

# Model-based Algorithms for Nonlinear System Identification

by

Fredrik Petter Skantze

Submitted to the Department of Mechanical Engineering  
in partial fulfillment of the requirements for the degree of

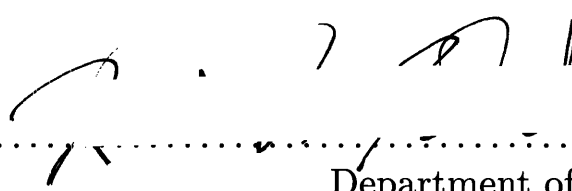
Master of Science in Mechanical Engineering

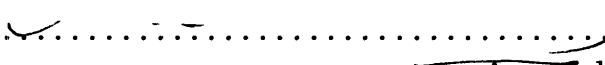
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1996

© Massachusetts Institute of Technology 1996. All rights reserved.

Author  .....  
Department of Mechanical Engineering  
February 8, 1996

Certified by  .....  
Anuradha M. Annaswamy  
Assistant Professor  
Thesis Supervisor

Accepted by  .....  
Ain A. Sonin  
Chairman, Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

MAR 19 1996

Eng.

LIBRARIES

# Model-based Algorithms for Nonlinear System Identification

by

Fredrik Petter Skantze

Submitted to the Department of Mechanical Engineering  
on February 8, 1996, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Mechanical Engineering

## Abstract

In this thesis three algorithms for the estimation of parameters which occur nonlinearly in dynamic systems are presented. The first algorithm pertains to systems in discrete-time regression form. It is shown that the task of finding an update law for the parameter estimates can be solved numerically by the formulation of a quadratic programming problem. The algorithm does not depend on analytical knowledge of the regressor function. In particular, a neural system model can be used to approximate the required regression form for systems which cannot easily be transformed into this form analytically. The second algorithm makes use of model-based parameterizations. It is shown that if some of the system parameters occur linearly, or enter the model multiplicatively, an update law for these parameters can be found analytically. The third algorithm makes use of convex properties of the regression function and applies to a class of continuous-time systems. It is demonstrated how the algorithms can be modified to make them robust in the presence of a bounded disturbance. The performance of the algorithms and the nature of the parameter convergence are illustrated in simulations of a magnetic bearing system and a low velocity friction model.

Thesis Supervisor: Anuradha M. Annaswamy

Title: Assistant Professor

## Acknowledgments

The three semesters that I spent in the Adaptive Control Laboratory at MIT have been a tremendously exciting and rewarding experience. I would like to thank my advisor, Professor Anuradha Annaswamy, for sharing with me her enthusiasm for research and teaching me the rewards of rigorous analytical work.

I am deeply grateful to Ssu-Hsin Yu for patiently helping me at virtually every stage of my research. Our many discussions are reflected in every page of this thesis.

I would also like to thank Ssu-Hsin and Rienk Bakker for their friendship and support which helped make my initial experience at MIT less overwhelming. Many thanks also to the other students of the Adaptive Control Laboratory, Steingrimur Karason, Kathy Misovec and Aleksandar Kojic for creating a pleasant atmosphere in the lab.

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Motivation . . . . .	6
1.2	Previous Work . . . . .	7
1.3	Contribution of the Thesis . . . . .	9
1.4	Notation . . . . .	10
1.5	Organization of the Thesis . . . . .	10
<b>2</b>	<b>A Recursive Parameter Estimation Algorithm</b>	<b>11</b>
2.1	Introduction . . . . .	11
2.2	The Algorithm . . . . .	12
2.3	Robustness to a Bounded Disturbance . . . . .	20
2.4	Using a Neural System Model . . . . .	23
<b>3</b>	<b>Partially Linear Systems</b>	<b>26</b>
3.1	Introduction . . . . .	26
3.2	The Algorithm . . . . .	27
3.3	When $\theta$ is a Scalar . . . . .	30
3.4	Robustness to a Bounded Disturbance . . . . .	31
<b>4</b>	<b>A Continuous-time Algorithm</b>	<b>33</b>
4.1	Introduction . . . . .	33
4.2	The General Case . . . . .	34
4.2.1	The Algorithm . . . . .	34

4.2.2 Application to a Magnetic Bearing System <sup>1</sup> . . . . . 38

4.3 When  $f$  is concave or convex in  $\theta$  . . . . . 40

4.3.1 The Algorithm . . . . . 40

4.3.2 Application to a Low-velocity Friction Model . . . . . 45

**5 Conclusion** . . . . . **51**

---

<sup>1</sup>This example was prepared in collaboration with Dr. Ssu-Hsin Yu, Postdoctoral Fellow in the Adaptive Control Laboratory at MIT.

# Chapter 1

## Introduction

### 1.1 Motivation

In many problems concerned with identification and prediction in complex engineering systems, models that capture the dominant system behavior are available. These models can be derived using physical laws such as conservation equations or constitutive relations and the constants which characterize the behavior of these differential equations have a physical meaning. In a large number of applications, including magnetic bearing systems, low velocity friction compensation, pH control and fermentation and combustion processes, some of the parameters enter the model nonlinearly. Due to changes in the operating conditions and variations in the system characteristics, there is uncertainty associated with these parameters. However, no general analytical solution is available for the identification of parameters which occur nonlinearly in dynamic systems. In order to achieve analytical tractability, the original system model is usually replaced by a linear black-box model. An extensive theory has been developed for the estimation of the parameters of these models [11, 7] and the vast majority of publications in the system identification literature are concerned with these algorithms and their statistical properties. Although linear model structures work well for many applications, they limit the ability of the model to replicate the actual physical characteristics of the process, since these often exhibit nonlinear features. As a result, the model will only be valid for a limited range of operating

conditions.

## 1.2 Previous Work

Recently, considerable attention has been given to the development of nonlinear black-box models for system identification. In particular, approaches based on neural networks have been investigated by a number of researchers, for example [13, 18, 19]. An attractive feature of these computational structures is that they are universal approximators which can model any continuous function to any desirable degree of accuracy [8, 15]. The neural network architecture usually includes a large number of parameters, some of which occur nonlinearly while others might occur linearly. These parameters are estimated based on input/output data from the nonlinear function to be approximated. The neural network is most efficient as a function approximator if the parameters entering its architecture nonlinearly are adjusted. For example, multi-layered neural networks with sigmoidal activation functions achieve an integrated squared error of order  $\frac{1}{n}$ , where  $n$  is the number of nodes [3]. Thus the approximating error of the multi-layered network is independent of the input dimension of the function to be approximated, which is a desirable property. The parameters of the neural network can be estimated off-line with a nonlinear optimization method such as the backpropagation technique or the Gauss-Newton algorithm. The resulting model must then be validated to ensure that the optimization problem has not converged to a local minimum. If only the neural network parameters occurring linearly are adjusted, the approximating capabilities of the neural network are somewhat deteriorated and the integrated square approximation error of the neural network cannot be made smaller than order  $\left(\frac{1}{n}\right)^{\frac{2}{d}}$ , where  $n$  is the number of basis functions and  $d$  is the input dimension of the nonlinear function being approximated [3]. Thus, in general, such a network will consist of more basis functions and parameters. When the system identification is to be performed on-line, this has proven a useful tradeoff for the sake of analytical tractability, since in this case standard results from the adaptive control literature can be used in the stability analysis. Results obtained with these

approaches indicate that when little prior information is available about the underlying system, they can lead to significant improvement. However, when a nonlinear system model is available, the problem of estimating a few parameters which occur nonlinearly in the original physical model is converted to the estimation of a large number of parameters of the neural network. For on-line identification, this tradeoff is questionable, since the estimation of a large number of parameters results in poor transient performance and nonuniqueness problems. Furthermore, the neural model provides little insight into the structure and dynamics of the actual system since the parameters of the network have no physical meaning.

In [1, 2] two novel, model-based uses of neural networks for system identification tasks are proposed. In the first approach, which is referred to as the block estimation method, the neural network is trained to learn the implicit function between the system variables and the values of the physical parameters,  $\theta$ , associated with the system. The second approach is a recursive estimation method where the neural network updates the parameter estimates on-line based on new samples of the system response. To distinguish these neural networks from those identifying the system input-output characteristics, they are referred to as  $\theta$ -adaptive neural networks (TANN), since they adapt to the system parameters,  $\theta$ . The training of the neural network is performed off-line, so that the parameter estimates of the system model can be obtained at each time instance by evaluating the neural network. As a result, fewer parameters are estimated on-line, which results in better transient performance. The resulting system model will be compact and the estimated parameters will have a physical interpretation. This is desirable since many powerful control techniques exist for nonlinear systems whose parameters are known [9, 21]. Furthermore, estimates of the physical parameters of a process contain valuable information for tasks such as fault detection and diagnosis.

Since, in general, it is not possible to derive analytical update laws for the estimates of parameters which occur nonlinearly in dynamic systems, the neural network in the recursive algorithm in [1, 2] has no direct target which can be used in its training. Instead, the amount by which it is desired to decrease the squared norm of



the parameter estimation error is used as a distal target. The weights of the neural network are adjusted by backpropagating the difference between the desired and the actual decrease in the parameter error using the training with a distal teacher procedure [10]. This is inherently a nonlinear optimization problem even if only the linear weights of the neural network are adjusted. Furthermore, the neural network has to be trained for variations in all the system variables and grows exponentially in size when the number of variables is increased.

### **1.3 Contribution of the Thesis**

In this thesis it is shown that the task of finding an update law for the estimates of parameters which occur nonlinearly in dynamic systems can be solved numerically by the formulation of a quadratic programming problem. If a model in discrete-time regression form is available or a simulation model can be used on-line, then this optimization problem can be solved on-line to yield a parameter update law at each sampling instance. Analytical update laws are derived for parameters which enter the regression equation linearly and parameters which multiply a nonlinear function which depends on unknown parameters. Therefore the dimension of the quadratic programming problem only depends on the number of parameters occurring nonlinearly. This greatly reduces the computational resources needed to implement the algorithm. If it is difficult to transform the system model into regression form, or if a simulation model is only available off-line, a neural system model can be constructed to approximate the unknown regression function. This neural network can then be used as a system model in the algorithm. The algorithm is then applicable to a large class of nonlinear systems. A modification to the estimation laws is presented which guarantees the stability of the algorithms in the presence of a bounded additive disturbance. The capabilities of the algorithms are demonstrated through simulation results for a magnetic bearing system and a low velocity friction model.

## 1.4 Notation

In order to enhance readability the following notation will be adhered to throughout the paper. The estimate of a quantity  $x$  will be denoted by  $\hat{x}$  and the estimation error of  $x$  will be denoted by  $\tilde{x}$ , where  $\tilde{x} = \hat{x} - x$ . The change in  $x$  between time  $t - 1$  and  $t$  is given by  $\Delta x_t = x_t - x_{t-1}$ . A function of several variables which are themselves functions of time is denoted as  $f_t$  if its arguments are evaluated at time  $t$  and possibly at past instants  $t - 1, t - 2, \dots$ . For example,  $f(x_{t-1}, y_{t-2})$  will be denoted as  $f_{t-1}$  and so on.  $\hat{f}_t$  corresponds to an estimate of  $f_t$  with all of its arguments that are not measurable replaced by their estimates.

## 1.5 Organization of the Thesis

The thesis is organized as follows. In Chapter 2 an algorithm for the estimation of parameters occurring nonlinearly in dynamic systems is presented with a stability proof. A modified version of this algorithm, which takes into account model based parameterizations is given in Chapter 3. In Chapter 4 a continuous-time algorithm is developed for systems where only one parameter occurs nonlinearly. The Thesis is summarized in Chapter 5.

# Chapter 2

## A Recursive Parameter Estimation Algorithm

### 2.1 Introduction

In this chapter we are concerned with dynamic systems which can be represented in the regression form

$$y_t = f(\phi_{t-1}, \theta), \quad (2.1)$$

where  $y_t$  is the output of the system at time  $t$ ,  $\phi_{t-1}$  represents past values of the measurable system signals and  $\theta$  is a vector of unknown system parameters. It is assumed that  $f$  is sufficiently smooth and that  $\theta$  lies in the known,  $n$  dimensional box  $\Theta$  defined by

$$\Theta = \{\theta \mid \theta_{i_{\min}} \leq \theta_i \leq \theta_{i_{\max}}, i = 1, \dots, n\},$$

where  $\theta_i$  is the  $i$ th component of  $\theta$ . The task is to estimate  $\theta$ . In particular, the focus is on systems where  $f$  is a nonlinear function of the parameter vector  $\theta$ , since in this case few analytical tools are available for the identification of the parameter vector. Under some standard assumptions on the system functions and the input, a large number of models can be transformed into the form of Eq. (2.1) [2].

In Section 2.2 an algorithm is presented which applies when  $f$  is either known analytically or a simulation model of the system is available on-line. In many cases,

it is difficult to find the transformation from a given model structure to the regression form. Section 2.4 outlines how the existing system model can be used to train a neural network to approximate  $f$  in Eq. (2.1). The neural model can then be used in conjunction with the algorithm in Section 2.2 to estimate  $\theta$ . In particular, the method is applicable to nonlinear continuous-time and discrete-time state space models when the state variables are not accessible. The algorithm also applies when a simulation model of the system is only available off-line.

