Adaptive Extremum Control

X. Ma*, N. K. Poulsen *,

February 18, 1997

Abstract

A number of industrial processes have the behaviour that the process should have as high efficiency as possible. To track a varying maximum or minimum is called extremum control. One major characteristic in extremum control is that the static response curve relating input to output is inherently nonlinear, and we therefore assume that the process can be divided into the linear part and nonlinear part in series. If the nonlinear part is placed before linear part, it is called input nonlinearity. On the contrary it is output nonlinearity. A Hammerstein model is employed for a process with input nonlinearity, and an extremum control law is then derived based on the model. For a process with output nonlinearity, two cases will be discussed according to that the intermediate signal between linear part and nonlinear part can be measured or not. If the intermediate signal can not be measured, a state-space model will be used for parameter estimation, and the EKF, RPE and a modified recursive prediction error method with line search will be implemented.

Keywords

Adaptive Extremum control, Input nonlinearity, Output nonlinearity, Extended Kalman filter, Prediction error method, Line search.

**IMM, Building 321, The Technical University of Denmark, 2800 Lyngby, Denmark.*

1 Introduction

In most control problems, the task of the regulator is to keep some variables at constant values, or to make them follow reference signals. With such problems the reference values are often easily determined. It can be the desired altitude of an airplane, or the desired concentration of a product. In these cases the control action is to optimize a cost function by involving an estimated system model.

On other occasions, It can be more difficult to find the suitable reference values or the best operating points of a process. A number of industrial processes have the behaviour that the process should have as high efficiency as possible, their performance can be improved by adjusting plant parameters so as to maximize or minimize the performance criterion. To track a varying maximum or minimum is called *extremum control*. There are several examples of practical systems that exhibit this type of behaviour, e.g., power generation systems, chemical and combustion process. One application is spark-ignition automotive engine. The fuel consumption of a car depends, among other things, on the ignition angle. The task of adaptive extremum control is to adjust spark ignition angle and operate the engine at a predetermined optimum value.

Extremum control systems have one major characteristic in common. In the absence of disturbance, the static response curve relating the outputs to inputs is nonlinear and has at least one extremum. The objective of extremum control is to keep the process operating at, or in the vicinity of, the extremum point of the performance function or process output despite changes in the process or the influence of disturbances.

[Blackman 1962] is a good introduction to the field of extremum control. The classification used in this report is *perturbation systems*, *switching systems* and *self driving systems*. In a so called *switching system*, the input is driven by a constant speed flip-flop with two states. If the extremum is passed, the direction of input drift is then reversed according to some fixed rules. The system can then be operated in the neighbourhood of the extremum. The *self driving systems* use measurements of the time derivative of the output to determine the input. If the process is started in the correct direction, it will continue until extremum is reached. In the *perturbation systems*, the effect at output from a known signal added to the input is used to derive information about the slope of the nonlinearity. The information is then used to drive the system and keep the slope as close to zero as possible.

A fourth class of methods that is not described in [Blackman 1962] have been developed later on. It is based on using a parameterized model in combining parameter identification and extremum control.

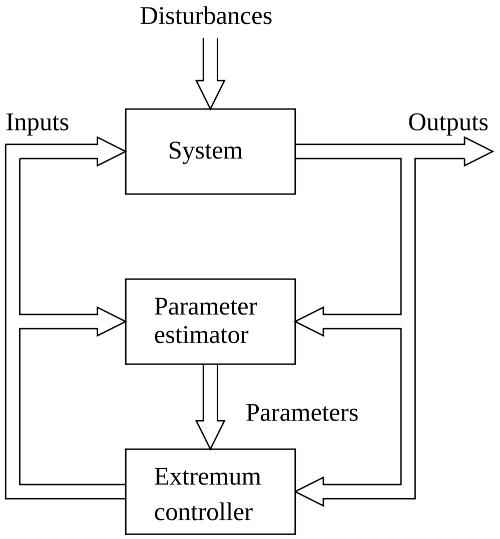

Figure 1: Block diagram of an adaptive extremum controller

In case the model of process dynamic is unknown, if the assumed structure is correct, the on-line parameters estimation should be combined with on-line extremum control. This leads to model-based adaptive extremum control. Model-based extremum control methods generate the control action by making use of a model obtained by some kind of recursive identification methods. The recursive least squares method (RELS) and recursive maximum likelihood method (RML) are normally good choices for estimation

problem. The current best estimates are used to determine the new input value. This idea is illustrated in Figure 1.

The extremum control problem treated here will be assumed to have unknown nonlinearities or at least partly unknown nonlinearities. One way to simplify the problem is to assume that the process can be separated into the linear part and the nonlinear part in series. Models with different properties can be obtained if the nonlinear part is placed before, after, or in the middle of the linear part. If the nonlinear part is placed before the linear part, it is called input nonlinearity. If the nonlinear part is placed after the linear part, it is output nonlinearity. Model behaviour will be different between input nonlinearity and output nonlinearity. The adaptive extremum control applied on process with input and output nonlinearity will be given in section 2 and section 3 respectively.

