

ISSN 0280-5316  
ISRN LUTFD2/TFRT-5492--SE

# Introduction of Adaptive Active Filters in HVDC Systems

Tor Göransson

Department of Automatic Control  
Lund Institute of Technology  
December 1993

<b>Department of Automatic Control</b> <b>Lund Institute of Technology</b> P.O. Box 118 S-221 00 Lund Sweden	<i>Document name</i> MASTER THESIS	
	<i>Date of issue</i> December 1993	
	<i>Document Number</i> ISRN LUTFD2/TFRT--5492--SE	
<i>Author(s)</i> Tor Göransson	<i>Supervisor</i> Anders Åberg (ABB) and Karl Johan Åström	
	<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Introduction of Adaptive Active Filters in HVDC Systems		
<i>Abstract</i> <p>When converting electric power from AC to DC, a 12-pulse converter is often used. Unfortunately, the converter creates harmonics superimposed on the DC voltage. These are removed by filters. A passive filter is normally used, but active filters may be added for improved performance. However, the drift in component values due to fluctuations in temperature and the change in load, makes it difficult to obtain filters with consistently good performance. Initially, tests and measurements are required to adjust the filters to the environment. To circumvent these drawbacks, adaptation and adjustment during operation is needed. Different adaptive control concepts are investigated in this report. Two of them, feedforward with estimation using the MIT rule and feedback using an indirect self-tuning regulator, are also tested in simulations. The ability to estimate the whole system, including the effect of reflections from the transmission net, is partially analyzed. Estimation is done using the recursive least-mean-square algorithm and the stochastic approximation algorithm. Many problem can arise in designing adaptive controllers for a HVDC System. In this report, some of them are discussed and different solutions are suggested.</p>		
<i>Key words</i> HVDC, Active filters. Adaptive control, Parameter estimation.		
<i>Classification system and/or index terms (if any)</i>		
<i>Supplementary bibliographical information</i>		
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 62	<i>Recipient's notes</i>
<i>Security classification</i>		

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.



# Contents

<b>1</b>	<b>Summary</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
2.1	Background . . . . .	4
2.2	The Lindome Station . . . . .	4
2.3	Report Contents . . . . .	5
2.4	Assumptions . . . . .	6
<b>3</b>	<b>The Model</b>	<b>7</b>
3.1	The Station . . . . .	7
3.2	Continuous Mathematical Model . . . . .	8
3.3	Discrete Mathematical Model . . . . .	8
3.4	The Transmission Net . . . . .	10
<b>4</b>	<b>Feedforward Control Using the MIT Rule</b>	<b>11</b>
4.1	The Test Signals . . . . .	11
4.2	Controller Without Adaptation . . . . .	12
4.3	Adaptive Control of the Feedforward Gain . . . . .	13
4.4	Adaptive Control of the Entire $H_{ff}$ . . . . .	15
4.4.1	The Denominator . . . . .	15
4.4.2	The Numerator . . . . .	17
4.4.3	Both the Numerator and the Denominator . . . . .	17
4.5	Conclusions . . . . .	18
<b>5</b>	<b>Estimating the Models Using RLS</b>	<b>19</b>
5.1	Estimating the Transfer Function $G(q)$ . . . . .	19
5.2	Estimating Both $G(q)$ and $H(q)$ . . . . .	23
5.3	Conclusions . . . . .	25
<b>6</b>	<b>Feedback Control</b>	<b>26</b>
6.1	Feedback Using Pole Placement Design . . . . .	26
6.1.1	Pole Placement Design . . . . .	26
6.1.2	The Result Without Estimator . . . . .	27
6.1.3	The Result With the Estimator Included . . . . .	30
6.2	Conclusions . . . . .	31
<b>7</b>	<b>Conclusions</b>	<b>32</b>
7.1	Conclusion . . . . .	32
7.2	Extensions . . . . .	33
7.3	Tools . . . . .	34

<b>A Appendix</b>	<b>36</b>
A.1 The model of the net . . . . .	36
A.2 Extend the Model of the Net . . . . .	38
A.3 The Extended Least-Squares Algorithm . . . . .	39
<b>B Appendix</b>	<b>40</b>
B.1 Programs Common For All Simulations . . . . .	40
B.2 MIT Program . . . . .	43
B.3 Programs Common For the RLS and RST Simulations . . . . .	49
B.4 RLS Program . . . . .	54
B.5 RST Program . . . . .	57

# Chapter 1

## Summary

In this report two concepts to install adaptive active filters in HVDC systems are studied.

The first concept is feedforward with adaptive parameters. These are adjusted using the MIT rule. The result of the simulations show very good performance. However, this is obtained using a simplified model of the real system. More investigations are needed to determine the usability of this concept in reality.

The second concept is feedback using an indirect self-tuning regulator. This regulator uses estimated parameters in its design. To estimate the system the recursive least-mean-squares algorithm and the stochastic approximation algorithm was used. Simulations showed the ability to estimate almost the whole system, even when the transmission net was connected. An important remark is that this was done in the simplified model mentioned before. The number of parameters to estimate increase in reality. Whether these are possible to estimate or not, further tests have to show. The controller used in this concept was designed to control in a wide range of frequencies. The simulations showed serious difficulties using this strategy. The conclusion is that another concept is preferable. In this, several controllers, each with small bandwidth, can be used in feedback.

## Chapter 2

# Introduction

### 2.1 Background

In the process of converting electric power from AC to DC in an HVDC system, harmonics are created in the DC voltage. Some of the harmonics (300 to 5000 Hz) interfere with the telephone signals and are definitely not wanted. To prevent these from entering the transmission line, different type of electrical filters are installed. In most stations of converting, only passive filters are used. To improve the performance, active filters have been developed, but so far, these have only been installed in two stations: the Lindome station in Sweden and Skagerrak 3 in Denmark. In these stations the passive filters are retained and the active filters are installed as complements. This due to the difficulties to annihilate the most power-full disturbances with an active filter. The combination of passive and active filters seems to be a good solution and it works quite well, though, there is one drawback with the active filter. The parameters used in the controller to determine its performance don't adjust to the environment. This will result in a bad performance if the model of the net changes, for instance, because of heat or a new load. It also requires that the frequency response of the net initially is measured in order to adjust the parameters. To avoid these drawbacks, an adaptive control concept is needed. In this report the controller in the Lindome station is studied and extended with different adaptive concepts.

### 2.2 The Lindome Station

One of the stations where an active filter has been installed is the Lindome station. This is shown in figure 2.1.

To the left, a 12-pulse converter is shown. This converts the AC signal to DC. During the conversion, a lot of disturbances are created. Basically, these

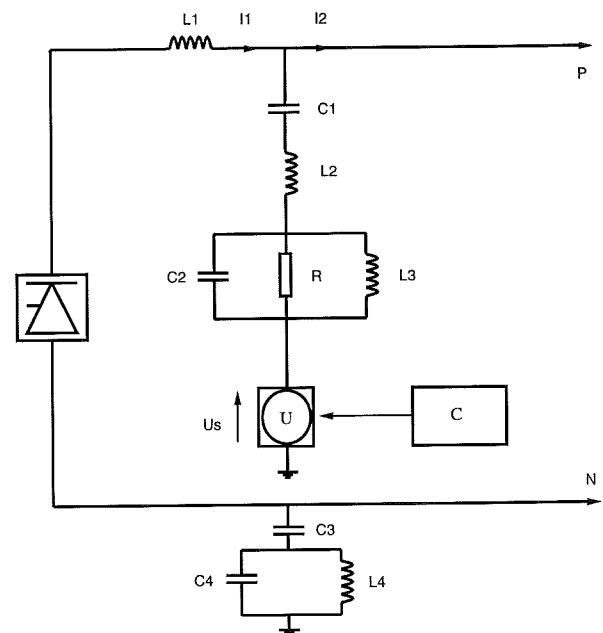


Figure 2.1: The filters in the Lindome station.

have the harmonic frequencies  $f = 12nf_0$  where  $f_0 = 50$  Hz and  $n = 1, 2, 3, \dots$ . In some cases, also the frequencies  $f = 3nf_0$  occur with small harmonic amplitudes. It is also observed that the largest power is found in the disturbances with the lowest frequencies. In figure 2.2 are the measured values of  $I_1$  and  $I_2$  shown. In this case there is no active control running, i.e.,  $U_s$  in figure 2.1 is equal to zero. Notice the periodicity in the disturbance, as described. It can be seen that there also are disturbances with frequencies below 300 Hz, but these are, however, not to be removed by the active filter. The reason is limitations in the equipment. For more information about disturbances in conversion using a 12-pulse converter, see [Zhang], [Shore], and [Dickmander].

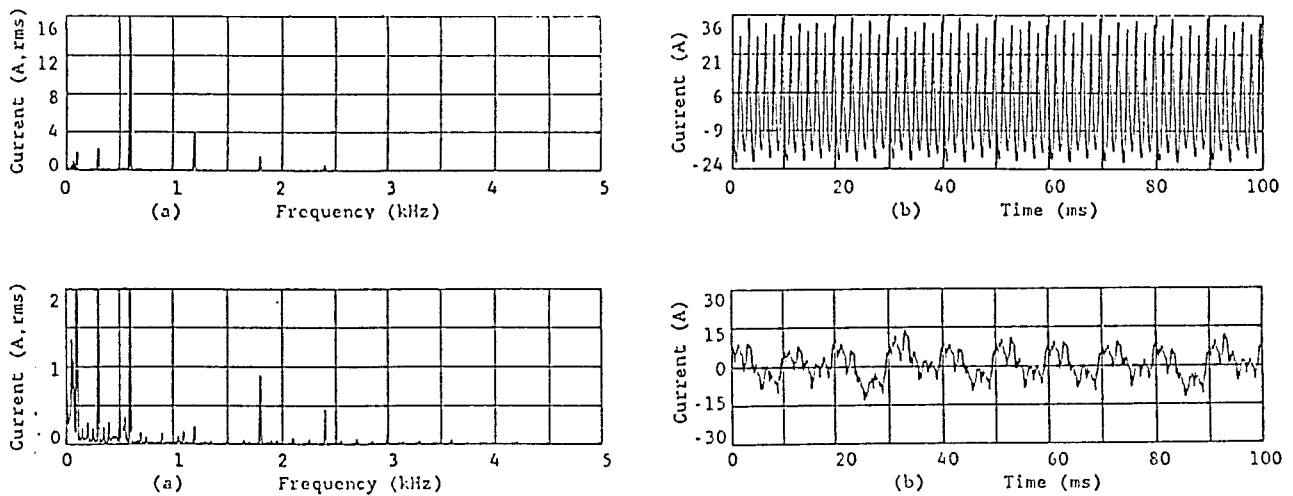


Figure 2.2: The currents  $I_1$  (the upper pair of figures) and  $I_2$  (the lower pair), when no active control is installed. To the left the currents are given as functions of frequency and to the right as functions of time.

To remove the harmonics created in the conversion, the current is filtered. This is partly done by the smoothing reactor L1 and the passive filter (C1, C2, L2, L3, R). The latter filter removes the 12th and 24th harmonics (that is 600 and 1200 Hz). The disturbances that not are removed by the passive filter are subject to removal by the active one.

The active filter is implemented with a high power PWM amplifier as voltage source and a controller. The voltage source, beneath the passive filter in the figure, is controlled by the controller (C), which is a program running on a high-speed digital signal processor (DSP). In principle, the controller works as shown in figure 2.3.

In the first module of the controller, the DC components and offsets in the measured currents  $I_1$  and  $I_2$  are filtered out. In the second module, the signal passes an inverse model of the whole plant. This gives a 'standardized' signal that is independent of the station. In order to create this inverse model, the frequency response of the station is measured before starting up. This is one of the drawbacks mentioned before. The third module comprises an algorithm suppressing all the repetitive signals. Finally, the signal is filtered in the fourth module (only frequencies over 300 Hz passes through) and is limited in the fifth.

In the station, there are also many systems for protection and supervision, but these are not described in the model and will not be considered in this report.

## 2.3 Report Contents

There are, of course, many ways of controlling a system like this. But in order to be able to do so, a model of the system plants needed. In chapter 3, both a continuous and a discrete model are designed and calculated. The different control concepts in the report are based on these models.

The first concept is a feedforward controller with estimated parameters. The MIT rule is used to estimate the parameters. This is described in chapter 4.

In chapter 5, almost all parameters of the system is estimated. The algorithm used in the estimator to do this is the recursive least-squares (RLS).

In chapter 6, a control concept utilizing the models and the parameter estimates is introduced and partially tested. More work is needed before this concept can be used in practice.

Finally, the conclusion in chapter 7 gives the results of this work. Here is also the different working tools commented on.



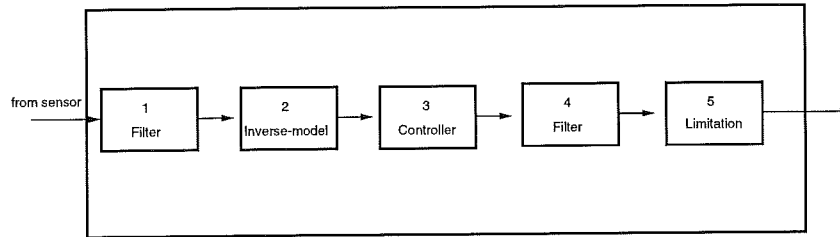


Figure 2.3: The modules in the controller.

## 2.4 Assumptions

The investigations and tests in this report are based on some assumptions of an ideal plant. Still, the assumed plant involves many of the difficulties of a real plant and, thus, the results can be used for at least discussing the principles. The assumptions are:

- All disturbances from the converter are supposed to be removed by the control. This means that only signals with frequencies in the range 300 to 5000 Hz are present in the measured signals. To accomplish this in reality, either a band-pass filter or many notch filters, can be used. In order not to interfere with the control system, it is important to choose a filter that gives minimum contributions in the phase. How this is done, is not further examined in this report. It is also assumed that the measured signals do not contain any noise.
- In reality, there are many delays, for instance, in measuring the current  $i_{\text{out}}$  (see figure 3.1). These delays are assumed to be zero in the following calculations, but because almost all signals are periodic with the period of 1/50 seconds, the values measured one period ago can be used. The alternative is to rewrite the algorithms in this report to use predicted values. Which alternative to choose depends on how periodic the signals are and how long the delays are.
- In order to create the feedforward controllers described in chapter 4 and 6 and the estimators in chapter 5 the assumption that it is possible to measure the voltage  $u_{\text{in}}$  (figure 3.1) is made. In reality this is not possible, but it is possible to calculate or approximate this signal if one or more other signals are measured.

## Chapter 3

# The Model

The true model of the station described in chapter 2.2 is very complicated and to develop and examine new concepts of controlling, a simplified model is to prefer. In the section 3.1 this simplified station is introduced and explained. Then the mathematical models are stated. Both the continuous model and then its discrete equivalent are calculated. The discrete model is calculated only to give a frame of reference for estimation, while the continuous one was implemented in the simulation program SIMNON. In reality the station is connected to a complex transmission net. When the disturbances enter the net, an echo is created. The echo will interfere with the new disturbances and will give contributions in the measured signal  $i_{out}$  in figure 3.1. This transmission net have also to be modeled to make the different concepts of controlling realistic. This is done in section 3.4. Notice that this model is very simplified, but in an initial phase of development, this is satisfying.

### 3.1 The Station

There were made some approximations in the model, in order to get it manageable. The major approximations are the removal of the 24th harmonic filter and the return path (C3, C4, L4 in figure 2.1); this yields a system with size  $3 \times 3$  instead of  $8 \times 8$ . This simplified filter is also used by [Zhang] and does not effect results in terms of principles.

The simplified plant is showed in figure 3.1. To the left, there is a voltage source which represent the disturbance signal from the converter. From now on, this signal will only have frequencies in the range 300–5000 Hz, that is, only signals that are to be filtered out enter the system. This signal passes through the smoothing reactor L1 to the resonance filter (L2, C1, and R). Using the values in table 3.1, its resonance becomes 600 Hz. Below the filter in

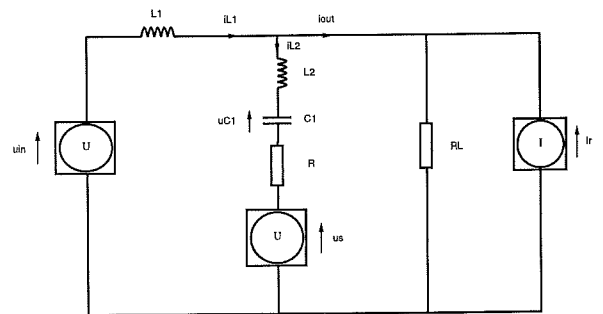


Figure 3.1: The approximated model of the HVDC station.

the figure, the controllable voltage source is shown. The voltage  $u_s$  is to be calculated in the controller.

In reality a current pulse in ( $i_{out}$ ) will be reflected back as an echo and these reflections are rather undamped. This echo effect is modeled by substituting the external net with a load  $R_L$  for the wave impedance and a controllable current source,  $I_r$ . This way of representing a complex net is explained in section 3.4.

$L_1$	$L_2$	$C_1$	$R$	$R_L$
200 mH	70.36 mH	1.0 $\mu$ F	1.0 $\Omega$	320 $\Omega$

Table 3.1: Parameters in the model. These give a resonance at 600 Hz.

## 3.2 Continuous Mathematical Model

The model in figure 3.1 is analyzed using Kirchhoff's laws [Claesson, chapter 3] giving the following state-space model.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = i_{\text{out}} = Cx(t) + Du(t) \end{cases}$$

where

$$x(t) = \begin{pmatrix} i_{L_1}(t) \\ i_{L_2}(t) \\ u_{C_1}(t) \end{pmatrix}$$

$$u(t) = \begin{pmatrix} u_{\text{in}}(t) \\ u_s(t) \\ i_r(t) \end{pmatrix}$$

$$A = \begin{pmatrix} -R_L/L_1 & R_L/L_1 & 0 \\ R_L/L_2 & -(R_L + R)/L_2 & -1/L_2 \\ 0 & 1/C_1 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1/L_1 & 0 & -R_L/L_1 \\ 0 & -1/L_2 & R_L/L_2 \\ 0 & 0 & 0 \end{pmatrix}$$

$$C = (1 \quad -1 \quad 0)$$

$$D = 0$$

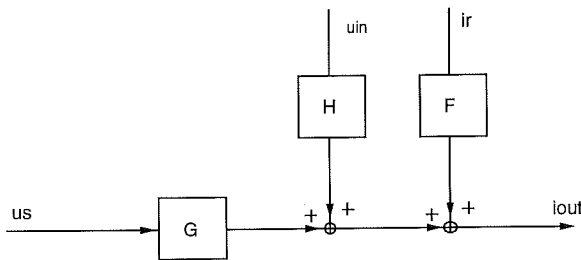


Figure 3.2: The transfer functions in the system and how they are connected to each other.

The input-output models are then (compare figure 3.2):

$$u_{\text{in}} \text{ to } i_{\text{out}}: H(s) = B_h(s)/A_h(s)$$

where

$$\begin{aligned} B_h(s) &= (5.0000s^2 + 71.0631s + 7.1063 \times 10^7) \\ A_h(s) &= s^3 + 6.1623 \times 10^3 s^2 + 1.4235 \times 10^7 s + 2.2740 \times 10^{10} \end{aligned}$$

$$u_s \text{ to } i_{\text{out}}: G(s) = B_g(s)/A_g(s)$$

where

$$B_g(s) = 14.2126s^2 - 5.5879 \times 10^{-9}s - 1.5259 \times 10^{-5}$$

$$A_g(s) = A_h(s)$$

$$i_r \text{ to } i_{\text{out}}: F(s) = B_f(s)/A_f(s)$$

where

$$B_f(s) = -6.1480 \times 10^3 s^2 - 2.2740 \times 10^4 s - 2.2740 \times 10^{10}$$

$$A_f(s) = A_h(s)$$

The bodeplots for all three transfer functions are plotted in figure 3.3. Notice the dip at 600 Hz for the function  $H(s)$ . It is also interesting to know the locations of poles and zeros. These are the following (given in the  $s$ -plane):

$$\text{root } B_h = 10^3[-0.0071 \pm i3.7700]$$

$$\text{root } B_g = [0.0010, -0.0010]$$

$$\text{root } B_f = 10^3[-0.0018 \pm i1.9232]$$

$$\text{root } A = 10^3[-4.0301, -1.00661 \pm i2.1221]$$

Notice, that there is one zero in the transfer function  $B_g$  in the right half- $s$ -plane. This non-minimum phase might be a problem when a controller is to be designed. In some cases, we make the approximation that  $B_g$  has its zeroes in the origin or just inside the left half- $s$ -plane.

## 3.3 Discrete Mathematical Model

The system can also be given in discrete values. Using a zero-order hold sampler [Åström90, chapter 3]. The sample rate is chosen to be the one used in the Lindome case. The sample rate is synchronized with the period of the 50 Hz signal and there are 512 samples every period. The sample frequency then becomes  $50 \times 512$  Hz and the sample period  $39.0625 \mu\text{s}$ . This period is used throughout the report.