## 2.2 The Algorithm

In this section it is assumed that  $f$  is either known analytically or that a simulation model is available such that, given  $\phi_t$  and  $\theta$ ,  $y_t$  can be evaluated on-line. The recursive estimation problem can be stated as follows. We want to find a function  $F_t$  such that the algorithm

$$\hat{\theta}_t = \hat{\theta}_{t-1} - F_t \tag{2.2}$$

ensures that  $\hat{\theta}_t \rightarrow \theta$  as  $t \rightarrow \infty$ . If the system is linearly parameterized such that Eq. (2.1) can be written as

$$y_t = f(\phi_{t-1})^T \theta,$$

then several algorithms are available for the estimation of  $\theta$  and conditions under which accurate identification can be carried out are well known. The exact choice of  $F$  depends on the criterion of best fit. If  $F$  is chosen so as to minimize the cost function

$$J = \frac{1}{2} |\tilde{\theta}_t - \tilde{\theta}_{t-1}|^2,$$

subject to the constraint that  $\hat{\theta}$  lies on the hyperplane

$$H = \{\hat{\theta} : y_t = f(\phi_{t-1})^T \hat{\theta}\},$$

then we obtain the well know projection algorithm (e.g. [6])

$$\hat{\theta}_t = \hat{\theta}_{t-1} - k_{t-1} \tilde{y}_t f(\phi_{t-1}) \quad (2.3)$$

where

$$\begin{aligned} \tilde{y}_t &= f(\phi_{t-1})^T \hat{\theta} - y_t \\ k_{t-1} &= \frac{1}{\lambda + \phi_{t-1}^T \phi_{t-1}} \end{aligned}$$

and  $\lambda$  is a small positive constant.

If  $f$  in Eq. (2.1) is a nonlinear function of  $\theta$ , then, in general, it is not possible to find an analytical form for  $F_t$ . Therefor we propose to solve the problem numerically. For this purpose, the following recursive algorithm is proposed to estimate  $\theta$

$$\hat{\theta}_t = \hat{\theta}_{t-1} - k_t d_t, \quad (2.4)$$

where

$$k_t = \frac{1}{d_t^T d_t}, \quad (2.5)$$

which represents a normalized version of Eq. (2.2).  $d_t$  is a vector to be determined numerically at each time instance  $t$ . A natural question to ask is under what conditions there exists a vector  $d_t$  which makes the parameter error decrease. This is addressed in the following theorem.

**Theorem 1** For the system in Eq. (2.1) and the parameter estimation algorithm defined by Eqs. (2.4) and (2.5), a vector  $d_t$  which guarantees that  $|\tilde{\theta}_t| - |\tilde{\theta}_{t-1}| < 0$  can be found if and only if there exists a vector  $a$  and scalars  $b$  and  $\epsilon$  with  $\epsilon > 0$ , such that  $a^T \hat{\theta}_{t-1} > b + \epsilon$  and  $a^T \theta < b$  for all  $\theta$  in  $\Theta$  which satisfy the equation  $y_t = f(\phi_{t-1}, \theta)$ .

**Proof** Let

$$V_t = \tilde{\theta}_t^T \tilde{\theta}_t. \quad (2.6)$$

The change in  $V_t$  is given by

$$\Delta V_t = V_t - V_{t-1} = \tilde{\theta}_t^T \tilde{\theta}_t - \tilde{\theta}_{t-1}^T \tilde{\theta}_{t-1}.$$

If  $\Delta V_t < 0$ , then it follows that  $|\tilde{\theta}_t| - |\tilde{\theta}_{t-1}| < 0$ . After some algebraic manipulation  $\Delta V_t$  can be expressed as

$$\Delta V_t = 2\Delta\hat{\theta}_t^T \tilde{\theta}_{t-1} + \Delta\hat{\theta}_t^T \Delta\hat{\theta}_t,$$

where  $\Delta\hat{\theta}_t = \hat{\theta}_t - \hat{\theta}_{t-1}$ . Substituting for  $\Delta\hat{\theta}_t$  from Eq. (2.4) gives

$$\Delta V_t = -2k_t d_t^T \tilde{\theta}_{t-1} + k_t^2 d_t^T d_t. \quad (2.7)$$

By making use of the identity  $k_t d_t^T d_t = 1$ , Eq. (2.7) can be written as

$$\Delta V_t = k_t(-2d_t^T \tilde{\theta}_{t-1} + 1). \quad (2.8)$$

Let  $L_{y_t}$  be the level set

$$L_{y_t} = \{\theta \in \Theta \mid f(\phi_{t-1}, \theta) = y_t\}.$$

$\Delta V_t$  is negative if and only if

$$d_t^T(\hat{\theta}_{t-1} - \theta) > \frac{1}{2} \quad (2.9)$$

for all  $\theta$  in  $L_{y_t}$ . If  $d_t$  is chosen as

$$d_t = \frac{a}{2\epsilon},$$

then

$$d_t(\hat{\theta}_{t-1} - \theta) = \frac{a}{2\epsilon}(\hat{\theta}_{t-1} - \theta) > \frac{1}{2\epsilon}(b + \epsilon - b) = \frac{1}{2}.$$

This establishes sufficiency. Necessity follows from the fact that if there exists no vector  $a$  and scalars  $b$  and  $\epsilon$  with  $\epsilon > 0$ , such that  $a^T \theta < b$  for all  $\theta$  in  $L_{y_t}$  and  $a^T \hat{\theta}_{t-1} > b + \epsilon$ , then neither can a vector  $a$  and scalars  $b$  and  $\epsilon$  with  $\epsilon > 0$  be found

such that

$$a^T(\hat{\theta}_{t-1} - \theta) > \epsilon.$$

If we choose  $a = d_t$  and  $\epsilon = \frac{1}{2}$ , then this contradicts Eq. (2.9). □

The theorem implies that a  $d_t$  which makes the algorithm stable can only be found if there exists two parallel hyperplanes which separate  $\hat{\theta}_{t-1}$  from all  $\theta$  in  $L_{y_t}$  by a finite amount. This is illustrated in Figure 2.1. If  $\hat{\theta}_{t-1}$  is located anywhere in the shaded region, then it is not possible to find a  $d_t$  such that  $\Delta V_t < 0$ . We then set  $d_t = 0$  to ensure such that  $\Delta V_t = 0$ .

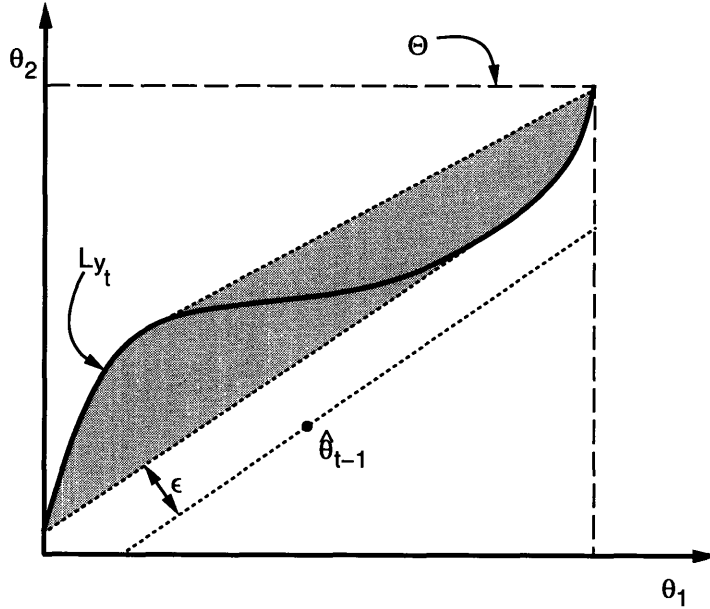


Figure 2-1: Graphical illustration of Theorem 1.

The problem of determining an update law for the parameter estimates consists of first determining the existence of a  $d_t$  which makes  $\Delta V_t$  negative and then finding such a  $d_t$ . Since these tasks are difficult to solve analytically, we propose the following numerical procedure for finding  $d_t$ . Let  $\theta_i, i = 1, \dots, n$  be the components of  $\theta$ . For each  $\theta$  in  $L_{y_t}$  we have  $n - 1$  degrees of freedom in choosing the components of  $\theta$ . For each  $i, i \neq j$ , divide  $\theta_i$  into  $q$  equally spaced intervals between  $\theta_{i_{\min}}$  and  $\theta_{i_{\max}}$  and form all possible  $q^{n-1}$  combination of these components. The  $j$ th component of  $\theta$  is then specified implicitly by Eq. (2.1). Since  $j$  can be any integer between 1 and  $n$ , it

is often possible to pick  $j$  such that  $\theta_j^r$  can be solved for explicitly as

$$\theta_j = g(y_t, \phi_{t-1}, \theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_n)$$

If this is not possible, or if the system model is not available in analytical form, then, assuming that  $f$  can be evaluated at different values of  $\theta$ ,  $\theta_j$  can be found using a one-dimensional root finding algorithm, in which case the index  $j$  should be chosen such that

$$\frac{\partial f(\phi_{t-1}, \theta)}{\partial \theta_j} \neq 0 \quad (2.10)$$

for all  $\theta$  in  $\Theta$ . Then  $\theta_j$  is unique and the bisection method is guaranteed to find the solution with a linear convergence rate [17]. If  $\theta_j$  does not satisfy Eq. (2.10), then there might exist multiple solutions. If these roots are bracketed appropriately, the bisection method will still find them. We now have  $q^{n-1}$  samples of  $\theta$  which lie in  $L_{y_t}$ . Denote these by  $\theta^1, \dots, \theta^R$ , where  $R = q^{n-1}$  and let  $T$  be the set defined by

$$T = \bigcup_{1 \leq r \leq R} \theta^r. \quad (2.11)$$

If we define

$$\tilde{\theta}_{t-1}^r = \hat{\theta}_{t-1} - \theta^r$$

and let

$$A_t = \begin{bmatrix} \tilde{\theta}_{t-1}^{1T} \\ \vdots \\ \tilde{\theta}_{t-1}^{RT} \end{bmatrix}, \quad \text{and} \quad b_t = \begin{bmatrix} \frac{1}{2} \\ \vdots \\ \frac{1}{2} \end{bmatrix}, \quad (2.12)$$

then the constraint given by Eq. (2.9), as applied to  $\theta^r$ ,  $r = 1, \dots, R$ , can be compactly expressed as

$$A_t d_t \geq b_t, \quad (2.13)$$

which represents a set of linear inequality constraints. Eq. (2.13) can have infinitely



many solutions. Of these we wish to choose the one which minimizes the cost function

$$J = d_t^T d_t, \quad (2.14)$$

since the magnitude of  $\Delta\hat{\theta}_t$  is proportional to  $\frac{1}{|d_t|}$ . The task of minimizing the cost function given by Eq. (2.14) subject to the constraints in Eq. (2.13) is a quadratic programming problem. It can be solved efficiently with a modified version of the well known simplex procedure for linear programming and is guaranteed to either find the optimal solution in a finite number of iterations or to detect that the problem is overconstrained [5, 22]. An overconstrained problem corresponds to a  $\hat{\theta}_{t-1}$  in the shaded region in Figure 2.1, which implies that there exists no two parallel hyperplanes which separate  $\hat{\theta}_{t-1}$  from all  $\theta$  in  $L_{y_t}$  by a finite amount. If no feasible solution exists,  $d_t$  is set to zero such that  $\hat{\theta}_t = \hat{\theta}_{t-1}$ , which ensures stability. Computer code for solving the quadratic programming problem is available in the Optimization Toolbox in MATLAB.

If  $\theta$  is in  $T$ , the above procedure provides a stable algorithm for estimating  $\theta$ . However, for a real system, the probability of  $\theta$  being in  $T$  is zero. For this reason we wish to modify the algorithm such that the nonincreasing property of the parameter error is guaranteed for those  $\theta$  in  $L_{y_t}$  but not in  $T$ . How this can be done is described in the following theorem.

**Theorem 2** Let  $\nu$  be a positive constant. If  $|d_t| \leq \nu$  and

$$d_t^T (\hat{\theta}_{t-1} - \theta^r) \geq \frac{1}{2} + \nu D \quad (2.15)$$

for all  $\theta^r$  in  $T$ , where

$$D = \sup_{\theta \in L_{y_t}} \left( \min_{1 \leq j \leq R} |\theta - \theta^j| \right), \quad (2.16)$$

then

$$d_t^T (\hat{\theta}_{t-1} - \theta) \geq \frac{1}{2}$$

for all  $\theta$  in  $L_{y_t}$ .

**Proof** Choose any  $\theta$  in  $L_{y_t}$ . From Eq. (2.16) it follows that there exists a  $\theta^r$  in  $T$  such that  $|\theta - \theta^r| \leq D$ . Eq. (2.15) then gives

$$d_t^T(\hat{\theta}_{t-1} - \theta^r) - \nu D \geq \frac{1}{2}$$

Since  $|d_t| \leq \nu$  and  $|\theta - \theta^r| \leq D$ , we can rewrite this as

$$d_t^T(\hat{\theta}_{t-1} - \theta^r) - |d_t||\theta - \theta^r| \geq \frac{1}{2}. \quad (2.17)$$

From the triangle inequality it follows that  $|d_t||\theta - \theta^r| \geq d_t^T(\theta - \theta^r)$ . Substituting this into Eq. (2.17) gives

$$d_t^T(\hat{\theta}_{t-1} - \theta) \geq \frac{1}{2}.$$

□

A potential problem with the algorithm in Eqs. (2.4) and (2.5) is the possibility of division by zero when  $d_t$  is zero. This can be avoided by adding a small positive constant,  $\lambda$ , to the denominator of  $k_t$ . Furthermore, for some systems, the convergence rate for the different parameters can differ considerably. One way to solve this problem is to include adaptation gains in the parameter update laws. It is readily verified that these two modifications do not change the stability properties of the algorithm.

