . .	8
<b>4 Modellanpassning</b>	<b>10</b>
4.1 ARMAX-modell . . . . .	10
4.2 Box-Jenkins modell . . . . .	11
4.3 Instrumentvariabel modell IV4 . . . . .	11
<b>5 Simulering och Reglering</b>	<b>14</b>

## 1 Sammanfattning

Vår uppgift var att identifiera en process, som bestod av en fläkt med ett konstant varvtal. Vid fläktens början satt ett värmeelement som värmden upp luften och detta luftflöde transporterades sedan vidare via ett rör till en temperaturgivare. Denna givare presenterade processens utsignalen som en spänning mellan 0 - 10 V. Insignalen till värmeelementet kunde varieras mellan 0 - 10 V.

Datainsamling skedde med hjälp av en lämpligt utformad PRBS-signal som exciterade de önskade frekvenserna i processen. Vi bestämde en fysikalisk modell och lät kravet på den skattade modellen vara så att denna inte skulle överstiga ett andra ordningens system. Resultatet blev att vi ur data valde en ARMAX modell enligt

$$A(z^{-1})y(n) = B(z^{-1})u(n) + C(z^{-1})e(n)$$

$$A(z^{-1}) = 1 - 1.5683z^{-1} + 0.6140z^{-2}$$

$$B(z^{-1}) = 0.0058z^{-4} + 0.0045z^{-5}$$

$$C(z^{-1}) = 1 - 0.3719z^{-1} + 0.1469z^{-2}$$

Modellen testades genom simulering med en framtagen Smith-regulator i simnon, och efter detta applicerades denna regulator direkt på processen via realtidssimnon. Vid simuleringen blev resultatet riktigt bra med en stigtid på c:a 0.4 sekunder. Reglering på den verkliga processen gav inte samma resultat, utan man ser här att processen inte uppför sig stabilt i sina parametrar.

## 2 Fysikalisk modell av processen

Processen består av en fläkt som producerar ett konstant luftflöde. På fläkten sitter ett luftinsugsspjäll, som kan ställas från  $0^\circ$  till  $180^\circ$ . Vi valde att ställa detta på  $60^\circ$ . Luftflödet transporteras vidare genom ett rör, och vid dess början sitter ett värmelement, vars temperatur påverkas av processens insignal  $u$ . Denna kan varieras mellan 0 till 10 volt. Den uppvärmda luften registreras senare vid rörets slut av ett termoelement, och omvandlas till en spänning mellan 0 till 10 volt. Denna spänning representerar processens utsignal  $y$ .

Temperaturen  $T_1$  vid rörets början och den uppmätta temperaturen  $T_2$  vid rörets slut ges av

$$T_1 = \frac{k_1}{(s+a)}u(s) \quad T_2 = T_1 e^{-s\tau}$$

där  $T_2$  omvandlas åter till en utsignalspänning  $y$  enligt

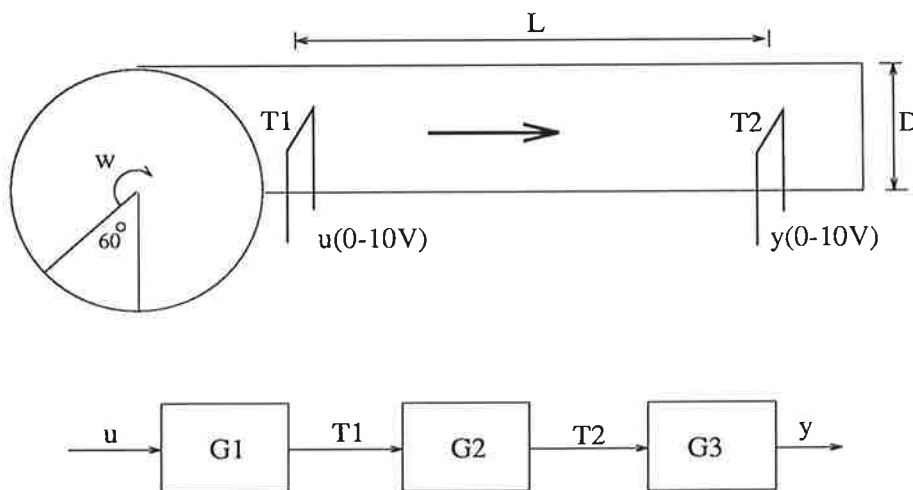
$$y(s) = \frac{k_2}{(s+b)}T_2$$

Dvs, vi får överföringsfunktionen från  $u$  till  $y$  som

$$y(s) = \frac{K e^{-s\tau}}{(s+a)(s+b)}u(s) = G_{process}(s)u(s)$$

där  $K = k_1 k_2$ . Vi har då antagit att  $\tau$  svarar för tidsfördröjningen, dvs den tid det tar innan  $u$  påverkar  $y$ .

### Fläktprocess



Figur 1: Insugsöppningen är ställd till  $60^\circ$  och de geometriska måtten är  $L = 28.5$  cm och  $D = 4.4$  cm. Fläktens vinkelhastighet  $\omega$  är konstant.

### 3 Identifiering av fläktprocessen

#### 3.1 Frekvenssvarsanalys

Om processen är ett linjärt tidsinvariant och stabilt system kan Bodediagrammet erhållas genom frekvenssvarsanalys. Man skickar då in en sinussignal  $u$  till processen och får ut en förstärkt eller dämpad samt fasförskjuten sinussignal  $y$  enligt figur 2. Denna utsignal multipliceras med en sinus- respektive en cosinussignal som har samma frekvens som insignalen  $u$ , och därefter sker en integration av dessa signalerna. I de flesta fall har man även en additiv störning som kommer in efter process-blocket och stör utsignalen. För att minska brusets inverkan används en mättid  $T$  för integrationen som är  $k$  hela perioder av testsignalen  $u$ , dvs  $T = 2\pi k/\omega$ . Ökad mättid minskar variansen hos skattningen  $\hat{G}$ , där  $G$  är processens frekvenssvar, då bruset är vitt och har man en sinusformad störning med frekvens skild från testfrekvensen påverkar denna inte estimaten alls. För en testfrekvens  $\omega$  ger sinuskanalen

$$S_T = \int_0^T y(t) \sin(\omega t) dt = \frac{1}{2} T |G(i\omega)| \cos(\arg(G(i\omega)))$$

och cosinuskanalen

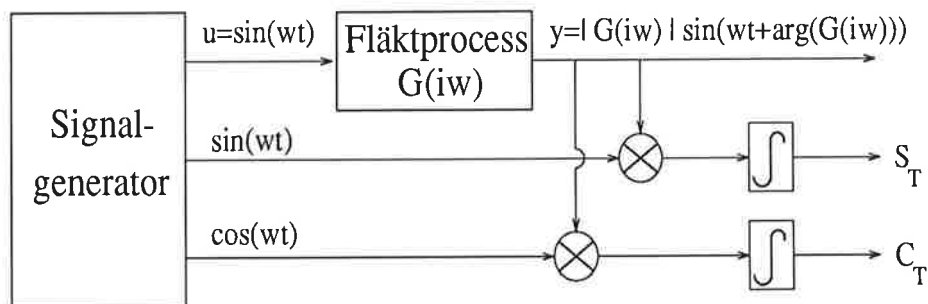
$$C_T = \int_0^T y(t) \cos(\omega t) dt = \frac{1}{2} T |G(i\omega)| \sin(\arg(G(i\omega)))$$

Estimaten av processens förstärkning  $|G(i\omega)|$  samt fasförskjutning  $\Phi(\omega)$  ges nu av

$$|\hat{G}(i\omega)| = \frac{2}{T} \sqrt{S_T^2(\omega) + C_T^2(\omega)}$$

$$\hat{\Phi}(\omega) = \arg(\hat{G}(i\omega)) = \arctan\left(\frac{C_T(\omega)}{S_T(\omega)}\right) + k\pi$$

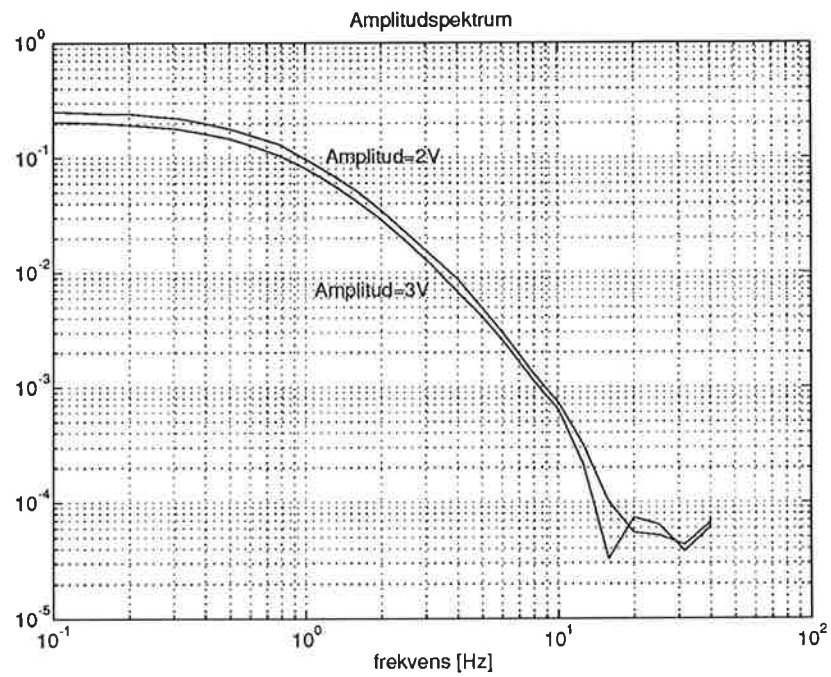
Ovanstående ger en punkt i Bodediagrammet och förfarandet kan upprepas för så brett frekvensområde man är intresserad av. Testsignalen  $u$  ges då den frekvens man söker i Bodediagrammet.



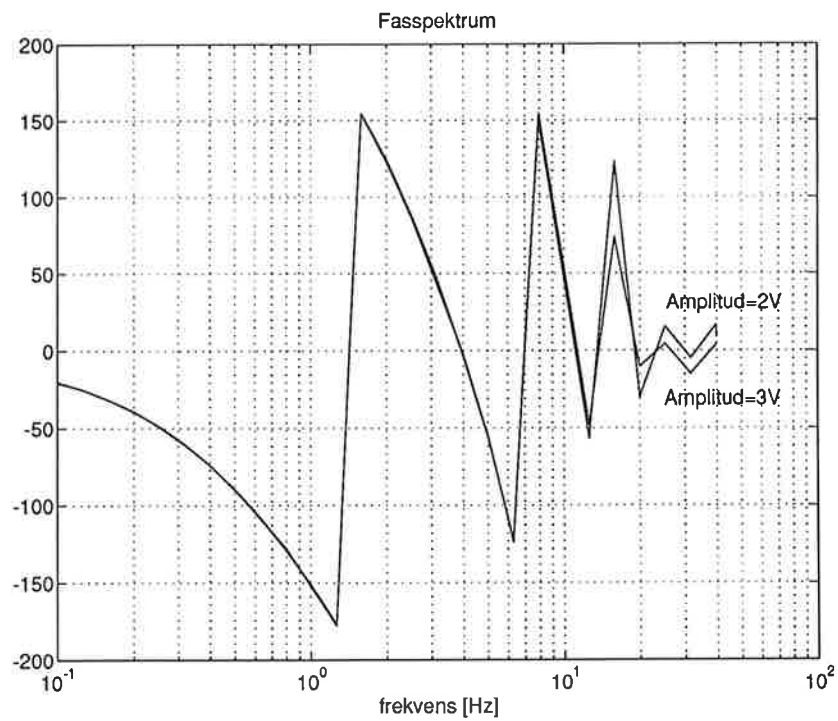
Figur 2: Frekvenssvarsanalys. Block som omger processen finns i en Schlumberger Solartron 1250 Frekvenssvarsanalysator

Vid frekvenssvarsanalysen av fläktprocessen användes en sinussignal med medelvärdet 5V och amplituden 3V respektive 2V. Frekvensområdet som undersöks är 0.1-40 Hz och integrationstiden är 10s med en fördröjningstid på 5s innan mätningen startar. De resulterande Bodediagrammen ges som ett amplitudspektrum (figur 3) och ett fasspektrum (figur 4).





Figur 3: Amplitudspektrum för fläktprocessen



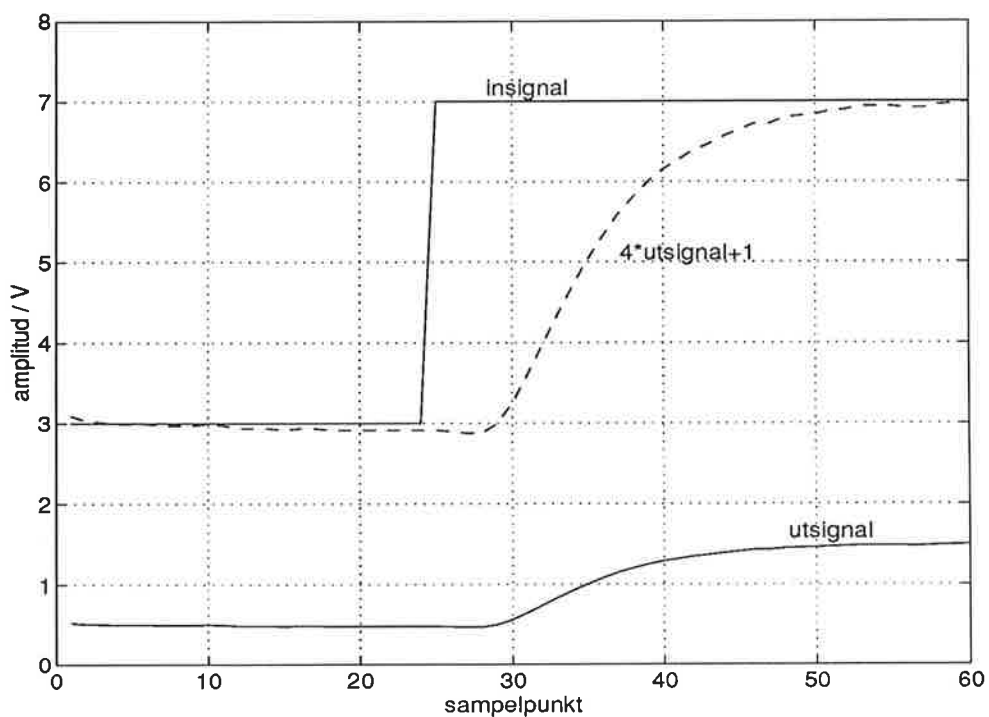
Figur 4: Fasspektrum för fläktprocessen

Jämför man Bodediagrammen för de olika amplitudvärdena ser man att dessa är väldigt lika vilket gör processen approximativt linjär. Den statiska förstärkningen vid låga frekvenser ligger mellan 0.2 och 0.3 för dessa amplitudvärden.

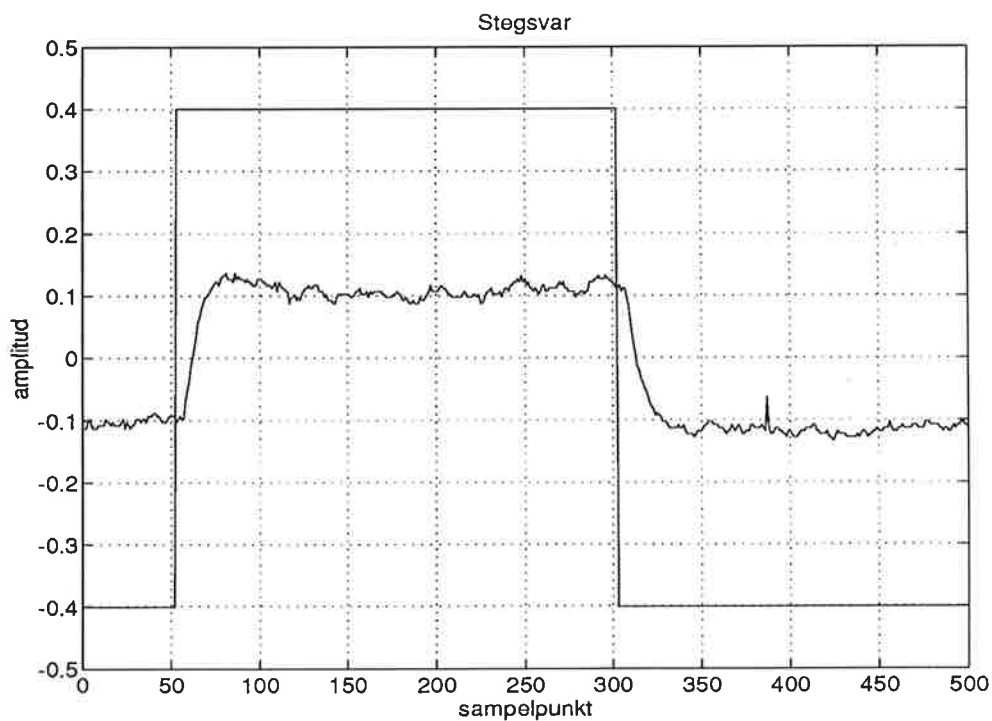
### 3.2 Stegvarsanalys

Genom att skicka in steg med olika amplitud till processen kan en hel del egenskaper hos processen erhållas. I figur 5 har ett steg med amplituden 3V och medelvärdet 5V skickats in till processen. Utsignalen får utseendet enligt den heldragna utsignalkurvan och med hjälp av denna kan man utläsa processens statiska förstärkning till ungefär 0.2 ggr. För att enklare kunna utläsa tidsfördröjningen och stigtiden har utsignalen multiplicerats med 4 och adderats med 1 så att denna blir i storlek med insignalen. Denna omskalade utsignal ges av den streckade kurvan i figuren och här ser man att tidsfördröjningen innan utsignalen reagerar på insignalsteget blir cirka 4 sampel, vilket motsvarar 200ms då samplingsperioden är 50ms. Snabbare än detta kan vi inte reglera systemet och vi låter detta vara vår undre snabbhetsgräns. Utsignalens stigtid blir ungefär 13 sampel (650ms) och den totala tiden från stegpåverkan till det att utsignalen stigit klart blir ungefär 1s. Vi vill reglera processen snabbare än detta och väljer vår stigtid till 500ms. En tumregel är att man minst ska ha tio sampelpunkter på en stigtid och vi måste därmed minst sampla med  $t_{samp} = 50ms$ . Detta värde var det vi startade med och därmed behåller vi detta även i fortsättningen.