An important remark is that this sample rate might be too low to control signals with frequencies round 5000 Hz. If a zero order hold sampler is used, the phase added due to the hold circuit is approximately  $\omega h/2$  rad [Åström90, page 219]. Here,  $\omega$  is the crossover frequency of the continuous system. If the

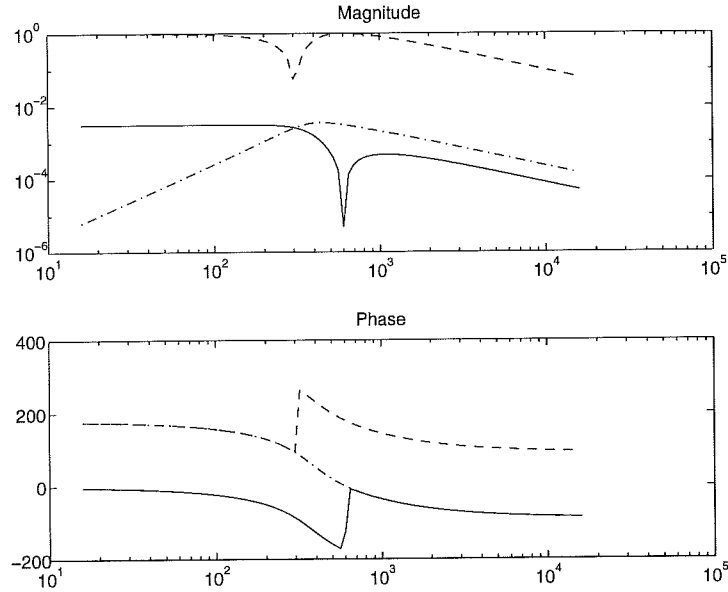


Figure 3.3: The bodeplot of  $H(s)$  (—),  $G(s)$  (---) and  $F(s)$  (-.-). The frequency axes are graded in Hz.

phase margin is allowed to be decreased with  $5^\circ$  to  $15^\circ$ , the sample period can be between  $0.17/\omega$  and  $0.52/\omega$ . To annihilate disturbances with frequencies up to 5000 Hz, the sample period has to be between  $5.4 \times 10^{-6}$  and  $16.7 \times 10^{-6}$  seconds, according to the rule above. This is a factor 2 to 7 faster than today. With the present period, the phase loss at 5000 Hz is  $35^\circ$ .

A too high sample rate will, however, increase the load on the computer. Not only the number of computations are increased, but also the accuracy of each computation. This to prevent numerical errors, because higher sample rate results in discrete roots closer to the unit circle.

The disturbances used in this report are in the frequency range 300-1800 Hz, as will be discussed in section 4.1. The loss in phase at 1800 Hz, using the sample period  $h = 39.0625 \mu\text{s}$ , is  $13^\circ$ . The zero-order hold sampler, with this sample rate, gives the following system:

$$\begin{cases} x(kh+h) = \Phi x(kh) + \Gamma u(kh) \\ y = i_{\text{out}} = Cx(kh) + Du(kh) \end{cases}$$

where

$$\Phi = \begin{pmatrix} 0.9445 & 0.0553 & -1.5959 \times 10^{-5} \\ 0.1572 & 0.8322 & -5.0687 \times 10^{-4} \\ 3.1918 & 35.6636 & 0.9898 \end{pmatrix}$$

$$\Gamma = \begin{pmatrix} 1.8937 \times 10^{-4} & -1.5959 \times 10^{-5} & -0.0555 \\ 1.5959 \times 10^{-5} & -5.0687 \times 10^{-4} & 0.1571 \\ 2.1171 \times 10^{-4} & -0.0102 & 3.1918 \end{pmatrix}$$

and  $C$  and  $D$  are the same as before.

From now on, the sample period  $h$  is set to 1, when systems are described as above. For instance,  $x(kh+h)$  is written as  $x(k+1)$ . The forward-shift operator, denoted by  $q$ , is used throughout the report.

The input-output models are obtained as follows:

$$u_{\text{in}} \text{ to } i_{\text{out}}: H(q) = B_h(q)/A_h(q)$$

where

$$\begin{aligned} B_h(q) &= 10^{-3} (0.1734q^2 - 0.3429q + 0.1733) \\ A_h(q) &= (q^3 - 2.7665q^2 + 2.5541q - 0.7864) \end{aligned}$$

$$u_s \text{ to } i_{\text{out}}: G(q) = B_g(q)/A_g(q)$$

where

$$\begin{aligned} B_g(q) &= 10^{-3} (0.4909q^2 - 0.9819q + 0.4910) \\ A_g(q) &= A_h(q) \end{aligned}$$

$i_r$  to  $i_{out}$ :  $F(q) = B_f(q)/A_f(q)$

where

$$\begin{aligned} B_f(q) &= (-0.2126q^2 + 0.4239q - 0.2126) \\ A_f(q) &= A_h(q) \end{aligned}$$

Just like in the continuous model, it is interesting to know where the poles and zeroes are located. (These are given in the  $Z$ -plane).

$$\begin{aligned} \text{root } B_h &= [0.9888 \pm i0.1464] \\ \text{root } B_g &= [1.0001, 1.0000] \\ \text{root } B_f &= [0.9971 \pm i0.075] \\ \text{root } A &= [0.8546, 0.9560 \pm i0.0793] \end{aligned}$$

The roots are plotted in figure 3.4. All zeroes are close to the unit circle, which is a result of the high sample rate and the system itself. Notice that the non-minimum phase in  $B_g$  still exists and the roots of  $A$  show that the system is stable.

The fact that the discrete roots are close to the unit circle may cause numerical problems. This is a problem that occurs when fast sample rate is used. To avoid this, other concepts can be used in converting the continuous system to a discrete equivalent. The two concepts proposed in [Åström89] are the delta and the Tustin operators. These use

$$\delta = \frac{q-1}{h}$$

and

$$\Delta = \frac{1}{2h} \frac{q-1}{q+1}$$

respectively. Still, the zero-order hold sampler is used throughout the report.

### 3.4 The Transmission Net

In reality, the converter is connected to a net that can change its characteristics during operation. This happens, for example, when the net is extended with new consumers or when the present net is connected in a different way. One problem at the converter when such a net is added is that the net reflect echos of the current sent out from the converter. The reflections also depend on the characteristics of the net. Thus, it is important to have a good model of the net.

Here the model consists of a load and a controllable current source, as can be seen to the right in figure 3.1. An expression for the current  $i_r$  is obtained

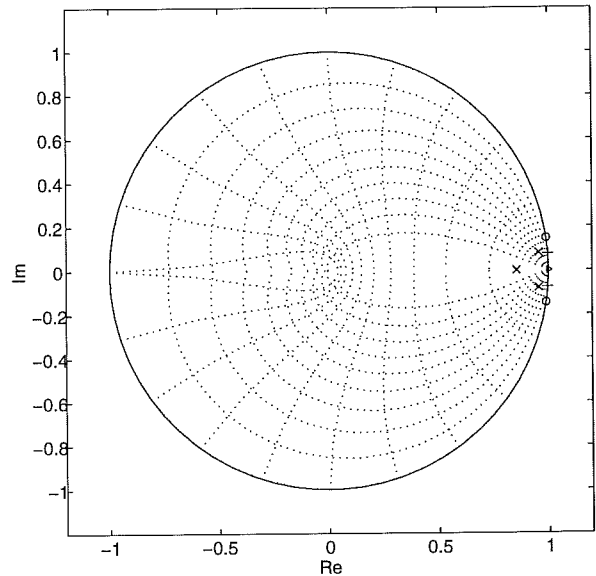


Figure 3.4: Poles and zeroes of the discrete system sampled with  $h = 39.0625\mu s$ . The roots of  $B_h$  are circles,  $B_g$  triangles,  $B_f$  plus signs, and  $A$  are crosses.

when the equation A.4 (appendix A.1) of wave propagation on a terminated transmission line (according to [Johnson]), is compared to a two-terminal-network equivalence. The result is:

$$i_r(t) = -a(i_r(t - \Delta) + 2i_{out}(t - \Delta))$$

where  $\Delta$  is the delay time for the echo to return and  $a$  is a factor ( $0 < a < 1$ ), containing line damping and end terminal reflection coefficient. The complete calculations are shown in appendix A.1. In this calculations it is assumed that  $\Delta$  and  $a$  do not depend on the frequency. This will be further commented on in appendix A.2, where the relation to frequency is studied. In the SIMNON simulation system, this model is implemented as a discrete system.

## Chapter 4

# Feedforward Control Using the MIT Rule

In this chapter feedforward control of the previously described system is investigated. The feedforward controller is using the estimated parameters obtained by the MIT rule. In the first section, a reference case is established with the true parameters. In the next section, the number of estimated parameters is increased. At the end of the chapter, all of the parameters are estimated.

To simulate the system, some sort of input signal is needed. This is described in section 4.1. Then, different feedforward controllers are described in sections 4.2, 4.3 and 4.4. All of them are implemented in the SIMNON simulator as discrete controllers.

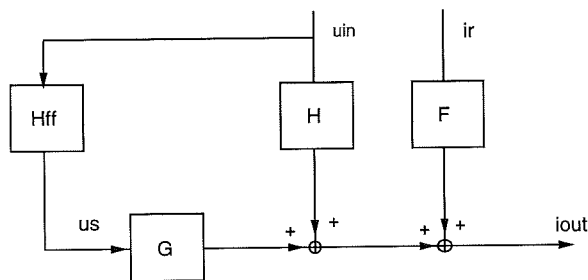


Figure 4.1: A true feedforward controller placed in the system.

### 4.1 The Test Signals

A signal  $u_{in}$  simulating the disturbances was implemented. As mentioned in section 2.2, this basically contains the frequencies  $12nf$  Hz, where  $f = 50$  Hz and  $n = 1, 2, 3, \dots$ , with the highest power at low frequencies. In some cases, the frequencies  $3nf$  also occurs with small harmonic amplitudes.

The sum of four sinusoidal signals (300, 600, 1200 and 1800 Hz) were used to simulate the input signal. In the program, the signal with the frequency of 300 Hz has the amplitude 2 while the others have the amplitude 5. The reason why the 300 Hz signal is included, is that the signal has to be persistently exciting in order to estimate the parameters in the transfer functions. A frequency below 600 Hz has to be present in the input signal. To realize this, the transfer functions in figure 3.3 can be studied. For example, if  $H(s)$  is to be estimated, there has to be information how the function acts on the frequencies below the resonance frequency 600 Hz. More information about persistently exciting is found in [Åström89].

The output signal ( $i_{out}$ ) without controller and without reflections from the transmission net ( $i_r = 0$ ) is shown in figure 4.2. In figures 4.3 and 4.4 reflections are included. The number of delays is set to 20. This corresponds to a transmission line with a length of about 117 km with the speed of the signals approximated with the speed of light. This is a good approximation if the transmission net conductor is in air. The damping factor  $a$  is chosen to 0.8. Later on other results will be compared to the three figures 4.2, 4.3, and 4.4.

If the power spectra with and without the transmission net added are compared, it is obvious that some frequencies are damped by the net and others are amplified. This can be understood by examining the transfer function from  $u_{in}$  to  $i_{out}$  with the transmission net added. Some calculations give the function

$$H_r = \frac{B_h(1 + aq^{-\Delta})}{A(1 + aq^{-\Delta}) + 2aB_fq^{-\Delta}}$$

which has the bodeplot showed in figure 4.5 (if  $\Delta = 20$  and  $a = 0.8$ ). Notice that the frequencies 600

and 1800 Hz are damped more than the signal with frequency 1200 Hz and that the 300 Hz signal is approximately the same.

In the SIMNON simulator, the signal generator was implemented as a continuous signal, except for some tests in chapter 6. Numerical problems occurred and the signal generator had to be changed to a discrete system. We will comment on this when the change is made.

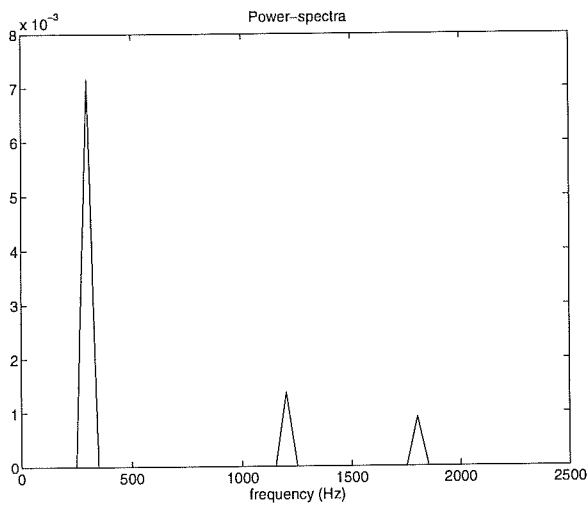


Figure 4.2: The power spectrum of  $i_{out}$  without controller and reflections.

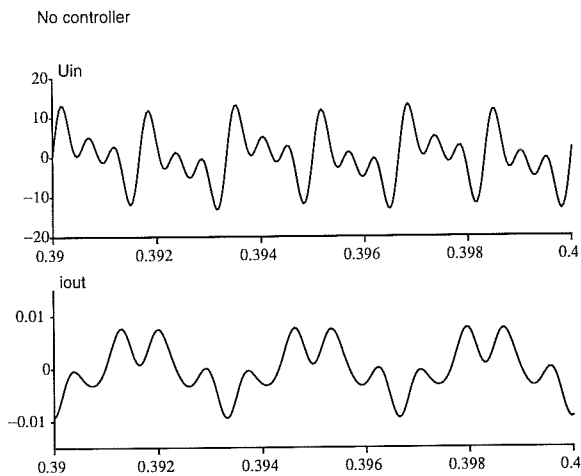


Figure 4.3: The current  $i_{out}$  without controller but with reflections ( $\Delta = 20$   $a = 0.8$ ). The time-scales are in seconds.

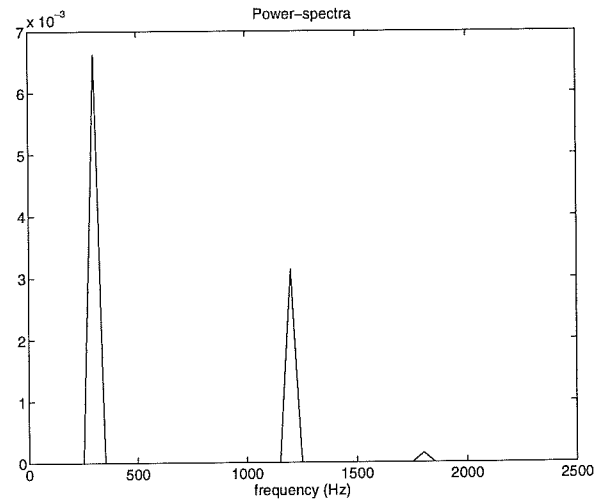


Figure 4.4: The power spectrum of  $i_{out}$  without controller, but with reflections ( $\Delta = 20$   $a = 0.8$ ).

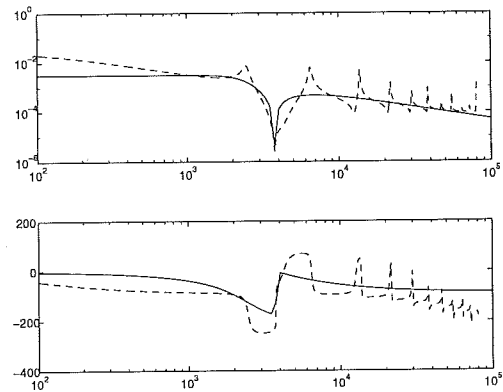


Figure 4.5: The transfer function from  $u_{in}$  to  $i_{out}$  when reflections are added (‘-’) and without reflections (‘—’). The frequencies are given in rad/s.

## 4.2 Controller Without Adaptation

The first controller to be studied was an ordinary feedforward controller such as the one in figure 4.1. If  $i_r = 0$ , the output current will be

$$i_{out} = (H_{ff}G + H)u_{in}$$

From this, the ideal feedforward controller can be

obtained, because it is obvious that

$$H_{\text{ff}}(q) = -G^{-1}H = -\frac{A_g(q) B_h(q)}{B_g(q) A_h(q)}$$

gives minimum output. Using this,

$$A_g(q) = A_h(q)$$

gives

$$H_{\text{ff}}(q) = -\frac{B_h(q)}{B_g(q)}$$

where

$$B_h(q) = (0.1734q^2 - 0.3429q + 0.1733) \quad (4.1)$$

and

$$B_g(q) = (0.4909q^2 - 0.9819q + 0.4910)$$

as calculated. The factor  $10^{-3}$  is present in the calculated  $B_g(q)$  and  $B_h(q)$ , but is left out when creating  $H_{\text{ff}}(q)$ . The polynomial  $B_g(q)$  has a root outside the unit circle making  $H_{\text{ff}}(q)$  unstable. Therefore, we make the approximation that  $B_g$  has a double pole. This gives

$$B_g(q) = a_f (q - p)^2 \quad (4.2)$$

where  $a_f$  is 0.4909. The calculated value of  $p$  is approximately equal to unit, but in this controller it is chosen to a slightly smaller value to prevent instability. To be sure not to come close to instability,  $p$  was chosen to 0.997.

The result of using this controller without any reflections from the net is shown in figure 4.6. Compared to the result shown in figure 4.2, it can be seen that the damping varies between 25 and 400 times, depending on the frequency. The reason why lower frequencies are more annihilated is the choice of the double pole  $p$ . If  $p$  is increased to a value very close to unit, the higher frequencies will be more annihilated. This will be seen in later simulations and then also be further commented on.

The result with the reflections added is shown in figure 4.7 and 4.8. This also has a strongly damping effect on lower frequencies.

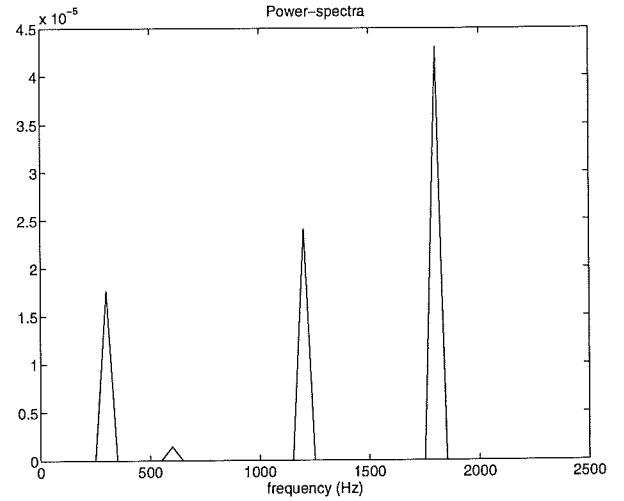


Figure 4.6: The power spectra of  $i_{\text{out}}$  when feedforward controller without adaptation and reflections is used. The calculated values are used.

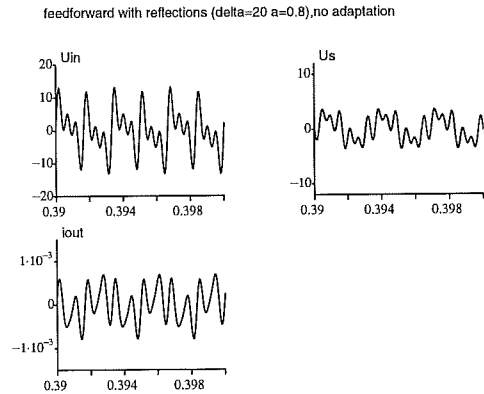


Figure 4.7: Feedforward controller without adaptation and with reflections included ( $\Delta = 20$  and  $a = 0.8$ ). The time scales are given in seconds.

### 4.3 Adaptive Control of the Feedforward Gain

The first controller, where the transfer functions have to be known, is not very useful in reality. To obtain a better controller, adaptation is needed. In this section, the constant  $a_f$  in equation 4.2 is estimated. The double pole has the same value as before. This means that the controller is

$$H_{\text{ff}}(q) = k \frac{B_h(q)}{(q - 0.997)^2}$$



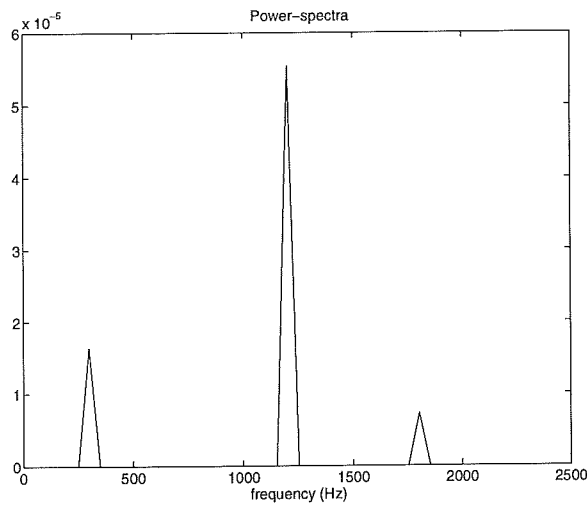


Figure 4.8: The power spectrum of  $i_{\text{out}}$  when feed-forward controller without adaptation and with reflections included, is used.

where  $k$  is estimated. To do this, the MIT rule is used according to [Åström89, chapter 3]. This rule is based on gradient search according to

$$\dot{\Theta}(t) = -\gamma e(t) \frac{\partial e(t)}{\partial \Theta(t)}$$

where in this case the error  $e$  is equal to the output current ( $i_{\text{out}}$ ). However, in practical use a little modification is needed. To make the estimating rate independent of the magnitude of the input signal ( $u_{\text{in}}$ ), normalization is introduced.