We are now ready to present a modified version of the parameter estimation algorithm as follows

$$\hat{\theta}_t = \hat{\theta}_{t-1} - k_t \Gamma d_t,$$

where

$$k_t = \frac{1}{\lambda + d_t^T \Gamma d_t},$$

$\lambda$  is a small positive constant,  $\Gamma$  is a diagonal matrix of positive adaptation gains and

$d_t$  is given by the solution of the quadratic programming problem

$$\begin{aligned} & \text{minimize} && d_t^T d_t \\ & && \end{aligned} \tag{2.18}$$

$$\text{subject to} \quad A_t d_t \geq b_t,$$

where

$$A_t = \begin{bmatrix} \tilde{\theta}_{t-1}^{1T} \\ \vdots \\ \tilde{\theta}_{t-1}^{RT} \end{bmatrix}, \quad \text{and} \quad b_t = \begin{bmatrix} \frac{1}{2} + \nu D \\ \vdots \\ \frac{1}{2} + \nu D \end{bmatrix},$$

$\theta^1, \dots, \theta^R$  are the elements of the set  $T$  as given by Eq. (2.11),  $\nu > 0$  and

$$D = \sup_{\theta \in L_{y_t}} \left( \min_{1 \leq j \leq R} |\theta - \theta^j| \right).$$

$d_t$  is defined to be zero if no feasible solution vector exists or if the solution vector does not satisfy

$$|d_t| \leq \nu.$$

Since we know that  $\theta$  lies in  $\Theta$  it is reasonable to restrict  $\hat{\theta}_t$  to lie in  $\Theta$  as well. This can be done by implementing a parameter projection as discussed in [7]. If this is done,  $V_t$  in Eq. (2.6) will retain its non-increasing property. In fact, the convergence properties of the algorithm will be improved by restricting  $\hat{\theta}_t$  to the compact set in which it is known that  $\theta$  lies. The modified version of Eq. (2.4) for the  $i$ th component would be

$$\hat{\theta}_{i_t} = \begin{cases} \theta_{i_{\min}} & \text{if } \hat{\theta}_{i_{t-1}} - k_t d_{i_t} \leq \theta_{i_{\min}} \\ \theta_{i_{\max}} & \text{if } \hat{\theta}_{i_{t-1}} - k_t d_{i_t} \geq \theta_{i_{\max}} \\ \hat{\theta}_{i_{t-1}} - k_t d_{i_t} & \text{otherwise.} \end{cases} \tag{2.19}$$

If sufficient computational power to solve the optimization problem in Eq. (2.18) at each time instance is not available, then it is possible to let the computation run for a number of sampling periods and set  $d_t$  equal to zero during this time. When the computation has been performed the parameter estimates can be updated and a new set of samples collected. The nonincreasing property of  $V_t$  will be retained,

but the rate of convergence will be slower than if the parameter estimates were updated at each time instance. However, this approach allows the on-line estimation of the parameters with reasonable computational resources while using a large enough sampling rate such that the model remains an accurate description of the system.

## 2.3 Robustness to a Bounded Disturbance

The algorithm described in Section 2.2 was developed based on the assumption of a perfect system model. This assumption rarely holds in practice, because of the inevitable presence of uncertainty due to disturbances, modeling errors and measurement noise. An approach to modeling these error terms is in the form of an additive disturbance signal,  $w_t$ . Eq. (2.1) then takes on the form

$$y_t = f(\phi_{t-1}, \theta) + w_t. \quad (2.20)$$

If the algorithm described in Section 2.2 is used to estimate  $\theta$ , then the presence of  $w_t$  could cause instability. In particular, we can no longer guarantee that  $\Delta V_t < 0$ . The objective of this section is to modify the algorithm such that  $V_t$  retains its nonincreasing property in the presence of  $w_t$ . It is assumed that there exists a known positive constant  $\Omega$  such that

$$\sup |w_t| < \Omega.$$

Furthermore, it is assumed that the parameter projection as given by Eq. (2.19) has been implemented such that  $\hat{\theta} \in \Theta$ . In Section 2.2 we made use of the fact that if the true system is given by

$$y_t = f(\phi_{t-1}, \theta),$$

then it is known that  $\theta$  lies in the level set

$$L_{y_t} = \{\theta \in \Theta \mid f(\phi_{t-1}, \theta) = y_t\}.$$

This was illustrated in Figure 2.1. However, when  $w_t$  is present as in Eq. (2.20),  $\theta$  is no longer restricted to  $L_{y_t}$ , but can lie anywhere in the compact set  $\Theta_w$  defined by

$$\Theta_w = \{\theta \in \Theta \mid f(\phi_{t-1}, \theta) = y_t + w, |w| < \Omega\}.$$

This is illustrated in Figure 2.3. It is clear that when  $|\tilde{y}_t| < \Omega$ , it is not possible to find a  $d_t$  which guarantees that  $\Delta V_t < 0$  since  $\hat{\theta}_{t-1}$  then lies in  $\Theta_w$  and we could have  $\hat{\theta}_{t-1} = \theta$ . Therefore we set  $d_t = 0$  when  $|\tilde{y}_t| < \Omega$ .

Given  $y_t$  and  $\phi_{t-1}$ ,  $\theta$  could lie anywhere in  $\Theta_w$ . Therefore, one way to robustify the algorithm of Section 2.2 in the presence of  $w_t$  is to let  $T$  in Eq. (2.11) be made up of samples of the entire set  $\Theta_w$  instead of samples of  $L_{y_t}$ .  $\Theta_w$  is of a larger dimension than  $L_{y_t}$  and, as a result,  $T$  would be considerably larger. The following theorem shows that in order to derive a stable update law, it is only necessary that  $T$  consists of samples of the the level set

$$L_\Omega = \{\theta \in \Theta \mid f(\phi_{t-1}, \theta) = y_t + \text{sgn}(\tilde{y}_t)\Omega\}.$$

Thus the size of  $T$  need not be larger than in the case where no disturbance is present.

**Theorem 2** For the system in Eq. (2.20), if  $|\tilde{y}_t| > \Omega$  and if a  $d_t$  in Eq (2.4) is such that  $\Delta V_t < 0$  for for all  $\theta$  in  $L_\Omega$ , then  $\Delta V_t < 0$  for all  $\theta \in \Theta_w$ .

**Proof** Choose any  $\theta = \theta_w \in \Theta_w$ , then  $\theta_w$  lies on the level set

$$L_w = \{\theta \in \Theta \mid f(\phi_{t-1}, \theta) = y_t + w\},$$

for some  $w$ ,  $|w| < \Omega$ .  $\hat{\theta}_{t-1}$  lies on the level set

$$L_{\hat{y}_t} = \{\theta \in \Theta \mid f(\phi_{t-1}, \theta) = \hat{y}_t\},$$

where  $\hat{y}_t = f(\phi_{t-1}, \hat{\theta}_{t-1})$ . Since  $|w| < \Omega$  and  $|\hat{y}_t - y_t| > \Omega$ ,

$$\text{sgn}(\tilde{y}_t)(y_t + w) < \text{sgn}(\tilde{y}_t)y_t + \Omega < \text{sgn}(\tilde{y}_t)\hat{y}_t,$$

$L_\Omega$  lies between  $L_w$  and  $L_{\hat{y}_t}$ . This combined with the convexity of  $\Theta$  ensures that a line segment between  $\theta_w$  on  $L_w$  and  $\hat{\theta}_{t-1}$  on  $L_{\hat{y}_t}$  must intersect  $L_\Omega$  in at least one point. Let  $\theta_\Omega$  be one such point. Then

$$\hat{\theta}_{t-1} - \theta_\Omega = c(\hat{\theta}_{t-1} - \theta_w, ) \tag{2.21}$$

for some  $c < 1$ . In Theorem 1 it was established that if  $\theta = \theta_w$ , then  $\Delta V_t < 0$  if and only if

$$d_t^T(\hat{\theta}_{t-1} - \theta_\Omega) > \frac{1}{2}.$$

By substituting for  $(\hat{\theta}_{t-1} - \theta_\Omega)$  from Eq (2.21) we have

$$d_t^T(\hat{\theta}_{t-1} - \theta_w) > \frac{1}{2c} > \frac{1}{2},$$

and thus  $\Delta V_t < 0$  for an arbitrary  $\theta_w$  in  $\Theta_w$ . □

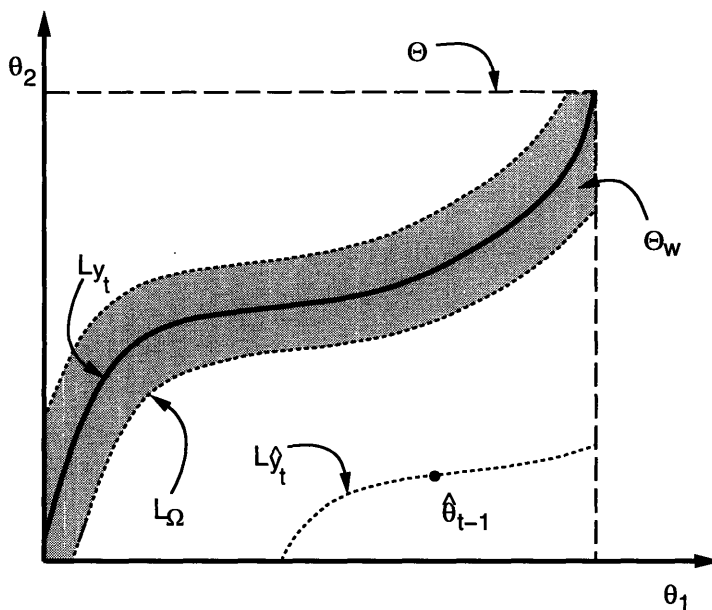


Figure 2-2: Graphical illustration of Theorem 2.

A similar technique to that described in Section 2.2 can be used to discretize  $L_\Omega$  into  $R$  approximately equally spaced points. We can then form  $A_t$  and  $b_t$  as in

Eq. (2.12) and obtain  $d_t$  by solving the quadratic programming problem in Eq. (2.18). If a solution exists, then we are guaranteed that the resulting  $d_t$  will cause  $V_t$  to decrease.

## 2.4 Using a Neural System Model

A large class of nonlinear systems can be represented by the model

$$\dot{x}_t = f_c(x_t, u_t, \theta) \tag{2.22}$$

$$y_t = h_c(x_t, u_t, \theta)$$

in continuous-time or

$$x_t = f_d(x_{t-1}, u_{t-1}, \theta) \tag{2.23}$$

$$y_t = h_d(x_{t-1}, u_{t-1}, \theta)$$

in discrete-time, where  $u_t$  is the input,  $x_t$  is the system state,  $y_t$  is the output and  $\theta$  is a vector of system parameters. The parameter estimation algorithm developed in the previous section requires the system model to be in the regression form

$$y_t = f(\phi_{t-1}, \theta), \tag{2.24}$$

such that  $y_t$  can be evaluated based on the parameter vector  $\theta$  and  $\phi_{t-1}$ , which represents a vector of past values of the measurable system signals. The models given by Eqs. (2.22) and (2.23) in general do not satisfy this condition if the state variables are not accessible. However, under some standard assumptions on the system functions and the input, the models given by Eqs. (2.22) and (2.23) can be transformed into the form of Eq. (2.24) [2]. If this transformation can be found analytically, the algorithm outlined in the previous section is applicable. In practice, this is often difficult. A neural model can then be used to generate  $y_t$  for desired values of  $\phi_{t-1}$  and  $\theta$ . This procedure is also applicable if a simulation model of the system is available off-line, or if the parameters of the system can be determined in controlled experiments on

the real system or a pilot plant, using extra sensors and measuring equipment. The off-line model or the experimental data can then be used to train the neural system model. This scenario is feasible in the aircraft industry, where tables of aerodynamic parameters and stability and control derivatives are created from wind tunnel experiments and flight tests using physically based aircraft models and calculations based on the structure and shape of the aircraft [16]. The input-output structure of the neural network is depicted graphically in Figure 2.4.

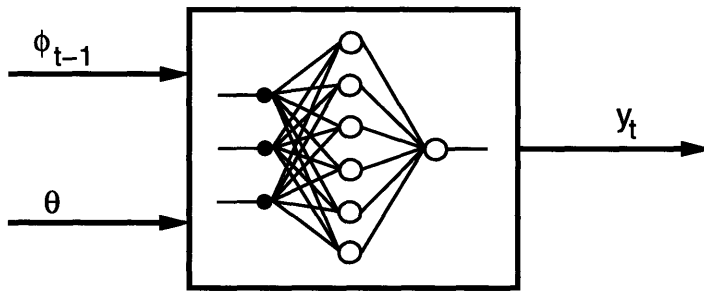


Figure 2-3: Input-output structure of the neural system model.

For the training of the neural network it is assumed that  $f$  is sufficiently smooth and that  $\phi_{t-1} \in \Phi \subset \mathbf{R}^p$ ,  $y_t \in \Psi \subset \mathbf{R}$  and  $\theta \in \Theta \subset \mathbf{R}^n$ , where  $\Phi$  and  $\Psi$  are known compact sets and  $\Theta$  is a  $n$  dimensional box. It will also be assumed that the neural network satisfies the property of a “universal approximator” [8].

In order to train the neural network it is necessary to form a training set. This can be done as follows. Set  $\theta = \theta^1 \in \Theta$  and measure the corresponding  $\phi$  and  $y$  for a number of samples and variations in the system input. Let these measurements be denoted  $\phi_1, \dots, \phi_p$  and  $y_1^1, \dots, y_p^1$  respectively. A typical set of data can then be formed as

$$T_1 = \{(y_i^1, \phi_i, \theta^1) \mid 1 \leq i \leq p\}$$

By repeating this procedure for other values of  $\theta$  in  $\Theta$  the data sets  $T_2, \dots, T_q$  can be formed corresponding to  $\theta^2, \dots, \theta^q$  respectively. The complete training set is then



given by

$$T_{train} = \bigcup_{1 \leq j \leq q} T_j.$$

The neural network can now be trained using  $\phi_i$  and  $\theta^j$  as inputs and  $y_i^j$  as the corresponding target. If a multi-layered neural network is used, a testing set,  $T_{test}$ , should also be formed for cross validation during training.