I figur 6 har insignalens amplitud sänkts till 0.45V och medelvärdet tagits bort både från insignalen och utsignalen. Utsignalen är påverkad av brus vars varians man kan uppskatta genom att studera dess amplitud. Denna är ungefär  $0.05/2$  vilket ger en brusvariens på 0.000625.



Figur 5: Stegsvär då insignalen har amplituden 2V och medelvärdet 5V. Samplingsperioden är 50ms.



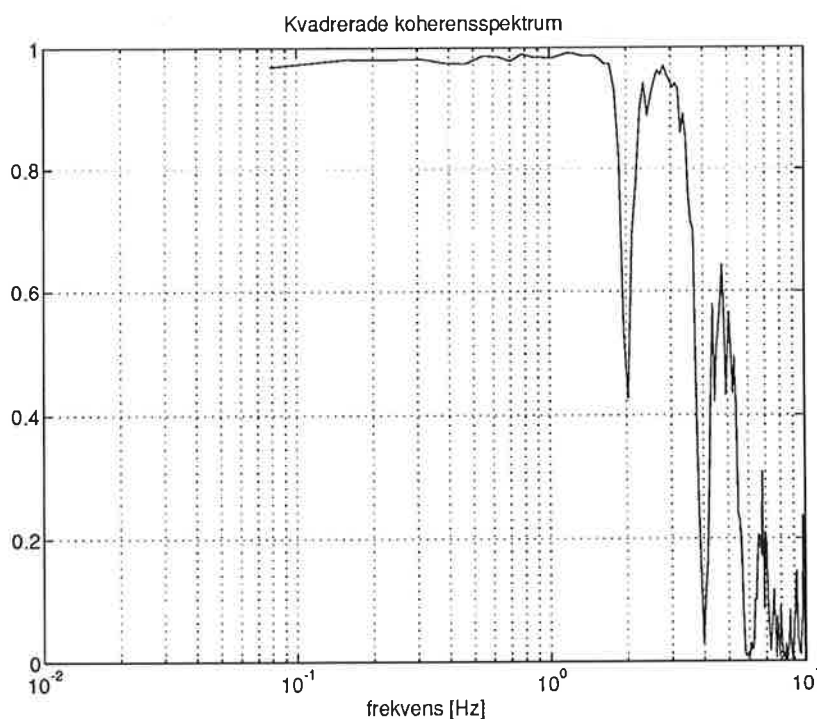
Figur 6: Stegsvär då insignalen har amplituden 0.45V och medelvärdet 5V. Samplingsperioden är 50ms

### 3.3 Undersökning av kvadrerade koherensfunktionen

Med hjälp av resultaten från stegvarsanalysen kan vi välja en lämplig skärfrekvens

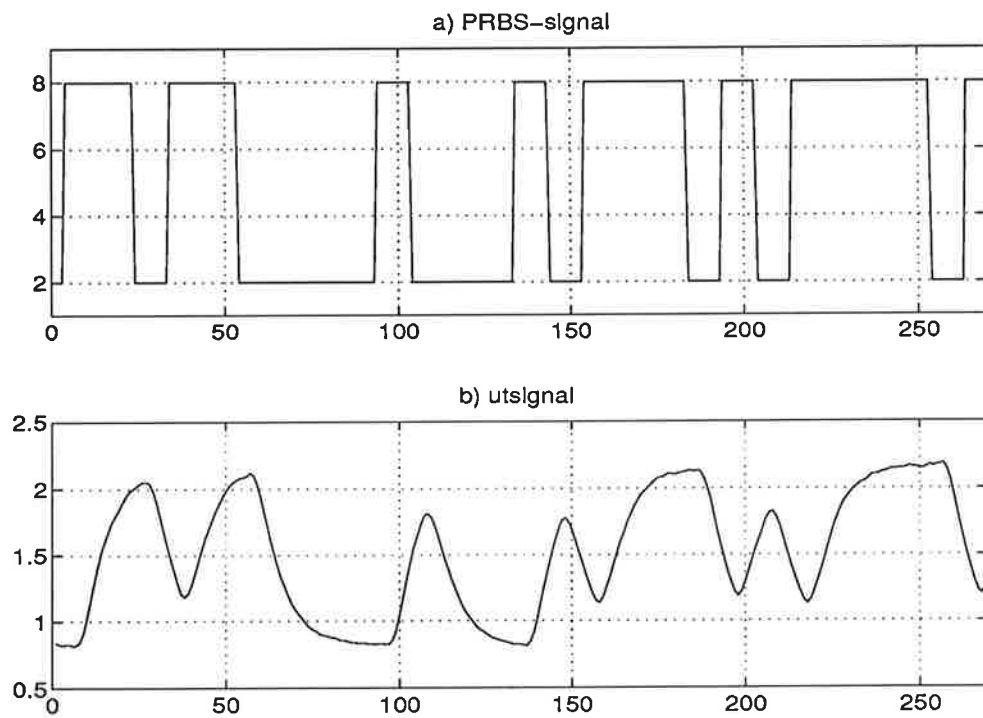
$$\omega = \frac{1}{\tau} = \frac{1}{500ms} \approx 1.33rad/s$$

Detta motsvarar en frekvens  $f=0.21$  Hz och det är vid denna frekvens som vi vill ha god excitation. Denna excitation kan undersökas med det kvadrerade koherensspektrat som antar värden nära 1 då detta krav är uppfyllt. Med detta krav kan en PRBS-signal (pseudo random binary sequence) designas så att denna exciterar de rätta frekvenserna i processen. En PRBS-signal med amplituden 3V, medelvärde 5V, PRBS-perioden 6 sampel, längden 800 sampel och samplingsperiod 50ms har ett kvadrerat koherensspektrum enligt figur ?? . Här ser man att frekvenser kring 0.2 Hz får god excitation ty  $\gamma^2 \approx 0.97$  och PRBS-signalen är därmed godkänd. Parametern som ändrar aktuellt frekvensområde är PRBS-perioden som här har valts till 6 sampel, dvs 300ms. Periodtiden (som beskriver den ungefärliga periodbredden) ska ligga någonstans i närheten av skärfrekvensen men den behöver ej vara identisk med denna.



Figur 7: Kvadrerade koherensspektrum med god excitotien för frekvenser kring 0.2 Hz då PRBS-perioden är 500ms

För den slutliga datainsamlingen valdes en PRBS-signal med amplitud 3V , medelvärde 5V, PRBS-period 10 sampel, längd 8000 sampel och sampelperiod 50ms. Denna har ett kvadrerat koherensspektrum enligt figur 7 och exciterar även den systemet bra vid 0.2 Hz. Studerar man en del av PRBS-signalen in till processen och utsignalen från denna har dessa utseendet enligt figur 8.



Figur 8: a) PRBS-signal in till processen b) utsignal från processen

## 4 Modellanpassning

Modellanpassningen sker på signalerna i figur 8 och vi använder oss av tre olika modeller att anpassa data till. Utgående från uppbyggnaden av den fysikaliska modellen och resultaten från tidigare analyser kan en del parametrar uppskattas. Tidsfördröjningen innan insignalen påverkar utsignalen blir ungefär 4 sampel (200ms) enligt stegsvarsanalysen och här ser man även att utsignalen inte stiger direkt (1:a ordn. system) utan har en "rundad" start vilket antyder ett 2:a ordningens system. Enligt Bodediagrammet har processen endast reella poler eftersom endast resonansstoppar ger upphov till komplexkonjugerade poler.

Kriterier som används vid beslut om modellordningen är följande:

1. Akaike FPE
2. Residualanalys
3. Simulering av modellen

Valet blir en avvägning mellan dessa kriterier där en låg modellordning prioriteras före en hög.

### 4.1 ARMAX-modell

ARMAX-modellen beskrivs av

$$A(z^{-1})y(n) = B(z^{-1})u(n) + C(z^{-1})e(n)$$

och vid modellanpassningen kom vi fram till att följande modell hade de bästa egenskaperna.

$$A(z^{-1}) = 1 - 1.5683z^{-1} + 0.6140z^{-2}$$

$$B(z^{-1}) = 0.0058z^{-4} + 0.0045z^{-5}$$

$$C(z^{-1}) = 1 - 0.3719z^{-1} + 0.1469z^{-2}$$

Detta är en ARMAX(2224)-modell med FPE-värdet  $4.46 \cdot 10^{-5}$  och där samtliga parametrar hade låg standardavvikelse. Residualanalys av felet mellan den verkliga utsignalen och utsignalen från modellen ger att korrelationsfunktionen ligger lite utanför konfidensintervallet nästan hela tiden men autokorrelationsfunktionen ligger innanför dessa gränser. För att residualanalysen ska vara helt godkänd måste minst en ARMAX(4234)-modell användas. FPE-värdet blir i detta fall  $3.689 \cdot 10^{-5}$ .

Studerar pol-nollställe diagrammet har man två reella poler i 2:a ordningens system samt ytterligare ett komplexkonjugerat polpar i 4:e ordningens system. Båda modellerna fungerar mycket bra i simuleringsavseende, dvs modellens utsignal följer den verkliga utsignalen mycket bra, och därför valde vi modellen av 2:a ordningen. Simuleringsresultatet för denna modell visas i figur 9.

## 4.2 Box-Jenkins modell

Box-Jenkins modellen beskrivs av

$$y(n) = \frac{B(z^{-1})}{F(z^{-1})}u(n) + \frac{C(z^{-1})}{D(z^{-1})}e(n)$$

och bäst resultat fick vi med följande polynom.

$$\begin{aligned} B(z^{-1}) &= 0.0058z^{-4} + 0.0047z^{-5} \\ F(z^{-1}) &= 1 - 1.5611z^{-1} + 0.6083z^{-2} \\ C(z^{-1}) &= 1 \\ D(z^{-1}) &= 1 - 1.0526z^{-1} + 0.0692z^{-2} \end{aligned}$$

Detta är en BJ(20224)-modell med FPE-värde  $4.003 \cdot 10^{-5}$  och även för denna modell hade samtliga parametrar låg standardavvikelse. Residualernas korrelationsfunktion ligger nu hela tiden i konfidensintervallet medan autokorrelationen har ett par områden utanför gränserna. Simulering med modellen ger ett mycket bra resultat enligt figur 10.

## 4.3 Instrumentvariabel modell IV4

IV4-modellen beskrivs av

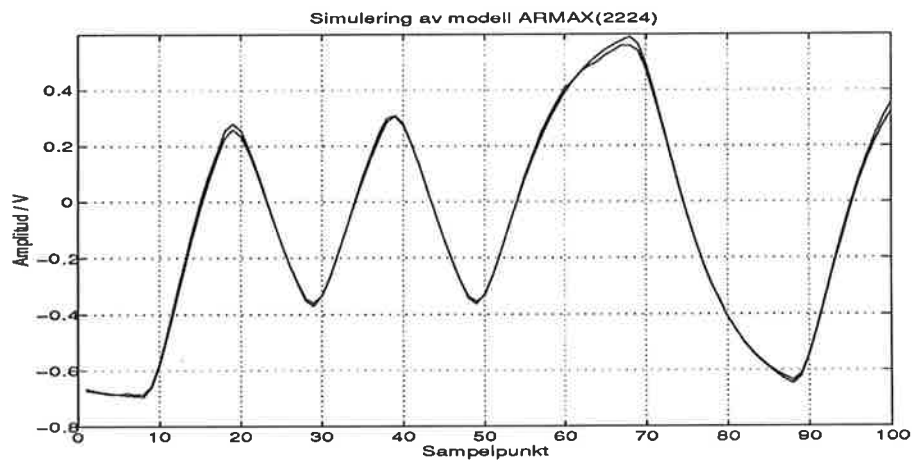
$$A(z^{-1})y(n) = B(z^{-1})u(n) + e(n)$$

och dess polynom blev efter modell Anpassning

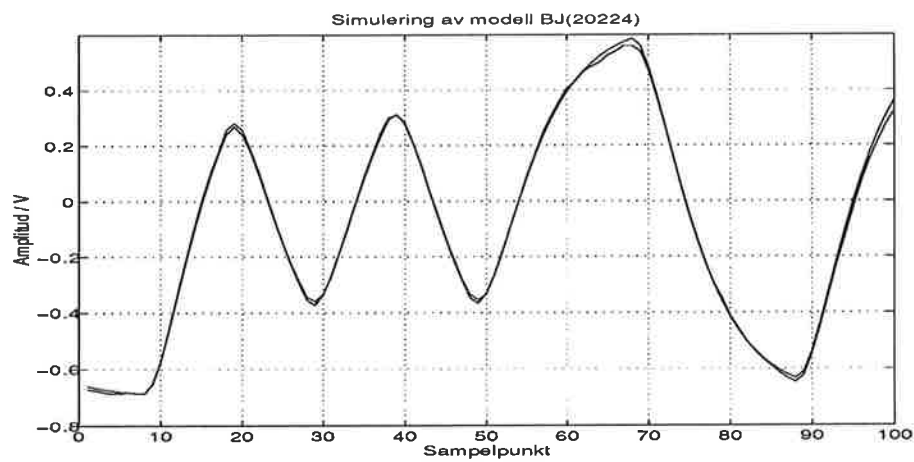
$$\begin{aligned} A(z^{-1}) &= 1 - 1.5709z^{-1} + 0.6170z^{-2} \\ B(z^{-1}) &= 0.0058z^{-4} + 0.0045z^{-5} \end{aligned}$$

Vi har valt en IV4(224)-modell med FPE-värde  $4.433 \cdot 10^{-5}$  och för denna modell hade samtliga parametrar låg standardavvikelse. Residualerna har dock en kraftig negativ korrelation på avståndet ett, som inte ens försvinner om man höjer modellordningen ett par steg. Simuleringarna fungerar däremot bra för modellen vilket visas i figur 11.

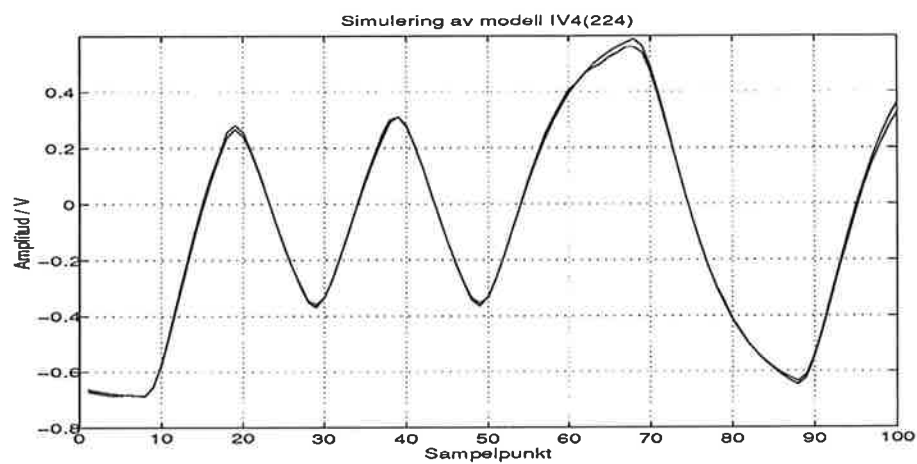
Box-Jenkins modellen har bäst FPE-värde, bäst residualegenskaper men vid simulering av modellerna visar det sig att de inte skiljer sig åt nämnvärt. Vi valde därför ARMAX-modellen eftersom omvandling till en kontinuerlig modell (via d2c i matlab etc.) ger en ökad modellordning då Box-Jenkins modell används. ARMAX-modellens residualegenskaper, pol-nollställes placering samt Bode-diagram finns i figurena 12-14.



Figur 9: Verklig utsignal samt utsignal från ARMAX-modellen

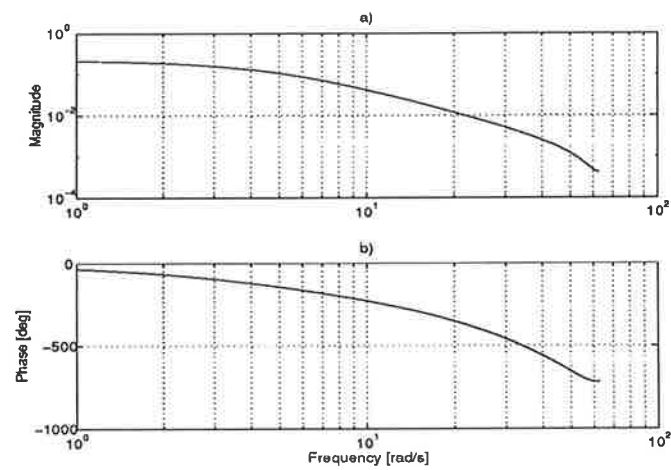


Figur 10: Verklig utsignal samt utsignal från Box-Jenkins modellen

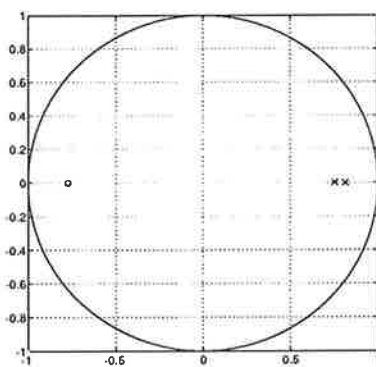


Figur 11: Verklig utsignal samt utsignal från IV-modellen

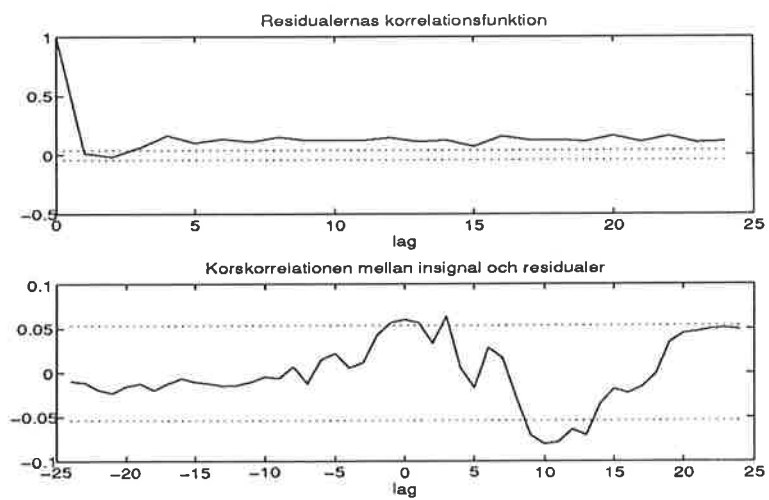




Figur 12: Bodediagram för ARMAX-modellen a) Amplitudkurva b) Faskurva



Figur 13: Pol-nollställes diagram för ARMAX-modellen



Figur 14: Residualensgenskaper för ARMAX-modellen

## 5 Simulering och Reglering

Eftersom processen innehåller ett antal tidsfördröjningar som ger upphov till kraftiga negativa fasförskjutningar, får det återkopplade systemet en sänkning av fasmarginen. System innehållande fasförskjutningar är därför svåra att reglera och därför har Smith-regulatorn specialdesignats för system med tidsfördröjningar. Systemet som ska regleras har utseendet

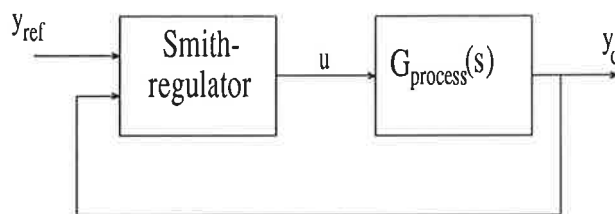
$$G_{process}(s) = G(s)e^{-s\tau}$$

där  $\tau$  är tidsfördröjningen på 0.2 sekunder. Smith-regulatorn kompenserar då denna tidsfördröjning så att regulatorn innehåller en intern tidsfördröjning. Regulatorn realiseras därmed enklast som en diskret regulator och ett krav är att man känner  $G(s)$ , dvs processen utan tidsfördröjningen. Hela processen beskrivs av

$$G_{process}(s) = \frac{0.0058s^2 + 0.2651s + 5.2314}{s^2 + 9.7552s + 23.2107} e^{-0.2s}$$

som har de reella polerna -5.64 och -4.12 samt den statiska förstärkningen 0.23. Motsvarande diskreta poler blir 0.75 och 0.81, som beskriver en ren tidsfördröjning.