The function minimized by the MIT rule is the sum of squares error, that is,

$$J = \sum e^2$$

This equation can sometimes have several minima. If the gain or the step-size in the gradient search, that is, the derivative of the parameter to change, is too big, the correct minima could be missed. To prevent this, a limitation of the derivative is introduced in the MIT rule.

Normalization and introduction of saturation give the following rule:

$$\dot{\Theta}(t) = -\gamma \text{sat} \left( \frac{i_{\text{out}} \frac{\partial i_{\text{out}}(t)}{\partial \Theta(t)}}{\alpha + \left( \frac{\partial i_{\text{out}}(t)}{\partial \Theta(t)} \right)^T \left( \frac{\partial i_{\text{out}}(t)}{\partial \Theta(t)} \right)}, \beta \right)$$

where

$$\text{sat}(a, \beta) = \begin{cases} -\beta & a < -\beta \\ a & |a| \leq \beta \\ \beta & a > \beta \end{cases}$$

In this case  $\Theta = k$ .

The following equivalence discrete version is

$$k(k) = -\gamma \text{sat} \left( \frac{i_{\text{out}}(k) f(k)}{\alpha + (f(k))^2} \right) + k(k-1)$$

where

$$f(k) = \frac{\partial i_{\text{out}}(t)}{\partial k(t)} = \frac{B_g(q)}{A(q)} \frac{B_h}{(q - 0.997)^2} u_{\text{in}}(k)$$

The estimated  $B_g(q)$  was used in the equation  $f(k)$  above, when this was implemented in the program. An alternative is to use the calculated one. This was tested, but no difference in the estimates or convergence rate was detected.

The term  $\alpha$  is to prevent a division by zero and in the first estimator  $\alpha = 0.5$  was used. In this estimator, the limiter  $\beta$  is equal to 0.01. This gave the result shown in figure 4.9. The final value of the constant was  $-1.9418$  and the resulting damping was now a little higher for the high frequency range and the same as before for the others.

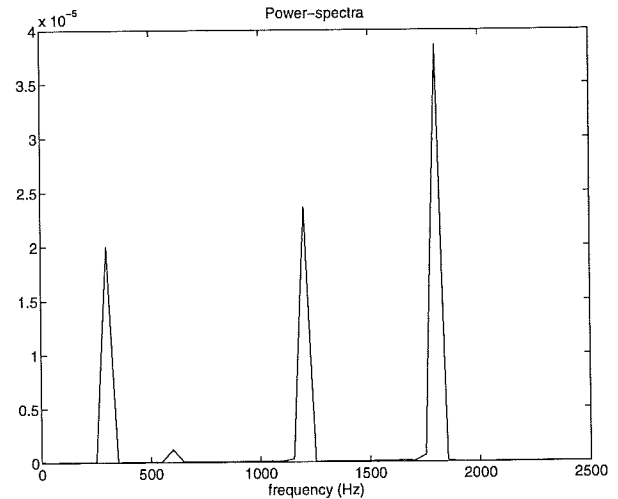


Figure 4.9: The power spectrum of  $i_{\text{out}}$  when feed-forward controller with MIT estimator is used. No reflections are added. Estimates the feedforward gain.

The factor  $\gamma$  controls how fast the estimation will converge and in figure 4.9  $\gamma = 1$ . It is possible to

have larger  $\gamma$  values before the estimator will be unstable, but this will cause ripple in the estimated parameter.

Note that the convergence rate also depends on  $\alpha$ , especially if the value is too large. A large  $\alpha$  also gives a small error in the MIT rule. If  $\alpha$  is too small, on the other hand, the denominator in the modified rule can be very close to zero and the derivative of the parameters will be large. This will result in a ripple in the estimates in the same way as a large  $\gamma$  will. Tests are required to find the optimum  $\alpha$ .

The adaptive controller worked quite well without reflections and theoretically it should work well even when these are added. This is because the MIT rule just changes the parameters to minimize the amplitude of the signal  $i_{\text{out}}$  and does not matter if there is another signal added.

Tests showed that the theory was correct. The result is shown in figures 4.10 and 4.11. The annihilation of the disturbances is good. If the power spectrum is compared to the one where the calculated values are used (figure 4.8), there is a little improvement. The parameters in the estimator were  $\gamma = 5$ ,  $\beta = 0.01$ ,  $\Delta = 20$ , and  $a = 0.8$ . This gave the estimated value  $k = -1.94053$ .

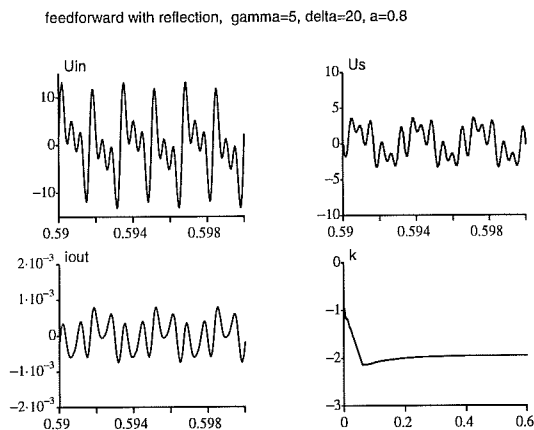


Figure 4.10: Feedforward controller with MIT estimator when reflections are added. ( $\gamma = 5$ ,  $\Delta = 20$  and  $a = 0.8$ . The final value of  $k$  is  $-1.94053$ . The time is given in seconds.)

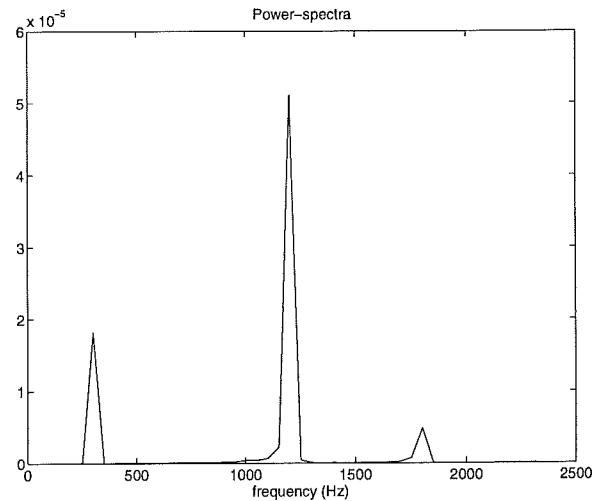


Figure 4.11: The power spectrum of  $i_{\text{out}}$ . A feed-forward controller with MIT estimator is used and the reflections are added.

## 4.4 Adaptive Control of the Entire $H_{\text{ff}}$

The first try was to estimate the coefficients in the numerator as well as in the denominator at the same time, but this was not very wise. If one of the four coefficients did not converge, the whole estimation procedure was disturbed. Therefore, the numerator and the denominator were examined separately, to finally be put together.

### 4.4.1 The Denominator

Using the same estimating rule as before, the constant  $a_f$  and the double pole  $p$ , in equation 4.2, were estimated. Because of the different size of the estimated parameters, different  $\gamma$ ,  $\alpha$ , and  $\beta$  in the MIT rule were used to each of the parameters.

The numerator in  $H_{\text{ff}}(q)$  was kept constant and equal to the one in equation 4.1. The error  $e = i_{\text{out}}$  was differentiated to create the estimation rule.

$$\frac{\partial i_{\text{out}}}{\partial a_f}(q) = G(q) \frac{B_h(q)}{a_f^2 (q-p)^2}$$

$$\frac{\partial i_{\text{out}}}{\partial p}(q) = -2G(q) \frac{B_h(q)}{a_f (q-p)^3}$$

In the equations above, the estimated parameters are present on the right-hand side (both in  $G(q)$  and

the denominator). To implement this in the control system, an approximation is needed. There are now two alternatives. The first is to replace the parameters with the calculated ones. As mentioned before, this can't be done because  $B_g$  has an unstable root. The second alternative is to use the estimated values one sample ago. This makes the estimator very complex and hard to analyze, but as simulations showed it worked quite well.

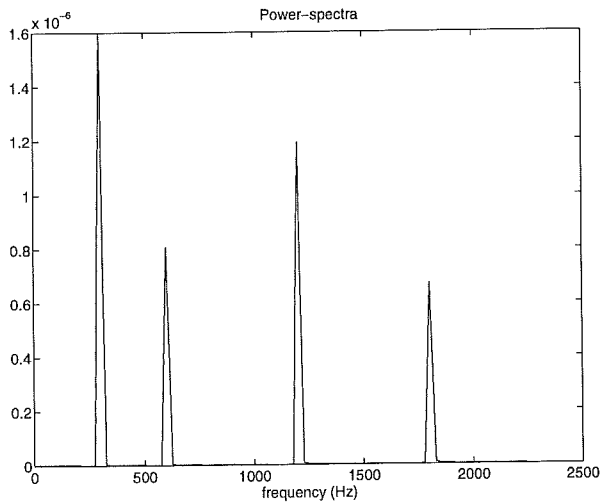


Figure 4.12: The power spectrum of the output current. A feedforward controller is used and the denominator is estimated. No reflections are added.

In figure 4.12, the power spectrum of the output current  $i_{out}$  is shown when no reflections are added. In the two figures 4.13 and 4.14, the reflections are included. In both simulations, the parameters controlling the estimator were  $\gamma_{a_f} = 0.001$ ,  $\gamma_p = 0.002$ ,  $\alpha_{a_f} = 0.5$ ,  $\alpha_p = 0.1$ ,  $\beta_{a_f} = 0.002$ , and  $\beta_p = 0.002$ . The final values were  $a_f = 5.0247 \times 10^{-4}$  and  $p = 0.999691$ .

As seen in figure 4.13, both parameters converge rather fast. Comparing to former results, it is also obvious that the controller has good performance. For example, when reflections are added, damping is 1500 to 6500 times. This is a very good result, but there is one problem, though.

The transfer function can not easily become unstable, because if there is a tendency for instability, the output and also the error  $i_{out}$  would grow. Then the estimator would decrease the value of the pole and make the system stable again. This is not a major problem.

The problem is instead the fact that the controller

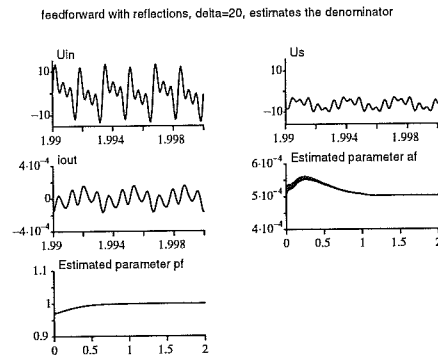


Figure 4.13: The feedforward controller when reflections are added. The final values of the estimates are  $a_f = 5.02472 \times 10^{-4}$  and  $p = 0.999691$ . Time is given in seconds.

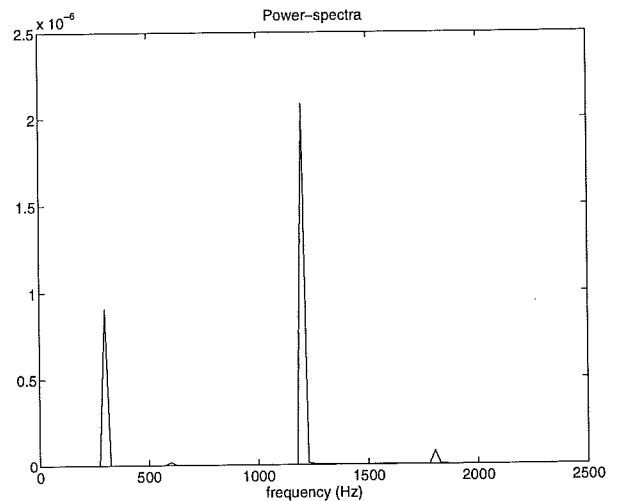


Figure 4.14: The spectrum of  $i_{out}$  when the denominator is estimated. Reflections are added.

will integrate the signal  $u_{in}$  when  $p$  comes close to one. If  $u_{in}$  contains any low frequency component, the output from the controller will increase. Such low frequencies can of course come from the signal  $u_{in}$ , but can also be the result of numerical errors (for example, round-offs) in the computations. This was the case in the simulations in this report.

Due to this problem, a DC off-set occurs in the control signal. This can, for example, be seen in figure 4.13. At first, the bias will not be noticed in  $i_{out}$ , because low frequencies are not transferred from  $u_s$  to  $i_{out}$  (see figure 3.3). It will, however, be noticed when the control signal reaches its limits. To

be able to use a controller like this one, some concept of protection is needed. Probably, a high-pass filter on the input of the controller will not be sufficient: since an ideal filter is not possible to make, some low frequencies will enter and the computational errors will remain. The protection has to be internal to the controller, observing the DC component in the states and sometimes resetting them to zero.

#### 4.4.2 The Numerator

The three parameters in equation 4.1 were estimated, while the parameters in the denominator were held constant ( $0.4909 \times 10^{-3}$ , 0.997). It was not easy to obtain good results, though. With a small amplification factors  $\gamma$ , the convergence was very slow. If the  $\gamma$ 's were made larger, ripple in the estimated parameters occurred, destroying convergence. If all parameters have to be estimated, another algorithm have to be used. A different and a better concept is to decrease the number of estimated parameters. By using the information that  $B_h(q)$  have complex roots on the unit circle, the following simplified model of the numerator can be stated:

$$B_h(q) = k(q^2 - a_f q + 1)$$

In simulations it were much easier to estimate only these two parameters. No result of these simulations are shown because the result is much the same as in the following section, where the numerator and the denominator were estimated at the same time.

#### 4.4.3 Both the Numerator and the Denominator

The following function was used, when putting together the numerator and the denominator:

$$H_{\text{ff}}(q) = \frac{k(q^2 - a_f q + 1)}{(q - p)^2} = \frac{B_{\text{ff}}}{A_{\text{ff}}}$$

Now were there only three parameters to estimate. These are  $k$ ,  $a_f$ , and  $p$ . The derivative of the error with respect to these parameters are:

$$\begin{aligned} \frac{\partial i_{\text{out}}}{\partial k} &= G \left[ \frac{q^2 - a_f q + 1}{(q - p)^2} u_{\text{in}} \right] \\ \frac{\partial i_{\text{out}}}{\partial a_f} &= G \left[ \frac{-kq}{(q - p)^2} u_{\text{in}} \right] \\ \frac{\partial i_{\text{out}}}{\partial p} &= G \left[ \frac{2k(q^2 - a_f q + 1)}{(q - p)^3} u_{\text{in}} \right] \end{aligned}$$

In section 4.4.1, when only the denominator was estimated, the estimated parameters on the right side were taken from the estimates one sample ago. The numerator ( $B_g$ ) in  $G$  was replaced with the estimates as well. This cannot be done in the same way here, because the constant  $K$  in  $B_g = K(q - p)^2$  cannot be separated from the estimated  $k$ . Therefore the calculated  $B_g = (0.4904 \times 10^{-3} q^2 - 0.9819 \times 10^{-3} q + 0.4910 \times 10^{-3})$  is used.

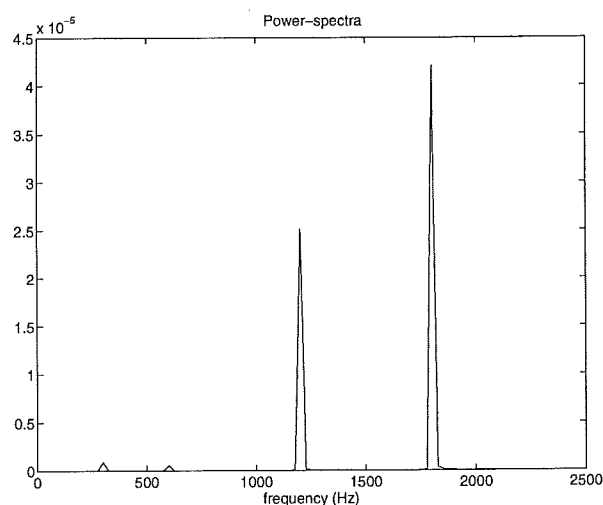


Figure 4.15: The power spectrum when the three parameters  $k$ ,  $a_f$  and  $p$  were estimated. Final values are  $k = -0.34845$ ,  $a_f = 1.9781$  and  $p = 0.99825$ .

The extended MIT rule was used and the control result became a little better than achieved using the estimation in section 4.3. The result with no reflections added is shown in figure 4.15. At disturbances are low frequencies 300 and 600 Hz more damped and at the other frequencies the damping is very much of the same. In figure 4.16 and 4.17, the estimation and control are shown, with reflections from the transmission net added. The reflection parameters are as before  $\Delta = 20$  and  $a = 0.8$ . If these are compared with the case when only the gain was estimated, the 300 Hz component is more annihilated. Unfortunately, the 1200 Hz signal became less annihilated. The situation is even worse if the results in section 4.4.1 (where the denominator was estimated) are compared to the results in this section. The conclusion is that the former estimator showed better performance.

In principle, this difference is not a result of the new estimator. It is only a question of the choice of parameters in the estimator and the time used to estimate. In the simulation done here, the parameters

are  $\gamma_k = 0.6$ ,  $\gamma_{a_f} = 0.02$ ,  $\gamma_p = 0.001$ ,  $\alpha_k = 0.1$ ,  $\alpha_{a_f} = 0.5$ ,  $\alpha_p = 0.1$ ,  $\beta_k = 0.001$ ,  $\beta_{a_f} = 0.01$ , and  $\beta_p = 0.01$ . This gave the estimated values  $k = -0.36708$ ,  $a_f = 1.97876$ , and  $p = 0.998121$ . Notice the big difference in the values of  $p$  in the different simulations. This is the reason behind the big differences in performance. If other parameters had been chosen, the estimated values would have been different. Probably a large increase of the time available for estimation will give similar results in the two cases, but this has not been investigated.

How to change the parameters is not obvious, though. A large gain  $\gamma$  will give a faster convergence, but can, especially regarding  $p$ , give less correct estimates. If the estimated value of  $p$  comes close to one too fast, the estimate will have difficulties finding the optimum value. It will have a variation of the value in the neighbourhood of the true value. To change  $\alpha$  is not easy either. A too large  $\alpha$  gives a kind of bias in the estimates and a too low  $\alpha$  will cause ripple in the estimates. This has been discussed in section 4.3. Finally, the parameter  $\beta$ : just as the other parameters, this has an upper and a lower limit. If it is too big, it becomes less significant and if it is too small, the rate of convergence is slow.

A better tuning of the MIT parameters would probably result in similar performance as in the case where only the denominator was estimated.

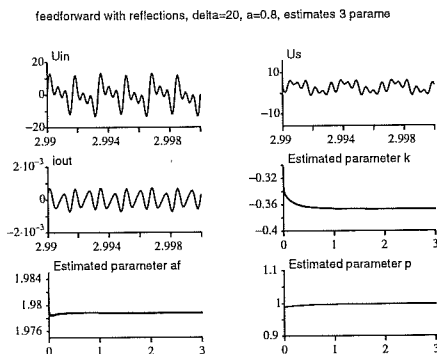


Figure 4.16: The feedforward controller when  $k$ ,  $a_f$ , and  $p$  are estimated and reflections are added. The final values are  $k = -0.36708$ ,  $a_f = 1.97876$ , and  $p = 0.998121$ . The time axes are in seconds.

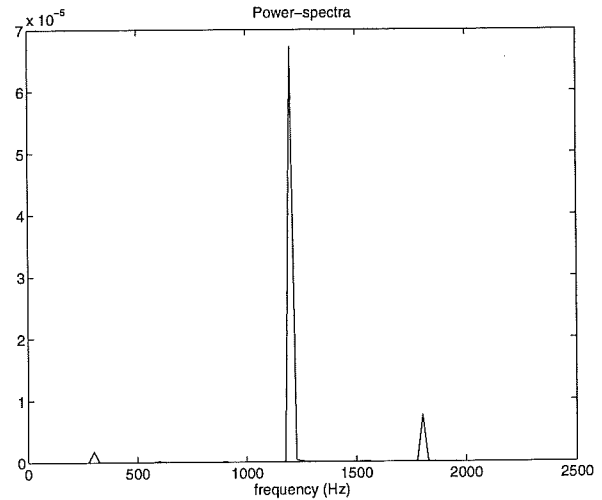


Figure 4.17: The feedforward controller when the reflections with  $\Delta = 20$  and  $a = 0.8$  is added. The signal is the power spectrum of the output current.

## 4.5 Conclusions

In this chapter a rather simple way of combining control and estimation has been used. In this simplified model of the complex reality, the system worked well. The performance of the controller depends, however, strongly on the choice of parameters in the estimator and these have to be chosen with precaution. The major benefit with this concept is that no model of the reflections is needed in the controller. As will be seen in later chapters, this problem may occur in other concepts. Another benefit using this concept is a rather stable controller, if the parameters mentioned before are chosen correctly.

One important thing is to implement some kind of resetting in the controller to prevent bias. This will not appear only for control concepts of this chapter. It will also be needed in chapter 6 where it will be discussed further.

The SIMNON program used in the last simulation is given in appendix B.