Because the neural network can only approximate the regression function  $f$ , there will be an approximation error associated with the neural model. As long as an upper bound on the magnitude of this error is available, the modified algorithm presented in the previous section can be used ensure the stability of the estimator by treating the approximation error as a bounded disturbance.

# Chapter 3

## Partially Linear Systems

### 3.1 Introduction

In this section, we consider the dynamic system

$$y_t = \sigma f(\phi_{t-1}, \theta) + \varphi_{t-1}^T \alpha, \quad (3.1)$$

where  $y_t$  is the output of the system at time  $t$ ,  $\phi_{t-1}$  and  $\varphi_{t-1}$  represent past values of measurable inputs and the output.  $\sigma$  is an unknown scalar and  $\theta$  and  $\alpha$  are vectors of unknown system parameters in  $\mathbf{R}^n$  and  $\mathbf{R}^m$  respectively. It is assumed that  $f(\phi_{t-1}, \theta)$  is sufficiently smooth and that  $\theta$  lies in the known, compact set  $\Theta$ . Furthermore, the sign of  $\sigma$  and an upper bound on its magnitude,  $\sigma_{\max}$ , must be known. Without loss of generality,  $\sigma$  will be assumed to be positive. The task is to estimate  $\sigma$ ,  $\theta$  and  $\alpha$ . This can be done by direct application of the algorithm presented in the previous section. However, this would result in a quadratic programming problem of dimension  $n + m + 1$  with  $n + m$  constraints. Instead, it would be desirable to exploit the fact that the subsystem  $\varphi_{t-1}^T \alpha$  is linear and that  $\sigma$  multiplies  $f(\phi_{t-1}, \theta)$  to reduce the complexity of the optimization problem.

## 3.2 The Algorithm

The prediction error for the system in Eq. (3.1) is given by

$$\tilde{y}_t = \hat{\sigma}_{t-1} \hat{f}_{t-1} - \sigma f_{t-1} + \varphi_{t-1}^T \tilde{\alpha}_{t-1},$$

where  $f_{t-1} = f(\phi_{t-1}, \theta)$  and  $\hat{f}_{t-1} = f(\phi_{t-1}, \hat{\theta}_{t-1})$ . After some algebraic manipulation  $\tilde{y}_t$  can be expressed as

$$\tilde{y}_t = \tilde{\sigma}_{t-1} \hat{f}_{t-1} + \sigma(\hat{f}_{t-1} - f_{t-1}) + \varphi_{t-1}^T \tilde{\alpha}_{t-1}. \quad (3.2)$$

Based on this error model the following parameter estimation algorithm is proposed

$$\hat{\sigma}_t = \hat{\sigma}_{t-1} - k_t \tilde{y}_t p \gamma_\sigma \hat{f}_{t-1} \quad (3.3)$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} - k_t \tilde{y}_t p \Gamma_\theta d_t \quad (3.4)$$

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} - k_t \tilde{y}_t p \Gamma_\alpha \varphi_{t-1} \quad (3.5)$$

where

$$k_t = \frac{1}{\lambda + \gamma_\sigma \hat{f}_{t-1}^2 + \sigma_m d_t^T \Gamma_\theta d_t + \varphi_{t-1}^T \Gamma_\alpha \varphi_{t-1}}, \quad (3.6)$$

$\gamma_\sigma$  is a positive adaptation gain and  $\Gamma_\theta$  and  $\Gamma_\alpha$  are diagonal matrices of positive adaptation gains.  $\lambda$  is a small positive constant and  $d_t$  is a vector to be determined.  $p$  is a scalar supplied by the user. It is bounded by  $0 < p < 1$  and will be discussed in more detail later.

As in the previous section we define a positive definite function of the parameter errors

$$V_t = \gamma_\sigma^{-1} \tilde{\sigma}^2 + \sigma \tilde{\theta}_t^T \Gamma_\theta^{-1} \tilde{\theta}_t + \tilde{\alpha}_t^T \Gamma_\alpha^{-1} \tilde{\alpha}_t.$$

Using the identity

$$\begin{aligned} \Delta V_t &= 2(\Delta \gamma_\sigma^{-1} \tilde{\sigma}_t \tilde{\sigma}_{t-1} + \sigma \Delta \tilde{\theta}_t^T \Gamma_\theta^{-1} \tilde{\theta}_{t-1} + \Delta \tilde{\alpha}_t^T \Gamma_\alpha^{-1} \tilde{\alpha}_{t-1}) \\ &+ (\gamma_\sigma^{-1} \Delta \tilde{\sigma}_t)^2 + \sigma \Delta \tilde{\theta}_t^T \Gamma_\theta^{-1} \Delta \tilde{\theta}_t + \Delta \tilde{\alpha}_t^T \Gamma_\alpha^{-1} \Delta \tilde{\alpha}_t \end{aligned}$$

and the fact that  $k_t(\gamma_\sigma^{-1}\hat{f}_{t-1}^2 + \sigma d_t^T \Gamma_\theta^{-1} d_t + \varphi_t^T \Gamma_\alpha^{-1} \varphi_t) < 1$ , the change in  $V_t$  can be bounded by

$$\Delta V_t \leq k_t p^2 (-2\tilde{y}_t p^{-1} [\hat{f}_{t-1} \tilde{\sigma}_{t-1} + \sigma d_t^T \tilde{\theta}_{t-1} + \varphi_t^T \tilde{\alpha}_{t-1}] + \tilde{y}_t^2).$$

Thus if there exists a vector  $d_t$  which satisfies the inequality

$$\text{sgn}(\tilde{y}_t) (\hat{f}_{t-1} \tilde{\sigma} + \sigma d_t^T \tilde{\theta}_{t-1} + \varphi_t^T \tilde{\alpha}_{t-1}) \geq p |\tilde{y}_t|, \quad (3.7)$$

then  $\Delta V_t \leq -k_t p^2 \tilde{y}_t \leq 0$ . Eq. (3.2) can be rewritten as

$$\hat{f}_{t-1} \tilde{\sigma} + \varphi_{t-1}^T \tilde{\alpha}_{t-1} = \tilde{y}_t - \sigma (\hat{f}_{t-1} - f_{t-1}),$$

Using this in Eq. (3.7) gives

$$\text{sgn}(\tilde{y}_t) d_t^T \tilde{\theta}_{t-1} \geq \text{sgn}(\tilde{y}_t) (\hat{f}_{t-1} - f_{t-1}) - \frac{1-p}{\sigma_{\max}} |\tilde{y}_t|. \quad (3.8)$$

Since we cannot solve for  $d_t$  analytically, we will again develop a numerical procedure for determining  $d_t$ . In the previous section, Eq. (2.1) restricted  $\theta$  to lie on the level set  $L_{y_t}$ . In Eq. (3.1), since  $\sigma$  and  $\alpha$  are unknown,  $\theta$  can lie anywhere in  $\Theta$ . Thus we now have  $n$  degrees of freedom in choosing the  $n$  components of  $\theta$ . The set  $T$  can therefore be formed as follows. Divide each component of  $\theta$ ,  $\theta_i$ , into  $q$  equally spaced points between  $\theta_{i_{\min}}$  and  $\theta_{i_{\max}}$  to obtain  $\theta_i^1, \dots, \theta_i^q$ . Use these to form all possible combinations of the vector  $\theta$  and denote these as  $\theta^1, \dots, \theta^R$ , with  $R = q^n$ . Then  $T$  is defined by

$$T = \bigcup_{1 \leq r \leq R} \theta^r.$$

Eq. (3.8) has to be satisfied for all  $\theta^r$  in  $T$ . Note that as in the previous section, we are somewhat conservative, since for stability we only need  $\frac{p}{2} |\tilde{y}_t|$  on the right hand side of Eq. (3.7). This is to ensure that  $\Delta V_t \leq 0$  for those  $\theta$  in  $\Theta$  but not in  $T$ .

Let  $\tilde{\theta}_{t-1}^r = \hat{\theta}_{t-1} - \theta^r$ . If we define the vector

$$a_{t,r} = \text{sgn}(\tilde{y}_t)\tilde{\theta}_{t-1}^r \quad (3.9)$$

and the scalar

$$b_{t,r} = \text{sgn}(\tilde{y}_t)(\hat{f}_{t-1} - f_{t-1}^r) - (1-p)|\tilde{y}_t|$$

and let

$$A_t = \begin{bmatrix} a_{t_1}^T \\ \vdots \\ a_{t_R}^T \end{bmatrix}, \quad b_t = \begin{bmatrix} b_{t_1} \\ \vdots \\ b_{t_R} \end{bmatrix}, \quad (3.10)$$

then the smallest  $d_t$  which satisfies the constraints given by Eq. (3.8) for all  $\theta_{t-1}^r$  can be found by solving the quadratic programming problem.

$$\begin{aligned} & \text{minimize} \quad d_t^T d_t \\ & \text{subject to} \quad A_t d_t \geq b_t. \end{aligned} \quad (3.11)$$

Thus we have reduced a  $n + m + 1$  dimensional optimization problem with  $q^{n+m}$  constraints to a  $n$  dimensional problem with  $q^n$  constraints. Clearly, if  $m$  is large the saving in computation time will be dramatic.

The role of  $p$  is clear from Eq. (3.8). If  $p = 1$ , then Eq. (3.8) reduces to  $\text{sgn}(\tilde{y}_t)d_t^T \tilde{\theta}_{t-1} \geq \text{sgn}(\tilde{y}_t)(\hat{f}_{t-1} - f_{t-1})$ . Since  $\theta$  is a linear function of  $d_t^T \tilde{\theta}_{t-1}$  and a nonlinear function of  $\hat{f}_{t-1} - f_{t-1}$ , in general, there exists no  $d_t$  which satisfies this inequality for all  $\theta$  in  $\Theta$ . However, as  $p$  decreases, the constraints become less stringent and a  $d_t$  might exist. However, since the bound on the change in  $V_t$  for all  $\theta = \theta^r$ ,  $r = 1, \dots, R$  is  $\Delta V_t \leq -k_t p^2 \tilde{y}_t^2$ , the smaller  $p$  is, the smaller the decrease in  $V_t$  will be. Thus there exists a tradeoff between how often  $V_t$  is decreased and by how much is decreased each time.

If the model under consideration does not have a parameter which appears multiplicatively like  $\sigma$  or a linear subsystem like  $\varphi_{t-1}^T \alpha$ , then the algorithm can easily be modified to accommodate for this. If both  $\sigma$  and  $\varphi_{t-1}^T \alpha$  are removed the algorithm

reduces to that presented in the chapter 2. However, as long as at least one parameter enters the regression model in the form of  $\sigma$  or  $\alpha$ , we do not need to or find points on the level surface defined Eq. (2.1), since  $\theta$  can be anywhere in  $\Theta$ . In general, it is advisable to exploit any type of special structure of the system model to simplify the identification problem. Additional examples are presented in section 3.3 and chapter 4.

### 3.3 When $\theta$ is a Scalar

This section is concerned with the case where  $\theta$  in Eq. (3.1) is a scalar. For simplicity,  $\sigma$  will be assumed to be known, but the algorithm can easily be modified if this is not the case. Eq. (3.1) then reduces to

$$y_t = f(\phi_{t-1}, \theta) + \varphi_{t-1}^T \alpha.$$

Again it is assumed that  $f$  is sufficiently smooth and that  $\theta$  lies in the known, compact set  $\Theta$  defined by  $\theta_{\min} \leq \theta \leq \theta_{\max}$ . For this system, Eqs. (3.3) through (3.6) reduce to

$$\begin{aligned} \hat{\theta}_t &= \hat{\theta}_{t-1} - k_t \tilde{y}_t p d_t \\ \hat{\alpha}_t &= \hat{\alpha}_{t-1} - k_t \tilde{y}_t p \varphi_{t-1} \end{aligned}$$

where

$$k_t = \frac{1}{\lambda + d_t^2 + \varphi_{t-1}^T \varphi_{t-1}}.$$

$p$  plays the same role as in the previous section and  $d_t$  should be chosen so as to ensure the stability of the algorithm. This is discussed below.