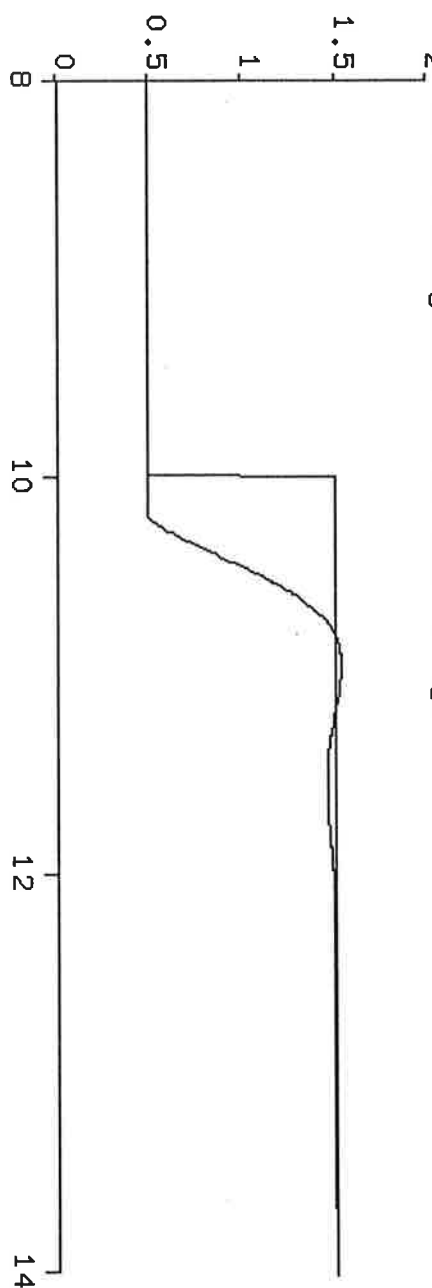
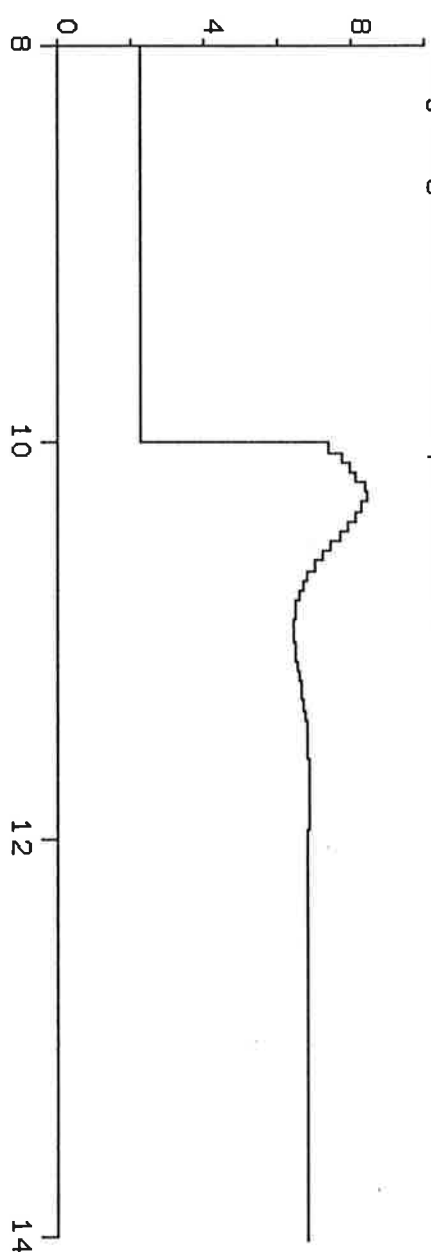
En diskret Smith-regulator implementeras lätt med hjälp av Simnon. Den reglerade process visas i figur 15 där Smith-regulatorn har integrationstiden 0.4 sekunder och förstärkningen 4.5 gånger.

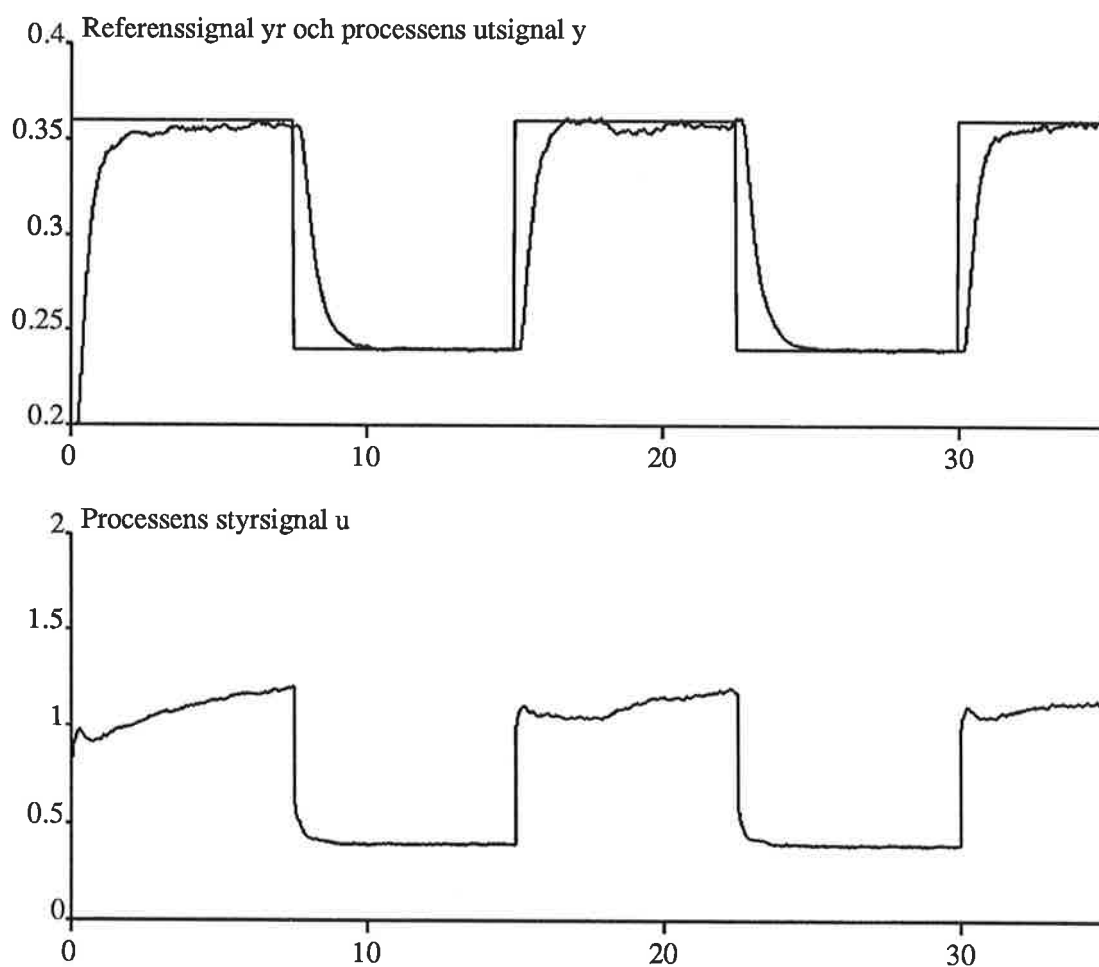


Figur 15: Fläktprocessen reglerad av en Smith-regulator

Eftersom processen dämpar insignalen  $u$  till en femtedel och denna är begränsad till intervallet 0-10 V, kommer referenssignalen  $y_{ref}$  inte att få överstiga 2 V.

Simuleras regleringen av modellen till fläktprocessen erhålles resultaten på sidan 15 där referenssignalen varierar mellan 0.5 och 1.5 V. Utsignalen gör en liten översläng innan den svänger in mot referensvärdet och styrsignalen håller sig innanför det tillåtna intervallet. Stigtiden bestämmas till cirka 0.4 sekunder vilket gör att systemet regleras snabbare än processen som hade en stigtid på 1 sekund. Figuren på sidan 16 visar resultatet efter att ha applicerat regulatorn på processen via realtidssimnon. Anledningen till att referensvärdena ligger som de gör, beror på att vid körning av realtidssimnon kan man endast variera referensvärdena mellan 0 till 1. Vidare ser man att följningen av referensvärdet inte ser ut som vid simuleringen. Detta beror på att processens parametrar inte ligger helt stilla, beroende på hur länge systemet har varit igång osv.

2 Referenssignalen  $y_{ref}$  och utsignalen  $y_d$ Styrsignalen  $u$  till processen



# Identifiering av kroppens balanssystem

Charlotta Johnsson, Doktorand  
Hans Nilsson, E-90  
Jerker Sjögren, E-89

Handledare: Rolf Johansson  
Institutionen för Reglerteknik

## Innehåll

<b>1. Inledning</b>	<b>3</b>
<b>2. Balanssystemet</b>	<b>4</b>
Receptorer — givare	4
Regulator — balanscentra + nervsystem	4
Ställdon — muskler	4
Kroppens ställning	4
<b>3. Experimentuppställning</b>	<b>6</b>
<b>4. Modeller</b>	<b>7</b>
Systemmodell	7
Processmodell	7
Regulatormodell	8
Modell av slutna systemet	9
<b>5. PRBS-signal</b>	<b>10</b>
<b>6. Experimentets utförande</b>	<b>11</b>
<b>7. Identifiering av slutna systemet</b>	<b>13</b>
Försök 1 — Två öppna ögon	13
Försök 2 — Slutna ögon	17
Försök 3 — Ett slutet och ett öppet öga	18
<b>8. Identifiering av balanssystemet</b>	<b>20</b>
<b>9. Resultat och utvärdering</b>	<b>21</b>

## 1. Inledning

I många medicinska sammanhang är kroppens balanssystem av stort intresse. I denna rapport har vi försökt att identifiera detta system genom olika mätningar gjorda hos neurofysiologiska institutionen vid Lunds Universitetssjukhus.

Målet med projektet var att identifiera kroppens balanssystem och att se hur detta påverkas av olika störningar. I vårt experiment har vi låtit försökspersonen ha både öppna och slutna ögon för att se hur synen påverkar balanssinnet. Vi har även ett försök med bara ett öga öppet för att undersöka djupseendets eventuella betydelse.

I avsnitt 2 ges en kort biologisk presentation av balanssystemet. Mer om experimentets uppställning och utörande återfinns i avsnitt 3 och 6. I avsnitt 4 behandlas de mekaniska och reglertekniska modellerna och i avsnitt 5 diskuteras den pålagda störningen. Slutligen, i avsnitt 7 och 8, identifieras och verifieras vår erhållna modell av kroppens balanssystem.

Modellbygge och validering har skett med hjälp av MATLABs Identification Toolbox.

## 2. Balanssystemet

För att få lite förståelse för hur kroppens balansfunktion biologiskt fungerar redogörs här kort för detta.

Kroppens balansreglering har, i likhet med många andra reglersystem, tre funktioner, nämligen:

**Givare** i form av receptorer. Dessa kan t.ex. vara ögon, innerörats bäggångar, m.m.

**Reglermekanism** som kan sägas vara balanscentret i hjärnan och nervsystemets funktion.

**Ställdon** i form av muskler.

### Receptorer — givare

I innerörat sitter örats bäggångar som kan liknas vid tre stycken halva toroider fyllda med vätska. Dessa sitter ortogonalt mot varandra och representerar kroppens koordinatsystem. I dessa finns givare i form av vestibularisorgan som, p.g.a. vätskans tröghet alstrar signaler då huvudet rör sig i "det rumsfasta koordinatsystemet". På så sätt kan kroppen uppfatta en vinkelacceleration. Dock är signalen från balansorganet mer proportionell mot hastigheten än mot accelerationen. Detta p.g.a. vätskans tröghet som introducerar en integrator mellan huvudrörelsen och nervimpulsen från balansorganet. (Kan ses som ett lågpas filter med en tidskonstant runt 2ms).

På vestibularisorganet sitter även s.k. öronstenar som har till uppgift att känna av statiskt läge och linjär acceleration. Det är dessa som gör att vi kan känna skillnad på upp och ner under inverkan av jordens tyngdacceleration.

Samtliga signaler är kopplade till ögonen vilket gör att man, trots huvudrörelser i olika riktningar, kan fixera ögonen på en bestämd punkt, till skillnad från t.ex. en videokamera som lätt kan uppfattas som "stötig".

### Regulator — balanscentra + nervsystem

Balanssystemet som ska modelleras styrs av balanscentret via nervsystemet och kan liknas vid en regulator.

### Ställdon — muskler

Kroppens ställdon utgörs av musklerna. Dessa får signaler från hjärnan via nervtrådar vilka gör att musklerna spänns (förkortas) eller slappnas av. De snabbaste nervimpulserna transporteras med en hastighet av ca 120 m/s. Med en kroppslängd på 1.70 m, blir tiden det tar för en nervimpuls att ta sig från balanscentret till vadmuskulerna ungefär 0.01 sekunder. Dock blir denna tid i praktiken något längre p.g.a. fördröjningar i nervsystemets synapser. Denna tidsfördröjning blir totalt sett relativt liten.

### Kroppens ställning

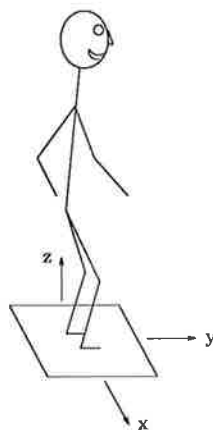
Balanscentret i hjärnan får information om kroppens position från olika håll, främst ifrån muskler, ledband och leder. Dock är det så att lederna inte har så hög prioritet. Om tex musklerna signalerar om en positionsförändring men inte lederna, så tror hjärnan på informationen från musklerna. Ett sätt att visa detta är att sätta vibratorer på vadmuskulerna. Vibrationerna gör att informationen från musklerna till hjärnan störs och hjärnan tror att musklerna är



längre än de egentligen är. I fallet med vadmuskelerna, uppfattar hjärnan detta som om en upprätt stående personen lutar framåt. Att fotlederna samtidigt skickar information om en oförändrad kroppsposition ignoreras av hjärnan, och hjärnan signalerar till musklerna: bli kortare!, i ett försök att komma ifrån vad den tolkat som en framåtlutande ställning.

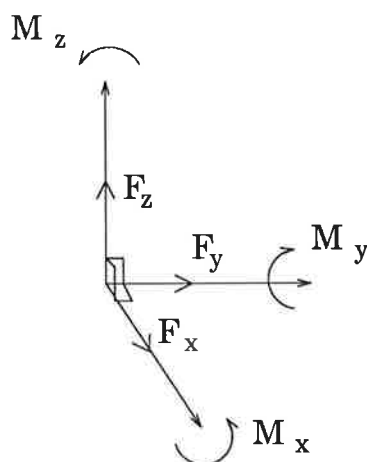
### 3. Experimentuppställning

För att kunna se hur kroppens balanssystem påverkas av olika typer av störningar har vi under experimenten haft tillgång till en s.k "balans-platta". En försöksperson står upprätt på plattan, vilken med hjälp av tre stycken trådtöjningsgivare mäter de krafter som försökspersonen ger upphov till då denna reglerar sin ställning. Uppställningen illustreras i figur 1.



Figur 1. Experiment med ståplatta.

Balansplattans tre trådtöjningsgivare är placerade under metallplattan i en liksidig triangel. På detta sätt kan krafterna i de tre riktningarna; x, y och z, mätas, se figur 2. De tillhörande momenten kan sedan beräknas.

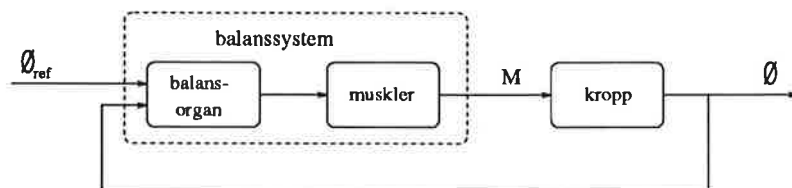


Figur 2. Krafter och moment som kan mätas och beräknas vid experimentuppställningen.

## 4. Modeller

### Systemmodell

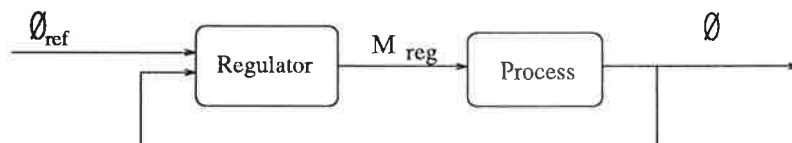
Systemet som skall undersökas är kroppen och dess balanssystem (balanscentra, nervsystem och muskler), se figur 3.



Figur 3. system

Insigheten till systemet är en referens-vinkel,  $\phi_{ref}$ , vilken normalt motsvarar stående lodrät ställning för försökspersonen,  $\phi_{ref} = 0$ . Kroppen känner av i vilken vinkel den verkligen står och med hjälp av felet, dvs skillnaden mellan dessa två vinklar,  $\phi - \phi_{ref} = \phi$ , beräknas ett moment. Musklerna i vaderna spänns eller slappnas av, beroende på tecknet på felet, och kroppens läge ändras.