## Chapter 5

# Estimating the Models Using RLS

In the previous chapter, the numerators of the transfer functions  $G(q)$  and  $H(q)$  were estimated using the MIT rule. This rule has the major drawback of being rather slow. In this chapter, a recursive version of the least-mean-squares method with forgetting factor is used to improve the convergence rate. It should be noticed that the convergence in some cases is rather slow, though. This is a result of the great amount of estimated parameters and the use of a simplified version of the algorithm, in which the covariance matrix is replaced by a scalar. The advantages of this simplification are less computations and no need of matrices in the program (SIMNON does not support vectors and matrices).

reflections are estimated. This is a secondary effect and more work has to be done if a good model of the net is required. The simulations show, however, the ability to estimate a rather simple and theoretical net, like the one used in this report.

In figure 5.1, the estimator is shown together with the system. The filters are not always utilized, but when they are, this will be commented on. The input signal  $u_{in}$  is either zero or the same as in chapter 4, that is, a sum of sinusoidals. The signal  $d$  in the figure is white Gaussian noise, to ensure persistent excitation when the transfer function  $G(q)$  is to be estimated [Åström89].

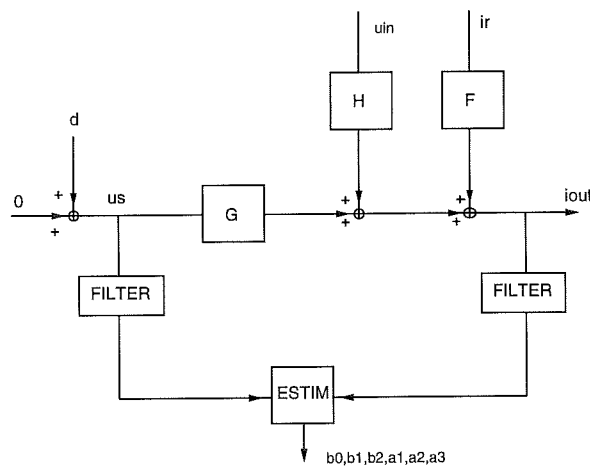


Figure 5.1: The whole system with no reflections added.

In order to create different adaptive control concepts, it is important to have the transfer functions  $G(q)$  and  $H(q)$  available. Initially,  $G(q)$  is estimated in section 5.1, after which both  $G(q)$  and  $H(q)$  are estimated in section 5.2. If the reflections are added, also most of the parameters in the model of the net

## 5.1 Estimating the Transfer Function $G(q)$

### No Reflections From the Transmission Net

In this section, the reflections are excluded, that is,  $i_r = 0$  in figure 5.1. During estimation of the parameters in  $G(q)$ , the signals  $u_{in}$  and  $d$  were zero and white Gaussian noise, respectively. The result can be seen in figure 5.2. Unfortunately, it is very difficult to see the convergence of the parameters in this figure. It is, however, possible to see how the difference between the true output ( $i_{out}$ ) and the estimated one ( $\varphi^T \Theta$ ) decreases. The convergence can be seen more easily if the absolute values of the poles and zeroes are plotted; this is done in figure 5.3. To be able to clearly see the convergence, the initial values of  $B_g$  and  $A$  are departed a little from the calculated ones. The final values of the parameters are very close to calculated ones and the rate of convergence is fast.

The algorithm used is an ordinary recursive least-mean-squares with forgetting factor. The algorithm

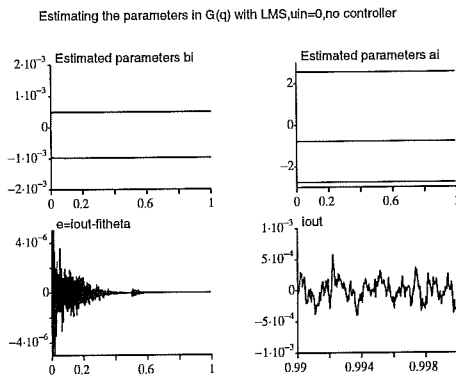


Figure 5.2: Estimating the transfer function  $G(q)$  when no disturbances are added, that is,  $u_{in} = 0$ . The signal  $d$  is white Gaussian noise, which explains the signal  $i_{out}$ . The signal  $e$  is the error between the correct output and the estimated. The time is given in seconds.

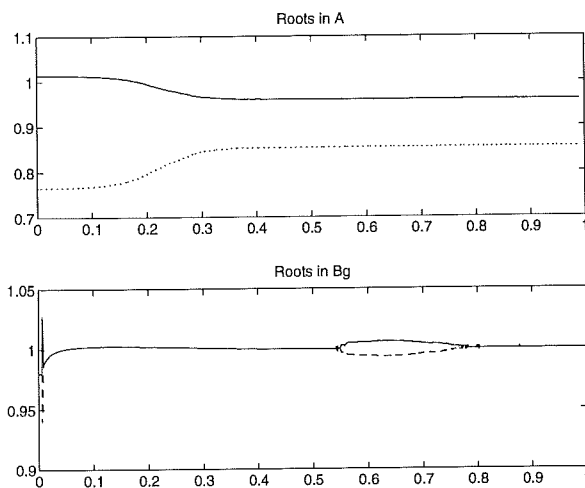


Figure 5.3: The absolute values of the poles and zeroes. The full line in  $A$  contains two poles which are complex.

is given by

$$\begin{aligned}\hat{\Theta}(t) &= \Theta(t) + K(t) (i_{out}(t) - \varphi^T(t)\hat{\Theta}(t-1)) \\ K(t) &= P(t)\varphi(t) = \\ & P(t-1)\varphi(t) (\lambda I + \varphi^T(t)P(t-1)\varphi(t))^{-1} \quad (5.1) \\ P(t) &= (I - K(t)\varphi^t(t)) P(t-1)/\lambda\end{aligned}$$

where in this case

$$\Theta(t) = [a_1(t) \ a_2(t) \ a_3(t) \ b_0(t) \ b_1(t) \ b_2(t)]$$

$$\varphi^T(t) = [i_{out}(t-1) \ i_{out}(t-2) \ i_{out}(t-3) \\ u_s(t-1) \ u_s(t-2) \ u_s(t-3)]$$

and

$$G(q) = \frac{b_0q^2 + b_1q + b_2}{q^3 + a_1q^2 + a_2q + a_3}$$

More information about this algorithm is found in [Åström89, chapter 3].

The initial value of the covariance matrix  $P$  was set to a diagonal matrix. The diagonal elements related to the current  $i_{out}$  were given the initial value 100, while the elements related to the control signal  $u_s$  were set to 0.1.

Here,  $\lambda$  is the exponential forgetting factor. The algorithm approximately remembers  $n = 2/(1 - \lambda)$  samples of old information. In other words, the value sampled  $n$  samples ago gives almost no contribution to the present calculations in the estimator. For the value  $\lambda = 0.999$ , which is used in this simulations,  $n$  will be 2000. This equals 0.078 seconds of memory with a chosen sample period  $h = 39.0625 \times 10^{-6}$  seconds.

The performance of the RLS algorithm was not satisfying when the signal  $u_{in}$  was added (the same test signal as in section 4.1 was used). This unmodeled signal creates a bias in the estimates, because the estimator tries to do a model for both  $G(q)$  and the new signal. There are now three different alternatives to solve this problem.

The first is to create a model for the signal. For instance,

$$i_{out} = \frac{B_g}{A} u_s + \frac{C}{D} e$$

where  $e$  is Gaussian random noise, can be used. Then, the extended least-squares (ELS) algorithm can be used. This is an extension of the equation 5.1 which results in estimation of the transfer functions  $C(q)$  and  $D(q)$ . The algorithm is described in appendix A.3. A serious drawback of using the ELS in this example is that  $C(q)$  and  $D(q)$  are of high order and in order to estimate all parameters, much more computation is needed. Therefore, if there are other alternative, they should prove preferable.

The second alternative is to assume that  $u_{in}$  can be measured. Then, a model for  $H(q)$  can be stated and estimated. This is done in section 5.2.

The third alternative is to separate the signals  $u_{in}$  and  $d$  in the frequency spectrum. This way, the information needed in the estimator can be filtered

out from the measurable output  $i_{out}$ . To compensate for the filter's influence on the  $d$ -part in  $i_{out}$ , a duplicate of the filter is used on the signal  $d$  (figure 5.1).

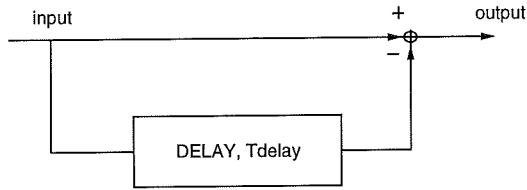


Figure 5.4: A simple FIR filter to annihilate the periodic signal in  $u_{in}$ .

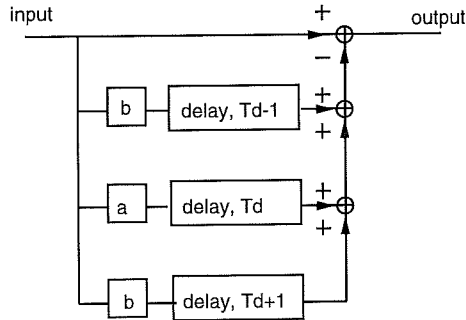


Figure 5.5: An extended FIR filter.

Using the simplified model of a HVDC station, the signal  $d$  can be noise with frequencies below 300 Hz and above 5000 Hz. Signals within this spectrum are sufficient to estimate the parameters. In this case, the filters in figure 5.1 are only band-stop filters.

In reality, the transfer functions are not this simple and information about the frequencies between 300 and 5000 Hz is needed in order to estimate the parameters. Then one can use the fact that the signal  $u_{in}$ , and also all control signals to compensate for this disturbance, are periodic with a cycle of 1/50 seconds. If this is assumed, it is possible to annihilate these signals with a FIR filter such as the one in figure 5.4. The delay-box delays the signal with one period, that is,  $T_{delay} = 1/50$  seconds. If this simple filter is not enough, several identical filters can be connected in cascade. In reality, where the frequencies of the disturbance can fluctuate, a wider filter like the one in figure 5.5 might be preferable.

The added disturbance  $d$  can now be Gaussian random noise and the information between the annihi-

lated frequencies can be used to estimate the transfer function  $G(q)$ . The signals that pass are violated both in amplitude and phase. Therefore, it is important to filter both the signals  $i_{out}$  and  $d$  through the same type of filter.

Note that if a non-periodic signal enters the system via  $u_{in}$ , the estimator must be turned off in order to protect the estimated values. This safety algorithm is not implemented in this report.

Unfortunately, the estimation with these FIR filter included could not be tested in simulations. The reason is the difficulties to implement such a long delay in SIMNON. The program supplies a function called DELAY, but this is not accurate enough to use in long delays. In a program supporting vectors and matrices it is easier to create a delay. The system, including the filter, can then be implemented.

### Including the Reflections

The problem with unmodeled dynamics that arose when  $u_{in}$  was added, occurred in this section when the transmission net was connected ( $i_r \neq 0$ ). In this case, the new filters needed are much more complicated. The effect of reflections on the transfer function  $H(s)$  can be seen in figure 4.5. The other transfer functions are changed similarly. For example, see the transfer function  $G(s)$  in figure 5.6.

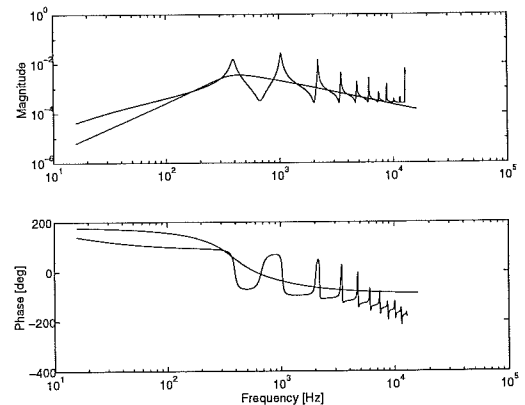


Figure 5.6: The transfer function  $G(s)$  describing the connection between  $u_s$  and  $i_{out}$  with and without reflections from the transmission net.

Instead of filtering the signals, the concept to model the reflections was used. This is done by combining

$$i_r(k) = -a(2i_{out}(k - \Delta) + i_r(k - \Delta))$$



with

$$i_{\text{out}}(k) = \frac{B_g}{A} u_s(k) + \frac{B_h}{A} u_{\text{in}}(k) + \frac{B_f}{A} i_r(k)$$

Since the FIR filter could not be implemented in the program, the input signal  $u_{\text{in}}$  was set to zero.

This yields

$$i_{\text{out}}(k) = \frac{B_g(1 + aq^{-\Delta})}{A(1 + aq^{-\Delta}) + 2B_f aq^{-\Delta}} u_s(k)$$

The regressor vector then became

$$\varphi^T(k) = \begin{bmatrix} i_{\text{out}}(k-1) \dots i_{\text{out}}(k-3) \\ i_{\text{out}}(k-\Delta) \dots i_{\text{out}}(k-\Delta-3) \\ u_s(k-1) \dots u_s(k-3) \\ u_s(k-\Delta-1) \dots u_s(k-\Delta-3) \end{bmatrix}$$

and the vector of estimated parameters

$$\hat{\Theta}^T = [a_1 \dots a_3 \quad ap_1 \dots ap_4 \quad b_{g0} \dots b_{g2} \quad b_{gp0} \dots b_{gp2}]$$

where

$$\begin{aligned} ap_1 &= a \\ ap_2 &= a(a_1 + 2b_{f0}) \\ ap_3 &= a(a_2 + 2b_{f1}) \\ ap_4 &= a(a_3 + 2b_{f2}) \\ b_{gp0} &= ab_{g0} \\ b_{gp1} &= ab_{g1} \\ b_{gp2} &= ab_{g2} \end{aligned}$$

Here, it is assumed that the delay  $\Delta$  is known. In reality, this parameter has to be found out through an initial test. If there are several reflections, more parameters have to be estimated. As long as there is no reflections contributing to the signals delayed between one and three samples, the polynomials  $A$  and  $B_g$  can be estimated.

It can be seen that not all of the parameters have to be estimated. The parameters  $b_{gp1}$ – $b_{gp3}$  can be calculated from  $ap_1$ – $ap_4$  and  $b_{g0}$ – $b_{g2}$ . This is not used in the simulations here. In reality, the factors  $a$  and  $\Delta$  depend on the frequency and the simple analytical connections between the parameters do not exist.

A simplified version of the RLS (equation 5.1) was used for estimation. This is called the *Stochastic approximation algorithm* [Åström89]. Following equation can be stated, if this algorithm is extended with a forgetting factor.

$$\hat{\Theta}(k) = \hat{\Theta}(k-1) + P(k)\varphi(k) \left( i_{\text{out}}(k) - \varphi^T(k)\hat{\Theta}(k-1) \right)$$

where

$$P(k) = \left( \sum_{i=1}^k \lambda^{k-i} \varphi^T(i)\varphi(i) \right)^{-1}$$

The benefit of this method is that the covariance matrix becomes a scalar, which, as mentioned before, requires less computation. The disadvantage is slower convergence. To make it faster and to ensure convergence, covariance resetting is used. Because of different size of the signals  $i_{\text{out}}$  and  $u_s$ , different initial values of the covariance ( $P$ ) are used for the two signals.

The result is shown in figure 5.7. Initially,  $B_g$  has a double zero at 0.98 and  $B_f$  has a double zero at 0.992. The denominator  $A(q)$  is set to the same as the one computed in chapter 3, except for one of the parameters ( $a_1$ ), which has been changed from  $-2.7665$  to  $-2.760$ . The forgetting factor is set to 0.999 and the initial covariances to 0.1 and 1000 for  $u_s$  and  $i_{\text{out}}$ , respectively.

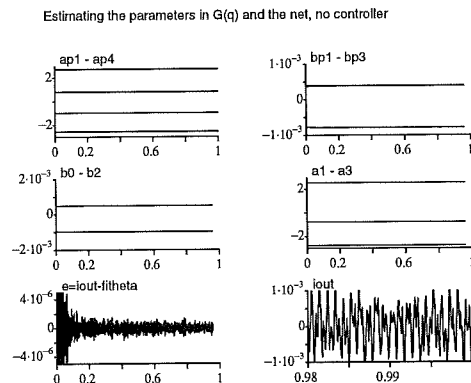


Figure 5.7: The estimation of  $G(q)$  with reflections added. The input to the system is only white Gaussian noise in the signal  $u_s$ . The time is given in seconds.

Even small changes in the parameters make large changes in the locations of the poles and zeroes. In figure 5.8, the absolute values of the poles and zeroes are shown. To make it easier to see the convergence. A faster convergence can be obtained if the initial covariance is increased, but then the estimates are more noisy, especially before all parameters have come close to their correct values.

As mentioned before, the number of parameters needed to describe a net are, in reality, large. Tests were made to simplify the implemented estimator.

Instead of using all parameters to model the influence of the reflected signal, only one or two were used, for example  $ap_1$  and  $i_{out}(k - \Delta)$ . The result was not encouraging, though. The bias in A and  $B_g$  reminded.

Another way to avoid the large amount of estimated parameters can be to use the ELS algorithm, described earlier. Instead of using the delayed  $i_{out}$  as regressor in the estimator, the error between the true output and the estimated one might be used. This idea has not been tested.

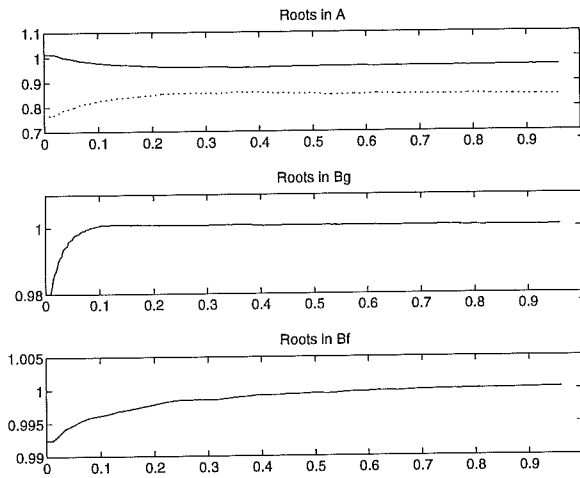


Figure 5.8: The absolute values of the poles and zeroes.  $B_f(q)$  is calculated through the estimates  $ap_1$  to  $ap_4$ . Time is given in seconds.

## 5.2 Estimating Both $G(q)$ and $H(q)$

### No Reflections From the Transmission Net

In this section, the signal  $u_{in}$  in figure 5.1 was measured and included. Just like in chapter 4,  $u_{in}$  was a sum of sinusoidals with frequencies 300, 600, 1200, and 1800 Hz. Even the amplitudes were set to the same values (2, 5, 5, and 5 respectively). The white Gaussian noise was still added to signal  $d$ .

The stochastic approximation algorithm with forgetting factor was used as before, but the regressor and the vector of estimated parameters were extended with

$$\begin{aligned} \varphi^T(k) &= [u_{in}(k-1) \dots u_{in}(k-3) \\ &\quad u_{in}(k-\Delta-1) \dots u_{in}(k-\Delta-3)] \\ \hat{\Theta}^T &= [b_{h0} \dots b_{h2} \quad b_{hp1} \dots b_{hp3}] \end{aligned}$$

As before, there are analytical connections between the first and the last three parameters in the vector  $\hat{\Theta}^T$ . These are

$$\begin{aligned} b_{hp0} &= ab_{h0} \\ b_{hp1} &= ab_{h1} \\ b_{hp2} &= ab_{h2} \end{aligned}$$

Firstly, the estimator was tested without the reflections. The result of this simulation is shown in figures 5.9 and 5.10. Some of the parameters and most of the initial values are changed compared to the previous simulations and some new parameters have been added:

- The forgetting factor is changed from 0.999 to 0.9995 to decrease the ripple.
- The initial value of  $B_h$  is such that there is one zero in 0.99 and one in 0.98.
- The initial covariances are  $P_{i_{out}} = 10$ ,  $P_{u_{in}} = 0.001$  and  $P_{u_s} = 6 \times 10^{-4}$ .
- $A(q)$  is initialized to the computed values.

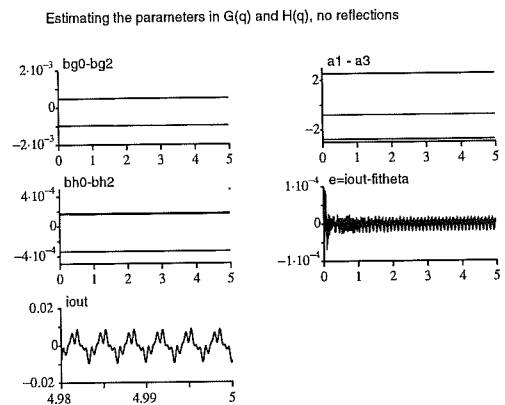


Figure 5.9: The estimated parameters, the error in the estimator, and the output current.  $A$ ,  $B_g$ , and  $B_h$  are estimated. No reflections are added. The time is given in seconds.