Let

$$V_t = \tilde{\theta}_t^2 + \tilde{\alpha}_t^T \tilde{\alpha}_t.$$

Using a similar procedure as in the previous section the following condition for the

stability of the algorithm can be obtained

$$\text{sgn}(\tilde{y}_t)d_t\tilde{\theta}_{t-1} \geq \text{sgn}(\tilde{y}_t)(\hat{f}_{t-1} - f_{t-1}) - (1-p)|\tilde{y}_t|,$$

which is equivalent to

$$\text{sgn}(\tilde{y}_t\tilde{\theta}_{t-1})d_t \geq \frac{1}{|\tilde{\theta}_{t-1}|}[\text{sgn}(\tilde{y}_t)(\hat{f}_{t-1} - f_{t-1}) - (1-p)|\tilde{y}_t|]. \quad (3.12)$$

Let  $T$  be the set  $\theta^1, \dots, \theta^q$  obtained by dividing  $\Theta$  into  $q$  equally spaced points between  $\theta_{\min}$  and  $\theta_{\max}$  such that  $\theta^r \neq \hat{\theta}_{t-1}$  for  $r = 1, \dots, q$ . Let  $T^+$  be the subset of  $T$  consisting of all  $\theta^r$  such that  $\tilde{y}_t\tilde{\theta}_{t-1} > 0$  and let  $T^-$  be the subset consisting of those  $\theta^r$  satisfying  $\tilde{y}_t\tilde{\theta}_{t-1} < 0$ . Furthermore, let

$$d_{\min} = \max_{\theta^r \in T^+} \left( \frac{1}{\tilde{\theta}_{t-1}^r} [(\hat{f}_{t-1} - f_{t-1}) - (1-p)\tilde{y}_t] \right)$$

and

$$d_{\max} = \min_{\theta^r \in T^-} \left( \frac{1}{\tilde{\theta}_{t-1}^r} [(\hat{f}_{t-1} - f_{t-1}) - (1-p)\tilde{y}_t] \right),$$

then Eq. (3.12) can be compactly expressed as

$$d_{\min} \leq d \leq d_{\max}.$$

If  $d_{\min} > d_{\max}$ , then the problem is overconstrained and we set  $d_t = 0$  to ensure stability. Otherwise, the  $d_t$  which minimizes  $d_t^2$  is given by

$$d = \begin{cases} d_{\min} & \text{if } |d_{\min}| \leq |d_{\max}| \\ d_{\max} & \text{otherwise.} \end{cases}$$

### 3.4 Robustness to a Bounded Disturbance

Consider the regression equation

$$y_t = \sigma f(\phi_{t-1}, \theta) + \varphi_{t-1}^T \alpha + w_t,$$

where  $w_t$  is a disturbance bounded by  $\Omega$  as before. It is assumed that  $\sigma_{\min} \leq \sigma \leq \sigma_{\max}$ , where  $\sigma_{\min}$  and  $\sigma_{\max}$  are known positive constants. The update laws for the parameter estimates are given by Eqs. (3.3) through (3.5) as before. However, the prediction error is now given by

$$\tilde{y}_t = \tilde{\sigma}_{t-1} \hat{f}_{t-1} + \sigma(\hat{f}_{t-1} - f_{t-1}) + \varphi_{t-1}^T \tilde{\alpha}_{t-1} - w_t.$$

Thus the condition for the stability of the algorithm becomes

$$\text{sgn}(\tilde{y}_t) \sigma d_t^T \tilde{\theta}_{t-1} \geq \text{sgn}(\tilde{y}_t) [\sigma(\hat{f}_{t-1} - f_{t-1}) - w_t] - (1-p)|\tilde{y}_t|,$$

which can be expressed as

$$\text{sgn}(\tilde{y}_t) d_t^T \tilde{\theta}_{t-1} \geq \text{sgn}(\tilde{y}_t) (\hat{f}_{t-1} - f_{t-1}) + \frac{(1-p)|\tilde{y}_t| + \text{sgn}(\tilde{y}_t)\Omega}{\sigma^*},$$

where

$$\sigma^* = \begin{cases} \sigma_{\min} & \text{if } (1-p)|\tilde{y}_t| + \text{sgn}(\tilde{y}_t)\Omega \geq 0 \\ \sigma_{\max} & \text{otherwise.} \end{cases}$$

If we let  $a_{t_r}$  be defined as in Eq. (3.9) and let

$$b_{t_r} = \text{sgn}(\tilde{y}_t) (\hat{f}_{t-1} - f_{t-1}) + \frac{(1-p)|\tilde{y}_t| + \text{sgn}(\tilde{y}_t)\Omega}{\sigma^*},$$

with  $A_t$  and  $b_t$  as defined by Eq. (3.10), then  $d_t$  can be obtained by solving the optimization problem given by Eq. (3.11). If this problem has a feasible solution, then the resulting  $d_t$  is guaranteed make  $\Delta V_t$  negative. Thus algorithm can be modified to ensure stability in the presence a bounded disturbance.



# Chapter 4

## A Continuous-time Algorithm

### 4.1 Introduction

In the algorithms developed for discrete-time regression models in the previous sections,  $\Delta\hat{\theta}_t$  is obtained at each sample time by the solution of a quadratic programming problem. In general it is difficult to extend these algorithms to continuous-time since in this case  $\dot{\hat{\theta}}_t$  must be available at all time and not just at discrete instances. However, in Section 3.3 it was shown that when  $\theta$  is a scalar, the task of finding a parameter update law is simplified substantially. In Section 4.2 this algorithm is extended to the continuous-time case. It is shown that the task of finding a parameter update law can be formulated as a linear programming problem to be solved on line. To simplify the computational burden a modified algorithm is presented where the parameter update law is found by solving a one-dimensional least squares problem. In [14] it was shown how the convexity of  $f$  can be utilized to derive an adaptive controller capable of stabilizing a nonlinear system. This controller is however not capable of tracking a reference input or identifying the parameters with a general input signal. In Section 4.3 it is shown that if all parameters except one occur linearly, and the regression function  $f$  is convex or concave as a function of the nonlinear parameter, then the parameter update laws can be derived analytically.

## 4.2 The General Case

### 4.2.1 The Algorithm

Consider the system

$$\dot{y}_t = \sigma f(\phi_t, \theta) + \varphi_t^T \alpha, \quad (4.1)$$

where  $y_t$  is system state and  $\phi_{t-1}$  and  $\varphi_{t-1}$  represent known functions of the  $y_t$  and the inputs.  $\sigma$  and  $\theta$  are unknown scalars and  $\alpha$  is a vector of unknown system parameters in  $\mathbf{R}^m$ . It is assumed that  $f$  is sufficiently smooth and that  $\theta$  lies in the known, compact set  $\Theta$  defined by  $\theta_{\min} \leq \theta \leq \theta_{\max}$ . Furthermore, the sign of  $\sigma$  and an upper bound on its magnitude,  $\sigma_{\max}$ , are known. Without loss of generality,  $\sigma$  will be assumed to be positive. The task is to estimate  $\sigma$ ,  $\theta$  and  $\alpha$ .

Consider the estimator model

$$\dot{\hat{y}}_t = -a_0 \tilde{y}_{\epsilon t} - a_t \text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) + \hat{\sigma}_t f(\phi_t, \hat{\theta}_t) + \varphi_t^T \hat{\alpha}_t,$$

where  $\tilde{y}_t = \hat{y}_t - y_t$ ,

$$\tilde{y}_{\epsilon t} = \tilde{y}_t - \epsilon \text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) \quad (4.2)$$

and  $\epsilon$  is a small positive constant. The relationship between  $\tilde{y}_t$  and  $\tilde{y}_{\epsilon t}$  is shown graphically in Figure 4.2.1.  $a_t$  is a positive function to be determined. The dynamics of the prediction error is given by

$$\dot{\tilde{y}}_t = -a_0 \tilde{y}_{\epsilon t} - a_t \text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) + \hat{\sigma}_t \hat{f}_t - \sigma f_t + \varphi_t^T \tilde{\alpha}_t, \quad (4.3)$$

where  $f_t = f(\phi_t, \theta)$  and  $\hat{f}_t = f(\phi_t, \hat{\theta})$ . After some algebraic manipulation, Eq. (4.3) can be written as

$$\dot{\tilde{y}}_t = -a_0 \tilde{y}_{\epsilon t} - a_t \text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) + \tilde{\sigma}_t \hat{f}_t + \sigma(\hat{f}_t - f_t) + \varphi_t^T \tilde{\alpha}_t.$$

Based on this expression for  $\dot{\tilde{y}}_t$  the following estimator structure is proposed

$$\begin{aligned}\dot{\hat{\sigma}}_t &= -\tilde{y}_{\epsilon t} \gamma_{\sigma} \hat{f}_t \\ \dot{\hat{\theta}}_t &= -\tilde{y}_{\epsilon t} \gamma_{\theta} d_t \\ \dot{\hat{\alpha}}_t &= -\tilde{y}_{\epsilon t} \Gamma_{\alpha} \varphi_t\end{aligned}$$

where  $\gamma_{\sigma} > 0$ ,  $\gamma_{\theta} > 0$  are positive adaptation gains,  $\Gamma_{\alpha}$  is a diagonal matrix of positive adaptation gains and  $d_t$  is a scalar to be determined.

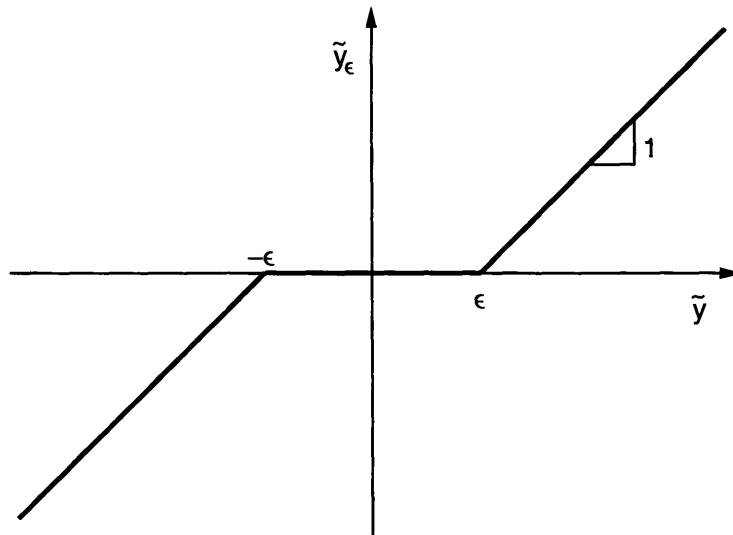


Figure 4-1: Relationship between  $\tilde{y}_t$  and  $\tilde{y}_{\epsilon t}$ .

Let  $V_t$  be the positive definite function of  $\tilde{y}_{\epsilon t}$  and the estimation errors given by

$$V_t = \frac{1}{2}(\tilde{y}_{\epsilon t}^2 + \gamma_{\sigma}^{-1} \tilde{\sigma}_t^2 + \gamma_{\theta}^{-1} \sigma \tilde{\theta}_t^2 + \tilde{\alpha}_t^T \Gamma_{\alpha}^{-1} \tilde{\alpha}_t)$$

We pause here to make some comments regarding the estimator structure and the choice of  $V_t$ . The form of the estimator and the adaptive laws resemble that used in [12] in linear adaptive control with the addition of the term  $a_t \text{sat}(\frac{\tilde{y}_t}{\epsilon})$  where  $\epsilon$  is a small positive constant. This term basically allows us to make the dynamics

of the estimator arbitrarily fast. The modification of  $\tilde{y}_t$  given by Eq. (4.2) closely resembles the modification of the sliding mode error term  $s_t$  used in the adaptive implementations of this algorithm [20]. As noted in [21], although the derivative of  $\tilde{y}_{\epsilon_t}$  is discontinuous, the derivative of  $\tilde{y}_{\epsilon_t}^2$  is continuous with

$$\frac{1}{2} \frac{d}{dt} \tilde{y}_{\epsilon_t}^2 = \tilde{y}_{\epsilon_t} \dot{\tilde{y}}_t.$$

In continuous-time the requirement that  $\Delta V_t$  be less than or equal to zero is replaced by the requirement that  $\dot{V}_t$  be nonpositive.  $\dot{V}_t$  is given by

$$\begin{aligned} \dot{V}_t &= \dot{\tilde{y}}_t \tilde{y}_{\epsilon_t} + \gamma_\sigma^{-1} \dot{\tilde{\sigma}}_t \tilde{\sigma}_t + \gamma_\theta^{-1} \sigma \dot{\tilde{\theta}}_t \tilde{\theta}_t + \dot{\tilde{\alpha}}_t \Gamma_\alpha^{-1} \tilde{\sigma}_t \\ &= -a_0 \tilde{y}_{\epsilon_t}^2 + \tilde{y}_{\epsilon_t} [-a_t \text{sat}(\frac{\tilde{y}_t}{\epsilon}) + \tilde{\sigma} \hat{f}_t + \sigma(\hat{f}_t - f_t) + \varphi_t^T \tilde{\alpha}] \\ &\quad + \gamma_\sigma^{-1} \dot{\tilde{\sigma}}_t \tilde{\sigma}_t + \gamma_\theta^{-1} \sigma \dot{\tilde{\theta}}_t \tilde{\theta}_t + \dot{\tilde{\alpha}}_t \Gamma_\alpha^{-1} \tilde{\sigma}_t \\ &= -a_0 \tilde{y}_{\epsilon_t}^2 + \tilde{y}_{\epsilon_t} [-a_t \text{sat}(\frac{\tilde{y}_t}{\epsilon}) + \sigma(\hat{f}_t - f_t - d_t \tilde{\theta}_t)]. \end{aligned}$$

Thus  $\dot{V}_t = 0$  if  $\tilde{y}_t \leq \epsilon$ , and

$$\dot{V}_t = -a_0 \tilde{y}_{\epsilon_t}^2 + \tilde{y}_{\epsilon_t} [-a_t + \sigma(\hat{f}_t - f_t - d_t \tilde{\theta}_t)]$$

when  $\tilde{y}_t > \epsilon$ . It follows that when  $\tilde{y}_t > \epsilon$ ,  $\dot{V}_t \leq 0$  if

$$\text{sgn}(\tilde{y}_{\epsilon_t}) \sigma d_t \tilde{\theta}_t \geq \text{sgn}(\tilde{y}_{\epsilon_t}) \sigma(\hat{f}_t - f_t) - a_t,$$

or, since  $\sigma \leq \sigma_{\max}$ ,  $\dot{V}_t < 0$  if

$$\text{sgn}(\tilde{y}_{\epsilon_t}) \sigma_{\max} d_t \tilde{\theta}_t + a_t \geq \text{sgn}(\tilde{y}_{\epsilon_t}) \sigma_{\max} (\hat{f}_t - f_t). \quad (4.4)$$