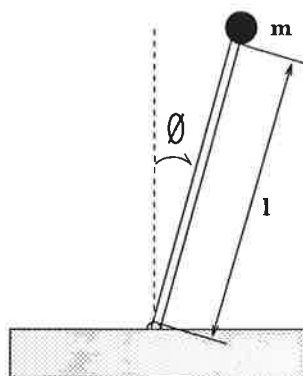
Ur reglerteknisk synvinkel kan kroppen betraktas som en process och balanssystemet kan tolkas som en regulator, se figur 4.



Figur 4. system

### Processmodell

Antag att kroppen kan modelleras som en inverterad pendel, d.v.s. kroppen anses helt stel och all massa samlad i en punkt, se figur 5.



Figur 5. Inverterad pendel

För en kropp av denna typ gäller:

$$J\ddot{\phi} = M \quad (1)$$

$$J = \int l^2 dm \quad (2)$$

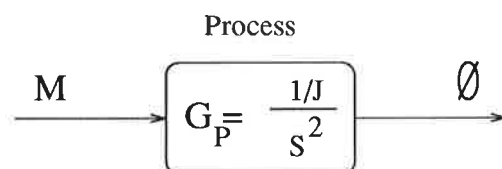
$$M = mgl \sin \phi \quad (3)$$

$$J\ddot{\phi} = M \implies J\ddot{\phi} = mgl \sin \phi \quad (4)$$

$$J\ddot{\phi} = M \implies ml^2\ddot{\phi} = mgl \sin \phi \quad (5)$$

Ur ekvation 1 kan överföringsfunktionen för processen, dvs den inverterade pendeln, tas fram:

$$\phi = \frac{1/J}{s^2} M \quad (6)$$



Figur 6. Processens överföringsfunktion

Vid små vinkelutslag,  $\phi \ll 1$ , gäller att  $\sin \phi \approx \phi$  och ekvation 5 kan då skrivas som:

$$ml^2\ddot{\phi} = mgl\phi \quad (7)$$

Denna differentialekvation har en lösning på formen:

$$\phi = A \sin \frac{2}{T} + B \cos \frac{2}{T}$$

där  $T$  är pendels periodtid:

$$T = 2\pi \sqrt{\frac{ml^2}{mgl}} = 2\pi \sqrt{\frac{l}{g}} \quad (8)$$

## Regulatormodell

Balanssystemet kan ses som systemets regulator. Insignalen till regulatorn är de båda vinklarna  $\phi$  och  $\phi_{ref}$  och dess utsignal är momentet  $M_{reg}$ . Med hjälp av felet, dvs skillnaden mellan vinklarna,  $\phi_{ref} - \phi = 0 - \phi = -\phi$ , beräknas således ett moment ut.

$$M_{reg} = -a\phi - b\dot{\phi} - c \int \phi dt \quad (9)$$

Kroppens balanssystem klarar av att mäta vinkelfelet,  $-\phi$ , ändringar i vinkelfelet,  $-\dot{\phi}$  och integralen av vinkelfelet,  $-\int \phi dt$ . En signal proportionell mot summan av dessa tre termer skickas från balanscentrat till musklerna.

Ur strukturen på ekvation 9 kan man se att balanssystemet kan liknas vid en PID-regulator.

### Modell av slutna systemet

För det slutna systemet gäller:

$$J\ddot{\phi} = M + M_{reg}$$

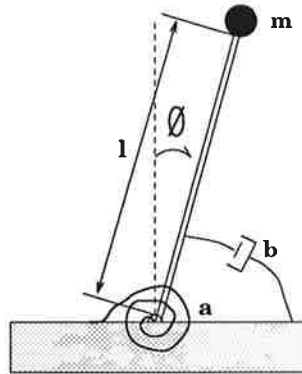
där  $M$  ges utav ekvation 3 och  $M_{reg}$  ges utav ekvation 9.

$$J\ddot{\phi} = mgl \sin \phi - a\phi - b\dot{\phi} - c \int \phi dt$$

$$J\ddot{\phi} + \underbrace{b\dot{\phi}}_1 + \underbrace{(a - mgl)\phi}_2 + \underbrace{c \int \phi dt}_3 = 0 \quad (10)$$

I denna ekvation kan term 1 tolkas som en dämpning, term 2 som en fjäderkraft och term 3 som en term vilken kompenserar för eventuella stationära fel.

Modellen för det slutna systemet kan nu ritas som en inverterad pendel utökad med dämpnings- och fjäder-egenskaper:

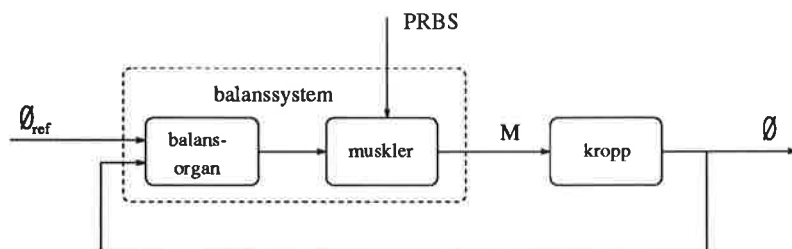


Figur 7. Mekanisk pendel

För att kunna identifiera systemet måste detta exciteras. Detta åstadkommes genom att en PRBS-signal (Pseudo Random Binary Sequence) läggs på.

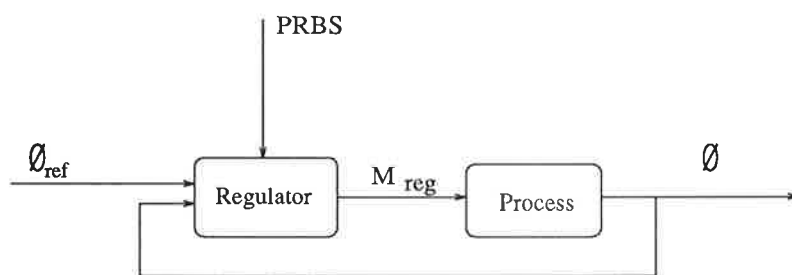
## 5. PRBS-signal

En vibrator, vilken alstrar PRBS-signaler, fästs på vardera vad. Dessa signaler ger upphov till störningar i vadmuskelns kommunikation med hjärnan och försökspersonen kommer att luta bakåt då vibratoren är på och sedan försöka nå upprätt ställning, då vibratorerna slår ifrån.



Figur 8. Systemet med pålaggd PRBS-signal

Dvs PRBS-signal ger upphov till en störning i systemets regulator.



Figur 9. Systemet med pålagd PRBS-signal

PRBS-signalen betecknas med  $v$ , störningen i regulatorn är proportionell mot denna och betecknas  $Kv$ . Ekvation 10 kan nu skrivas som:

$$J\ddot{\phi} + b\dot{\phi} + (a - mgl)\phi + c \int \phi dt - Kv = 0$$

Ur denna ekvation kan nu det slutna systemets överföringsfunktion plockas fram:

$$\phi(s) = \frac{\frac{1}{J}Ks}{s^3 + \frac{b}{J}s^2 + (a - mgl)\frac{1}{J}s + \frac{c}{J}}V(s)$$

Det är emellertid inte intressant att ha  $\phi$  som utsignal då det med balansplattan inte är möjligt att mäta denna. Utsignalen som balansplattan mäter är momentet  $M_{reg}$ . Enligt ekvation 4 gäller  $M_{reg} = J\ddot{\phi} - M = J\ddot{\phi} - mgl\phi$ , dvs:

$$M_{reg}(s) = (Js^2 - mgl)\phi(s)$$

Detta leder fram överföringsfunktionen:

$$M_{reg}(s) = \frac{\frac{1}{J}K(Js^2 - mgl)s}{s^3 + \frac{b}{J}s^2 + \frac{a - mgl}{J}s + \frac{c}{J}}V(s) \quad (11)$$

## 6. Experimentets utförande

Försökspersonens uppgift då denne står på plattan är endast att hålla balansen.

Försökspersonen har raka knän och är stel i höften, dvs endast vadmusklerna och tårna kan användas för att reglera den stående ställningen. För att reglera ställningen spänns eller slappnas vadmusklerna av, detta medför att tårna trycker med en viss kraft på plattan och det är denna kraft som mäts. Händerna hålls hela tiden stilla på ryggen. Anledningen till att knän och höft hålls stela och händer stilla intill kroppen är att den matematiska modellen av försökspersonen då kommer att var relativt enkel.

På försökspersonens båda vader fästs vibratorer. . Vibratorerna alstrar en PRBS-signal med frekvensen 50 Hz och effekten 120 mW. Dessa ger upphov till försökspersonens svaj (se kap 2)

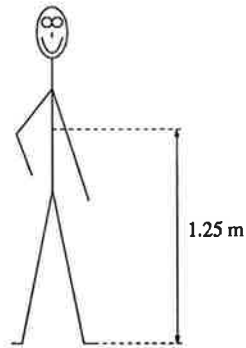
Under försökets har försökspersonen freestyle på sig där klassisk musik spelas. Detta för att utestänga yttre ljud och på så sätt eliminera hörselns eventuella inverkan på balansen.

Tre olika experiment gjordes:

1. Båda ögonen öppna, två ben på plattan och PRBS-signal med effekten 120 mW.
2. Båda ögonen slutna, två ben på plattan och PRBS-signal med effekten 120 mW.
3. Ett öga (vänster) öppet och ett slutet (höger), två ben på plattan och PRBS-signal med effekten 120 mW.

Varje experiment varade i 210 sekunder, varav PRBS-signalen under de 30 första sekunderna inte var inkopplad.

För att välja samplingsintervall,  $h$ , började vi med att beräkna pendelns (processens) tidskonstant. Denna kan beräknas med ekvation 8. Pendelns längd,  $l$ , valdes till 1.25m, vilket motsvarar övre magtrakten på försökspersonen, se figur 10.



**Figur 10.** Val av pendels längd

Pendelns periodtid:

$$T = 2.24 \text{ sek}$$

En bra tumregel vid val av samplingsintervall är:

$$0.2 \leq \omega h \leq 0.6$$

vilket i vårt fall medför:

$$\omega = \frac{2\pi}{T} \Rightarrow 0.07 \leq h \leq 0.2$$

Vi valde samplingsintervallet:

$$h = 0.1$$

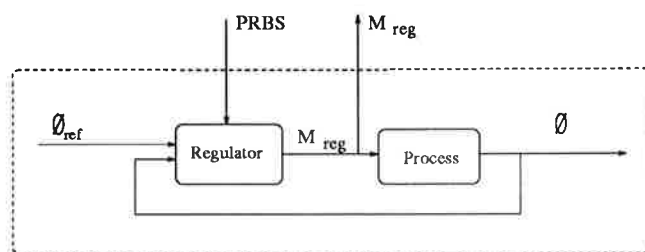


## 7. Identifiering av slutna systemet

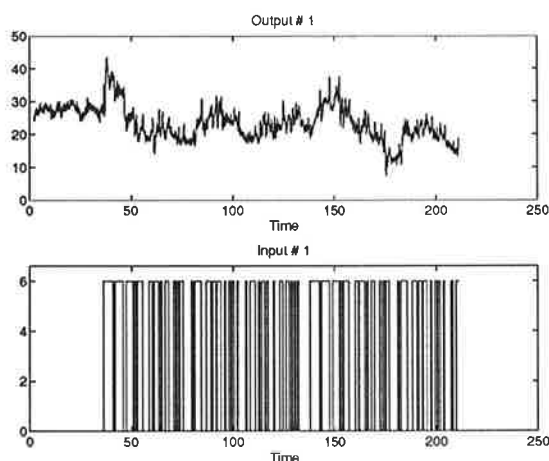
Slutna systemet har två insignaler,  $\phi_{ref}$  och  $PRBS$ -signalen.  $\phi_{ref}$  kommer från hjärnan och talar om att vi önskar stå rakt upp,  $PRBS$ -signalen kommer utifrån. Utsignalen är  $\phi$  vilken talar om i vilken vinkel vi står. Denna går emellertid inte att mäta utan istället väljs signalen  $M_{reg}$  som utsignal.

$PRBS$ -signalen som vibratorn genererar tolkas av hjärnan som om vad-musklerna vore längre än de är. Genom att trycka med tårna eller hälen mot golvet försöker försökspersonen reglera sin stående ställning. Denna kraft mäts och räknas om till ett moment,  $M_{reg}$ . In- och ut-signal visas i figur 12.

Systemet som skall identifieras har alltså  $PRBS$ -signalen som insignal och momentet  $M_{reg}$  som utsignal. Detta kan illustreras med blockschemat:



Figur 11. Slutna systemet



Figur 12. In- och ut-signal

Under de första 30 sekunderna av experimentet skickas ingen insignal in. Detta för att kunna identifiera bakgrundsbruset.

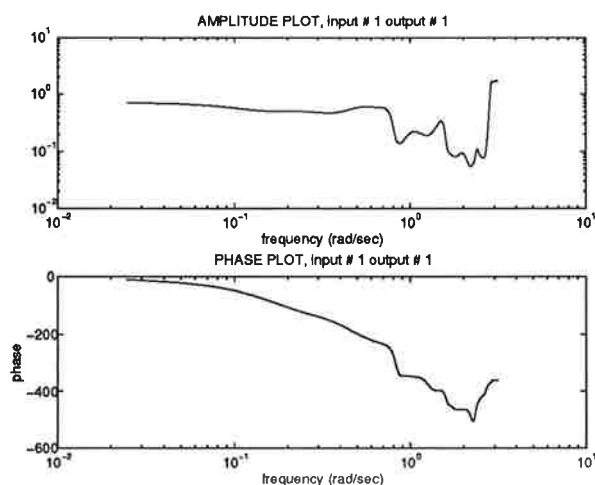
För att slippa systematiska fel vid identifieringen subtraheras in- och ut-signal med sina resp. medelvärden. Signalen är sedan lågpas-filtrerad med ett åttonde ordningens Tjebysjev-filter för att slippa vikningsfenomen. I den fortsatta analysen används de filtrerade versionerna av signalerna.

Det bästa av våra försök, med avseende på koherensspektrum, visade sig vara det med båda ögonen öppna och hög amplitud på vibrationerna. Därför analyseras detta försök grundligt i följande avsnitt och sedan följer en kort jämförelse med de övriga experimenten.

### Försök 1 — Två öppna ögon

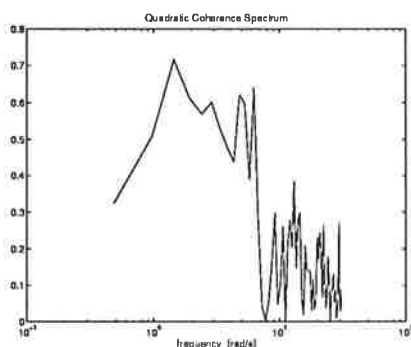
För att få en uppfattning om hur signalerna ser ut i frekvensplanet görs en bodeanalys samt en koherensanalys. Bodediagrammet för systemet visas i figur

13.



Figur 13. Amplitud- och fas-kurva

Frekvensintervall lämpligt för skattning ges av koherensfunktionen enligt figur 14. Denna visar att det är möjligt att göra en bra skattning i intervallet  $[0.8 \ 8]$  rad/sek.



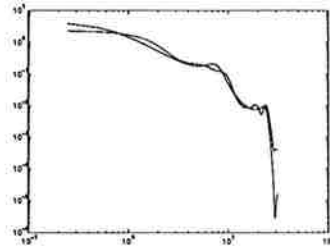
Figur 14. Coherens-kurva

Trots att det under de 30 första sekunderna i experimentet inte genereras någon *PRBS*-signal, dvs insignalen är identiskt lika med noll, så varierar utsignalen. Detta beror på att försökspersonen naturligt står och svajar lite. Detta naturliga svaj kan modelleras som brus med ARMA-modellen,

$$Ay(t) = Ce(t)$$

Den bästa modellen visade sig vara  $\deg(C) = 4$ ,  $\deg(A) = 6$ , vilken har använts i det fortsatta arbetet.

Spektrum för det verkliga och det modellerade bruset visas i figur 15



**Figur 15.** Spektrum för det verkliga och det modellerade bruset

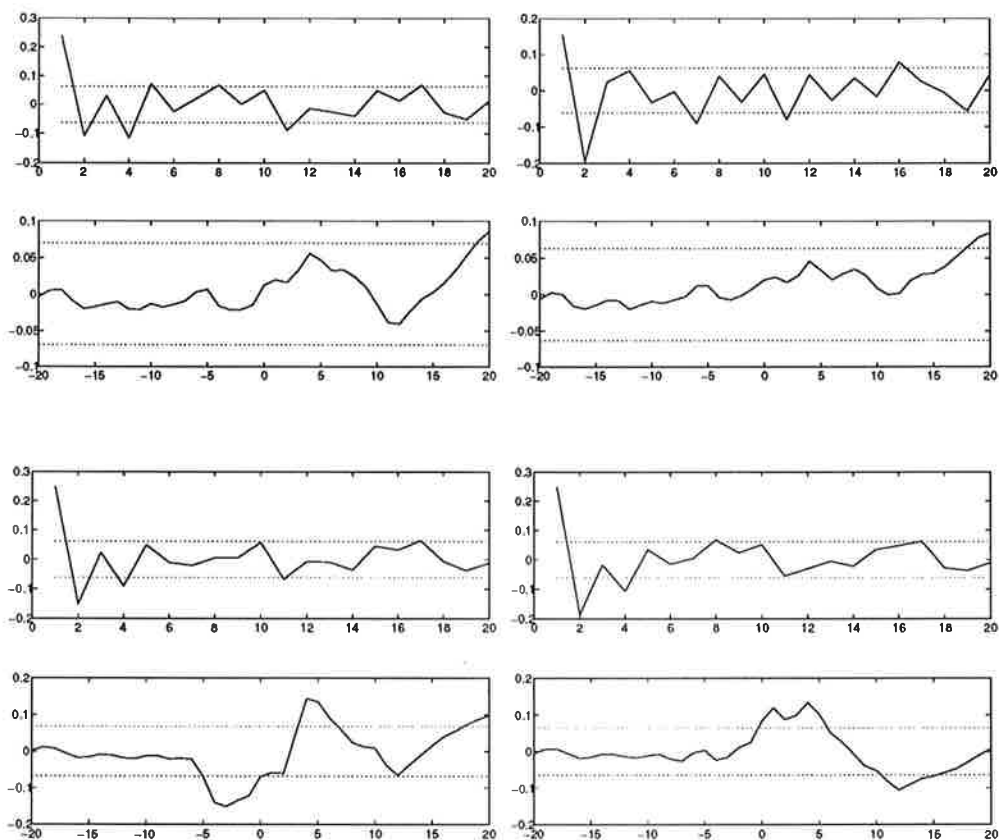
Box-Jenkins modell väljes sedan för att modellera det slutna systemet. Detta ger en möjlighet att modellera bruset dynamik oberoende av systemets.

$$y(t) = \frac{B}{F}u(t) + \frac{C}{D}e(t) \quad (12)$$

I avsnittet "Modeller" plockades en teoretisk modell för systemet fram med utseendet:

$$M_{reg} = \frac{\frac{1}{J}K(Js^2 - mgl)s}{s^3 + \frac{b}{J}s^2 + \frac{a-mgl}{J}s + \frac{c}{J}}V(s) \quad (13)$$

Den teoretiska modellen gör det troligt att systemets ordning ligger nära tre. Följdaktligen ansättes olika modeller omkring ordning tre vilket gav residualtester och pol- nollställe-diagram enligt figur 16 och 17. De fyra modellerna kallas för th3463, th3464, th2462 samt th2463. Index i namnet motsvaras av graden på B, C, D och F polynomen i modell 12.