2 Input nonlinearity

The key point of extremum control is the basic assumption of a model which describes the performance function or the process dynamics. A model for a process with input nonlinearity is given first, and an extremum control algorithm can then be derived based on the model.

2.1 Models for the process with input nonlinearity

A general description of a nonlinear discrete-time model is

$$y(t) = f(y(t-1), y(t-2), \cdots, u(t-d), u(t-d-1), \cdots, \theta, t) + \omega(t) \quad (1)$$

where f is a nonlinear function and assumed that an extremum value exists, $\{y(t)\}$, $\{u(t-d)\}$ and $\omega(t)$ denote respectively the output, input and random disturbance sequences. d is time delay, θ is a vector of unknown parameters that may change with time. The sequence $\{y(t)\}$ might be some measurement of the system output or a performance function.

It is assumed that the process can be divided into linear and nonlinear part in series, the input nonlinearity can then be defined if nonlinear part is placed before linear part. A process with input nonlinearity is illustrated in Figure 2.

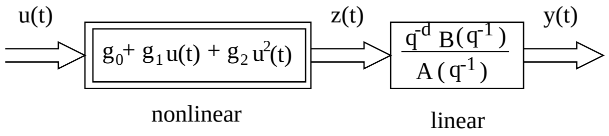

Figure 2: Process with input nonlinearity

The nonlinear block can be represented by

$$z(t) = g_0 + g_1 u(t) + g_2 u^2(t) \tag{2}$$

where $u(t)$ and $z(t)$ are input signal and intermediate signal at time t. The linear block can be represented by

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})z(t) \tag{3}$$

where $y(t)$ is output signal, d is time delay, and

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b} \end{aligned} \tag{4}$$

are the polynomials of backward shift operator q^{-1}.

Therefore the representation for the whole model is

$$A(q^{-1})y(t) = g_0 B(1) + g_1 B(q^{-1})u(t-d) + g_2 B(q^{-1})u^2(t-d) \tag{5}$$

This is a simple Hammerstein model, which can be generalized to a generalized Hammerstein model

$$A(q^{-1})y(t) = B_0 + B_1(q^{-1})u(t-d) + B_2(q^{-1})u^2(t-d) \tag{6}$$

with

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B_1(q^{-1}) &= b_{10} + b_{11} q^{-1} + \cdots + b_{1n_{b1}} q^{-n_{b1}} \\ B_2(q^{-1}) &= b_{20} + b_{21} q^{-1} + \cdots + b_{2n_{b2}} q^{-n_{b2}} \end{aligned} \tag{7}$$

The model of Hammerstein type has been used quite often in extremum control systems. The Hammerstein representation is very popular because it is a good picture of the reality, and linear in term of the unknown parameters of the system. Most identification methods are based on the assumption that the model is linear in parameters.

To be more specific, if the measurements of process output or performance is typically corrupted by noise, it is then necessary to take the noise model into account, therefore the generalized Hammerstein model (6) is modified by

$$A(q^{-1})y(t) = B_0 + B_1(q^{-1})u(t-d) + B_2(q^{-1})u^2(t-d) + C(q^{-1})e(t) \tag{8}$$

where $e(t)$ is a white noise with zero mean normal distribution, and

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B_1(q^{-1}) &= b_{10} + b_{11} q^{-1} + \cdots + b_{1n_{b1}} q^{-n_{b1}} \\ B_2(q^{-1}) &= b_{20} + b_{21} q^{-1} + \cdots + b_{2n_{b2}} q^{-n_{b2}} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c} \end{aligned}$$

2.2 Adaptive extremum control for Hammerstein model

The purpose of the extremum control is to maintain the output as close as possible to the extremum despite the influence of disturbances. It means that the control objective is to maximize or minimize the performance function

$$J(u(t)) = E\{y(t+d)\} \tag{9}$$

where E denotes the expectation operator, $y(t)$ is the output signal or a performance function described by (1). If it is assumed that model (1) has the first and second order derivatives, then the control law can be achieved by

$$\frac{dJ}{du} = 0 \tag{10}$$

Consider a system with input nonlinearity, which can be represented by a generalized Hammerstein model (8), the static characteristics of the process is given by

$$\bar{A}y = B_0 + \bar{B}_1 u + \bar{B}_2 u^2 \tag{11}$$

Since the extremum control is to make the steady-state value of output as low or as high as possible, the resulting controller will be derived from the static response to seek the optimal inputs.

$$u(t) = -\frac{\bar{B}_1}{2\bar{B}_2} \tag{12}$$

and optimal value of the output is

$$y_0 = \frac{B_0}{\bar{A}} - \frac{\bar{B}_1^2}{4\bar{A}\bar{B}_2} \tag{13}$$

The extremum controller is then a constant gain and no feedback controller.

If the parameters of the model are unknown, the idea is to use an on-line recursive estimation procedure to identify them, and at each adjustment step use the current best estimates to determine the new value of $u(t)$.