Let  $T$  be the set of  $q$  equally spaced points in  $\Theta$ ,  $\theta^1, \dots, \theta^q$ . Then Eq. (4.4) must be satisfied for each  $\theta^r$  in  $T$ . This leads to the following set of linear inequalities

$$A_t x_t \geq b_t, \quad (4.5)$$

where

$$x_t = \begin{bmatrix} d_t \\ a_t \end{bmatrix},$$

and

$$A_t = \begin{bmatrix} \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}\tilde{\theta}_t^1 & 1 \\ \vdots & \vdots \\ \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}\tilde{\theta}_t^q & 1 \end{bmatrix}, \quad \text{and} \quad b_t = \begin{bmatrix} \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}(\hat{f}_t - f_t^1) \\ \vdots \\ \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}(\hat{f}_t - f_t^1) \end{bmatrix},$$

In general there exists an infinite number of solution vectors which satisfy Eq. (4.5).  $a_t$  is a measure of the time constant of the estimator. In general, it is not desirable to have a very fast estimator since the parameter adaptation is driven by  $\tilde{y}_{\epsilon_t}$ . Thus it is desirable to choose the solution vector which minimizes  $a_t$ . This leads to the following linear programming problem

$$\text{minimize } a_t$$

$$\text{subject to } A_t x_t \geq b_t.$$

If it is computationally too demanding to solve this problem on-line at each time instance, then a suboptimal solution can be found by solving a one-dimensional least squares problem. To do this, set  $a_t$  equal to zero in Eq. (4.5) and equate the two sides. Let

$$h_t = \begin{bmatrix} \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}\tilde{\theta}_t^1 \\ \vdots \\ \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}\tilde{\theta}_t^q \end{bmatrix} \quad \text{and} \quad b_t = \begin{bmatrix} \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}(\hat{f}_t - f_t^1) \\ \vdots \\ \text{sgn}(\tilde{y}_{\epsilon_t})\sigma_{\max}(\hat{f}_t - f_t^1) \end{bmatrix}.$$

We then wish to find  $d_t$  such that

$$h_t d_t = b_t.$$

This is an overconstrained problem. The least squares solution is given by

$$d_t = \frac{h_t^T b_t}{h_t^T h_t}.$$

$a_t$  is then chosen as

$$a_t = \sigma_{\max} \max[\text{sgn}(\tilde{y}_{\epsilon_t})(b_t - h_t d_t)]$$

to ensure that Eq. (4.5) holds for all  $\theta$  in  $T$ . For both approaches it is necessary to add a positive constant to  $a_t$  to ensure that Eq. (4.5) is satisfied for those  $\theta$  in  $L_{y_t}$  but not in  $T$ .

## 4.2.2 Application to a Magnetic Bearing System <sup>1</sup>

In this section the algorithm is applied to a magnetic bearing system. We consider the thrust bearing in a turbo pump used to create a vacuum environment for semiconductor manufacturing. The relationship between the vertical displacement  $z$  and the control current  $u$  is given by [23]

$$\ddot{z}_t = K \frac{(i_0 + 0.5u_t)^2}{(h_0 - z_t)^2} - K \frac{(i_0 - 0.5u_t)^2}{(h_0 + z_t)^2} - g, \quad (4.6)$$

where

$$K = \frac{n^2 \mu_0 A}{4m}.$$

The numerical values of the parameters are given by

$$\begin{aligned} h_0 &= 4.0 \times 10^{-4} \text{ m} && \text{(nominal bearing gap)} \\ n &= 133 \text{ turns} && \text{(number of turns)} \\ A &= 7.0 \times 10^{-4} \text{ m}^2 && \text{(pole face area)} \\ i_0 &= 0.5 \text{ Amps} && \text{(bias current)} \\ m &= 2.2 \text{ kg} && \text{(rotor mass)} \end{aligned}$$

---

<sup>1</sup>This example was prepared in collaboration with Dr. Ssu-Hsin Yu, Postdoctoral Fellow in the Adaptive Control Laboratory at MIT.

$$\mu_0 = 1.26 \times 10^{-6} \text{ H/m} \quad (\text{permeability of air})$$

The magnetic bearing system is open loop unstable. Therefore, a controller must be used to stabilize it to allow for the parameters to be estimated. By multiplying out Eq. (4.6) and completing squares the following feedback linearizing controller can be derived

$$u_t = \begin{cases} h_0^2(2Ki_0)^{-1}(g - c_2\dot{z}) & \text{if } z = 0 \\ (h_0z_t)^{-1}\sqrt{i_0^2(h_0^2 - z_t^0)^2 + h_0z_t(h_0^2 - z_t^0)^2(h_0^2 + z_t^0)^2}\frac{1}{K}(g - c_2\dot{z}_t - c_1z_t + r_t) - \\ i_0(h_0z_t)^{-1}(h_0^2 - z_t^0) & \text{otherwise} \end{cases}$$

$c_1$  and  $c_2$  are positive constants which define the linearized dynamics and  $r_t$  is the reference input. In practice the parameters are unknown and only their estimates can be used in the control law. However, since the focus here is on identification, the nominal parameter values were used to avoid stability problems due to uncertainty in the control law.

The algorithm in the previous section was derived for a first order system. Since the state variables are accessible, we can construct a first order estimator model of the magnetic bearing system by defining a velocity variable  $v_t$  as

$$v_t = \dot{z}_t.$$

The estimator is then given by

$$\dot{\hat{v}}_t = -a_0 \tilde{v}_{\epsilon_t} - a_t \text{sat}\left(\frac{\tilde{v}_t}{\epsilon}\right) + \hat{K}_t f(u_t, z_t, \hat{h}_{0_t}) - g.$$

This leads to the error model

$$\dot{\tilde{v}}_t = -a_0 \tilde{v}_{\epsilon_t} - a_t \text{sat}\left(\frac{\tilde{v}_t}{\epsilon}\right) + \hat{K}_t f(u_t, z_t, \hat{h}_{0_t}) - K f(u_t, z_t, h_0),$$

which is in the form required by the algorithm. In order to speed up computation, the least squares solution was used to find  $d_t$  and  $a_t$ .  $r_t$  was chosen as a random signal

with standard deviation 0.1. The estimates of  $K$  and  $h_0$  are shown in Figures 4-2 and 4-3. The system is highly nonlinear. For this reason it was found that the parameter estimates depended substantially on the initial conditions used. As can be seen in the Figures, the parameter estimates converge towards their true values, but stop before they get there. This is because  $\tilde{v}_{\epsilon_t}$  goes to zero before the parameter errors. If the linear programming solution to the update laws had been used, the performance of the algorithm would most likely have been better.

## 4.3 When $f$ is concave or convex in $\theta$

### 4.3.1 The Algorithm

We consider here the same system as in Section 4.2, but with the additional assumption that, for given  $\phi_t$ ,  $f$ , satisfies the following condition

- (A)  $\forall \theta \in \Theta$ ,  $f(\phi_t, \theta)$  is either
- (a) convex with respect to  $\theta$  or
  - (b) concave with respect to  $\theta$ .

As we shall see, this allows us to derive analytical update laws for the parameter estimates. In the subsequent discussion the terms convex and concave will refer to the convexity and concavity of  $f$  with respect to  $\theta$ .

We will make use of the same estimator model and adaptation laws as in the previous section. Thus  $\dot{V}$  is given by

$$\dot{V}_t = -a_0 \tilde{y}_{\epsilon_t}^2 + \tilde{y}_{\epsilon_t} [-a_t \text{sat}(\frac{\tilde{y}_t}{\epsilon}) + \sigma(\hat{f}_t - f_t - d_t \tilde{\theta}_t)]. \quad (4.7)$$

We show below that time functions  $d_t$  and  $a_t$  can be found such that  $\dot{V}_t \leq 0$  everywhere. We present our argument by considering three separate cases (a)  $|\tilde{y}_t| \leq \epsilon$ , (b)  $\tilde{y}_t > \epsilon$  and (c)  $\tilde{y}_t < -\epsilon$ . The arguments are outlined for the case where  $f$  is convex in  $\theta$ . How  $d_t$  and  $a_t$  can be found when  $f$  is concave in  $\theta$  is mentioned at the end of the section.

(a) If  $|\tilde{y}_t| \leq \epsilon$ , then  $\tilde{y}_{\epsilon_t} = 0$  and thus  $\dot{V}_t = 0$ . Therefore stability is assured for any



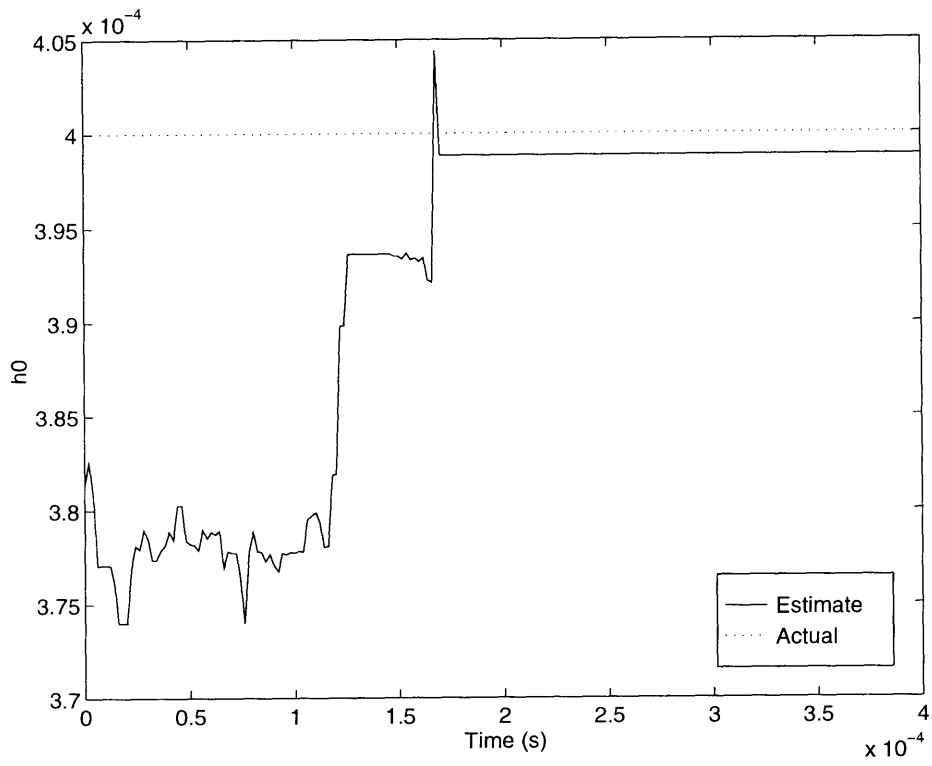


Figure 4-2: Estimates of  $h_0$

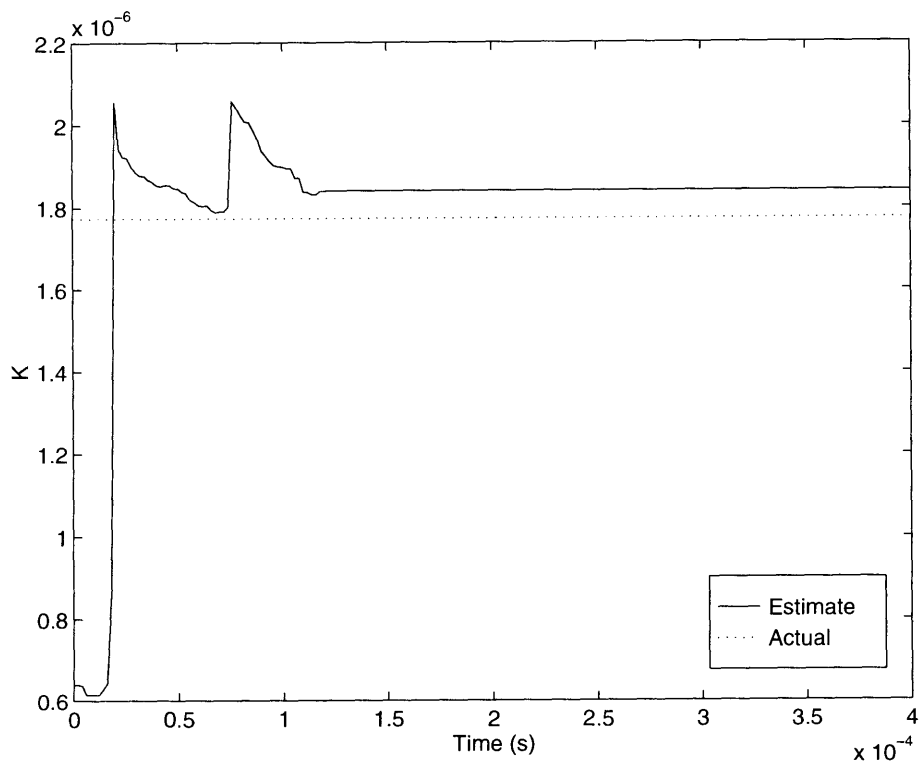


Figure 4-3: Estimates of  $K$

choice of  $d_t$  and  $a_t$ .