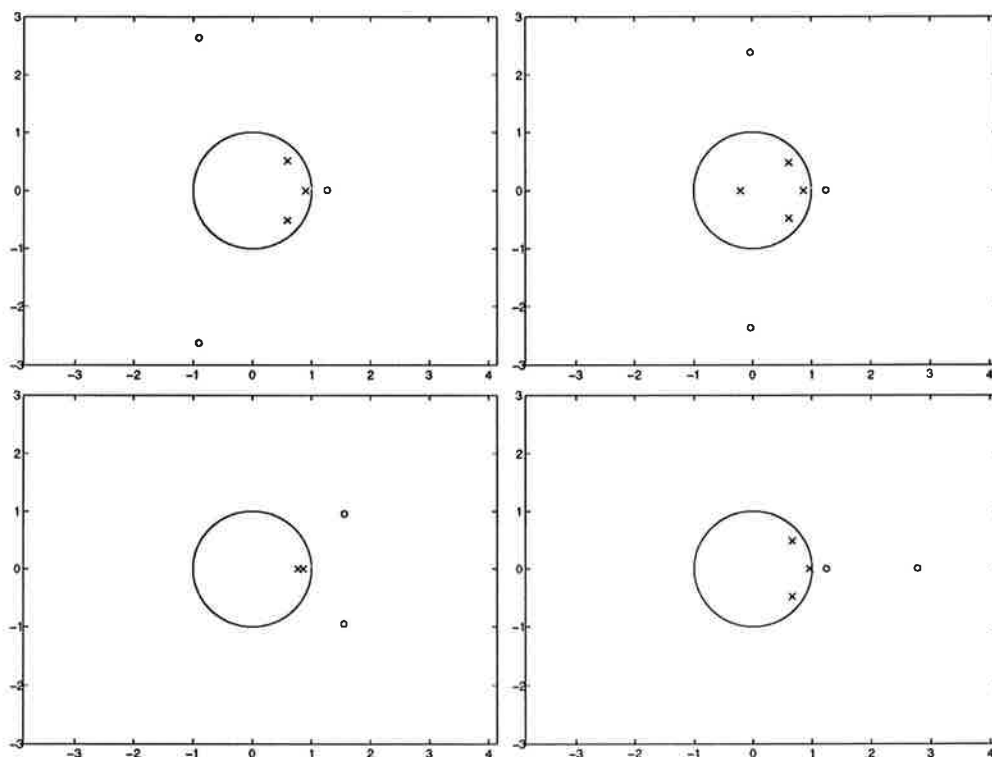


**Figur 16.** Residualtester med 99% konfidensintervall för våra modeller. Översta testen visar autokorrelationen och den undre visar korskorrelationen. Modellen (Ö.t.v.) svarar mot bj3463. (Ö.t.h.) bj3464, (N.t.h.) bj2462 samt (N.t.h.) bj2463.

Man ser att modell bj3463:

$$\deg(B) = 3, \deg(C) = 4, \deg(D) = 6, \deg(F) = 3$$

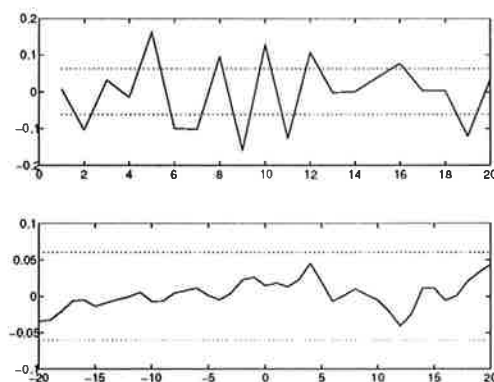
är den som verkar bäst. Detta stämmer bra överens med vår teoretiska modell.



**Figur 17.** Pol- nollställekonfiguration för processmodellerna. Brusets poler och nollställen finns alltså inte med. Modellen (ö.t.v.) svarar mot bj3463. (Ö.t.h.) bj3464, (n.t.h.) bj2462 samt (n.t.h.) bj2463.

## Försök 2 — Slutna ögon

Experiment på en person med slutna ögon gjordes för att se hur ögonen påverkar balanssystemet. Samma modell som tidigare ansattes, d.v.s. bj3463.

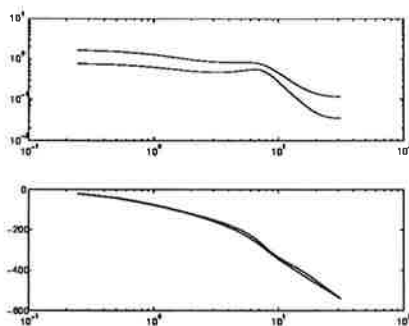


**Figur 18.** Autokorrelation (ovan) respektive korskorrelation (nedan) med 99 % konfidensintervall för vår modell med slutna ögon.

Som autokorrelationen i figur 18 visar, är detta ingen fullständig modell vilket kanske kan förklaras med att viss högfrekvent information finns kvar i utsignalen. Modellen används ändå eftersom den anses tillräckligt bra för fortsatta analyser.

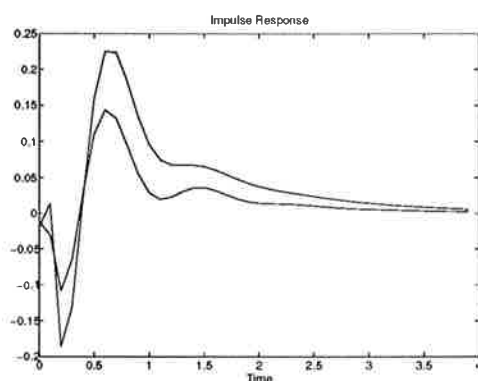
Bodediagrammet visar att det slutna systemet får en högre förstärkning främst i det höga och låga frekvensintervallet när man sluter ögonen, figur 19. Detta kan eventuellt förklaras med att man får en större osäkerhet i kroppens

lägeskontroll, d.v.s. kroppen reagerar ryckigare.



**Figur 19.** Bodediagram för slutna ögon (övre kurvan) samt öppna ögon (nedre kurvan).

En jämförelse mellan impulssvaren bekräftar bodediagrammet. Det slutna systemet, för försök 2, har en resonanstopp kring 1 Hz, samt en något högre förstärkning än tidigare försök, och är därmed ett något svängigare system. Se figur 20.

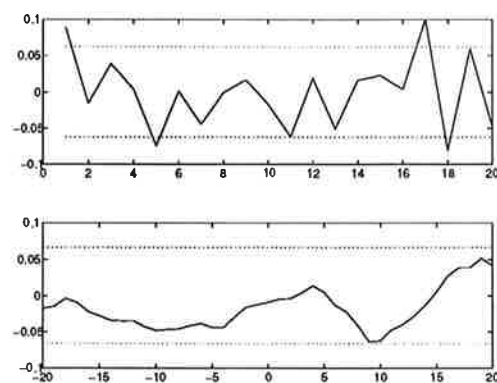


**Figur 20.** Impulssvar för systemet med slutna ögon (övre kurvan) samt öppna ögon (nedre kurvan)

### Försök 3 — Ett slutet och ett öppet öga

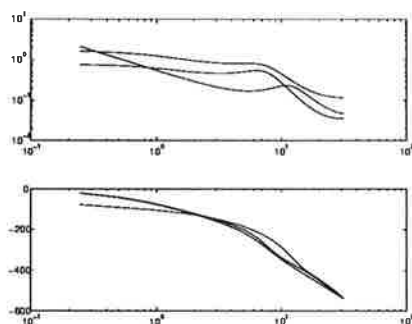
Det tredje experimentet utfördes på en försöksperson med bara ett öga öppet. Detta gjordes för att se om djupseendet eventuellt har någon inverkan på balanssystemet.

Residualanalys av systemet visar att modellen från tidigare försök stämmer väl överens med systemet, figur 21.

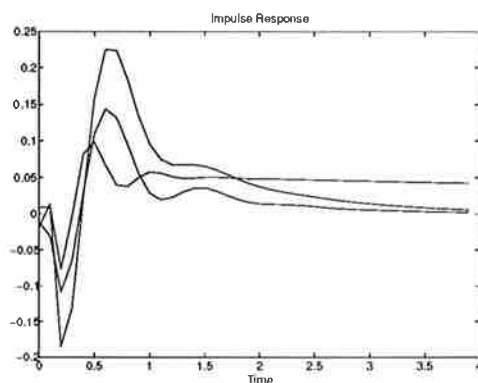


**Figur 21.** Autokorrelation (ovan) respektive korskorrelation (nedan) med 99 % konfidensintervall för vår modell med ett öga öppet.

Bodediagrammet och impulssvaret visas i figur 22 resp. figur 23 i jämförelse med de två tidigare försöken.



**Figur 22.** Bodediagram för samtliga tre försök med modell bj3436. Den avvikande kurvan med en resonanstopp kring 10 rad/sek visar systemet med ett öga.



**Figur 23.** Impulssvar från de tre försöken.

Man noterar speciellt att systemet med ett öga har ett statiskt fel till skillnad från de andra två systemen. Detta kan tänkas bero på kroppens svårighet att hitta lodlinjen då djupseendet störs. Man kan tänka sig att hjärnan uppfattar det som att den tar emot syninformation som är korrekt trots att så inte är fallet.

## 8. Identifiering av balanssystemet

I våra tidigare analyser har konstaterats att balanssystemet kan liknas vid kroppens regulator. I detta avsnitt tas regulatorparametrarna ut från det slutna systemet enligt följande:

Försök	$s^3 + \frac{b}{J}s^2 + \frac{a-mgl}{J}s + \frac{c}{J}$
Öppna ögon	$s^3 + 5.02s^2 + 60.98s + 39.20$
Ett öga	$s^3 + 7.78s^2 + 138.27s + 11.68$
Slutna ögon	$s^3 + 7.58s^2 + 64.20s + 54.27$

$$M_{reg} = - \underbrace{a}_{K_c} \phi - \underbrace{b}_{K_c T_D} \dot{\phi} - \underbrace{c}_{\frac{K_c}{T_i}} \int \phi dt$$

Identifiering och normering ger:

Försök	$K_c$	$K_c/T_i$	$K_c T_d$
Öppna ögon	5.9	3.4	0.4
Ett öga	12.5	1.0	0.7
Slutna ögon	6.2	4.7	0.7

Vi kan bl.a. notera en liten ökning av I- och D-parametrarna då man sluter ögonen. Vidare kan vi också se hur liten integraldelen är då djupseendet tas bort. Detta bekräftas även av figur 23 då man kan se ett stationärt fel.



## 9. Resultat och utvärdering

Vår modellansats med en regulator och en process visade sig stämma väl överens i de flesta fall. Tittar man på residualanalyserna kan man dock notera att viss högfrekvent information inte är fullt modellerad. Det kan tänkas att bruset är mer komplicerat än vi har trott.

Processmodellen är kanske även den i enklaste laget då kroppen endast modelleras som en inverterad pendel. Här skulle man naturligtvis kunna ansätta en mer komplex pendel för att modellera kroppens alla leder. Komplexiteten ökar dessvärre explosionsartat och beräkningarna skulle förmodligen falla p.g.a. numeriska problem.

I denna rapport har synens inverkan på kroppens balanssystem undersökts. Balanssystemet kan modelleras som en PID-regulator enligt:

$$M_{reg} = - \underbrace{a}_{K_c} \phi - \underbrace{b}_{K_c T_D} \dot{\phi} - \underbrace{c}_{\frac{K_c}{T_i}} \int \phi dt$$

Tre olika försöksserier har gjorts:

1. Båda ögonen öppna
2. Båda ögonen slutna
3. Ett öppet öga och ett slutet

För att titta närmare på hur kroppen adapterar sig då synförmågan ändras kan regulatorparametrarna vid de tre olika försöken studeras. I tabellen nedan är parametrarna översatta till normerade PID-parametrar.

Försök	$K_c$	$K_c/T_i$	$K_c T_d$
Öppna ögon	5.9	3.4	0.4
Ett öga	12.5	1.0	0.7
Slutna ögon	6.2	4.7	0.7

Man kan bl.a. notera en ökning av I- och D-parametrarna då man sluter ögonen. Vidare kan vi också se hur liten integraldelen är då djupseendet tas bort.

Av det material som vi här har presenterat kan man inte dra några direkta samband mellan själva regulatorn och det slutna systemet och vi ber läsaren att vara observant på detta. Vidare har vi konstaterat att koherensen har varit medelmåttig i vårt skattade intervall vilket naturligtvis gör analysen osäker.

Institutionen för Reglerteknik  
Lunds Tekniska Högskola

# Neurala nätverk för Identifiering och Reglering

Projekt i Adaptiv Reglering och Processidentifiering

Lund April 1994

Jesper Bergstrand  
Tord Björsne  
Handledare: Ulf Jönsson

## **Innehållsförteckning:**

- 1. Inledning
- 2. Neurala Nätverk
  - 2.1 Neuronen
  - 2.2 Topologi
  - 2.3 Prestandamått
  - 2.4 Konvergenshastighet och lokala minima
  - 2.5 Parametrar
- 3. Reglering mha Neurala Nätverk
  - 3.1 Emulatorn
  - 3.2 Träning av regulatorn
- 4 Andra ordningens dåligt dämpade system
  - 4.1 Styrbarhet
  - 4.2 Deadbeatregulatorn
  - 4.3 Resultat
  - 4.4 Olika parametrars betydelse
- 5 Styrning av backande lastbil
  - 5.1 Identifiering
- Referenser
- A Bilagor

## 1. Inledning

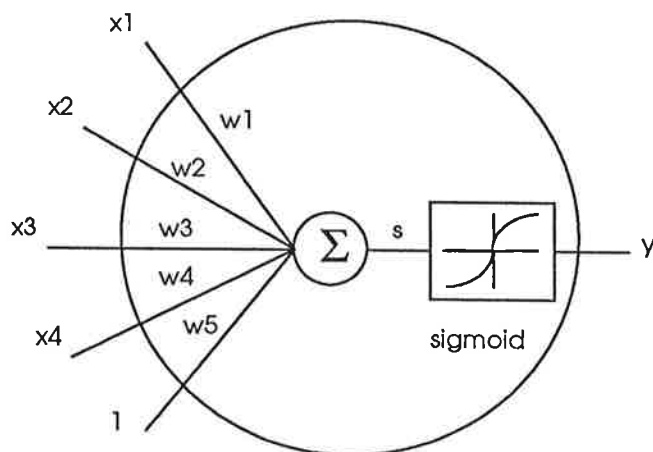
Denna rapport är resultatet av ett kombinerat projekt i kurserna Processidentifiering och Adaptiv Reglering på Institutionen för Reglerteknik vid Lunds Tekniska Högskola.

Vi undersöker här en tillämpning av neurala nätverk för identifiering och styrning av olinjära dynamiska system vars samtliga tillstånd är mätbara. Arbetet är ett försök att reproducera de resultat som redovisas i en artikel av Hguyen och Widrow [2].

## 2. Neurala nätverk

När man nämner "neurala nätverk" (n.n.) är det många som reagerar starkt, mycket beroende på okunskap. Vi skulle vilja avdramatisera ämnet något. Som vi ser det är n.n. i grunden en metod att approximerar olinjära funktioner. Det som gör den speciell är att man tränar in dess funktion m h a exempel. Ofta använder man n.n. när den funktion man försöker approximerar är okänd. Man kan alltså med god fog också påstå att n.n. är en identifieringsmetod. Ur den synpunkten är metoden speciellt intressant eftersom den "automatiskt" hanterar även olinjära system.

Det finns många användningsområden för n.n. Ett av de vanligaste är för klassificering. För många problem är det svårt eller för kostsamt att finna analytiska lösningar. I den här rapporten undersöker vi möjligheten att använda n.n. för identifiering och reglering av dynamiska system. Vi har utgått från två artiklar. Den första [1] är en översikt av grunderna medan [2] behandlar en speciell metod för olinjär reglering.



Figur 1. Neuronens uppbyggnad.

### 2.1 Neuronen

Neurala nätverk är uppbyggda av beräkningselement som kallas neuroner. Dessa är sinsemellan sammankopplade så att utgångarna från vissa neuroner är kopplade till ingångarna på andra. I denna rapport undersöker vi bara "feed-forward" nät, d v s sådana som inte är återkopplade. Som framgår av figur 1 multipliceras insignalerna med vikter. De viktade insignalerna summeras och resultatet är insignal till en olinjär funktion, i vårt fall en s.k. sigmoid. Denna funktion har egenskaperna att den är nära linjär i närheten av origo för att mjukt plana ut mot 1 eller -1 när argumentet går mot stora positiva resp. negativa värden. Observera att man vanligtvis låter en av insignalerna vara konstant ett. Detta möjliggör bias, vilket är nödvändigt för att kunna approximerar allmänna olinjära funktioner. Utsignalen kan lätt ställas upp som

$$y = \text{sgm}(WX) \quad (1)$$

där viktvektorn  $W$  och insignalvektorn  $X$  har utseendet

$$W = [w_1 \quad w_2 \quad \dots \quad w_M], \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}$$

och  $\text{sgm}(\cdot)$  är sigmoidfunktionen som i vårt fall är lika med  $\tanh(\cdot)$ . Man ser att utsignalen bestäms entydigt av viktvektorn  $W$ .