If the convergence of the poles and zeroes is studied in figure 5.10, it is obvious that  $B_g$  converges

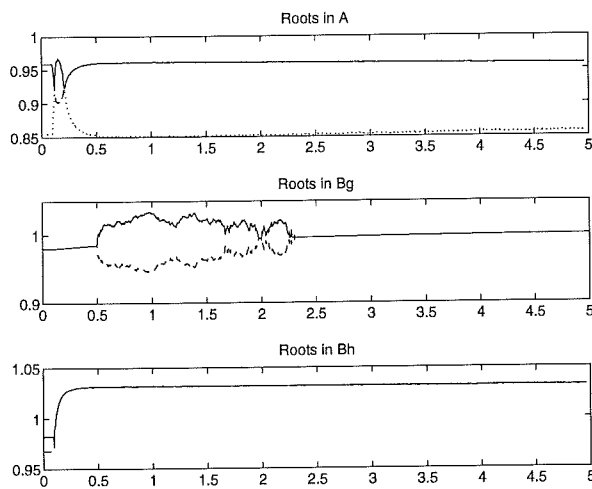


Figure 5.10: The absolute values of poles and zeroes when no reflections are present.

much slower than the other polynomials. A larger initial covariance can, however, not be used. This would increase the ripple between 0.2 and 2.3 seconds, which probably would slow down convergence. A too small initial covariance will of course also give a slow convergence and a proper value has to be found by tests.

### Including the Reflections

Finally, the reflections were included. The result was much of the same as before. The convergence was maybe a little slower in these simulations. In figure 5.11, the parameters are shown, while figure 5.12 contains the absolute values of the poles and zeroes in the transfer functions  $G(q)$ ,  $H(q)$ , and  $F(q)$ . The final value after 5 seconds are given in table 5.1. The initial values of the covariances were

$$\begin{aligned} P_{i_{out}} &= 1000 \\ P_{u_{in}} &= 0.02 \\ P_{u_s} &= 0.001 \end{aligned}$$

and the covariance resetting time was 0.1 seconds. Just as in other simulations in this chapter, the initial values of the parameters were a little different from the computed ones. The polynomial  $B_g$  has a double zero in 0.98,  $B_h$  has one zero in 0.98 and one in 0.97, and finally  $B_f$  has a double zero in 0.99. The polynomial  $A$  is exactly as calculated.

Notice the initial part in figure 5.12. The poles and zeroes have a tendency to move when the covariances are reset. Maybe, the values of the reset covariance (the same values as the initial ones) are too big. Smaller values would, however, slow down the convergence rate as discussed before.

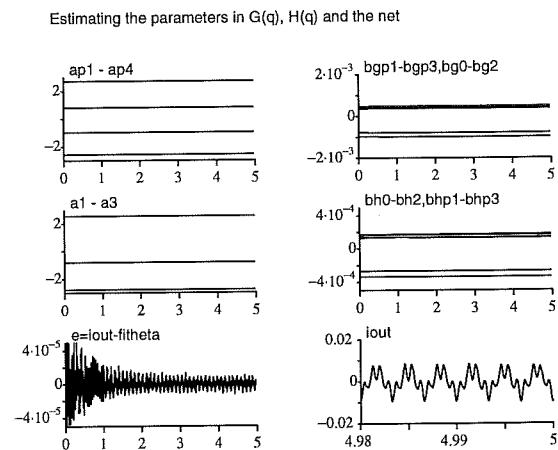


Figure 5.11: The estimated parameters, the error in the estimator, and the output current. The whole system is estimated. The time is given in seconds.

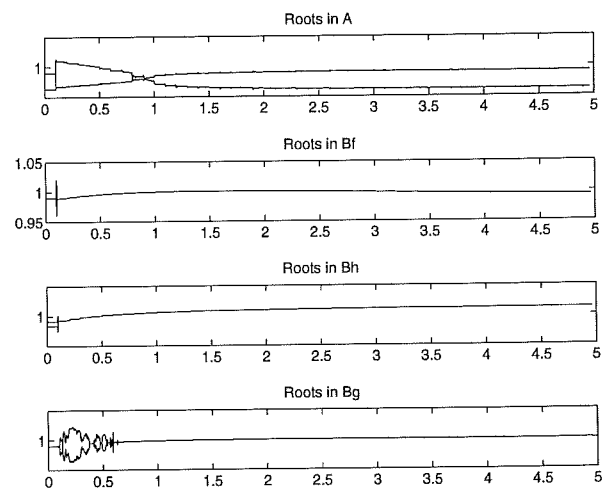


Figure 5.12: The absolute values of the poles and zeroes when the whole system is estimated.

	estimated	calculated
$a_1$	-2.76447	-2.7665
$a_2$	2.55341	2.5541
$a_3$	-0.78719	-0.7864
$b_{g0}$	$4.91841 \times 10^{-4}$	$4.909 \times 10^{-4}$
$b_{g1}$	$-9.83714 \times 10^{-4}$	$-9.819 \times 10^{-4}$
$b_{g2}$	$4.92558 \times 10^{-4}$	$4.910 \times 10^{-4}$
$b_{h0}$	$1.66863 \times 10^{-4}$	$1.734 \times 10^{-4}$
$b_{h1}$	$-3.3830 \times 10^{-4}$	$3.429 \times 10^{-4}$
$b_{h2}$	$1.74951 \times 10^{-4}$	$1.733 \times 10^{-4}$
$b_{f0}$	-0.210076	-0.2126
$b_{f1}$	0.416075	0.4239
$b_{f2}$	-0.20717	-0.2126
$a$	0.80201	0.8

Table 5.1: Estimated parameters after 10 seconds. All transfer functions in the system are estimated.

### 5.3 Conclusions

The simulations in this chapter clearly show that it is possible to estimate the transfer functions of this simplified system. This is done even if a reflection from the transmission net is present. How this method would work in reality is, however, uncertain. A major problem is the reflections from the transmission net. As shown in this chapter, it is important to model the reflections. In reality this can be very difficult, due to a very complex net.

The algorithm used is basically the least-mean-squares (LMS) in its recursive form (recursive least-mean-squares, RLS). In the first simulation, the full covariance matrix was used, while in the later part this matrix was simplified using a scalar. This algorithm is called the *Stochastic approximation algorithm*. In both algorithms, forgetting factors were used. When simplifying the covariance matrix to a scalar, the convergence rate is decreased to the benefit of a reduction in computer load. The RLS-algorithm is, of course, to prefer if the computer load is not too large. To increase the convergence rate when the Stochastic approximation rule is used, stochastic resetting is introduced.

The program used when the last simulation was done is shown in appendix B.

## Chapter 6

# Feedback Control

In this chapter, a feedback concept is developed and slightly tested. This concept uses an RST-controller, which has a pole placement design. In section 6.1, the system is implemented without any estimator; this is to only study the performance of the controller. The transfer functions needed in the design were the ones computed earlier. Finally, in section 6.1.3 the estimator and the controller are running simultaneously, which results in an indirect self-tuning regulator.

### 6.1 Feedback Using Pole Placement Design

When we earlier applied the MIT rule, the controller designed used was only feed-forward. This is a very good concept to annihilate known disturbances. There is one disturbance that can not be measured, though: the reflection from the transmission net  $i_r$ . If the estimator in the previous chapter is used,  $i_r$  can be calculated and feedforward can be used. The drawback is that if the signal from the net contains anything but the modeled reflections, this part will not be annihilated. Therefore, a feedback design is preferred. One problem with using feedback are delays in the measurement equipments. If the signal from the previous period is used, all disturbances having a period different from 1/50 seconds will be undamped. These signals will also create new disturbances throughout the controller, if no precautions are taken. The alternative is to use prediction. This is, however, not implemented in this application.

#### 6.1.1 Pole Placement Design

A standard RST-control is chosen to control this deterministic system. This is shown with the system connected in figure 6.1. To create the controller

(polynomials  $R$ ,  $S$  and  $T$ ), the diophantine equation

$$A_{\text{open}}R + B_{\text{open}}S = B_{\text{open}}^+ A_o A_m$$

is solved. In this report, the performance of the transfer function  $G(q)$  is studied in order to design the controller. This gives  $A_{\text{open}} = A$ ,  $B_{\text{open}} = B_g$ , and  $B_{\text{open}}^+ = B_g^+$ , if the reflections from the transmission net not are considered in the design.  $B_g^+$  is the part of  $B_g$  allowed to be canceled by closed loop poles. In this case  $B_g^+ = 1$ , because the roots of  $B_g$  are close to or outside the unit circle. If these are canceled by the controller, the closed loop would be unstable.

$A_m$  is the denominator of the required closed loop. The order of  $A_m$  have to be greater than the order of  $B_g$ , if the solution of the pole-placement design is to be causal [Åström90, chapter 10]. In this simulations, the order is chosen to 3. Its performance is given by the continuous parameters  $\omega_{m1}$ ,  $\omega_{m2}$ , and  $\zeta_m$  in the polynomial:

$$A_m(s) = (s + \omega_{m1})(s^2 + 2\omega_{m2}\zeta_m s + \omega_{m2})$$

This is sampled in the SIMNON program and the discrete polynomial is then used. The values used are  $\omega_{m1} = 1.3 \times 10^4$  rad/s,  $\omega_{m2} = 100$  rad/s and  $\zeta_m = 0.7$ , which makes a band-pass filter out of the closed loop from  $u_r$  to  $i_{\text{out}}$ . Comments regarding the choice of parameters are given later in the report, for example, together with the results in section 6.1.2.

The polynomial  $A_o$  is an observer polynomial, which in this case (and usually) is chosen to be a little faster than the closed loop. The order of  $A_o$  is chosen to 2, to get a causal solution to the design [Åström90, chapter 10]. This is also given in the continuous time scale as

$$A_o(s) = (s^2 + 2\omega_o\zeta s + \omega_o)$$

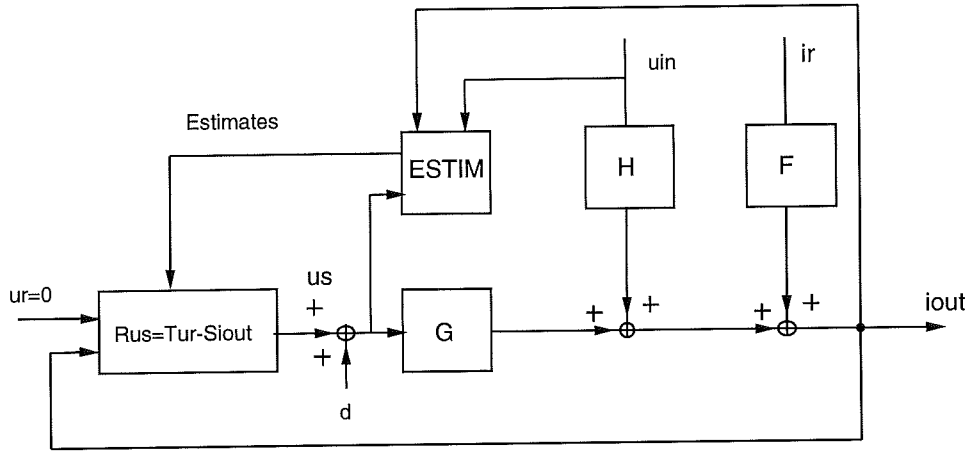


Figure 6.1: The whole system when the estimator and an RST-controller is included.

and then sampled in the program. The values used are  $\omega_o = 2 \times 10^4$  rad/s and  $\zeta = 0.9$ .

The transfer function  $B_g/(A_m A_o)$  is shown in figure 6.2. Notice that this is the function describing how the signal  $u_r$  transfers to  $i_{out}$  and not  $u_{in}$  to  $i_{out}$ . In other words, it changes the controller function  $G(s)$ , but does not improve  $H(s)$ , which is really wanted. To do this, the polynomials  $A_m$  and  $A_o$  are to be chosen differently. This is not done in this report, but is an extension worth trying.

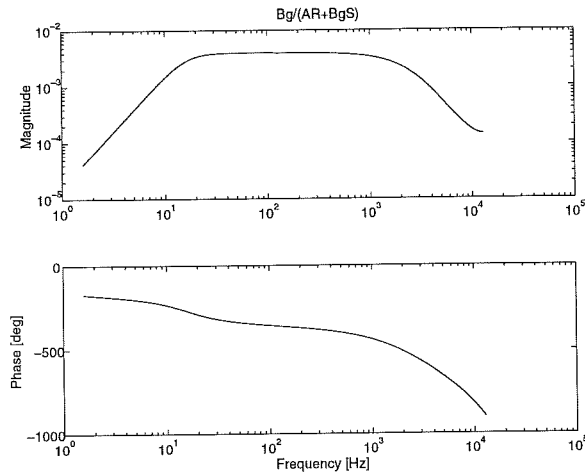


Figure 6.2: The transfer function for the signal  $u_r$  to  $i_{out}$  in closed loop. The frequency is given in Hz.

The  $R$  and  $S$  polynomials are chosen as follows:

$$\begin{aligned} R &= q^2 + qr_1 + r_2 \\ S &= q^2 s_0 + qs_1 + s_2 \end{aligned}$$

They have equal degrees, which gives no time delays in the controller. The diophantine equation is now solved. When running the estimator, the equation has to be solved for each sample, each time with a new  $A$  and  $B_g$  as input.

To prevent windup in the controller when the control signal reaches its limits, anti-reset windup was installed. This is done by changing the controller to

$$A_o v_s = T u_r - S i_{out} + (A_o - R) u_s$$

where

$$u_s = \begin{cases} u_{limit} & v_s \geq u_{limit} \\ v_s & -u_{limit} < v_s < u_{limit} \\ -u_{limit} & v_s \leq -u_{limit} \end{cases}$$

The use of this anti-reset windup is not shown in these simulations, but is definitely needed in reality.

### 6.1.2 The Result Without Estimator

To be able to study the behaviour of the controller, no estimation could be present. The transfer functions used in the controller are the ones calculated in chapter 3.

The signal  $u_{in}$  is exactly the same as before, that is, the sum of sinusoidals.

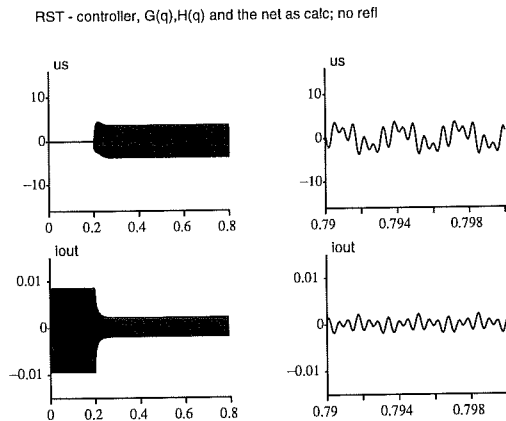


Figure 6.3: The RST-controller when no reflections are added. The controller is started at  $t = 0.2$  seconds. The time axes are given in seconds. Note the different time scales.

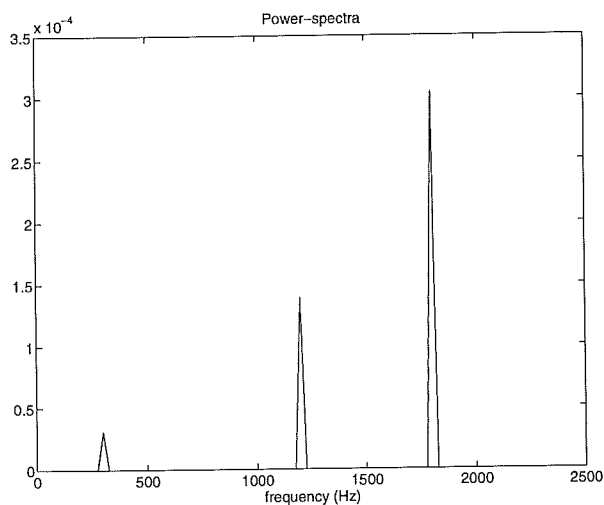


Figure 6.4: The power spectrum of the output current  $i_{out}$  with the RST-controller implemented.

If the controller is started without the reflections included, the result is as shown in figures 6.3 and 6.4. The power spectrum is to be compared to figure 4.2. The conclusion is that the controller annihilated the signals with a factor 220 at 300 Hz down to 3 at 1800 Hz. To understand how the controller changes the transfer function between  $u_{in}$  and  $i_{out}$ , see figure 6.5, can be studied. It can clearly be seen that frequencies between 20 and 2000 Hz are damped in close loop and that the performance of the controller is satisfying up to about 1500 Hz.

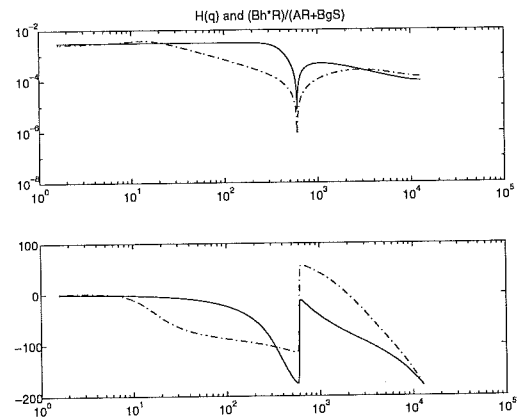


Figure 6.5: The transfer function between  $u_{in}$  and  $i_{out}$ , with ('- -') and without ('—') controller. The frequency axes are given in Hz.

The model of the net was connected to the system in the next test. The damping factor  $a$  in the model of the net is, however, 0.7. This is because the former value of 0.8 gave an unstable system. Why this happened, is not clear. It might depend on the fact that the reflections not were considered in the pole placement design. Another theory is that the model of the net maybe is too simple to work without trouble. Tests done with only the system and the model of the transmission net added showed that instability occurred even when  $a$  was less than one. Instability is impossible in reality, because  $a < 1$  naturally, which shows that there is something wrong with the model. Lack of time, however, made it impossible to examine the model further, but if more work is to be done in this field, this should be tested.

The result became as shown in figures 6.6 and 6.7, when the transmission net was connected. The power spectrum is to be compared to figure 4.4 in chapter 4, even if the factor  $a$  in this figure is 0.8. The difference in power spectra in the two cases is very small and not noticeable at the scales used here. The conclusion is that the result is much of the same as without reflections.

The parameters to design  $A_m$  and  $A_o$  used in these simulations are the same as the ones given before. Maybe, these could given other values to get an even better controller. No big efforts have been made to optimize the controller.

One parameter that seems to have a strange value is  $\omega_{m2}$ . In the simulation, this is chosen to 100 rad/s, that is, about 16 Hz. This is rather a low frequency,

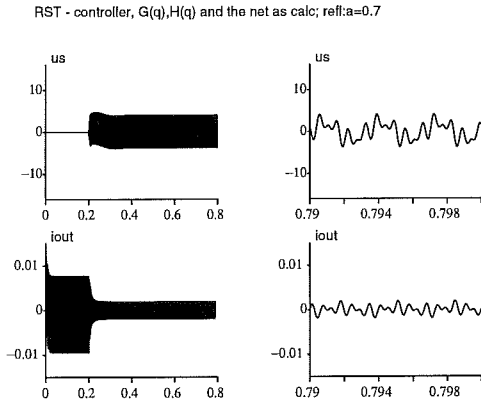


Figure 6.6: Control signal and output current if the RST-controller is used with reflections added. The controller was started at  $t = 0.2$  seconds. The time axes are given in seconds. Note the different time scales.

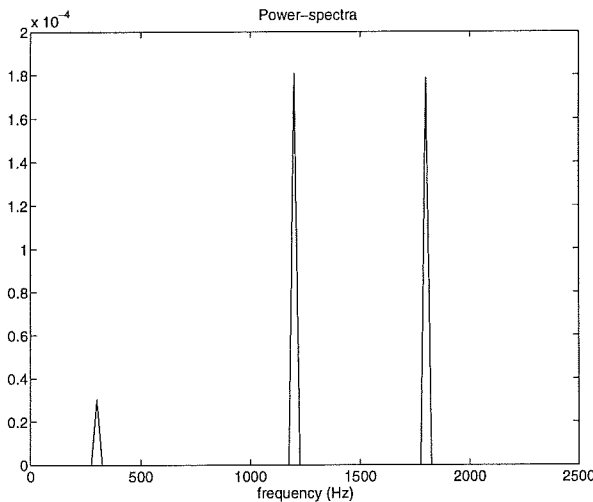


Figure 6.7: The power spectrum of the output current with reflections added.

considering that only disturbances in the range 300–5000 Hz are to be removed. Tests showed, however, that the system became unstable if a larger value of  $\omega_{m2}$  were used, without changing other parameters. Why this happened is not examined in this report.

At the end of chapter 4, there was a problem if low frequencies entered the controller: this would give the control signal a bias not noticeable in the output current. Finally, the control signal reached its limits and then  $i_{out}$  became larger. The same problem arose in these simulations. This can, for example,

be seen in figure 6.8, where  $u_s$  has a low frequency part that is not noticed in  $i_{out}$ . In this simulation, reflections are added (without reflections, no problem was noticed). This is explained by studying how the signals from  $u_{in}$  are transferred to  $u_s$  in closed loop with and without the net connected (see figure 6.9). When the transmission net is connected, the low frequencies are amplified a factor 10 more than without the net. If there somehow are any low frequencies present in  $u_{in}$ , these pose a bigger problem with the net connected.

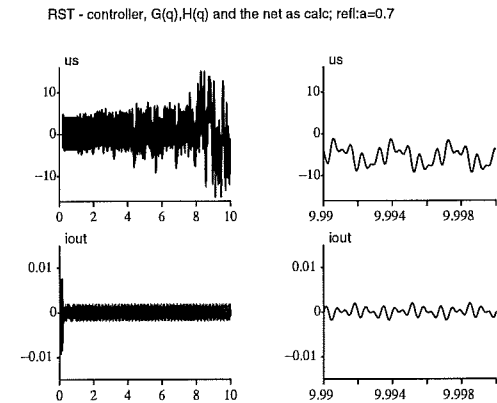


Figure 6.8: The RST-controller run for a longer time. The signal  $u_{in}$  is created in a continuous system. The controller was started at  $t = 0.2$  seconds. Time is given in seconds.