(b) From Eq. (4.7) it can be seen that  $\dot{V}_t \leq 0$  if the following condition holds

$$\text{sgn}(\tilde{y}_{\epsilon_t})\sigma d_t \tilde{\theta}_t \geq \text{sgn}(\tilde{y}_{\epsilon_t})\sigma(\hat{f}_t - f_t) - a_t,$$

or, since  $\sigma \leq \sigma_{\max}$ ,  $\dot{V}_t < 0$  if

$$\text{sgn}(\tilde{y}_{\epsilon_t})d_t \tilde{\theta}_t \geq \text{sgn}(\tilde{y}_{\epsilon_t})(\hat{f}_t - f_t) - \frac{a_t}{\sigma_{\max}}. \quad (4.8)$$

Since  $\phi_t$ ,  $f(\phi_t, \theta)$  is convex in  $\theta$ ,

$$\tilde{\theta}_t \left. \frac{\partial f(\phi_t, \theta)}{\partial \theta} \right|_{\theta=\hat{\theta}_t} \geq \hat{f}_t - f_t.$$

This is a result of the fact that for a convex function, the tangent line to  $f(\phi_t, \theta)$  at  $\hat{\theta}_t$  always lies below the graph of  $f(\phi_t, \theta)$ . This is illustrated in Figure 4.4. Since  $\tilde{y}_t > \epsilon$ ,  $\tilde{y}_{\epsilon_t} > 0$ . Hence, if we choose

$$d_t = \left. \frac{\partial f(\phi_t, \theta)}{\partial \theta} \right|_{\theta=\hat{\theta}_t} \quad \text{and} \quad a_t = 0, \quad (4.9)$$

then the inequality given by Eq. (4.8) is satisfied and hence  $\dot{V}_t \leq 0$ .

(c) Since  $\tilde{y}_{\epsilon_t} < 0$  in this case, the desired inequality becomes

$$d_t \tilde{\theta}_t \leq (\hat{f}_t - f_t) + \frac{a_t}{\sigma_{\max}}. \quad (4.10)$$

Ideally we would like to choose  $d_t$  such that  $d_t \tilde{\theta}_t \leq \hat{f}_t - f_t$ . In Figure 4.4, this corresponds to a line through  $\hat{\theta}$  with slope  $d_t$  which is always above the graph of  $f(\phi_t, \theta)$ . As can be seen from Figure 4.4, there exists no such  $d_t$ . The best we can do is to minimize the maximum distance by which the line is *below* the graph of  $f(\phi_t, \theta)$ . This corresponds to choosing a  $d_t$  which results in a minimum magnitude of  $a_t$ . This is desirable since  $a_t$  represents a measure of the time constant of the estimator. If the estimator is too fast, the parameter estimates will not have time to converge to their

true values before the  $\tilde{y}_{\epsilon_t}$  reaches zero. We can therefore pose the task of finding  $d_t$  and  $a_t$  as the following optimization problem

$$\begin{aligned} & \text{minimize } a_t \\ & \text{subject to } c_t \geq d_t(\hat{\theta}_t - \theta) + (f_t - \hat{f}_t), \end{aligned} \tag{4.11}$$

where

$$c_t = \frac{a_t}{\sigma_{\max}}.$$

The solution to this optimization problem is given by

$$d_t = \frac{f_{t_{\max}} - f_{t_{\min}}}{\theta_{\max} - \theta_{\min}} \tag{4.12}$$

$$a_t = \frac{\sigma_{\max}[(\hat{\theta}_t - \theta_{\min})(f_{t_{\max}} - \hat{f}_t) - (\theta_{\max} - \hat{\theta}_t)(\hat{f}_t - f_{t_{\min}})]}{(\theta_{\max} - \theta_{\min})}. \tag{4.13}$$

This is established in the following theorem.

**Theorem 3** The solution to the optimization problem in Eq. (4.11) is given by Eqs. (4.12) and (4.13).

**Proof** Since  $f$  is convex in  $\theta$  and  $d_t\theta$  is linear in  $\theta$ ,

$$d_t(\hat{\theta}_t - \theta) + (f_t - \hat{f}_t) \tag{4.14}$$

is convex in  $\theta$ . Therefore Eq. (4.14) takes on a maximum at either  $\theta_{\min}$  or  $\theta_{\max}$  or both. The constraint in Eq. (4.11) can be thus be expressed as

$$c_t \geq d_t(\hat{\theta}_t - \theta_{\min}) + (f_{\min} - \hat{f}_t) \tag{4.15}$$

$$c_t \geq d_t(\hat{\theta}_t - \theta_{\max}) + (f_{\max} - \hat{f}_t) \tag{4.16}$$

These two inequalities can be converted to equalities by adding slack variables  $\epsilon_1 > 0$

and  $\epsilon_2 > 0$ . This gives

$$\begin{aligned} c_t &= d_t(\hat{\theta}_t - \theta_{\min}) + (f_{\min} - \hat{f}_t) + \epsilon_1 \\ c_t &= d_t(\hat{\theta}_t - \theta_{\max}) + (f_{\max} - \hat{f}_t) + \epsilon_2 \end{aligned}$$

We now consider the three cases where a  $d_t$  and  $c_t$  have been found such that Eqs. (4.15) and (4.16) are satisfied with (a)  $\epsilon_1 > 0$ ,  $\epsilon_2 > 0$ , (b)  $\epsilon_1 > 0$ ,  $\epsilon_2 = 0$  and (c)  $\epsilon_1 = 0$ ,  $\epsilon_2 > 0$ .

(a) Let  $d_t^* = d_t$  and choose  $c_t^* = c_t - \max(\epsilon_1, \epsilon_2)$ . Then Eqs. (4.15) and (4.16) are satisfied by  $d^*$  and  $c_t^* < c_t$ . Thus  $c_t$  is not optimal.

(b) Choose

$$d_t^* = \frac{\epsilon_1}{\theta_{\max} - \theta_{\min}}$$

and let

$$c_t^* = c_t - \frac{\epsilon_1(\theta_{\max} - \hat{\theta}_{t-1})}{\theta_{\max} - \theta_{\min}},$$

then

$$c_t^* = d_t^*(\hat{\theta}_t - \theta_{\min}) + (f_{\min} - \hat{f}_t) \quad (4.17)$$

$$c_t^* = d_t^*(\hat{\theta}_t - \theta_{\max}) + (f_{\max} - \hat{f}_t). \quad (4.18)$$

Thus Eqs. (4.15) and (4.16) are satisfied by  $d^*$  and  $c_t^* < c_t$ . Therefore  $c_t$  is not optimal, since  $c_t^* < c_t$ .

(c) Let

$$d_t^* = \frac{\epsilon_2}{\theta_{\max} - \theta_{\min}}$$

and let

$$c_t^* = c_t - \frac{\epsilon_2(\hat{\theta}_{t-1} - \theta_{\max})}{\theta_{\max} - \theta_{\min}}.$$

Then  $d_t^*$  and  $c_t^*$  satisfy Eqs. (4.17) and (4.18) and therefore also Eqs. (4.15) and (4.16).

It follows that  $c_t$  is not optimal, since  $c_t^* < c_t$ .

Since  $c_t$  is not optimal for case (a), (b) or (c), it must be optimal when  $\epsilon_1 = \epsilon_2 = 0$ . Thus Eqs. (4.17) and (4.18) hold. These equations represent two linear simulta-

neous equations in the two unknown variables  $d_t$  and  $c_t$ . The solution to Eqs. (4.17) and (4.18) is given by Eqs. (4.12) and (4.13).  $\square$

In order to complete the algorithm we need to specify  $d_t$  and  $a_t$  when  $\tilde{y}_t \leq \epsilon$ . This should preferably be done in such a way that  $d_t$  and  $a_t$  are continuous. We then obtain the following algorithm

Estimator model:

$$\dot{\hat{y}}_t = -a_0 \tilde{y}_{\epsilon t} - a_t \text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) + \hat{\sigma}_t f(\phi_t, \hat{\theta}_t) + \varphi_t^T \hat{\alpha}_t$$

where

$$a_t = \begin{cases} 0 & \text{if } \tilde{y}_t \geq 0 \\ -\text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) \frac{\sigma_{\max}[(\hat{\theta}_t - \theta_{\min})(f_{t\max} - \hat{f}_t) - (\theta_{\max} - \hat{\theta}_t)(\hat{f}_t - f_{t\min})]}{(\theta_{\max} - \theta_{\min})} & \text{otherwise.} \end{cases} \quad (4.19)$$

Adaptation laws:

$$\begin{aligned} \dot{\hat{\sigma}}_t &= -\tilde{y}_{\epsilon t} \gamma_{\sigma} \hat{f}_t \\ \dot{\hat{\theta}}_t &= -\tilde{y}_{\epsilon t} \gamma_{\theta} d_t \\ \dot{\hat{\alpha}}_t &= -\tilde{y}_{\epsilon t} \Gamma_{\alpha} \varphi_t \end{aligned}$$

where

$$d_t = \begin{cases} \text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) \frac{\partial f(\phi_t, \theta)}{\partial \theta} \Big|_{\theta=\hat{\theta}_t} & \text{if } \tilde{y}_t \geq 0 \\ -\text{sat}\left(\frac{\tilde{y}_t}{\epsilon}\right) \frac{f_{t\max} - f_{t\min}}{\theta_{\max} - \theta_{\min}} & \text{otherwise.} \end{cases} \quad (4.20)$$

If  $f(\phi_t, \theta)$  is instead concave in  $\theta$ , then  $d_t$  and  $a_t$  should be chosen as given by Eq. (4.9) when  $\tilde{y}_t < 0$  and as given by Eqs. (4.12) and (4.13) when  $\tilde{y}_t \geq 0$ .

### 4.3.2 Application to a Low-velocity Friction Model

Friction often acts as a limiting factor for the performance of servo mechanisms designed for high precision positioning and low velocity tracking tasks. The use of a detailed friction model to predict and compensate for the effects of friction has the

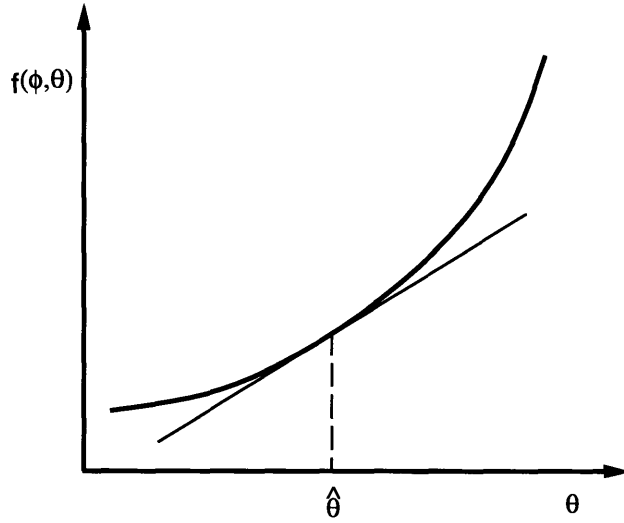


Figure 4-4: If  $f$  is a convex function of  $\theta$ , then the first order Taylor expansion around  $\hat{\theta}$  underestimates  $f$ .

potential to improve the performance of these machines considerably. However, the local nature of low velocity friction effects require these models to be nonlinearly parameterized, which makes it difficult to use conventional techniques to estimate the parameters associated with the models. In this section the algorithm described above is used to estimate the parameters of the following steady-state friction model from [4]

$$\frac{dv_t}{dt} = \tau_t - F_C \text{sgn}(v_t) - (F_S - F_C) \text{sgn}(v_t) e^{-\left(\frac{v_t}{v_0}\right)^2} \quad (4.21)$$

where  $v$  is the angular velocity of the motor shaft,  $\tau$  is the input torque,  $F_C$  represents the Coulomb friction,  $F_S$  stands for static friction and  $v_s$  is the Stribeck parameter. Considerable attention has been given to the estimation of parameters in models of this type. In particular, the term including the static and Stribeck friction parameters has been studied extensively since these parameters enter the model nonlinearly and therefore are difficult to estimate. Figure 4.3.2 shows a plot of the friction force as a function of velocity and illustrates the role of the friction coefficients. For simplicity the inertia of the load and the viscous friction coefficient have not been included in the model. If desired, these parameters can be included and estimated in a straight

forward manner since they enter the model linearly.

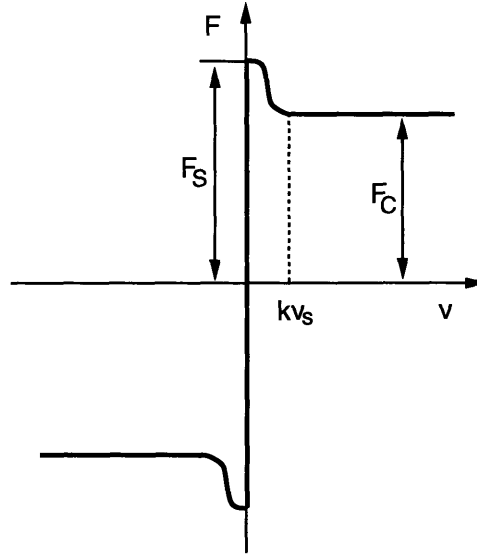


Figure 4-5: Friction force as a function of velocity.  $k$  is a constant indicating that the velocity at which the Stribeck curve flattens out is proportional to  $v_s$ .

$\tau_t$  is given by the proportional control law  $\tau_t = K(r_t - v_t)$ , where  $r_t$  is the reference trajectory and  $K$  is the gain of the controller. The model in Eq. (4.21) can be put into the form

$$\dot{y}_t = \sigma f(\phi_t, \theta) + \varphi_t^T \alpha,$$

where

$$\begin{aligned} \begin{bmatrix} \phi_t \\ \dot{y}_t \end{bmatrix} &= \begin{bmatrix} v_t \\ \dot{v}_t \end{bmatrix} & f(\phi_t, \theta) &= -\text{sgn}(y_t)e^{-\theta v_t^2} & \varphi_t &= \begin{bmatrix} \tau \\ \text{sgn}(v_t) \end{bmatrix} \\ \sigma &= F_S - F_C & \theta &= \frac{1}{v_s^2} & \alpha &= \begin{bmatrix} 1 \\ F_C \end{bmatrix} \end{aligned}$$

Simulation results are shown in Figures 4.6 through 4.9. The reference input was a random signal with variance 0.015. The actual parameter values are marked by

a dotted line in the Figures. Because of the local nature of the static friction and the Stribeck effect, the convergence of the parameter estimates to their actual values depends on the input. The parameter estimates were also fairly sensitive to the choice of adaptation gains.