Önskad utsignal från neuronerna erhålles som tidigare nämnts genom träning med par av insignaler och önskade utsignaler. Innan träningen påbörjas initieras vikterna slumpmässigt. Vi har, tämligen godtyckligt, valt att använda en normalfördelning med en standardavvikelse på ca 0.5 och väntevärdet 0. Första steget i varje träningsomgång är att beräkna utsignalen  $y$  från neuronerna för en given insignal  $X$  m h a (1). Därefter beräknas felet  $\varepsilon = d - y$ , där  $d$  är den önskade utsignalen. Till sist justeras vikterna  $W$  på sådant sätt att felet minskas något. Viktjusteringen bestäms med en minstakvadratfelsmetod. Vikterna justeras i motsatt riktning mot det kvadratiske felets gradient. Denna är naturligtvis inte känd men kan uppskattas på följande vis. Vi vill minimera förlustfunktionen

$$J = \frac{1}{2} \varepsilon^2 = \frac{1}{2} (d - y)^2$$

En uppskattning av felgradienten är

$$\hat{\nabla} = \frac{\partial J}{\partial W} = \frac{1}{2} \frac{\partial \varepsilon^2}{\partial W} = \varepsilon \frac{\partial \varepsilon}{\partial W} = \varepsilon \frac{\partial (d - y)}{\partial W} = -\varepsilon \frac{\partial y}{\partial W}$$

som kan utvecklas till

$$\hat{\nabla} = -\varepsilon \frac{\partial \text{sgm}(WX)}{\partial W} = -\varepsilon \text{sgm}'(WX) X^T$$

Detta ger viktjusteringen

$$W_{k+1} = W_k - \mu \hat{\nabla}_k = W_k + \mu (d_k - y_k) \text{sgm}'(W_k X_k) X_k^T \quad (2)$$

Där  $k$  är tidsargumentet och  $\mu$  är en parameter som avgör konvergenshastighet och stabilitet. Valet av adaptionsförstärkning  $\mu$  kan vara kritiskt. Om den väljs för låg konvergerar neuronerna för långsamt, för stor förstärkning ger instabilitet.

Det är intressant att jämföra neuronerna med adaptiva modellreferenssystem (MRAS). Dessa behandlas utförligt i text [3]. Där ges den så kallade MIT-regeln för uppdatering av vikter i kontinuerlig tid

$$\frac{d\theta}{dt} = -\gamma \varepsilon \frac{\partial \varepsilon}{\partial \theta}$$

Detta motsvarar exakt den metod som vi använder. Vi har funnit att uppdateringsformeln (2) är svåränvänd. En förklaring kan vara att konvergenshastighet och stabilitet beror på insignalens amplitud, se exempel 4.4 i [3]. För att komma runt detta problem använder vi en motsvarighet till den normaliserade MIT-regeln

$$\frac{d\theta}{dt} = -\gamma \frac{\varepsilon \frac{\partial \varepsilon}{\partial \theta}}{\alpha + \left( \frac{\partial \varepsilon}{\partial \theta} \right)^T \left( \frac{\partial \varepsilon}{\partial \theta} \right)}$$

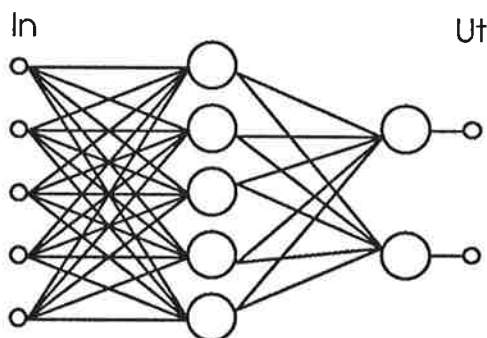
som också ges i [3]. Denna uppdateringsformel är okänslig för insignalamplituden. Konstanten  $\alpha > 0$  har införts för att undvika division med noll. Den ges också minskad förstärkning vid små insignaler.

I [1] ges en annan förbättrad variant av uppdateringsformeln. Där normaliseras felgradienten så att den alltid har samma Euklidiska längd. Vikterna justeras alltså lika långt för varje träningsiteration, oberoende av felets storlek eller insignalamplituden. Vi har inte provat denna metod men nämner den som referens.

Det är intressant att notera att neuronerna kan betraktas som linjär för små ut signaler. Då kan resultat från analys av linjära minsta kvadratmetoder användas med god noggrannhet. Man inser bl a att neuronerna (med brusfria in- och ut signaler) kan tränas till godtycklig noggrannhet, om bara insignalerna väljs tillräckligt små. Man kan också notera att färgat brus ställer till problem.

## 2.2 Topologi

Neurala nätverk skapas genom att flera neuroner förbinds. Detta kan ske på många olika sätt. Emellertid finns det ett trevligt resultat, refererat i [1], som säger att alla "tillräckligt snälla" olinjära funktioner kan approximeras godtyckligt noggrant med ett n.n. enligt figur 2, bara det finns tillräckligt många neuroner i första lagret. Andra mer komplicerade topologier används dock också då de kan ha bättre egenskaper när det t ex gäller konvergenshastigheter. Vi har valt att begränsa oss till den topologi som figur 2 visar, bl a därför att den är lätt att formulera på matrisform.



Figur 2. Topologi för neuralt nätverk.

Låt insignalen  $X$ , utsignalen från lager 1 och 2,  $Y_1$  respektive  $Y_2$ , vara kolonvektorer och låt vikterna i lager 1 och 2 vara matriserna  $W_1$  respektive  $W_2$ .

$$X = Y_0 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \quad Y_1 = \begin{bmatrix} y_{11} \\ y_{12} \\ \vdots \\ y_{1L} \end{bmatrix} \quad Y_2 = \begin{bmatrix} y_{21} \\ y_{22} \\ \vdots \\ y_{2N} \end{bmatrix}$$

$$W_1 = \begin{bmatrix} w_{111} & w_{112} & \cdots & w_{11M} \\ w_{121} & w_{122} & \cdots & w_{12M} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1L1} & w_{1L2} & \cdots & w_{1LM} \end{bmatrix} \quad W_2 = \begin{bmatrix} w_{211} & w_{212} & \cdots & w_{21L} \\ w_{221} & w_{222} & \cdots & w_{22L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{2N1} & w_{2N2} & \cdots & w_{2NL} \end{bmatrix}$$

Utsignalen kan då skrivas som

$$Y_1 = \text{sgm}(W_1 X)$$

$$Y_2 = \text{sgm}(W_2 Y_1)$$

eller kortare som

$$Y_2 = \text{sgm}(W_2 \text{sgm}(W_1 X)) \quad (3)$$

Sigmoidfunktionen i formeln ovan verkar elementvis på argumentvektorena. Formeln för träning av nätet härleds lätt på samma sätt som för den enkla neuron. För andra lagret blir formeln identiska, med undantaget att  $W_2$  och felgradienten nu är matriser istället för kolonvektorer. Givet den önskade utsignalen  $Y_d$  får vi utsignalfelet  $\epsilon_2 = Y_d - Y_2$ . Med detta erhålls

$$W_2(k+1) = W_2(k) - \mu \hat{V}_2(k) = W_2(k) + \mu \epsilon_2(k) \text{sgm}'(W_2(k)Y_1(k))Y_1(k)^T$$

Första lagrets vikter justeras på samma sätt. Problemet är att beräkna vilket fel  $\epsilon_1$  som ska användas, vi har ju inte något utsignalfel att tillgå. Vi provar att beräkna felgradienten för utsignalfelet med avseende på viktmatrisen  $W_1$  för första lagret.

$$\hat{V}_1 = \epsilon_2 \frac{\partial \epsilon_2}{\partial W_1} = -\epsilon_2 \frac{\partial \text{sgm}(W_2 Y_1)}{\partial W_1} = -\epsilon_2 \text{sgm}'(W_2 Y_1) W_2^T \frac{\partial Y_1}{\partial W_1}$$

Om vi definierar det ekvivalenta felet  $\epsilon_1$  för första lagret som

$$\epsilon_1 = \epsilon_2 \text{sgm}'(W_2 Y_1) W_2^T$$

erhålls felgradienten och uppdateringsformeln

$$\hat{V}_1 = -\epsilon_1 \frac{\partial Y_1}{\partial W_1} = -\epsilon_1 \frac{\partial \text{sgm}(W_1 X)}{\partial W_1} = -\epsilon_1 \text{sgm}'(W_1 X) X^T$$

$$W_1(k+1) = W_1(k) - \mu \hat{V}_1(k) = W_1(k) + \mu \epsilon_1(k) \text{sgm}'(W_1(k)X(k))X(k)^T$$

Den algoritm som vi visat brukar benämnas felbakåtpropagering (eng. back-propagation). Namnet kommer sig av att utsignalfelet sprids bakåt i nätet på det sätt som vi visat ovan. Observera att man på samma sätt som ovan kan beräkna ett ekvivalent insignalfel

$$\epsilon_0 = \epsilon_1 \text{sgm}'(W_1 X) W_1^T$$

Detta fel används i algoritmen nedan för träning av en neural regulator.

## 2.3 Prestandamått

För att bedöma det neurala nätets prestanda har vi funnit det naturligt att använda oss av medelkvadratiskfelet. Det är ju detta som felbakåtpropageringen minimerar. Vi har valt att betrakta utsignalfelet som en brusterm och använder därför naturligt signalbrusförhållandet (SNR) som prestandamått. Med de beteckningar vi använt ovan får vi

$$SNR = 10 \cdot \log \left( \frac{\sum_k \epsilon_2(k)^T \epsilon_2(k)}{\sum_k Y_d(k)^T Y_d(k)} \right)$$

uttryckt i decibel.

## 2.4 Konvergenshastighet och lokala minima

Vi har i resonemanget ovan utelämnat en del viktiga kommentarer när det gäller träning av neurala nätverk. För det första är det viktigt att påpeka att det kan krävas ganska många in/utsignalpar för att nätet ska konvergera till önskad precision. Det verkar vara så att ju grövre olinjäriteter man försöker lära nätet approximera, ju mer träning krävs.

En annan aspekt som vi inte nämnt är hur den kvadratiske felytan ser ut. Det är svårt att visualisera denna hyperyta eftersom den har lika många dimensioner som antal vikter i nätet. Ändock är det denna yta som är nyckeln till förståelse för

många av de problem som man råkar ut för när man jobbar med n.n. Det första man måste inse är att denna yta oftast har lokala miniman. Detta ställer till väldigt mycket problem om man använder den vanliga felgradientmetod som vi beskrivit här. Det är nämligen mycket lätt hänt att man fastnar i lokala miniman med den typen av algoritmer. Detta orsakar bland annat det kan vara svårt att reproducera sina resultat. Eftersom viktmatriserna initieras slumpmässigt kan man mycket väl hamna i ett annat minima om man gör om träningen, trots att man använder samma indata! Vi har inte funnit någon bra metod att hantera detta problem.

## 2.5 Parametrar

Det finns en hel del parametrar som måste väljas när man ska träna ett neuralt nätverk. Lämpliga val av dessa kan vara gynnsamt för prestanda och konvergens.

### Val av sigmoidfunktion

Vi har bara tittat på en faktor när det gäller sigmoidfunktionen, nämligen maximal utsignal. Säg att vi önskar en utsignal som har omfånget  $(-a, +a)$ . Vi har då använt

$$sgm(x) = a \tanh(x / a)$$

$$sgm'(x) = 1 - \tanh^2(x / a)$$

### Antal neuroner i första lagret

Detta är en mycket viktig parameter som avgör hur pass grova olinjäriteter som nätet kan approximera. Ju komplexare funktion, ju fler neuroner krävs. Man kan föreställa sig att varje neuron approximerar ett litet segment av funktionen. Om funktionen är besvärlig inser man att många segment krävs för god approximation (jfr linjär interpolering).

### Adaptionsförstärkningen $\mu$

Adaptionsförstärkningen är definitivt kritisk om man använder den enkla uppdateringsformeln (2). Vår erfarenhet är dock att den modifierade MIT-regeln ger ganska stora friheter vid val av  $\mu$  och rekommenderar därför att denna eller någon annan modifierad uppdateringsformel används.

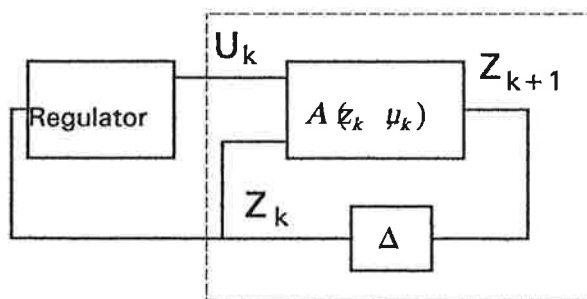
### Initieringen av viktmatriserna

Viktmatriserna initieras med normalfördelade slumpstal med väntevärde noll. Vi har provat olika varianser men inte funnit att valet det är kritiskt. Man kan inse att längre konvergenstider erhålles om variansen är mycket liten. Om vikterna är små blir nämligen felgradienten också liten. För stora vikter ger mättade utsignaler vilket också resulterar i långsam konvergens. Också i detta fall beror det på att felgradienten blir liten, ty sigmoidens derivata (som är liten när utsignalerna är mättade) är en faktor i gradienten.



### 3 Reglering

En allmän tillståndsrepresentation av ett system visas i figur 3.



figur 3 Process och regulator

Vektorn  $u_k$  representerar insignalerna till processen vid tidpunkten  $k$  och vektorn  $z_k$  representerar processens tillstånd vid samma tidpunkt. Funktionen  $A(z_k, u_k)$  ger nästa tillstånd från nuvarande tillstånd och insignal. Om processen är linjär övergår funktionen  $A$  i den vanliga linjära tillståndsekvationen, där  $F$  och  $G$  är matriser.

$$z_{k+1} = A(z_k, u_k) = Fz_k + Gu_k$$

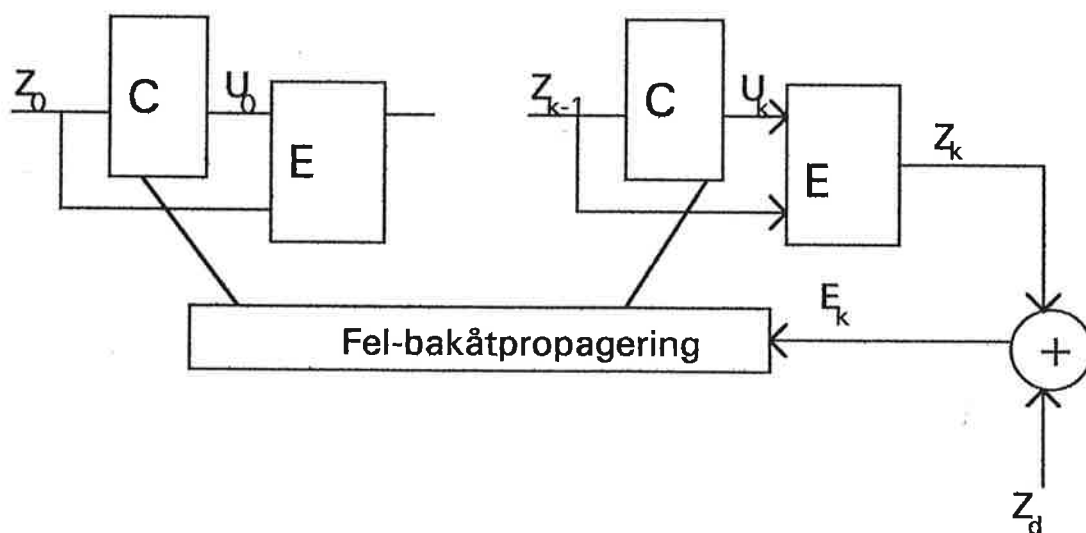
Ett vanligt problem inom reglerteknik är att ge korrekt insignal för att driva en olinjär process från ett ursprungligt tillstånd till ett önskat tillstånd. Normalt sett innebär detta linjarisering mm, vilket är omständigt. Ett annat sätt är att träna ett neuralt nätverk nn till att ge rätt insignal [2]. Vi har utgått från denna artikel för att på samma sätt försöka skapa en regulator. Fördelarna med denna metod är att man inte behöver känna processen och att den kan vara vilken olinjäritet som helst, bara man känner ordningen på systemet. Nackdelen är att det tar lång tid att träna regulatorn till önskat uppförande. Detta gäller i princip generellt för alla tillämpningar av nn. En svårighet med användandet av nn är de många parametrarna som måste väljas och det finns inga regler för hur de ska väljas, utan de måste provas fram empiriskt.

### 3.1 Emulator

Innan träning av regulatorn kan ske, måste ett annat neuralt nätverk tränas att uppföra sig som processen. Detta därför att i träningen bygger på bakåtpropagering av ett fel och det går inte att använda själva processen till detta.

### 3.2 Träning av regulatorn

Regulatorn tränas som visas i figur 4.



figur 4 Träning av regulator  
(C = Regulator E = emulator)

För att det ska vara möjligt att träna en regulator behövs en modell av processen. I detta kapitel utgås från att denna redan är framtagen (se kapitel 3.1). Regulatorn tränas att gå från ett initialtillstånd till ett önskat tillstånd på K steg. Träningen fortgår ett stort antal gånger, där man startar från ett initialtillstånd och fortsätter K steg tills man befinner sig i sluttillståndet  $z_k$ . Målet med träningen är att få fram de vikter som minimerar förlustfunktionen

$$J = \frac{1}{2} \varepsilon^2 = \frac{1}{2} (z_d - z_k)^2$$

där  $Z_d$  står för önskad utsignal och  $Z_k$  är åstadkommen utsignal. J ska dessutom medelvärdesbildas över alla initialtillstånd  $Z_0$ .