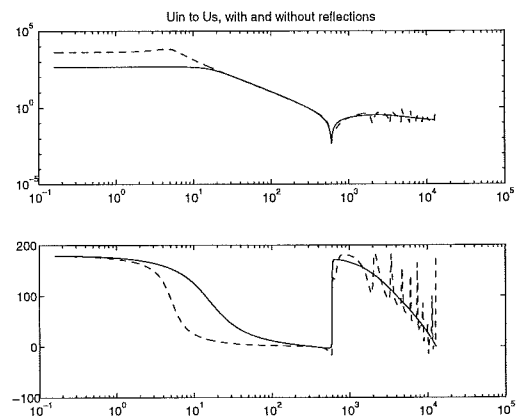


Figure 6.9: The transfer function that describes how the signal  $u_{in}$  are transferred to  $u_s$ , without reflections (“—”) and with reflections added (“- -”). The frequency is given in Hz.



Where do the low frequencies come from, then? In reality, it is natural to have them, because the filter to annihilate them is not ideal. This is, however, not the case in the simulations. The only added signals are the four sinusoids with frequencies 300, 600, 1200, and 1800 Hz. The answer to the question is that it is errors in the computations that generates the low frequency signals. If the system that created the signal  $u_{in}$  is implemented as a discrete system instead of a continuous, like before, the problem would disappear. This is probably a reminder of the numerical problems or synchronization problem that may occur in programs, when a high sample rate is used in a rather slow system.

As mentioned in chapter 4, some kind of watchdog is needed to prevent the control signal from getting biased. How this is to be implemented is, however, not examined here.

### 6.1.3 The Result With the Estimator Included

To introduce an adaptive concept in the RST-controller, the estimator in chapter 5 and the controller were run simultaneously. This concept is usually named 'indirect self-tuning regulator'. The estimator estimates all transfer functions in the system, but this particular regulator only uses  $G(q)$ . In each sample, the new values of the estimated parameters are used in the design of the controller. The initial values of the parameters in the estimator are all the calculated ones.

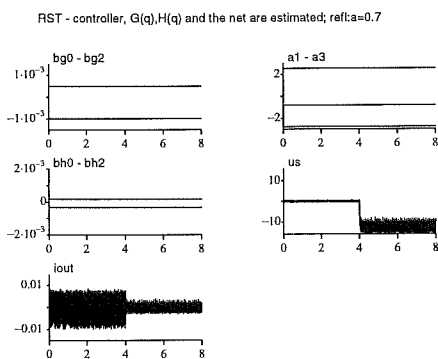


Figure 6.10: The RST-controller and the estimator run simultaneously. The controller was started at  $t = 4$  seconds. Time is given in seconds.

The first simulation used exactly the same controller as in section 6.1.2. The result is shown in figures 6.10

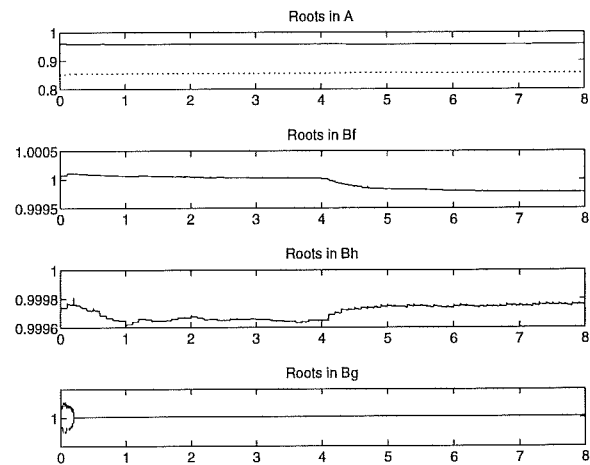


Figure 6.11: The absolute values of the estimated parameters. Time is given in seconds.

and 6.11. If the latter figure is studied, the conclusion can be drawn that the estimator still worked. The controller, however, did not work satisfactorily. This can be seen in figure 6.10. The control signal got a bias at once after the start of the controller. This behaviour is, in this case, related to an instability in the controller. The zeroes of the denominator,  $R(q)$ , are computed to be 1.011 and 0.229. When the calculated values of  $G(q)$  were used, the controller was stable but even a little difference between the computed and estimated values made it unstable. To stabilize the controller, different  $A_m(q)$  and  $A_o(q)$  are to be chosen. This is done in the last simulation. The new values are  $\omega_{m_1} = 300$ ,  $\omega_{m_2} = 9000$ , and  $\omega_o = 13000$  rad/s; in other words, a narrowing of the range of frequencies in which to control. Now the controller was stable, which can be seen in figure 6.12; unfortunately, the controller had a worse performance. This is obvious on examining the power spectrum of the output current in figure 6.13. The signal with the frequency of 1200 Hz was almost not damped at all compared to figure 4.4, where no controller was implemented. The reason behind this behaviour is not examined and a lot more work needs to be done.

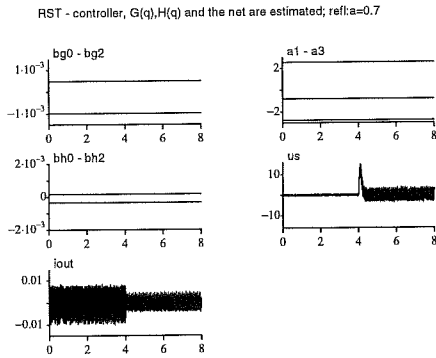


Figure 6.12: The controller and the estimator run simultaneously. The new  $A_m(q)$  and  $A_o(q)$  were used and the controller was started at  $t = 4$  seconds. Time is given in seconds.

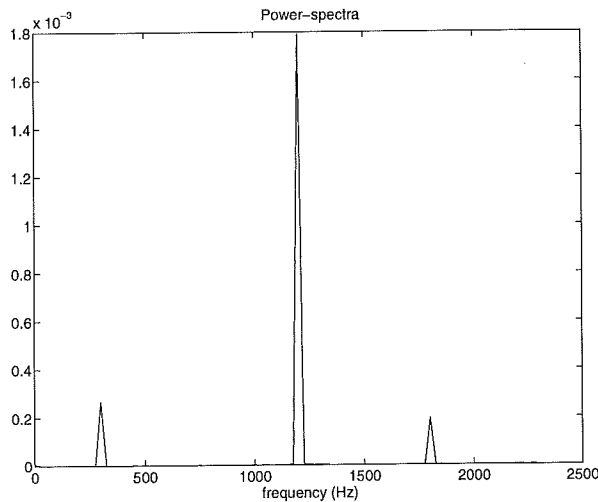


Figure 6.13: The power spectrum of the output current with the new controller.

## 6.2 Conclusions

In this chapter a feedback control concept is partially tested. The controller is designed (using  $A_m(q)$  and  $A_o(q)$ ) to control in a wide range of frequencies. This worked rather well when the calculated values of the parameters in  $G(q)$  were used. With present bandwidth to control within, the annihilation factor of the disturbances are between 220 for 300 Hz down to 3 for 1800 Hz. If more tests had been done with other choices of bandwidth, better performance could probably have been obtained.

However, when the estimated parameters in  $G(q)$

were used in section 6.1.3, the controller was unstable. The range of frequencies to control within had to be decreased to make the controller stable again. A lot more work is necessary before a proper concept can be obtained.

An important remark is the need of different safety nets, if this concept is to be useful. For example, is the need of resetting in the controller to prevent  $u_s$  to get biased obvious. Otherwise, the control signal reaches its limit and the output current grows. When the estimation and control is done simultaneously, there is also a need for a safety net. The variation in the estimated parameters may result in an unstable controller, if not precaution is taken.

The final results of the simulations in this chapter show the difficulties to design a controller that work properly in a wide range of frequencies. Perhaps, other concepts using several controllers, should be considered. One special concept is to have one controller for each frequency of disturbance.

The program used in the last simulation is shown in appendix B.

## Chapter 7

# Conclusions

This work has resulted in a couple of controllers and estimators, summarized in section 7.1. During the work on this project, several ideas arose that not were tested. Two of these ideas, not mentioned before in the report, are introduced in section 7.2. Finally, section 7.3 has comments about the tools used in the work.

### 7.1 Conclusion

The development of a well-performing adaptive controller is a large project. The results in this report are a step in the direction of an adaptive concept. The simulations show that the possibility exists. The fact that a simplified system is used is no drawback to the method, since the same problems arise as in the full system.

#### The Feedforward Concept

In principle, two different control concepts were tested in this report. The first was a feedforward controller with adaptive parameters, in which the MIT rule was employed to adjust the parameters. The error to minimize was the output current from the plant and the measured current was assumed to be filtered. In other words, only the signals to be removed entered the control system. The results of using this method were very good in this simplified system. If the parameters in the estimator were accurate, the disturbances were annihilated by a factor of 1500 or more. This positive result was obtained even when the model of the transmission net was connected. Notice that the transmission line do not have to be estimated in this concept, something which makes this application of the MIT rule very interesting. In many other concepts, the dynamics of the net have to be regarded. The drawback to this method is that the disturbances have to be

measured. In reality, this is not possible, though. However, an approximate value can be obtained, if one or more signals are measured. More investigations are needed in this area.

#### The Feedback Concept

The second concept is a feedback controller using pole placement design. The denominator of the closed system ( $A_m$ ) and the observer polynomial ( $A_o$ ) are chosen and determines the properties of the controller. In these tests,  $A_m$  and  $A_o$  were chosen to create a controller that work in a wide range of frequencies. This concept is only partially tested. The result when the calculated values of  $A(q)$  and  $B_g(q)$  were used, was an annihilation of the disturbances a factor 220 for 300 Hz down to 3 for 1800 Hz. The result was not as good as for the feedforward concept, but can be seen as a complement to this. The design of the controller made in this application, was not the optimal one and better performance is perhaps possible to obtain. One test was then made by running the controller and the estimator simultaneously. Each sample the estimated parameters were used in the design of the controller. The result in this single test was not satisfying. With the same control bandwidth as before, the controller was unstable. A decrease in the bandwidth was necessary to obtain stability. The decrease in bandwidth gave, however, less annihilation of the disturbances and is no solution to the problem. This performance show the difficulties when a single controller is used in a wide range of frequencies. Other concepts using several controllers is probably preferable.

#### The Estimation

This report also treats the estimation of all the transfer functions in the system. The algorithms used are the recursive least-mean-squares algorithm

and the stochastic approximation algorithm. The results of these simulations show that it is possible to estimate  $G(q)$ ,  $H(q)$ ,  $F(q)$ , and the damping factor  $a$ , in this simplified model, provided the signals are persistently exciting. One major problem of implementing this estimator in reality are reflections from the transmission net. To prevent bias in the other parameters, the dynamics of the net have to be modeled. In this report, the model of the net was simplified and the number of parameters to estimate due to reflections was small. In reality, however, the transmission net is very complex and a higher-order model has to be used. Whether this is possible to estimate, further tests have to show.

## 7.2 Extensions

During this project, many ideas arose that were not tested. In this section, two of these are mentioned.

### The Use of MIT Rule In a Feedback Concept

This is an idea not mentioned before in this report. In the same way as the feedforward rule, the MIT rule can be used in a feedback concept to get an adaptive controller combining the good properties of the feedback controller with the MIT rule. In other words, this is a system where the transmission net probably not have to be modeled, yet all the advantages of feedback may be used. To clearly show the idea, a simple example is given. An ordinary feedback controller is shown in figure 7.1.

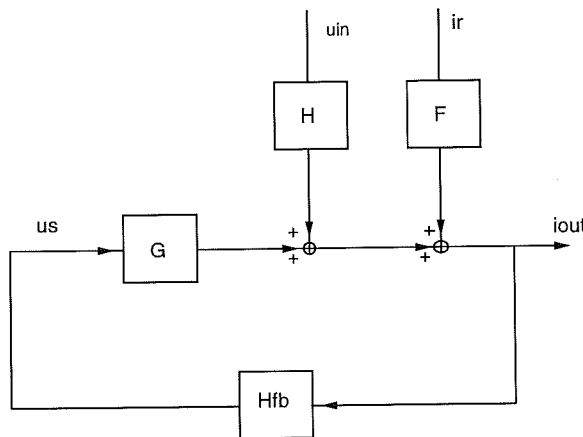


Figure 7.1: An ordinary feedback controller.

Proportional feedback is used, that is,

$$H_{fb} = -K$$

The expression of the output current in closed loop will be

$$i_{out} = \frac{B_g}{A + KB_g} u_{in}$$

if the  $i_r = 0$ . Differentiating this yields

$$\frac{\partial i_{out}}{\partial K} = -\frac{B_g^2}{(A + KB_g)^2}$$

Now the MIT rule can be used, exactly as in chapter 4. Other transfer functions  $H_{fb}$  can of course be chosen. Computations are done in the same way. Problems may occur when  $i_r \neq 0$ . It is not examined if the transmission net have to be modeled or not. This is a subject for further investigations.

### Feedforward From $i_r$

The model of the transmission net was estimated in chapter 5. This makes it possible to use this model to calculate the part from the net in the output current. This can then be used in a feedforward concept. The principle is shown in figure 7.2. The ideal transfer function  $H_{ff}$  is

$$H_{ff} = -\frac{B_f}{B_g}$$

where  $B_f$  and  $B_g$  are the numerators in the transfer functions  $G(q)$  and  $F(q)$ .

The result of a feedforward like this depends mainly on the quality of the model of the transmission net. Problems will probably occur if this is to be used in reality, where the net is very complex.

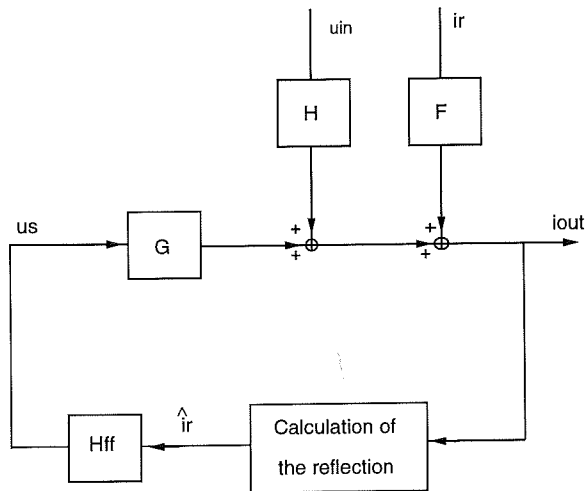


Figure 7.2: The part in  $i_{out}$  due to reflections in the transmission net is calculated. Then a feedforward controller is used.

### 7.3 Tools

There are some aspects of the working tools that demand comments. The tools used are the SIMNON simulation program and the MATLAB mathematical program.

#### SIMNON

This program has both positive and negative aspects.

First, the negative ones. The features are almost none. Missing functions needed from SIMNON during this particular work were for example vectors and matrices. Also, functions to calculate poles and zeroes had been to preferable to have. Another problem during the simulations was the function  $DELAY(a, t)$ . This is used to delay the signal  $a$  the time  $t$ . To do this, some kind of interpolation is used. Unfortunately, the accuracy of the interpolated signal was not always good enough. Other numerical problem also occurred, but these are general for most simulators. It is probably a result of a rather slow system sampled with a rather high frequency.

The positive aspects of SIMNON is its simplicity. It is rather easy to learn and to use.

#### MATLAB

This program was used to calculate and draw the transfer functions. It was also used to create the power spectra in chapter 4 and 6. MATLAB worked very well in most cases.

A positive aspect of MATLAB is the possibility to create functions. Many functions related to control design have already been written at the Department of Automatic Control, Lund Institute of Technology. These were often used and that made the work easier.

**References**

- [Zhang] Wenyan Zhang, Gunnar Asplund, Anders Åberg, Uno Jonsson, and Ove Lööf. *Active D.C. Filter for HVDC System - A Test Installation In The Konti-Skan D.C. Link at Lindome Converter Station*. IEEE Transaction on Power Delivery, Vol. 4, No. 3, July 1993.
- [Dickmänder] D.L. Dickmänder, K.J. Peterson. *Analysis of D.C. Harmonics Using the Three-Pulse Model For the Intermountain Power Project HVDC Transmission*. IEEE Transactions on Power Delivery, Vol. 2, No. 2, April 1989.
- [Johnson] Walter C. Johnson. *Transmission Lines and Networks*. International student edition.
- [Shore] N.L. Shore, G. Andersson, A.P. Canelhas, and G. Asplund. *A Three-Pulse Model of D.C. Side Harmonic Flow in HVDC Systems*. IEEE Transactions on Power Delivery, Vol. 4, No. 3, July 1989.
- [Åström89] Karl Johan Åström and Björn Wittenmark. *Adaptive Control*. Addison-Wesley Publishing Company, 1989.
- [Åström90] Karl Johan Åström and Björn Wittenmark. *Computer Controlled Systems: Theory and Design*. Prentice-Hall International Editions, 1990. Second edition..
- [Claesson] I. Claesson, P. Eriksson, B. Mandersson, and G. Salomonsson. *Analoga kretsar och signaler*. Studentlitteratur, 1989.

## Appendix A

### Appendix

#### A.1 The model of the net

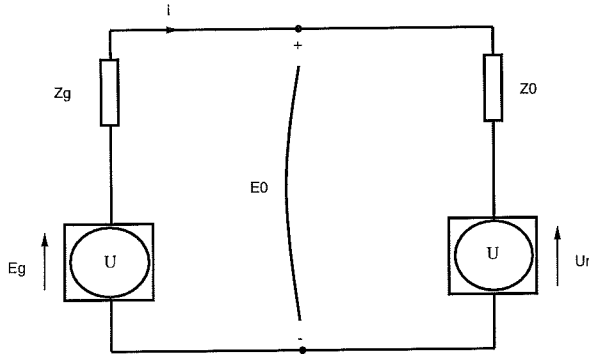


Figure A.1: Two ordinary two-terminal networks connected to each other.

For two ordinary two-terminal networks connected to each other as in figure A.1, following equations can be formulated.

$$E_0 = \frac{Z_0}{Z_g + Z_0} \left( E_g + \frac{Z_g}{Z_0} U_r \right) \quad (\text{A.1})$$

$$E_0 = E_g - i Z_g \quad (\text{A.2})$$

$$E_0 = U_r + i Z_0 \quad (\text{A.3})$$

Compare these equations with the true equations for a transmission line (see figure A.2). The equation A.4 and the basic theory is found in [Johnson].

$$E_0 = \frac{Z_0}{Z_0 + Z_g} \frac{E_g}{1 - k_r k_g e^{-2\gamma l}} (1 + k_r e^{-2\gamma l}) \quad (\text{A.4})$$

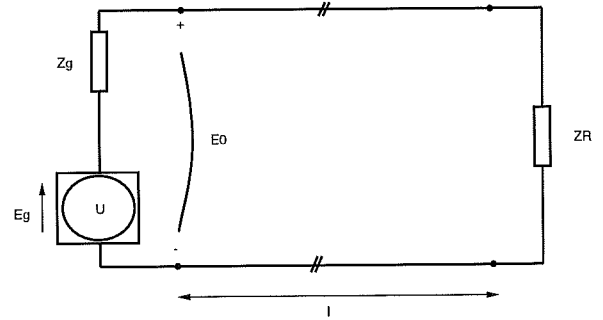


Figure A.2: A transmission line.

where

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)}$$

$$k_r = \frac{Z_R - Z_0}{Z_R + Z_0}$$

$$k_g = \frac{Z_g - Z_0}{Z_g + Z_0}$$

The quantities  $Z_0$  and  $\gamma$  are respectively, the characteristic impedance and the propagation constant of the line.