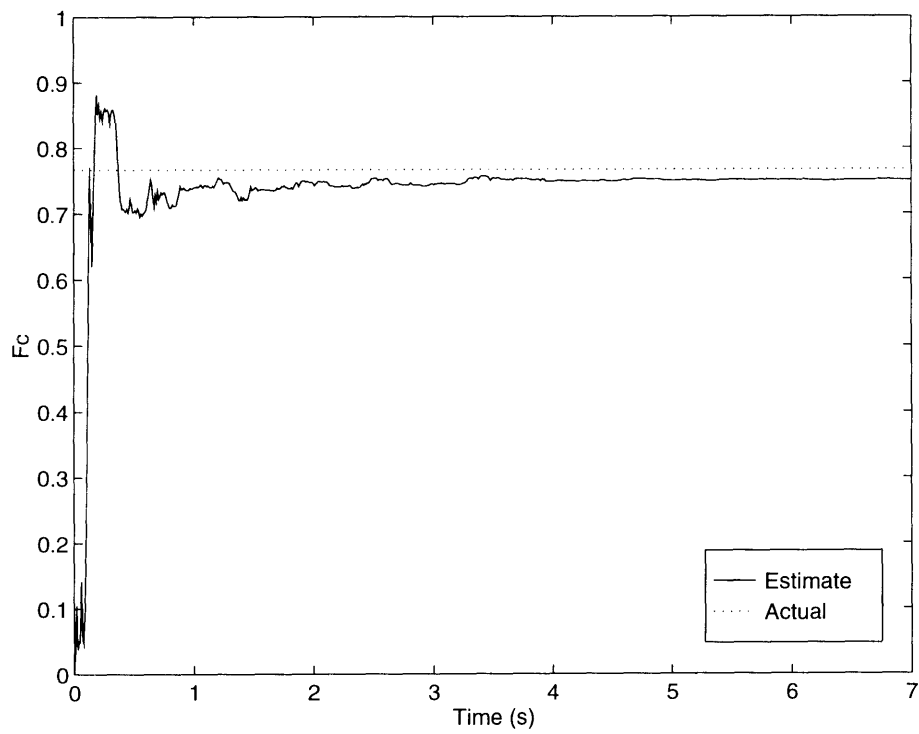


Figure 4-6: Estimates of  $F_C$

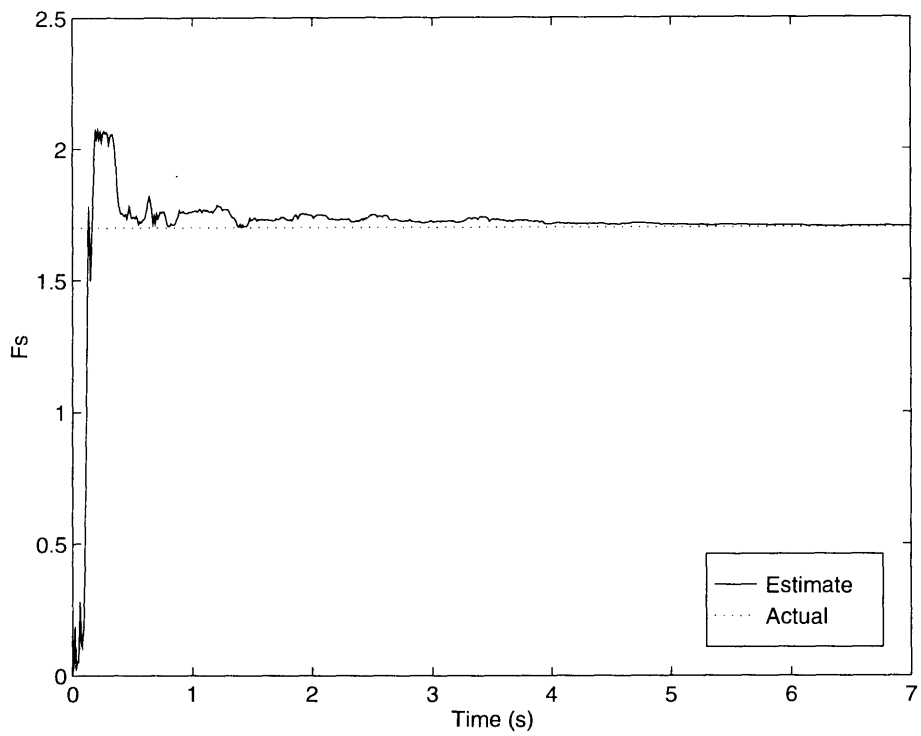


Figure 4-7: Estimates of  $F_S$

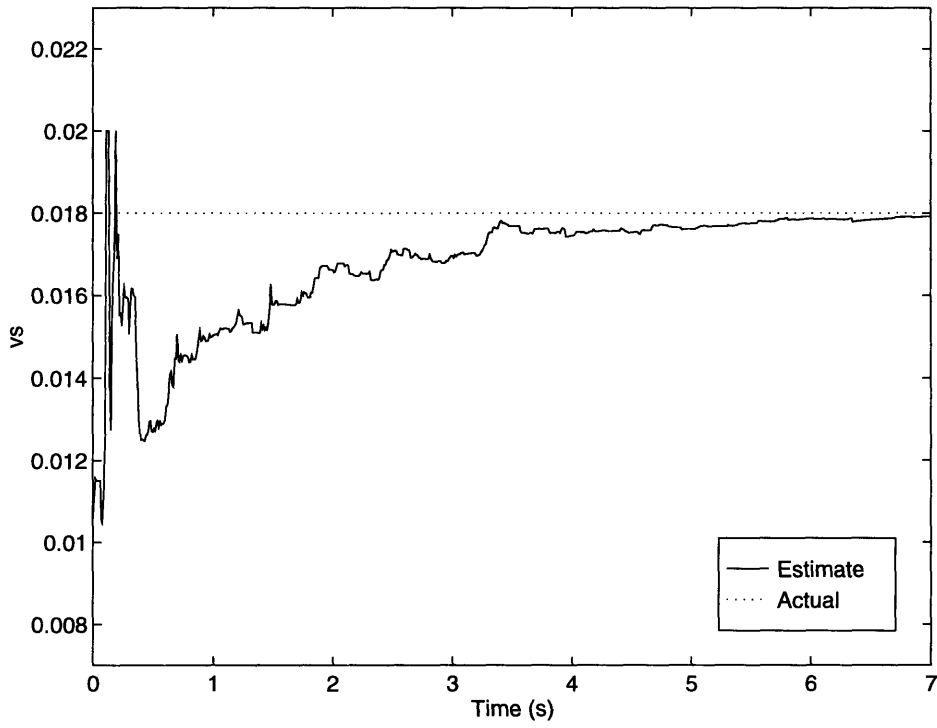


Figure 4-8: Estimates of  $v_s$

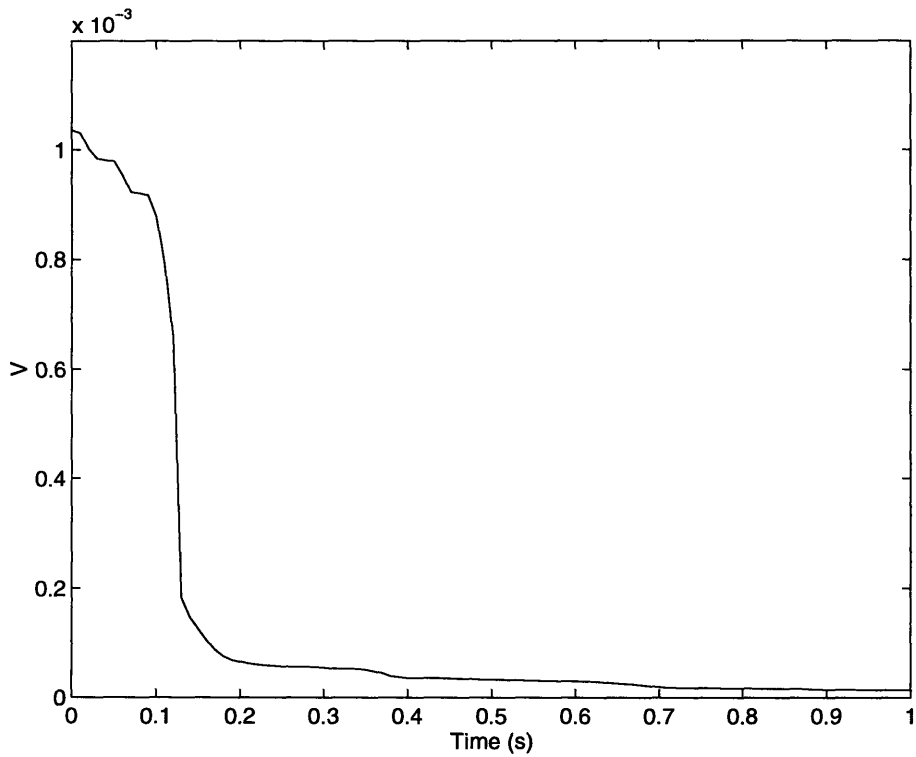


Figure 4-9:  $V$  as a function of time from  $t = 0$  to  $t = 1$

# Chapter 5

## Conclusion

Estimates of the physical parameters of a system provide valuable information for tasks such as control and fault detection. For systems which are nonlinearly parameterized however, few analytical techniques are available for parameter estimation. In [2] a novel, model-based approach to neural network based system identification was proposed where the neural network is trained to provide an estimate of the system parameters at each instant of time. This algorithm thus has the potential to improve the performance of control and fault detection schemes for dynamic systems in the presence of nonlinearities. However, the neural network has to be trained for variations in all the system variables and grows exponentially in size when the number of variables is increased. Furthermore the training of the neural network is a nonlinear optimization problem.

In this thesis it is shown that the task of finding an update law for the parameter estimates of the class of systems considered in [2] can be solved numerically by the formulation of a quadratic programming problem. The algorithm does not depend on analytical knowledge of the regressor function. In particular, a neural system model can be used to approximate the required regression form for systems which cannot easily be transformed into this form analytically. It is shown that if some of the system parameters occur linearly, or enter the model multiplicatively, an update law for these parameters can be found analytically. This greatly reduces the amount of computation needed to implement the algorithm. Algorithms are also presented

which apply to continuous-time systems where only one parameter occurs nonlinearly. An analytic solution is given for the case where the regression function is either concave or convex in the system parameter. It is demonstrated how the algorithms can be robustified with respect to a bounded disturbance. The performance of the algorithms is illustrated in simulations of a magnetic bearing system and a low velocity friction model.

# Bibliography

- [1] A. M. Annaswamy and S. Yu. Intelligent control using  $\theta$ -adaptive neural networks: A new approach to identification and control of nonlinear systems. In *Proceedings of the American Control Conference*, 1995.
- [2] A. M. Annaswamy and S. Yu.  $\theta$ -adaptive neural networks: A new approach to parameter estimation. *IEEE Transactions on Neural Networks*, (to appear) 1996.
- [3] Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.
- [4] Carlos Canudas de Wit, Henrik Olsson, Karl Johan Åström, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995.
- [5] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.
- [6] G. C. Goodwin and K. S. Sin. *Adaptive Filtering, Prediction, and Control*. Prentice-Hall, 1984.
- [7] Graham C. Goodwin and Kwai Sang Sin. *Adaptive Filtering Prediction and Control*. Prentice-Hall, Inc., 1984.
- [8] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

- [9] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, New York, NY, 2nd edition, 1989.
- [10] Michael I. Jordan and David E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992.
- [11] Lennart Ljung. *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [12] Kumpati S. Narendra and Anuradha M. Annaswamy. *Stable Adaptive Systems*. Prentice-Hall, Inc., 1989.
- [13] Kumpati S. Narendra and Kannan Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–26, March 1990.
- [14] Romero Ortega. Some remarks on adaptive neuro-fuzzy systems. *International Journal of Adaptive Control and Signal Processing*, (to appear) 1996.
- [15] J. Park and I. W. Sandberg. Universal approximation using radial-basis function networks. *Neural Computation*, 3:246–257, 1991.
- [16] Gerald Peterson, William Bond, Roger Germann, Barry Streeter, and James Urnes. Using neural networks for aerodynamic parameter modeling. In *Proceedings of the American Control Conference*, pages 1360–1361, Seattle, Washington, June 1995.
- [17] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.
- [18] Robert M. Sanner and Jean-Jacques E. Slotine. Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6):837–863, November 1992.

- [19] Jonas Sjöberg, Q. Zhang, Lennart Ljung, A. Benveniste, B. Deylon, P-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Non-linear black-box modeling in system identification. *Automatica*, December 1995.
- [20] J.-J. E. Slotine and J. A. Coetsee. Adaptive sliding controller synthesis for non-linear systems. *International Journal of Control*, 43(6):1631–1651, 1986.
- [21] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, NJ, 1991.
- [22] P. Wolfe. Some simplex-like non-linear programming procedures. *Operations Research*, 10(4), 1962.
- [23] Ting-Jen Yeh. *Modeling, analysis and control of magnetically levitated rotating machines*. PhD thesis, MIT, Massachusetts, MA, 1996.