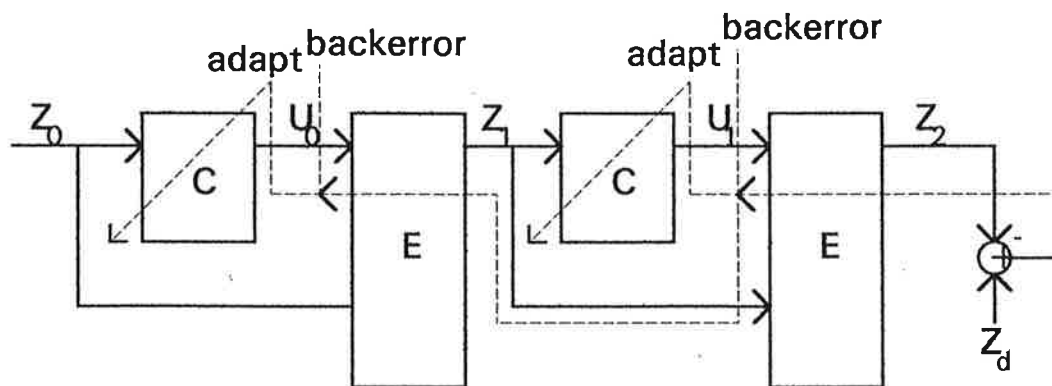
Träningsproceduren är illustrerad i figur 4. Den börjar med ett slumpvis valt initialtillstånd  $Z_0$  för neurala nätverksemulatorn. Eftersom regulatören i början är otränad kommer den att ge ut en felaktig signal till emulatorn och även till själva processen. Emulatorn kommer då att gå till nästa tillstånd  $Z_1$ , som givetvis i början är behäftat med ett fel pga den felaktiga insignalen. Denna procedur fortsätter i K steg. Då befinner sig processen i tillstånd  $Z_k$ . Antalet steg K är en av de parametrar som måste bestämmas av konstruktören och den bestämmer i princip regulatorns snabbhet, dvs K är antalet samplingsintervall som hinner gå innan processen har reglerats till önskat läge. För att minimera förlustfunktionen J måste vikterna i regulatören justeras så att det kvadratiske felet  $(Z_d - Z_k)^2$  blir mindre vid nästa försök. För att kunna träna regulatören måste vi känna till felet i regulatorns utsignal för varje samplingsintervall. Detta har vi inte tillgång till, och det är därför som en modell i form av ett neuralt nätverk behövs. Vi har tillgång till felet för det sista tillståndet (det som ska vara lika med önskat tillstånd). Detta felet kan med hjälp av vår modell bakåtoppropageras så att ett fel i regulatorns utsignal kan fås fram. Detta fås från ekvationerna (10) & (11) i [2] som i vårt fall med endast ett mellanlager blir

$$D_2 = \text{sgm}'(Y_2) \cdot \begin{bmatrix} z_{d1} - z_1 & 0 \\ 0 & z_{d2} - z_2 \end{bmatrix}$$

$$D_1 = \text{sgm}'(Y_1) \cdot W_2 D_2$$

där D är matriser av det bakåtoppropagerade felet med index angivande lager,  $\text{sgm}'$  derivatan av tanhyperbolikus, och där Y och W är matriser som beskrivs i kapitel 2.2 dvs Y elementets summerade och viktade insignaler, och W vikterna.

Detta fel kan då användas till att träna regulatören genom att använda ekvation 2. Emulatorn delar alltså upp felet i tillståndet k i ett fel för regulatorns utsignal och ett fel i tillståndet k-1. Den verkliga processen kan inte användas eftersom felet inte kan bakåtoppropageras genom den. Felet fortsätter att bakåtoppropageras genom alla K stegen av proceduren och justeringen av vikterna beräknas för varje steg. När alla stegen har gått igenom läggs slutligen alla ändringar till de riktiga vikterna. Det är här möjligt att direkt vid varje steg ändra vikterna, eftersom ändringarna vid varje steg är små i förhållande till själva vikterna. Av detta inses också att ett stort antal träningsrundor måste genomföras. Proceduren för en 2-stegsregulator är beskriven i figur 5.



Figur 5 Bildförklaring av regleringsalgoritm

Figur 5 visar alltså hur regleringen går till. Till en början befinner sig processen i tillståndet  $Z_0$  som ges som insignal till regulatoren. Regulatoren skickar en styrsignal till emulatoren, varvid nästa tillstånd  $Z_1$  erhålls. Båda dessa steg upprepas så att tillståndet  $Z_2$  fås. Detta tillstånd jämförs med önskat tillstånd  $Z_d$ . Skillnaden, dvs felet bakåtpropageras med hjälp av funktionen *backerror* så att ett fel erhålls i såväl  $U_1$  som  $Z_1$ . Felet i  $U_1$  används för att justera vikterna i regulatoren med funktionen *adapt* och felet i  $Z_1$  bakåtpropageras vidare på samma sätt som tidigare.

Algoritmen är en gradientmetod och därför kan man inte vara säker på att hitta det globala minimat, istället kan man fastna i ett lokalt minima. Normalt sett erhålls dock en bra lösning genom att använda ett stort antal element i mellanlagren i det neurala nätverket.

#### 4 Andra ordningens dåligt dämpade system

Den ena av de processer som undersökts är ett dåligt dämpat system av andra ordningen med överföringsfunktionen

$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

där  $\omega_0$  är 1 och  $\zeta$  är 0.3. Detta system samplades med ett samplingsintervall  $h$  på 0.1 s. Då fås tillståndsekvationen

$$Z(k+1) = \Phi Z(k) + \Gamma U(k)$$

med  $\Phi = \begin{bmatrix} 0.94 & -0.097 \\ 0.097 & 0.995 \end{bmatrix}$  och  $\Gamma = [0.097 \quad 0.0049]^T$

En fasplot visas i bilaga A2. Där kan man på ett ungefär se hur dåligt dämpat systemet är. Vad som inte syns är hur många samplingsintervall som hinner gå innan det önskade tillståndet (0,0) uppnås. Detta visas istället genom ett stegsvar från tillståndet (0.5,0.5) i bilaga A3.

#### 4.1 Styrbarhet

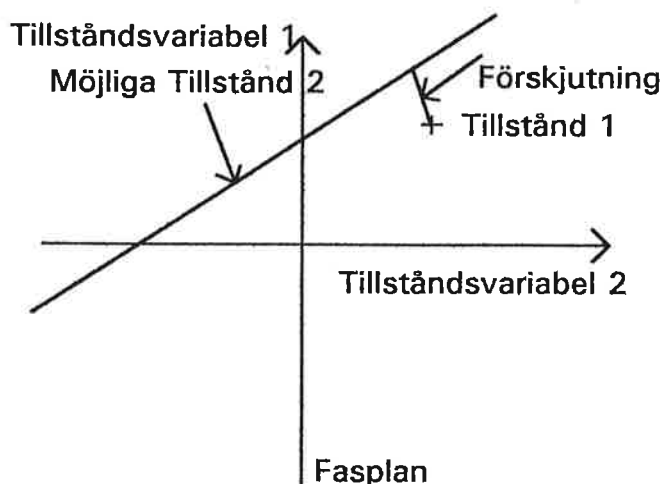
Den tillståndsekvation som systemet är beskrivet på går i fallet att ett specifikt tillstånd ska uppnås från ett annat specifikt tillstånd i ett steg att tolka som att nästa tillstånd ges av summan av en konstant och en viktad insignal, där vikten är  $\Gamma$ . För att förklara använder vi oss av ett exempel.

Exempel 1:

Antag att processen befinner sig i tillståndet (0.01, 0.01) och att den styrs till nästa tillstånd av insignalen  $U$  i ett steg. Då kommer nästa tillstånd att bli:

$$Z(k+1) = \begin{bmatrix} 0.94 & -0.097 \\ 0.097 & 0.995 \end{bmatrix} \cdot (0.01, 0.01)^T + [0.097, 0.0049]^T U = K + \Gamma U$$

vilket är en förskjutet rät linje i fasplanet, där förskjutningen är beroende av föregående tillstånd och linjens lutning på förhand är bestämd av tillståndsbeskrivningen. Detta illustreras i figur 6.



Figur 6: Styrning från ett tillstånd till nästa

Pga detta är det inte möjligt att i ett steg styra processen från alla möjliga tillstånd till önskat tillstånd. Därför är det intressant att jämföra med en deadbeatregulator.

## 4.2 Deadbeatregulatorn

Deadbeatregulatorn kännetecknas av att alla poler i den önskade tillståndsekvationen är placerade i origo. Därmed får den egenskapen att alla tillståndsvariabler kan fås att gå till noll på högst  $n$  steg, där  $n$  är antalet tillståndsvariabler. Den är enkel att designa eftersom endast samplingsintervallet behöver bestämmas. Samplingsintervallet bestäms så att styrsignalen får lämplig storlek. Nackdelarna med den är bl a att den endast går att konstruera i diskret tid och att den är störkänslig. För utförligare beskrivning hänvisas till [7]. En fasplot av hur deadbeatregulatorn fungerar från tillståndet  $(0.01, 0.01)$  visas i bilaga A4 och A5.

## 4.3 Resultat

En lyckad identifiering av en andra ordningens dåligt dämpade system genomfördes. I bilaga A6 visas felet mellan den predikterade utsignalen och den verkliga utsignalen.

En jämförelse mellan deadbeatregulatorn och neurala nätverks-regulatorn vald att reglera i två steg från ett initialtillstånd  $(0.01, 0.01)$  visar att de är identiska. NN-regulatorn har i detta fallet valts att tränas från enbart en position, då svårigheter uppträder vid en global träning. Dessa svårigheter bör bero på att nn-regulatorn fastnar i lokala minima istället för det globala, pga att processen endast går att styra i speciella riktningar (se figur 6). Botemedlet för detta är oftast att öka antalet element i mellanlagren i det neurala nätverket alternativt att öka antalet mellanlager. Tiden för en träningsperiod för ett sådant nätverk ryms dock ej inom ramarna för detta projekt. En annan lösning på problemet är att öka antalet steg som regulatorn har på sig att reglera processen till önskat tillstånd. Om antalet steg sätts lika med 20 så fungerar regulatorn utmärkt (se bilaga A2 och A3).

## 4.4 Olika parametrars betydelse

De viktigaste parametrarna vid konstruktion av ett neuralt nätverk är  $my$ , antalet mellanlager och antalet element i mellanlagren.

$My$  är en konstant som bestämmer konvergenshastighet och stabilitetsegenskaper i uppdateringen av vikterna. Den är svår att bestämma, men som en tumregel kan sägas att om systemet inte konvergerar så bör  $my$  minskas.

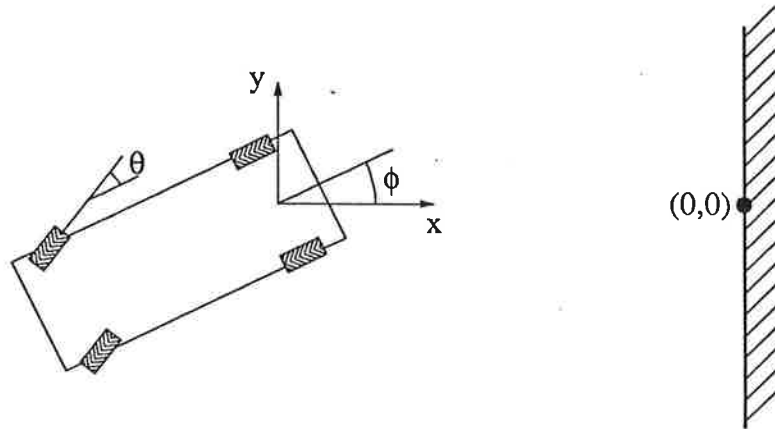
Antalet mellanlager är även detta svårt att bestämma, men som nämns i tidigare kapitel och i [1] så kan alla "tillräckligt snälla" olinjära funktioner approximeras med ett nätverk innehållande ett mellanlager, bara detta innehåller tillräckligt många neuroner. Vi har därför begränsat oss till att använda enbart ett mellanlager.

För antalet neuroner i mellanlagret finns heller inga tumregler, men om det inte är tillräckligt stort finns risk att fastna i lokala minima istället för det globala, vilket tex kan visa sig som att vid träning från olika initialtillstånd alla sluttillstånd hamnar på en linje istället för i önskad punkt i en fasplot.

En annan parameter som måste väljas är den som bestämmer hur sigmoiden ska begränsas. Denna är viktig, speciellt om man vill kunna reglera i ett fåtal steg, varvid stora styrsignaler kan behövas.

## 5. Styrning av backande lastbil

Att backa en lastbil är ett svårt och kraftigt olinjärt styrproblem. Det tjänar därför som ett bra exempel för den metod som vi beskrivit ovan. Uppgiften är att reglera framhjulens vinkel så att lastbilen når önskad position (t ex en lastbrygga) med rätt orientering.



Figur 7. Backande lastbil.

De samplade dynamiska ekvationerna för lastbilen blir enligt [6]

$$\begin{cases} x(k+1) = x(k) + \cos(\phi(k) + \theta(k)) + \sin(\theta(k)) \cdot \sin(\phi(k)) \\ y(k+1) = y(k) + \sin(\phi(k) + \theta(k)) - \sin(\theta(k)) \cdot \cos(\phi(k)) \\ \phi(k+1) = \phi(k) - \sin^{-1}\left(\frac{2}{b} \sin(\theta(k))\right) \end{cases} \quad (4)$$

Här är  $\theta(k)$  framhjulens styrvinkel och  $b$  avståndet mellan hjulaxlarna. Lastbilen backas med konstant hastighet.

### 5.1 Identifiering

Vi vill träna ett neuralt nätverk så att det beskriver lastbilens dynamik enligt (4). Detta är inte helt lätt. Utsignalerna från ett neuralt nätverk är begränsade (p g a sigmoidfunktionen). Följden av detta är att man inte kan träna en emulator som klarar hela planet. Vi har försökt lösa detta på två olika sätt.

#### Försök 1

Vi har här valt att begränsa emuleringen till en del av planet. Sigmoiden valdes så att emulatorns utsignalområdet blev  $(-40, 40)$ . Vi försökte därefter träna emulatorn på det sätt som beskrivits ovan. Resultaten var dock nedslående. Efter många dagars arbete kunde vi konstatera att det bästa försöket gav  $\text{SNR} = -36$  dB vilket är helt oacceptabelt.

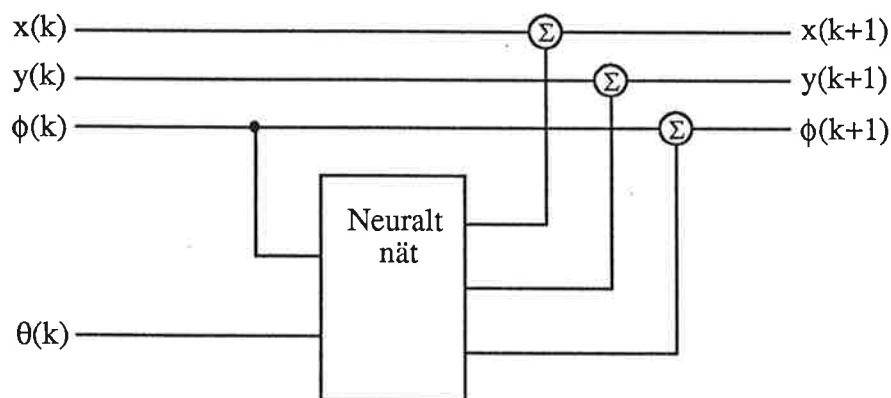
#### Försök 2

Man inser lätt att lastbilens dynamik måste vara oberoende av positionen och orienteringen i rummet. Vi har därför valt att realisera emulatornätet enligt figur 4 (där bara positionsberoendet utnyttjas). Här realiserar det neurala nätet endast *ändringen* från tiden  $k$  till  $k+1$ . (Frågan är hur detta nät ska tolkas vid den felbakåtpropagering som görs vid träningen av regulatorn. Vi lämnar detta problem eftersom detta tyvärr aldrig blev aktuellt.)



Vi blev dock tvungna att i även detta fall konstatera att det inte är så lätt att träna neurala nätverk som vi hoppats på. Efter mycket arbete erhöll vi resultat runt  $\text{SNR} = -45$  dB vilket fortfarande är alldeles för dåligt.

Eftersom vi inte lyckades träna någon emulator kunde vi inte heller träna någon regulator. Vi vet inte vad vi gör för något fel men det måste vara något. Författarna till [2], varifrån vi hämtat exemplet, har nämligen lyckats. Den mest troliga orsaken till våra problem är att träningen fastnar i lokala minima. Förmodligen borde vi vid träning använda någon annan variant av uppdateringsformel för vikterna.



Figur 8. Emulator för lastbilsdynamik.

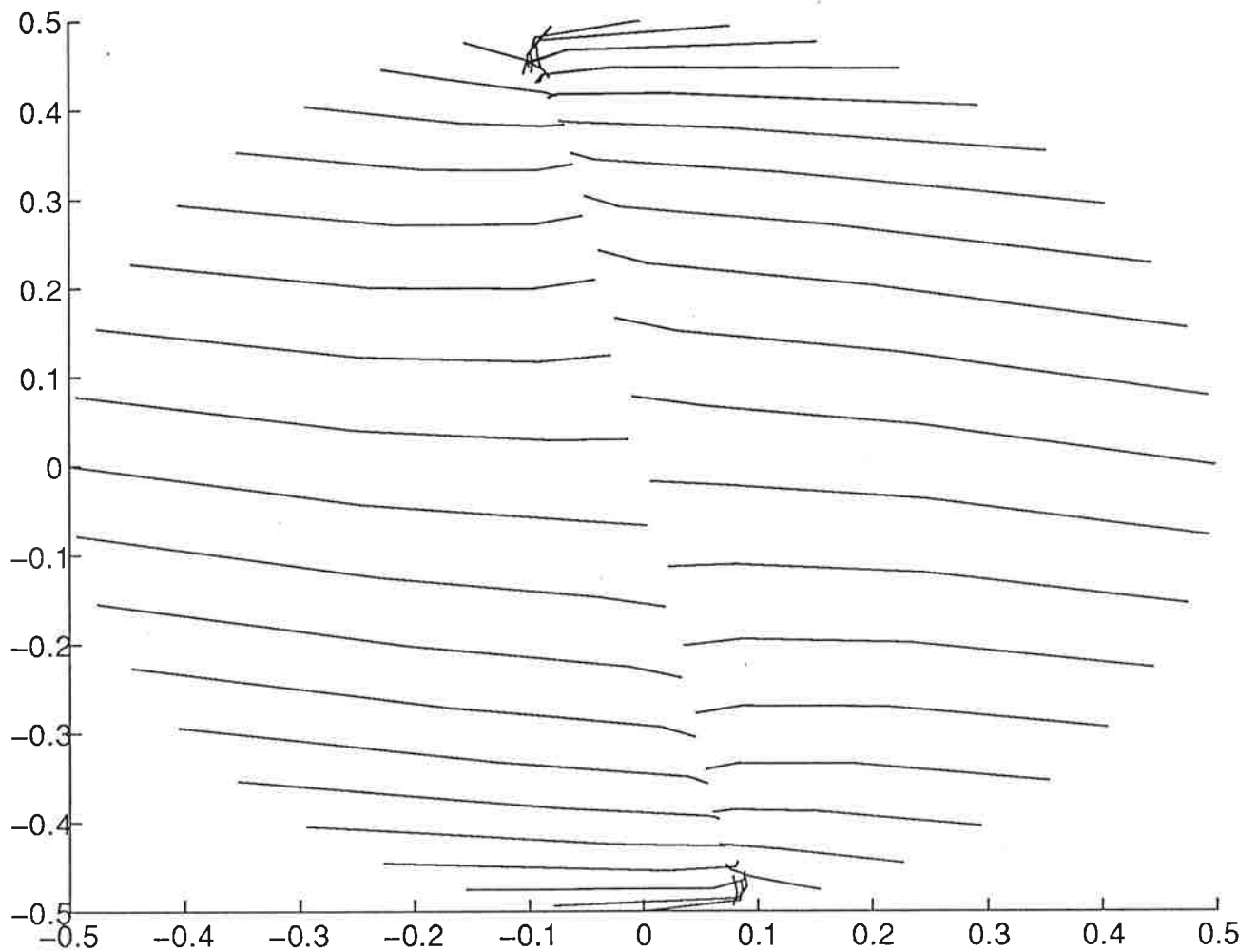
## Referenser

Vissa figurer och formler i denna rapport är hämtade från [5]. Övriga referenser är använda i texten.