Now the factor  $e^{-2\gamma l}$  is approximated with a damping factor ( $b$ ) and a delay ( $\Delta$ ), i.e

$$e^{-2\gamma l} = bq^{-\Delta}$$

Then the equation A.4 is rewritten in several steps to

$$E_0 = \frac{Z_0}{Z_0 + Z_g} (E_g + E_g k_r b q^{-\Delta} +$$

$$E_0 k_r k_g b q^{-\Delta} \frac{Z_0 + Z_g}{Z_0}$$

But

$$k_g \frac{Z_0 + Z_g}{Z_0} = \frac{Z_g - Z_0}{Z_0}$$

gives that

$$E_0 = \frac{Z_0}{Z_g + Z_0} \left( E_g + E_g k_r b q^{-\Delta} + E_0 k_r b q^{-\Delta} \frac{Z_g - Z_0}{Z_0} \right) \quad (\text{A.5})$$

Now we want to have the two systems identically. This means that equation A.1 is to be the same as equation A.5. The result is

$$U_r = k_r b q^{-\Delta} \left( \frac{Z_0}{Z_g} E_g + E_0 \frac{Z_g - Z_0}{Z_g} \right)$$

Equation A.2 gives that

$$U_r = k_r b q^{-\Delta} (E_0 + Z_0 i)$$

and if we use equation A.3 we will get

$$U_r = k_r b q^{-\Delta} (U_r + 2Z_0 i)$$

i.e., if we rewrite it

$$U_r = \frac{k_r b q^{-\Delta}}{1 - k_r b q^{-\Delta}} 2Z_0 i$$

If we want to express the line with a current source parallel with the load  $Z_0$  instead of a voltage source serial with the load, it is no problem. The current will in this case be

$$i_r = \frac{k_r b q^{-\Delta}}{1 - k_r b q^{-\Delta}} 2i \quad (\text{A.6})$$

We can see that  $k_r$  is factor close to  $-1$ , because  $Z_0 \gg Z_R$ . This means that equation A.6 can be written as

$$i_r = \frac{-a q^{-\Delta}}{1 + a q^{-\Delta}} 2i \quad (\text{A.7})$$

and if we rewrite it one last time it will be like

$$i_r(t) = -a (i_r(t - \Delta) + 2i(t - \Delta)) \quad (\text{A.8})$$

where  $0 < a < |k_r| < 1$  because  $0 < b < 1$



## A.2 Extend the Model of the Net

When creating the model of the transmission net in appendix A.1, some assumptions were made. The most important were made when the factor  $a$  and delay  $\Delta$  were assumed to be independent of the frequency. If the factor  $\gamma$  in  $e^{-2l\gamma}$  is rewritten as

$$\gamma = \alpha + j\beta$$

where

$$\alpha = \sqrt{RG - \omega^2 LC}$$

and

$$\beta = \sqrt{\omega LG + \omega RC}$$

the factor  $e^{-2l\gamma}$  may be written as

$$e^{-2l\gamma} = e^{-2l\alpha} e^{-j2l\beta}$$

It is then obvious that  $b = e^{-2l\alpha}$  and  $\Delta = -2l\beta$  are functions of the frequency. The relation between  $b$  and  $a$  is

$$a = k_r b$$

where

$$k_r = \frac{Z_g - Z_0}{Z_g + Z_0}$$

The variations of  $b$  and the impedances  $Z_g$  and  $Z_0$  with frequency, make the factor  $a$  dependent on the frequency as well. Consequently, to get a more accurate model of the transmission net, the model of the transmission net has to be extended with  $a$  and  $\Delta$  that are functions of the frequency. However, this is not examined in this report.

### A.3 The Extended Least-Squares Algorithm

The principle of the extended least square algorithm (ELS) is described in this section.

Consider the model

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (\text{A.9})$$

where  $e(t)$  is white Gaussian noise. The problem with estimation of this model is that you can't measure the noise  $e(t)$ . To create an regression model the approximation

$$e(t) = \varepsilon(t) = y(t) - \varphi^T \hat{\Theta}(t-1)$$

where

$$\Theta(t) = [a_1 \dots a_n \quad b_1 \dots b_n \quad c_1 \dots c_n]$$

and

$$\varphi^T(t) = [-y(t) \dots -y(t-n+1) \quad u(t) \dots u(t-n+1) \quad \varepsilon(t) \dots \varepsilon(t-n+1)]$$

must be done. With this approximation, the ordinary least square algorithm can be applied. For example the simplified (and slower) algorithm can be used. This is given by

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + P(t)\varphi(t-1)\varepsilon(t)$$

$$P^{-1}(t) = P^{-1}(t-1) + \varphi(t-1)\varphi^T(t-1)$$

If the model is like

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

instead of the one in equation A.9 the parameters for this model can be calculated in the same way.

## Appendix B

# Appendix

In this chapter is the SIMNON code used in this report shown. The first section contains programs that were common for all simulations. Then the ones specific for the MIT simulation in chapter 4 are shown. These are shown in section B.2. There are also programs that only are common in the simulations done in the chapters 5 and 6. These are shown in section B.3. Finally the programs used specific for chapter 5 and 6 are shown in sections B.4 and B.5, respectively.

### B.1 Programs Common For All Simulations

The SIMNON code identical in all simulations is shown in this chapter. The modules are 'noise', 'refl' and 'sys'.

```

CONTINUOUS SYSTEM sys
"
" A model of a HVDC-station
" Simulate the system with the inputs Uin,Us and i3.
" The output is iout.
" To prevent numerical problems in the integration, the
" states have to be approximately equal size. Therefore
" is the factor 'intfact' used.
"
"
STATE iL1 iL2 Uc
DER diL1 diL2 dUc

INPUT Uin Us i3          " Inputs
OUTPUT iout              " Output

iout=iL1-iL2

diL1=-(RL/L1)*iL1+RL*iL2/L1+Uin/L1-RL*i3/L1
diL2=(RL/L2)*iL1-(RL+R)*iL2/L2-Uc/(intfact*L2)-Us/L2+RL*i3/L2
dUc=intfact*iL2/C

R:1
RL:320
L1:0.2000                " Parameters that give a resonance
L2:70.36E-3              " at 600 Hz between Uin and iout.

```

C:1E-6  
intfact:3E-3

END

---

DISCRETE SYSTEM refl

"  
" Simulates the reflection from the line  
" The delay used is 20 samples.  
"  
" Parameter to vary: rfact  
"  
"

STATE ir1 ir2 ir3 ir4 ir5 ir6 ir7 ir8 ir9 ir10  
NEW nir1 nir2 nir3 nir4 nir5 nir6 nir7 nir8 nir9 nir10

STATE ir11 ir12 ir13 ir14 ir15 ir16 ir17 ir18 ir19 ir20 ir21  
NEW nir11 nir12 nir13 nir14 nir15 nir16 nir17 nir18 nir19 nir20 nir21

STATE i1 i2 i3 i4 i5 i6 i7 i8 i9 i10  
NEW ni1 ni2 ni3 ni4 ni5 ni6 ni7 ni8 ni9 ni10

STATE i11 i12 i13 i14 i15 i16 i17 i18 i19 i20  
NEW ni11 ni12 ni13 ni14 ni15 ni16 ni17 ni18 ni19 ni20

TIME t " Time  
TSAMP ts

INPUT iout " Input  
OUTPUT irefl " Output

ioutd=i20  
irefld=ir20

irefl=-rfactor\*(2\*ioutd+irefld) " Calculations

nir1=irefl  
nir2=ir1  
nir3=ir2  
nir4=ir3  
nir5=ir4  
nir6=ir5  
nir7=ir6  
nir8=ir7  
nir9=ir8  
nir10=ir9  
nir11=ir10 " Updating the states  
nir12=ir11 " of reflected current.  
nir13=ir12  
nir14=ir13  
nir15=ir14  
nir16=ir15  
nir17=ir16  
nir18=ir17  
nir19=ir18

```

nir20=ir19
nir21=ir20

ni1=iout
ni2=i1
ni3=i2
ni4=i3
ni5=i4
ni6=i5
ni7=i6
ni8=i7
ni9=i8
ni10=i9           " Updating the states
ni11=i10          " of output current
ni12=i11
ni13=i12
ni14=i13
ni15=i14
ni16=i15
ni17=i16
ni18=i17
ni19=i18
ni20=i19

ts=t+h           " Update the time

h:39.0625000E-6   " Parameters
rfactor:0.8

END

```

---

```

DISCRETE SYSTEM noise
"
" Creates white Gaussian noise with mean value "mean" and
" the deviation "stdev"
"
"

TIME t           " Time
TSAMP ts

OUTPUT d         " Output

d=stdev*NORM(t)+mean  " Calculations

ts=t+h          " Update the time

h:39.0625000E-6   " Parameters
mean:0
stdev:0

END

```

## B.2 MIT Program

The SIMNON program used in chapter 4, when the MIT rule was used.

```

MACRO hvdc1
"
" Feedforward controller using MIT estimator.
"
"
"

SYST sys signal fforward festim refl2 noise connect

LET stoptime=3
,stime=2.99
,storet=2.8           " Declaration of constants
,steptime=0.01       " in this macro
,hcent=39.0625E-6

PAR ulim:15
,h[fforward]:hcent
,h[festim]:hcent      " Sets the sample rate
,h[refl]:hcent
,h[noise]:hcent
,A0[signal]:2
,A1[signal]:5         " The amplitudes of the
,A2[signal]:5         " simulated disturbances
,A3[signal]:5
,step[signal]:0       " Can have a step in the
,sept[signal]:steptime " input signal
,f0[signal]:6
,f1[signal]:12        " The frequency of the
,f2[signal]:24        " disturbances
,f3[signal]:36
,styr[fforward]:1     " Starts the controller
,gammak[festim]:0.2   " Estimating gains
,gammaaf[festim]:0.01
,gammaap[festim]:0.05
,alfak[festim]:0.1
,alfaa[festim]:0.1    " To prevent division by zero
,alfap[festim]:0.1
,betak[festim]:1E-3
,betaaf[festim]:5E-4 " Limits the step in estimates
,betap[festim]:2E-4
,bg0[festim]:0.4909E-3
,bg1[festim]:-0.9819E-3 " Initial value of Bg
,bg2[festim]:0.4910E-3
,rfactor[refl]:0.8    " Reflection-factor
,stdev[noise]:0       " Deviation of the noise

INIT k1[festim]:-0.3532
,af1[festim]:1.977
,p1[festim]:0.99
,iL1[sys]:-1.2176E-2

```

```

,iL2[sys]:-1.20608E-2          " Initial values
,Uc[sys]:-6.00927E-3
,Uin1[fforward]:-5.95631
,Uin2[fforward]:-9.35805
,Us1[fforward]:0.89354
,Us2[fforward]:1.82396

STORE Uin[sys] iout[sys] Us[fforward] k[festim] af[festim] p[festim]
SPLIT 3 2
AREA 1 1
AXES H stime stoptime V -20 20
TEXT 'Uin'
PLOT Uin[sys]
SIMU 0 stoptime /plotlist hcent " Simulates and plots
AREA 1 2
AXES H stime stoptime V -16 16
SHOW Us[fforward] /plotlist
TEXT 'Us'
AREA 2 1
AXES H stime stoptime V -2E-3 2E-3
SHOW iout[sys] /plotlist
TEXT 'iout'
AREA 2 2
AXES H 0 stoptime V -0.4 -0.3
SHOW k[festim] /plotlist
TEXT 'Estimated parameter k'
AREA 3 1
AXES H 0 stoptime V 1.975 1.985
SHOW af[festim] /plotlist
TEXT 'Estimated parameter af'
AREA 3 2
AXES H 0 stoptime V 0.9 1.1
SHOW p[festim] /plotlist
TEXT 'Estimated parameter p'
MARK A 2 15
MARK "feedforward with reflections, delta=20, a=0.8, estimates 3 parameters.
EXPORT plotlist < plotlist 1000 storet /0 " Transforms the file

END

```

---

```

DISCRETE SYSTEM festim
"
" A estimator to estimate k,af and p in the feedforward-controller.
" Uses the modified MIT-rule(normalized and saturation function).
" When G(q) is needed in the calculations the analytical values
" are used.
"
" Parameters: gammak,gammaaf,gammap,betak,betaaf,betap,alfak,alfaaf
" and alfap
"
"
STATE ip11 ip12 ip13 i11 i12 i13 ip21 ip22 ip23 i21 i22 i23
NEW nip11 nip12 nip13 ni11 ni12 ni13 nip21 nip22 nip23 ni21 ni22 ni23

STATE ip31 ip32 ip33 i31 i32 i33 k1 af1 p1 Uin1 Uin2 Uin3
NEW nip31 nip32 nip33 ni31 ni32 ni33 nk1 naf1 np1 nUin1 nUin2 nUin3

```

```

INPUT Uin iout          " Inputs
OUTPUT k af p          " Outputs

TIME t                " Time
TSAMP ts

p2=p*p

                                " Estimates k:

ip10=Uin-af*Uin1+Uin2+2*p*ip11-p2*ip12
i10=bg0*ip11+bg1*ip12+bg2*ip13-a1*i11-a2*i12-a3*i13
Ep1=(iout*i10)/(alfak+i10*i10)
E1=IF Ep1>betak THEN betak ELSE (IF Ep1<-betak THEN -betak ELSE Ep1)
k=-gammak*E1+k1

                                " Estimates af:

ip20=-k*Uin1+2*p*ip21-p2*ip22
i20=bg0*ip21+bg1*ip22+bg2*ip23-a1*i21-a2*i22-a3*i23
Ep2=(iout*i20)/(alfaa+i20*i20)
E2=IF Ep2>betaaf THEN betaaf ELSE (IF Ep2<-betaaf THEN -betaaf ELSE Ep2)
af=-gammaaf*E2+af1

                                " Estimates p:

ip30=2*k*(Uin1-af*Uin2+Uin3)+3*p*ip31-3*p2*ip32+p*p2*ip33
i30=bg0*ip31+bg1*ip32+bg2*ip33-a1*i31-a2*i32-a3*i33
Ep3=(iout*i30)/(alfap+i30*i30)
E3=IF Ep3>betap THEN betap ELSE (IF Ep3<-betap THEN -betap ELSE Ep3)
p=-gammap*E3+p1

nip11=ip10
nip12=ip11
nip13=ip12
ni11=i10
ni12=i11
ni13=i12
nip21=ip20
nip22=ip21
nip23=ip22                                "Update the states
ni21=i20
ni22=i21
ni23=i22
nip31=ip30
nip32=ip31
nip33=ip32
ni31=i30
ni32=i31
ni33=i32

nk1=k
naf1=af
np1=p

```



```

nUin1=Uin
nUin2=Uin1
nUin3=Uin2

ts=t+h           " Update the time

alfak:0.1
alfaa:0.1       " Prevent division by zero
alfap:0.1
betak:1E-4
betaaf:1E-4    " Limits the step in
betap:1E-4     " the estimator
h:39E-6        " Sample rate
gammak:0.001
gammaaf:0.001  " Estimating gain
gammap:0.001

a1:-2.7665
a2:2.5541      " Parameters in A(q)
a3:-0.7864
bg0:0.4909E-3
bg1:-0.9819E-3 " Parameters in BG(q)
bg2:0.4910E-3

END

```

---

```

DISCRETE SYSTEM fforward

```

```

"
" This is a feedforward-controller using that
" Bh(q) has its zeroes on the unit circle and
" that Bg(q) has a double-pole close to +1.
" This gives the feedforward-controller
"
"  $H_{ff}(q) = k \cdot (q^2 - af \cdot q + 1) / (q - p)^2$ 
"

```

```

STATE Uin1 Uin2 Us1 Us2
NEW  nUin1 nUin2 nUs1 nUs2

```

```

INPUT  Uin0 k af p           " Inputs
OUTPUT Us                    " Output

```

```

TIME t                        " Time
TSAMP ts

```

```

" The controller:

```

```

Us0=k*(Uin0-af*Uin1+Uin2)+2*p*Us1-p*p*Us2
Ushelp=IF (styr<1) THEN 0 ELSE Us0
Us=IF Ushelp<-ulim THEN -ulim ELSE (IF Ushelp>ulim THEN ulim ELSE Ushelp)

```

```

nUin1=Uin0
nUin2=Uin1           " Update the states
nUs1=Us

```

```
nUs2=Us1
```

```
ts=t+h           " Update the time
```

```
h:39E-6
ulim:10         " The parameters
styr:0
```

```
END
```

---

```
CONTINUOUS SYSTEM signal
```

```
"
```

```
" Creates the disturbances to simulate the converter.
```

```
"
```

```
INPUT d         " Input
OUTPUT Uin      " Output
```

```
TIME t         " Time
```

```
pi=arccos(-1)
omega=2*pi*f
help=A3*sin(f3*omega*t)
Uhvdc=A0*sin(omega*f0*t)+A1*sin(f1*omega*t)+A2*sin(f2*omega*t)+help
Uindet=IF t>stept THEN (Uhvdc+step) ELSE Uhvdc
Uin=Uindet+d
```

```
A0:2
A1:5           " Amplitudes of the
A2:5           " sinusoidal signals
A3:5
step:5         " Amplitude of the step
stept:0        " Time to begin the step
f0:6
f1:12
f2:24         " Frequencies of the
f3:36         " sinusoidal signals
f:50
```

```
END
```

---

```
CONNECTING SYSTEM connect
```

```
"
```

```
" Connects the hole system
```

```
"
```

```
"
```

```
d[signal]=d[noise]
Uin[sys]=Uin[signal]
```

```
Uin0[fforward]=Uin[signal]
Us[sys]=Us[fforward]
k[fforward]=k[festim]
af[fforward]=af[festim]
p[fforward]=p[festim]
```

```
Uin[festim]=Uin[signal]  
iout[festim]=iout[sys]
```

```
iout[refl]=iout[sys]  
i3[sys]=0  
"i3[sys]=irefl[refl]
```

```
END
```

### B.3 Programs Common For the RLS and RST Simulations

The SIMNON code identical in the simulations made in chapter 5 and 6 is shown in this section. The modules are 'estim' and 'noise2'.

```

DISCRETE SYSTEM estim
"
" Estimates the parameters in
"
"  $G(q)=(bg_0q^2+bg_1q+bg_2)/(q^3+a_1q^2+a_2q+a_3)$ ,
"
"  $H(q)=(bh_0q^2+bh_1q+bh_2)/(q^3+a_1q^2+a_2q+a_3)$  and
"
" The net
"
" using Stochastic approximation algorithm with forgetting factor
"
"
" Parameter: lam = forgettingfactor
"           alfai, alfaii, alfaus = initial covariances
"
"
STATE us1 us2 us3 i1 i2 i3
NEW nus1 nus2 nus3 ni1 ni2 ni3

STATE us4 us5 us6 us7 us8 us9 us10 us11 us12 us13 us14 us15
NEW nus4 nus5 nus6 nus7 nus8 nus9 nus10 nus11 nus12 nus13 nus14 nus15

STATE us16 us17 us18 us19 us20 us21 us22 us23 us24
NEW nus16 nus17 nus18 nus19 nus20 nus21 nus22 nus23 nus24

STATE i4 i5 i6 i7 i8 i9 i10 i11 i12 i13 i14 i15
NEW ni4 ni5 ni6 ni7 ni8 ni9 ni10 ni11 ni12 ni13 ni14 ni15

STATE i16 i17 i18 i19 i20 i21 i22 i23 i24
NEW ni16 ni17 ni18 ni19 ni20 ni21 ni22 ni23 ni24

STATE a11 a21 a31 ap11 ap21 ap31 ap41 bg01 bg11 bg21
NEW na11 na21 na31 nap11 nap21 nap31 nap41 nbg01 nbg11 nbg21

STATE bgp11 bgp21 bgp31 bh01 bh11 bh21 bhp11 bhp21 bhp31
NEW nbgp11 nbgp21 nbgp31 nbh01 nbh11 nbh21 nbhp11 nbhp21 nbhp31

STATE uin1 uin2 uin3 uin4 uin5 uin6 uin7 uin8 uin9 uin10 uin11
NEW nuin1 nuin2 nuin3 nuin4 nuin5 nuin6 nuin7 nuin8 nuin9 nuin10 nuin11

STATE uin12 uin13 uin14 uin15 uin16 uin17 uin18 uin19 uin20
NEW nuin12 nuin13 nuin14 nuin15 nuin16 nuin17 nuin18 nuin19 nuin20

STATE uin21 uin22 uin23 uin24
NEW nuin21 nuin22 nuin23 nuin24

STATE pinv11 pinvui1 pinvus1 n1
NEW npinv11 npinvui1 npinvus1 nn1

INPUT iout us uin " Inputs

```

```

OUTPUT a1 a2 a3 bg0 bg1 bg2      " Outputs

TIME t                          " Time
TSAMP ts

INITIAL

pinvii=1/alfai
pinvui1=1/alfau1                " Initial values of the
pinvus1=1/alfaus                " covariances

SORT

fi1=-i1
fi2=-i2
fi3=-i3
fi4=-i20
fi5=-i21
fi6=-i22
fi7=-i23
fi8=us1
fi9=us2                          " Gives the fi vector
fi10=us3                          " its values
fi11=us21
fi12=us22
fi13=us23
fi14=uin1
fi15=uin2
fi16=uin3
fi17=uin21
fi18=uin22
fi19=uin23

gotime=ngo*h                     " Time to begin the estimation
runtime=nrun*h                   " Time to run the estimation

go=IF (t>gotime) THEN (IF t<(gotime+runtime) THEN 1 ELSE 0) ELSE 0

fifii=fi1*fi1+fi2*fi2+fi3*fi3+fi4*fi4+fi5*fi5+fi6*fi6+fi7*fi7
fifius=fi8*fi8+fi9*fi9+fi10*fi10+fi11*fi11+fi12*fi12+fi13*fi13
fifiuin=fi14*fi14+fi15*fi15+fi16*fi16+fi17*fi17+fi18*fi18+fi19*fi19
fifiu=fifius+fifiuin
fifi=fifii+fifiu
fitheta1=fi1*a11+fi2*a21+fi3*a31+fi4*ap11+fi5*ap21+fi6*ap31+fi7*ap41
fitheta2=fi8*bg01+fi9*bg11+fi10*bg21+fi11*bgp11+fi12*bgp21+fi13*bgp31
fitheta3=fi14*bh01+fi15*bh11+fi16*bh21+fi17*bhp11+fi18*bhp21+fi19*bhp31
fitheta=fitheta1+fitheta2+fitheta3
pinvui=IF (go>0.5) THEN (pinvui1*lam+fifiuin) ELSE pinvui1
pinvus=IF (go>0.5) THEN (pinvus1*lam+fifius) ELSE pinvus1
pinvi=IF (go>0.5) THEN (pinvi1*lam+fifii) ELSE pinvi1
e=iout-fitheta                  " The estimating error
helpui=go*e/pinvui
helpus=go*e/pinvus
helpi=go*e/pinvi

a1=a11+fi1*helpi
a2=a21+fi2*helpi
a3=a31+fi3*helpi

```

```

ap1=ap11+fi4*helpi
ap2=ap21+fi5*helpi
ap3=ap31+fi6*helpi
ap4=ap41+fi7*helpi
bg0=bg01+fi8*helpus
bg1=bg11+fi9*helpus
bg2=bg21+fi10*helpus
bgp1=bgp11+fi11*helpus      " Calculate the estimated
bgp2=bgp21+fi12*helpus      " parameters
bgp3=bgp31+fi13*helpus
bh0=bh01+fi14*helpui
bh1=bh11+fi15*helpui
bh2=bh21+fi16*helpui
bhp1=bhp11+fi17*helpui
bhp2=bhp21+fi18*helpui
bhp3=bhp31+fi19*helpui

k=ap1
bf0=(ap2-ap1*a1)/(2*ap1)    " Calculate the dampin factor k
bf1=(ap3-ap1*a2)/(2*ap1)    " and the polynomial Bf(q)
bf2=(ap4-ap1*a3)/(2*ap1)

nus1=us
nus2=us1
nus3=us2
ni1=iout
ni2=i1
ni3=i2

na11=a1
na21=a2
na31=a3
nap11=ap1
nap21=ap2
nap31=ap3
nap41=ap4
nbg01=bg0
nbg11=bg1
nbg21=bg2
nbgp11=bgp1                  " Update the estimated
nbgp21=bgp2                  " parameters
nbgp31=bgp3
nbh01=bh0
nbh11=bh1
nbh21=bh2
nbhp11=bhp1
nbhp21=bhp2
nbhp31=bhp3

" Covariance resetting
reset=IF ((ts/restime)>n1) THEN 1 ELSE 0
n=IF (reset>0.5) THEN (n1+1) ELSE n1

npinvui1=IF (reset>0.5) THEN (1/alfai) ELSE pinvui
npinvus1=IF (reset>0.5) THEN (1/alfaus) ELSE pinvus
npinvii1=IF (reset>0.5) THEN (1/alfai) ELSE pinvi

nn1=n
nus4=us3
nus5=us4

```

nus6=us5  
nus7=us6  
nus8=us7  
nus9=us8  
nus10=us9  
nus11=us10  
nus12=us11  
nus13=us12  
nus14=us13  
nus15=us14  
nus16=us15  
nus17=us16  
nus18=us17  
nus19=us18  
nus20=us19  
nus21=us20  
nus22=us21  
nus23=us22  
nus24=us23

" Update the states us

ni4=i3  
ni5=i4  
ni6=i5  
ni7=i6  
ni8=i7  
ni9=i8  
ni10=i9  
ni11=i10  
ni12=i11  
ni13=i12  
ni14=i13  
ni15=i14  
ni16=i15  
ni17=i16  
ni18=i17  
ni19=i18  
ni20=i19  
ni21=i20  
ni22=i21  
ni23=i22  
ni24=i23

" Update the states i

nuin1=uin  
nuin2=uin1  
nuin3=uin2  
nuin4=uin3  
nuin5=uin4  
nuin6=uin5  
nuin7=uin6  
nuin8=uin7  
nuin9=uin8  
nuin10=uin9  
nuin11=uin10  
nuin12=uin11  
nuin13=uin12  
nuin14=uin13  
nuin15=uin14  
nuin16=uin15  
nuin17=uin16

" Update the states uin

```
nuin18=uin17
nuin19=uin18
nuin20=uin19
nuin21=uin20
nuin22=uin21
nuin23=uin22
nuin24=uin23
```

```
ts=t+h          " Update the time

lam:1           " Estimating gain
h:39.062500E-6 " Sample period
alfai:100
alfai:100      " Prevent division by zero
alfaus:100
ngo:1100      " Starts the estimator
runtime:1E20  " Keep the estimator running
restime:2E-1  " Resets the covariance
```

```
END
```

---

```
DISCRETE SYSTEM noise2
```

```
"
" Generates white Gaussian noise with mean value "mean" and
" deviation "stdev".
"
```

```
TIME t          " Time
TSAMP ts

OUTPUT d        " Output

d=stdev*NORM(t)+mean " Calculate the noise

ts=t+h          " Update the time

h:39.0625E-6    " Sample rate
mean:0          " Mean value
stdev:0.1       " Deviation
```

```
END
```



## B.4 RLS Program

The SIMNON program used in chapter 5, when the stochastic approximation algorithm was used.

```

MACRO hvdc5
"
" Estimating the transfer functions G(q), H(q) and F(q) plus the
" damping factor a in the net.
" Are using the Stochastic approximation algorithm with forgetting
" factor.
"
SYST sys signal refl2 noise noise2 estim5 connect

LET stoptime=5           " Stops the simulation
,stime=4.96             " Stops part 1/ starts part 2
,shtime=4.98           " Showtime
,storet=0               " Storetime
,steptime=0.01          " Steptime
,hc=39.0625E-6          " Sample rate

PAR A0[signal]:2
,A1[signal]:5           " Amplitudes of the simulated
,A2[signal]:5           " disturbance
,A3[signal]:5
,step:0                 " Amplitude of the step
,stepb:steptime         " Steptime
,f0[signal]:6
,f1[signal]:12          " Frequencies of the
,f2[signal]:24          " simulated disturbance
,f3[signal]:36
,rfactor[refl]:0.8      " Damping factor
,stdev[noise]:0         " Deviation of the noise1
,stdev[noise2]:0.2     " Deviation of the noise2
,lam:0.999              " Estimation gain
,alfai:1000
,alfai:2E-2             " Prevents division by zero
,alfaus:1E-3
,ngo[estim]:2500        " Starts the estimator
,runtime[estim]:1E10   " Runs the estimator
,restime[estim]:1E-1   " Resets the covariance
,h[refl]:hc
,h[noise]:hc            " Sets the sample rate in
,h[noise2]:hc           " the whole system
,h[estim]:hc

" Initial values of the parameters:

INIT bg01[estim]:0.4909E-3
,bg11[estim]:-0.9622E-3 "Double zero in 0.98
,bg21[estim]:0.4715E-3 "The calculated values
,a11[estim]:-2.7665
,a21[estim]:2.5541
,a31[estim]:-0.7864
,ap11[estim]:0.8
,ap21[estim]:-2.55336   "Bf double zero in 0.99
,ap31[estim]:2.71672
,ap41[estim]:-0.96256

```

```
,bgp11[estim]:0.39272e-3
,bgp21[estim]:-0.76976E-3          "0.8*Bg
,bgp31[estim]:0.3772E-3
,bh01[estim]:0.1734e-3
,bh11[estim]:-0.3381E-3          "One in 0.98 and one in 0.97
,bh21[estim]:0.1648E-3
,bhp11[estim]:0.13872e-3
,bhp21[estim]:-0.27048E-3        "0.8*Bh
,bhp31[estim]:0.13184E-3

" Simulate and plot

STORE bg01[estim] ap21[estim] ap31[estim] ap41[estim]
STORE bg11[estim] bg21[estim] a11[estim] a21[estim] a31[estim] -ADD
STORE ap11[estim] bgp11[estim] bgp21[estim] bgp31[estim] -ADD
STORE bh01[estim] bh11[estim] bh21[estim] bhp11[estim] -ADD
STORE bhp21[estim] bhp31[estim] e[estim] iout[sys] -ADD
SPLIT 3 2
AREA 1 1
AXES H 0 stime V -3 3
PLOT ap11[estim] ap21[estim] ap31[estim] ap41[estim]
TEXT 'ap1 - ap4'
SIMU 0 stime /plotlist 3.90625E-3
AREA 1 2
AXES H 0 stime V -2e-3 2E-3
SHOW bgp11[estim] bgp21[estim] bgp31[estim] bg01[estim] bg11[estim] bg21[estim] /plotlist
TEXT 'bgp1-bgp3,bg0-bg2'
AREA 2 1
AXES H 0 stime V -3 3
SHOW a11[estim] a21[estim] a31[estim] /plotlist
TEXT 'a1 - a3'
AREA 2 2
AXES H 0 stime V -5e-4 5e-4
SHOW bh01[estim] bh11[estim] bh21[estim] bhp11[estim] bhp21[estim] bhp31[estim] /plotlist
TEXT 'bh0-bh2,bhp1-bhp3'
SIMU stime stoptime -CONT /plotlis2 3.90625E-6
AREA 3 1
AXES H 0 stoptime V -5e-5 5e-5
SHOW e[estim] /plotlist
TEXT 'e=iout-fitheta'
AREA 3 2
AXES H shtime stoptime V -2e-2 2e-2
SHOW iout[sys] /plotlis2
TEXT 'iout'
MARK A 2 15
MARK "Estimating the parameters in G(q), H(q) and the net
EXPORT plotlist < plotlist 5000 storet /0 " Converts the file

END
```

---

CONTINUOUS SYSTEM signal

```
"
" Creates the signal to simulate the disturbances from the converter
" This is a sum of sinusoidals. It as possible to add some white Gaussian
" noise to the signal.
"
INPUT d          " Input
OUTPUT Uin      " Output
```

```

TIME t                " Time

" Calculations:

pi=arccos(-1)
omega=2*pi*f
Uhvdc=A0*sin(omega*f0*t)+A1*sin(f1*omega*t)+A2*sin(f2*omega*t)+Uhelp
Uhelp=A3*sin(f3*omega*t)
Uindet=IF t>steptb THEN (IF (t<stepts) THEN (Uhvdc+step) ELSE Uhvdc) ELSE Uhvdc
Uin=Uindet+d

A0:2
A1:5                  " The amplitudes of the
A2:5                  " disturbance
A3:5
step:0               " Amplitude of the step
steptb:0             " Step starts
stepts:1E6           " Step stops
f0:6
f1:12
f2:24                " Frequence of the
f3:36                " disturbances
f:50

END

```

---

```

CONNECTING SYSTEM connect
"
"Connects the hole system
"
"
d[signal]=d[noise]
Uin[sys]=Uin[signal]
Us[sys]=d[noise2]
us[estim]=Us[sys]
iout[estim]=iout[sys]
uin[estim]=Uin[signal]
iout[ref1]=iout[sys]
"i3[sys]=0
i3[sys]=irefl[ref1]

END

```

## B.5 RST Program

The SIMNON-program used in chapter 6, when the RST controller was used. This version is the one used in section 6.1.3, where both the estimator and the controller were run simultaneously.

```

MACRO hvdc6
"
" Controlling using a RST-controller, designed with Am and Ao.
" Uses the calculated or estimated values of A and Bg, depending
" on the chosen value of ngo. This parameter controls when to start
" the estimator.
" The estimator uses the stochastic approximation algorithm with
" forgetting factor.
"
SYST sys signal2 ref12 noise noise2 estim6 control2 connect2

LET stoptime=8           " Stoptime 1
,newstop=8.1            " Stoptime 2
,storet=0                " Storetime 1
,store2=0                " Storetime 2
,stepon=0                " Starts the step
,stepoff=0              " Stops the step
,hcent=39.0625000E-6    " Sample rate

PAR A0[signal]:2
,A1[signal]:5           " The amplitude of the
,A2[signal]:5           " disturbances
,A3[signal]:5
,step:0                 " The amplitude of the step
,f0[signal]:6
,f1[signal]:12          " The frequencies of the
,f2[signal]:24          " disturbances
,f3[signal]:36
,rfactor[ref1]:0.7     " Damping factor
,stdev[noise]:0         " Deviation of the noise1
,stdev[noise2]:0.2     " Deviation of the noise2
,lam:0.9995            " Estimation gain
,alfai:1000
,alfai:1E-2             " Prevents division by zero
,alfaus:1E-3           " in the estimator
,omo[control]:1E4      " Sets the observer Ao(s)
,zetao[control]:0.9
,omm1:0.8E4
,omm2:300              " Sets the wanted denominator Am(s)
,zetam2:0.7
,ngo[estim]:10         " Starts the estimator
,nrun[estim]:100000E6  " Runs the estimator
,styrt[control]:4      " Starts the controller
,ulim[control]:15      " Limits the control signal
,restime[estim]:1E-1   " Resets the covariance
,h[ref1]:hcent
,h[noise]:hcent
,h[noise2]:hcent
,h[estim]:hcent
,h[control]:hcent
,h[signal]:hcent

```

```

INIT bg01[estim]:0.4909E-3
,bg11[estim]:-0.9819E-3
,bg21[estim]:0.4910E-3
,a11[estim]:-2.7665
,a21[estim]:2.5541
,a31[estim]:-0.7864
,ap11[estim]:0.7
,ap21[estim]:-2.23419           " Initially values of the
,ap31[estim]:2.38133           " parameters
,ap41[estim]:-0.84812
,bgp11[estim]:0.34363E-3
,bgp21[estim]:-0.68733E-3
,bgp31[estim]:0.3437E-3
,bh01[estim]:0.1734e-3
,bh11[estim]:-0.3429E-3
,bh21[estim]:0.1733E-3
,bhp11[estim]:0.12138E-3
,bhp21[estim]:-0.24003E-3
,bhp31[estim]:0.12131E-3

" Simulate and plots

ERROR 1E-6
STORE bg01[estim] ap21[estim] ap31[estim] ap41[estim]
STORE bg11[estim] bg21[estim] a11[estim] a21[estim] a31[estim] -ADD
STORE ap11[estim] bh01[estim] bh11[estim] bh21[estim] -ADD
STORE us[control] iout[sys] -ADD
SPLIT 3 2
AREA 1 1
AXES H 0 stoptime V -1.5E-3 1.5E-3
PLOT bg01[estim] bg11[estim] bg21[estim]
TEXT 'bg0 - bg2'
SIMU 0 stoptime /plotlist 3.90625E-3
SIMU stoptime newstop -CONT /plotlis2 hcent
AREA 1 2
AXES H 0 stoptime V -3 3
SHOW a11[estim] a21[estim] a31[estim] /plotlist
TEXT 'a1 - a3'
AREA 2 1
AXES H 0 stoptime V -2E-3 2E-3
SHOW bh01[estim] bh11[estim] bh21[estim] /plotlist
TEXT 'bh0 - bh2'
AREA 2 2
AXES H 0 stoptime V -16 16
SHOW us[control] /plotlist
"PLOT us[control]
TEXT 'us'
AREA 3 1
AXES H 0 stoptime V -1.5E-2 1.5E-2
SHOW iout[sys] /plotlist
TEXT 'iout'
"AREA 3 2
"AXES H 0 stoptime V -5E-2 5E-2
"SHOW i3[sys] /plotlist
"TEXT 'irefl'
MARK A 2 15
MARK "RST - controller, G(q),H(q) and the net are estimated; refl:a=0.7
EXPORT plotlist < plotlist 6000 storet /0 " Converts the file
EXPORT plotlis2 < plotlis2 5000 stoptime /0 " Converts the file

```

END

DISCRETE SYSTEM control

```

"
"
" Indirect RST-controller connected as a feedforward-controller.
" Have anti-reset windup
"
" Inputs are the parameters in G and H, the polynom Am in the
" transfer function wanted for the closed loop  $G_m=B_m/A_m$  and
" the observer  $A_o$ 
"
"  $B_m=B$  because no zeroes can be cancelled
"
"  $A_m=q^3+am_1q^2+am_2q+am_3$  is given in continuous time by the parameters
"  $omm_1, omm_2$  and  $zeta_m2$  in  $A(s)=(s+omm_1)(s^2+2*omm_2*zeta_m2*s+omm_2*omm_2)$ 
"
"  $A_o=q^2+ao_1q+ao_2$  is given in continuous time by the parameters
"  $omo$  and  $zeta_o$  in  $A_o(s)=(s^2+2*omo*zeta_o*s+omo*omo)$ 

```

" NOTICE:

```

" There are no protection against poles outside the
" unit disc and it is presumed that there are no common factors
" in A and Bg.
"
"

```

```

STATE uin1 uin2 uin3 iout1 iout2 us1 us2
NEW nuin1 nuin2 nuin3 niout1 niout2 nus1 nus2

```

```

STATE r11 r21 usd1 usd2
NEW nr11 nr21 nusd1 nusd2

```

```

INPUT iout uin d bg0 bg1 bg2 a1 a2 a3 " Inputs
OUTPUT us r1o r2o am1o am2o am3o ao1o ao2o " Outputs

```

```

TIME t " Time
TSAMP ts

```

INITIAL

```

wm=omm2*sqrt(1-zeta_m2*zeta_m2)
alpham=exp(-zeta_m2*omm2*h)
betam=cos(wm*h) " Calculation of the
cm=exp(-omm1*h) " desired polynomial Am(q)
am1=-2*alpham*betam-cm
am2=2*alpham*betam*cm+alpham*alpham
am3=-cm*alpham*alpham

```

```

wo=omo*sqrt(1-zeta_o*zeta_o)
alphao=exp(-zeta_o*omo*h)
betao=cos(wo*h) " Calculation of the
ao1=-2*alphao*betao " observer polynomial Ao(q)
ao2=alphao*alphao

```

SORT

M1=ao1+am1-a1

```

M2=ao2+am1*ao1+am2-a2
M3=am1*ao2+am2*ao1+am3-a3
M4=am2*ao2+am3*ao1
M5=am3*ao2

L1=M2-a1*M1
L2=M3-a2*M1
L3=M4-a3*M1

K1=L1-M5/a3
K2=L2-a1*M5/a3
K3=L3-a2*M5/a3

Q0=bg2-a2*bg0+(bg1/bg0)*(a1*bg0-bg1)
Q1=bg1*bg2/(bg0*a3)+bg0-a1*bg2/a3
Q2=-a3*bg0+(bg2/bg0)*(a1*bg0-bg1)
Q3=bg2*bg2/(bg0*a3)+bg1-bg2*a2/a3

s2=(K3-bg2*K1/bg0-(Q2/Q0)*(K2-bg1*K1/bg0))/(Q3-Q1*Q2/Q0)
s1=(K1+bg2*s2/a3+(a1*bg0-bg1)*s0)/bg0
s0=(K2-bg1*K1/bg0-Q1*s2)/Q0           " Calculate S(q)

r2=(M5-bg2*s2)/a3                     " Calculate R(q)
r1=M1-bg0*s0

" RST - CONTROLLER, uref=0, anti-reset windup

tcon=IF (ts>styrt) THEN (ts-styrt) ELSE 0
us0=- (s0*iout+s1*iout1+s2*iout2)-ao1*us1-ao2*us2+(ao1-r1)*usd1+(ao2-r2)*usd2
usd=IF us0<-ulim THEN -ulim ELSE (IF us0>ulim THEN ulim ELSE us0)
ushelp=(1-exp(-50*tcon))*usd
us=ushelp+d

am1o=am1
am2o=am2
am3o=am3
ao1o=ao1           " Creates output signals
ao2o=ao2
r1o=r11
r2o=r21

nr11=r1           " Update R(q)
nr21=r2

nuin1=uin
nuin2=uin1       " Update uin
nuin3=uin2
niout1=iout     " Update iout
niout2=iout1

nus1=us0
nus2=us1         " Update us
nusd1=usd
nusd2=usd1      " Update usd

ts=t+h           " Update the time

h:39.0625000E-6 " Sample rate
styrt:1E10      " Starts the controller

```

```

ulim:15                " Limits the control signal
omm1:1
omm2:1                " Initially value of Am(s)
zetam2:1              " and Ao(s)
omo:1
zetao:1

```

END

---

DISCRETE SYSTEM signal

```

"
" This is a discrete generator of disturbances, which is a sum
" of four sinusoidal signals.
"
INPUT d                " Input
OUTPUT Uin            " Output

TIME t                " Time
TSAMP ts

pi=arccos(-1)
omega=2*pi*f
Uhvdc=A0*sin(omega*f0*ts)+A1*sin(f1*omega*ts)+A2*sin(f2*omega*ts)+Uhelp
Uhelp=A3*sin(f3*omega*ts)
Uindet=IF ts>stept THEN (Uhvdc+step) ELSE Uhvdc
Uin=Uindet+d

ts=t+h                " Update the time

h:39.0625E-6          " Sample rate
A0:2
A1:5                  " The amplitudes of
A2:5                  " the disturbance
A3:5
step:0                " The amplitude of the step
stept:0               " Steptime
f0:6
f1:12                 " The frequency of the
f2:24                 " disturbance
f3:36
f:50

```

END

---

CONNECTING SYSTEM connect

```

"
"Connects the whole system
"
d[signal]=d[noise]
Uin[sys]=Uin[signal]
Us[sys]=us[control]
d[control]=d[noise2]
iout[control]=iout[sys]
uin[control]=Uin[signal]
us[estim]=Us[sys]
iout[estim]=iout[sys]

```



```
uin[estim]=Uin[signal]
```

```
iout[ref1]=iout[sys]
```

```
"i3[sys]=0
```

```
i3[sys]=iref1[ref1]
```

```
bg0[control]=bg0[estim]
```

```
bg1[control]=bg1[estim]
```

```
bg2[control]=bg2[estim]
```

```
a1[control]=a1[estim]
```

```
a2[control]=a2[estim]
```

```
a3[control]=a3[estim]
```

```
END
```