- [1] Bernard Widrow och Michael A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", Proceedings of the IEEE, Vol. 78, No. 9, pp 1415-1442, September 1990
- [2] Derrick J. Hguyen och Bernard Widrow, "Neural Networks for Self-Learning Control Systems", IEEE Control Systems Magazine, April 1990, pp 18-23
- [3] Karl Johan Åström och Björn Wittenmark, "Adaptive Control", Addison-Wesley, 1989
- [4] Rolf Johansson, "System Modeling and Identification", Prentice Hall, 1993
- [5] Tord Björsne och Christer Gertman, "Klassificering av Fonem m h a Neuralt Nätverk", Projektrapport i kursen Optimal Signalbehandling vid Institutionen för Teletransmissionsteori, 1993
- [6] Li-Xin Wang och Jerry M. Mendel, "Generating Fuzzy Rules by Learning from Examples", IEEE Transactions on System, Man and Cybernetics, Vol 22, No 6, 1992
- [7] Karl Johan Åström och Björn Wittenmark, "Computer-controlled Systems — Theory and design", Prentice Hall, 1990

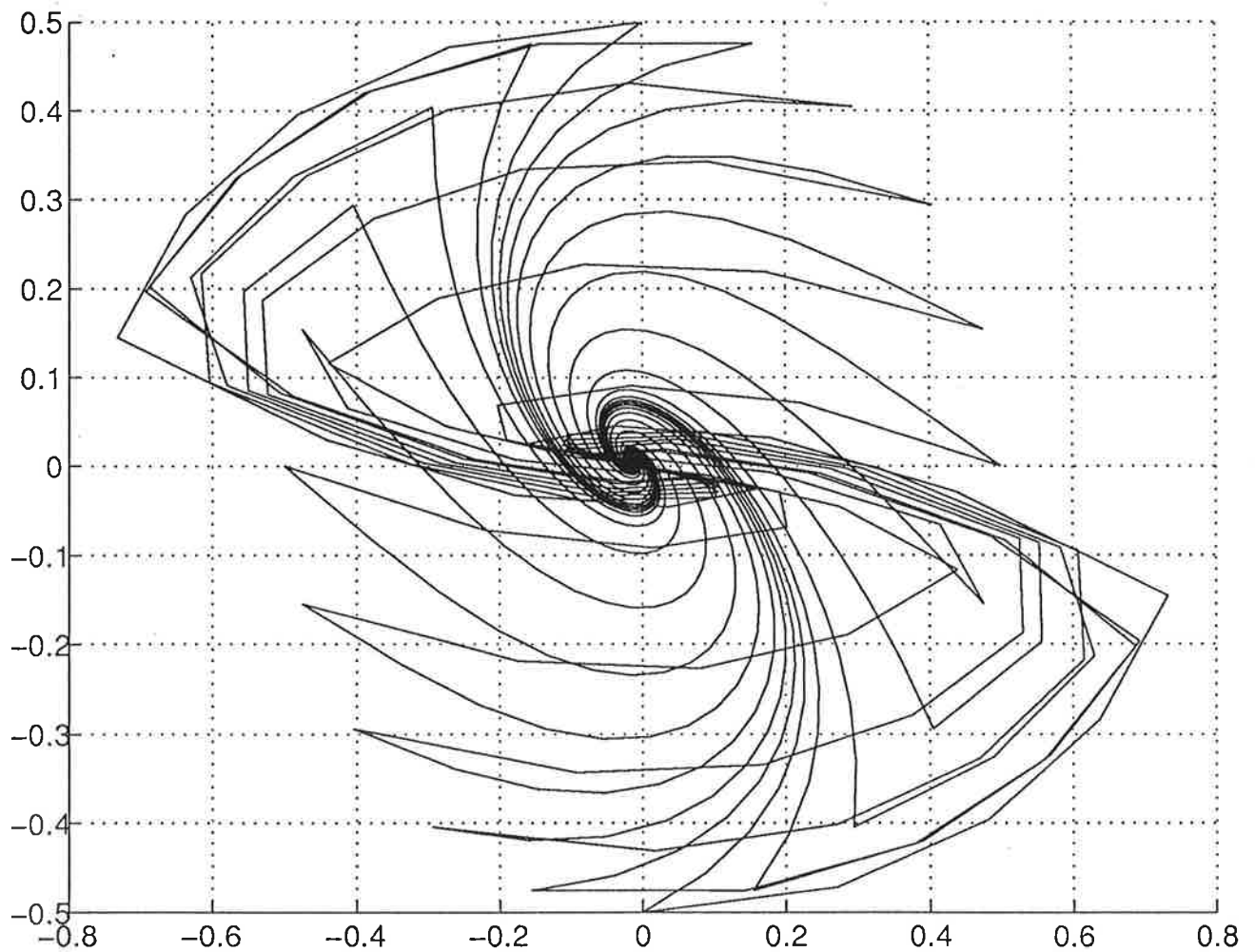
# Fasplot av tränad regulator

## Försök till deadbeatregulator



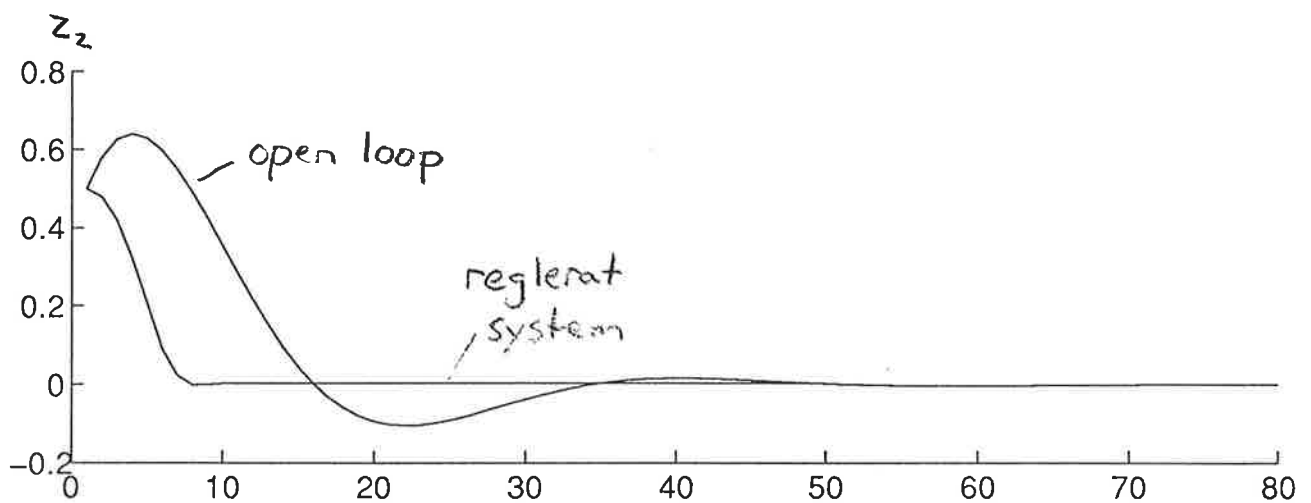
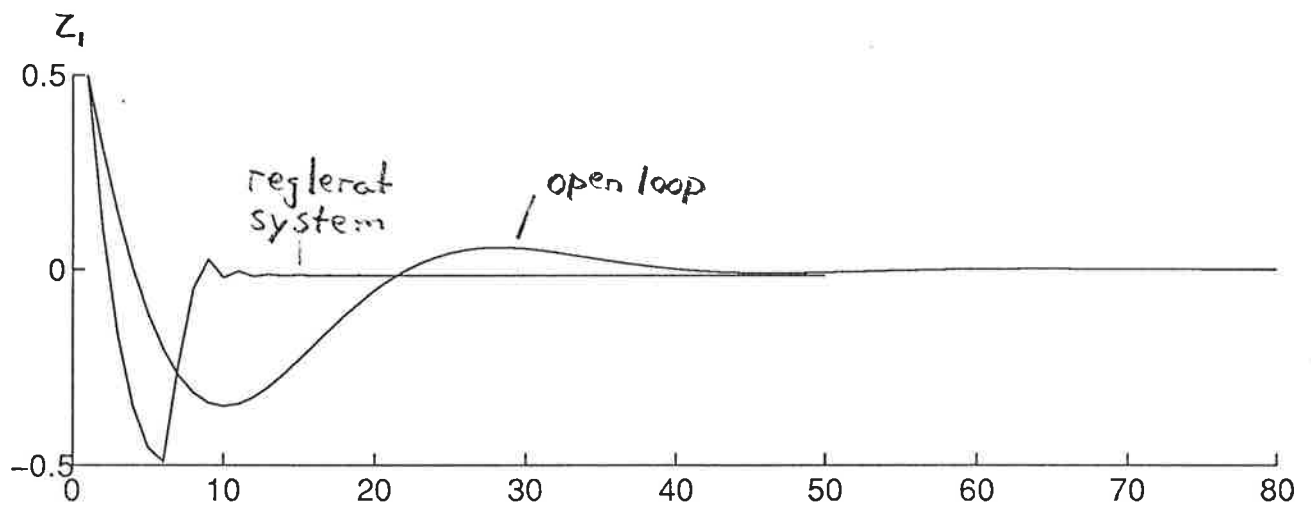
Fasplot av tränad regulator  
och openloop system

Regulatorn reglerar processen till (0,0)  
på 20 samplingsintervall

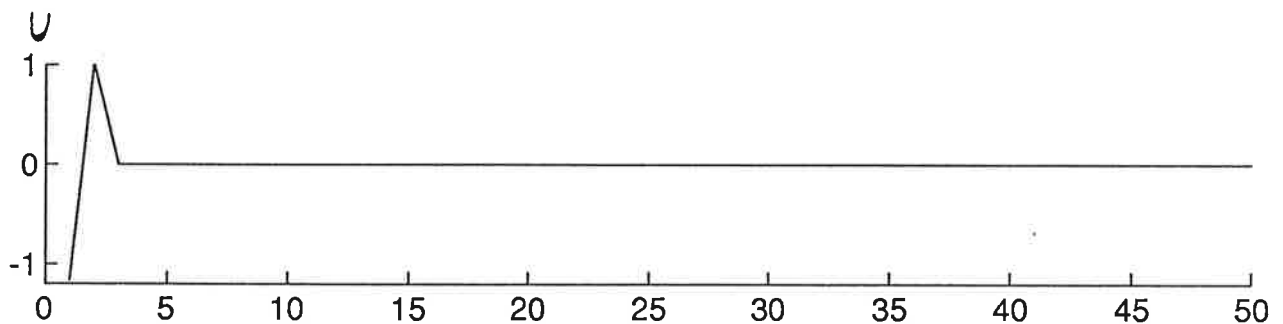
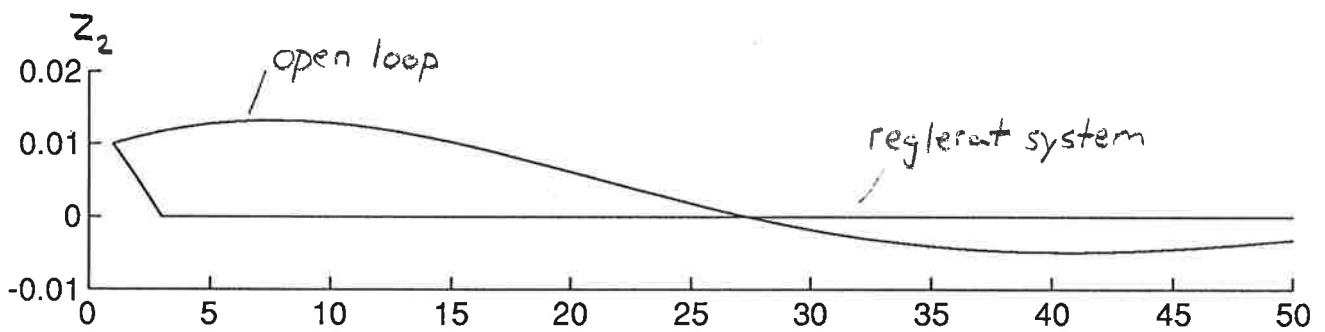
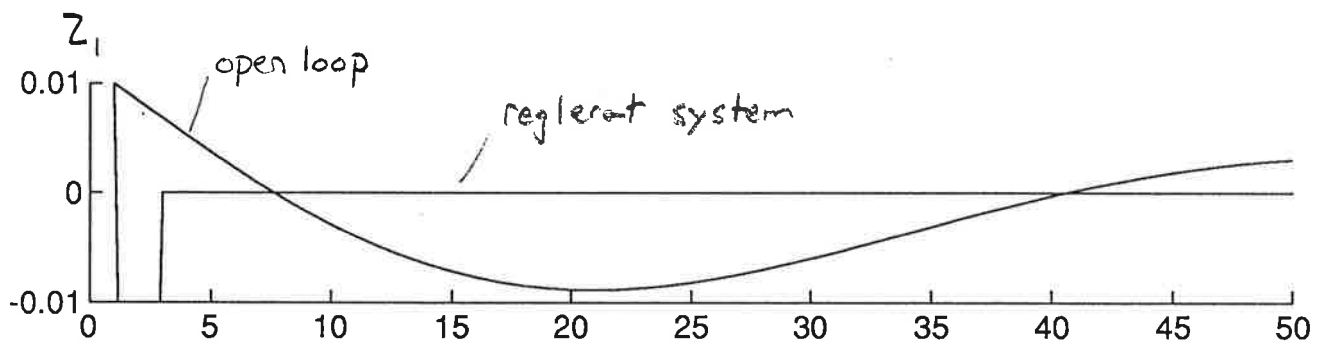


Stegsvar från positionen (0.5, 0.5)

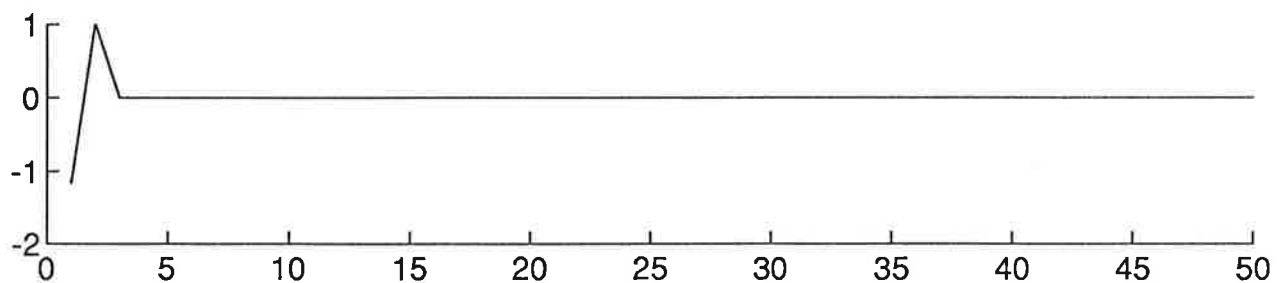
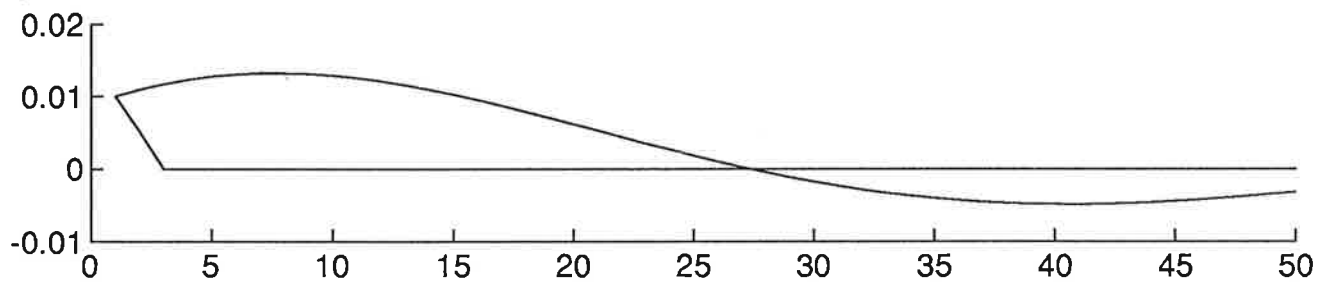
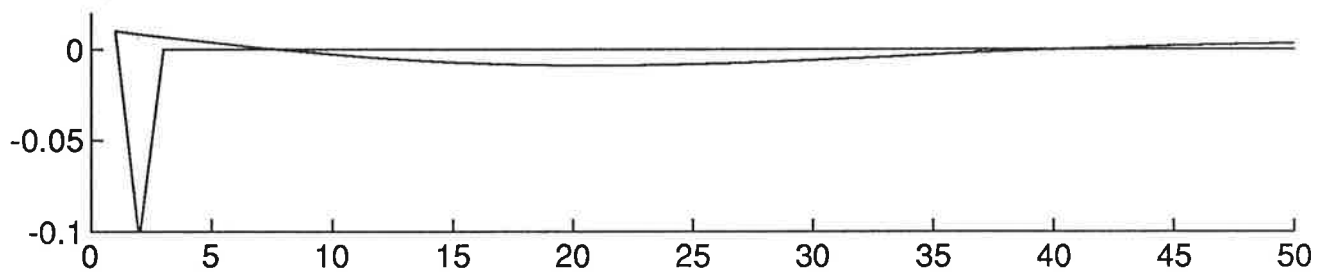
I övrigt samma som A2



Stegsvar från (0,01,0,01) av tränad regulator.  
Regulatorn reglerar processen till (0,0) på  
2 samplingsintervall.



Samma som A4



# Prediktionsfel för processemulator

Process: Andragrads linjär med dämpning 0.3