If $C(q^{-1}) = 1$ in the model (8), then the ordinary least-squares method can be used to estimate parameters. If $C(q^{-1}) \neq 1$, then it is necessary to approximate and use e.g. the recursive extended least-squares (RELS) method or the recursive maximum likelihood (RML) method. The model (8) can then be written as

$$y(t) = \varphi^T(t)\theta + \epsilon(t) \tag{14}$$

where

$$\begin{aligned} \varphi(t) &= [-y(t-1), \cdots, -y(t-n_a), 1, u(t-d), \cdots, u(t-n_{b1}-d), \\ &\quad u^2(t-d), \cdots, u^2(t-n_{b2}-d), \epsilon(t-1), \cdots, \epsilon(t-n_c)]^T \end{aligned} \tag{15}$$

$$\theta = [a_1, \cdots, a_{na}, b_0, b_{10}, \cdots, b_{1nb1}, b_{20}, \cdots, b_{2nb2}, c_1, \cdots, c_{nc}]^T$$

and

$$\epsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1) \tag{16}$$

The estimates using RELS method are given by equations

$$\begin{aligned} \hat{y}(t) &= \varphi^T(t)\hat{\theta}(t-1) \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + k(t)[y(t) - \hat{y}(t)] \\ k(t) &= P(t)\varphi(t)[\lambda(t) + \varphi^T(t)P(t)\varphi(t)]^{-1} \\ P(t+1) &= [I - k(t)\varphi^T(t)]P(t)/\lambda(t) \end{aligned} \tag{17}$$

where $\lambda(t)$ is an exponential forgetting factor.

The estimated parameters may then be used in the control law (12) instead of the true parameter values

$$u(t) = -\frac{\hat{\bar{B}}_1(t)}{2\hat{\bar{B}}_2(t)} \tag{18}$$

where $-\hat{\bar{B}}_1(t)/2\hat{\bar{B}}_2(t)$ is the current estimated value of the optimal input.

The adaptive extremum control algorithm can be summarized as

Step 1: Apply input $u(t)$ to the system

$$y(t+d) = f(u(t), \theta, t) + e(t)$$

and measure the output $y(t+d)$.

Step 2: Use RELS or RML algorithm to estimate parameters for the model (8).

Step 3: Extremum control law is

$$u(t+1) = -\frac{\hat{\bar{B}}_1(t)}{2\hat{\bar{B}}_2(t)}$$

Step 4: Increment the time $t \to t+1$ and return to **Step 1**.

Such a scheme has to be used with great care, since the control input depends only on the estimated values. When $u(t)$ is actively varying by large amounts, i.e., when the self-tuning is in progress, the identifiability problems do not arise. When the system is nominally tuned, the $u(t)$ converges to a constant value. This will reduce the excitation of the process and identifiability may be lost, and the parameter estimates may converge to some wrong values. This problem has been discussed by [Sternby 78].

In practice the adaptive extremum control algorithm will be modified by adding an extra perturbation signal to the input. The test perturbation signal can be a zero mean random variable with small variance

$$u(t) = -\frac{\hat{B}_1(t)}{2\hat{B}_2(t)} + u_p(t) \tag{19}$$

where u_p is a test perturbation signal.

2.3 Case studies

In this section the behaviour of adaptive extremum control algorithm discussed above will be shown by some simulation examples.

Example 2.1 *A nonlinear dynamic system*

Now a nonlinear dynamic system with second-order Hammerstein model is considered

$$y(t) + ay(t-1) = b_0 + b_1 u(t-1) + b_2 u^2(t-1) + e(t) \tag{20}$$

with $a = -0.8$, $b_0 = 100$, $b_1 = 2$ *and* $b_2 = -0.01$. *The maximum attainable value of the output is* $y^* = 1000$, *and optimal input is* $u^* = 100$.

The system can be modelled by

$$y(t) + \theta_1 y(t-1) = \theta_2 + \theta_3 u(t) + \theta_4 u^2(t) \tag{21}$$

The linear regression can be written as

$$y(t) = \varphi^T(t)\hat{\theta} + \epsilon(t)$$

with

$$\begin{aligned} \varphi(t) &= [-y(t-1),\ 1,\ u(t-1),\ u^2(t-1)]^T \\ \hat{\theta} &= [\hat{\theta}_1,\ \hat{\theta}_2,\ \hat{\theta}_3,\ \hat{\theta}_4]^T \end{aligned} \tag{22}$$

The extremum control law is given by

$$u(t) = -\frac{\hat{\theta}_3}{2\hat{\theta}_4} + u_p(t)$$

and $u_p(t)$ *is a PRBS signal which ensure the persistent excitation of the input signal.*

Figure 3 and Figure 4 give the results of simulation. The estimated parameters are given in Figure 3, where $th(1) = \hat{a}$, $th(2) = \hat{b}_0$, $th(3) = \hat{b}_1$ *and* $th(4) = \hat{b}_2$. *It has been shown that the estimated parameters converge to the correct values very fast, and output* $y(t)$ *achieves the maximum value after 20 steps, which is slower than the convergence speed of estimated parameters. This is because of the dynamics in the system.*

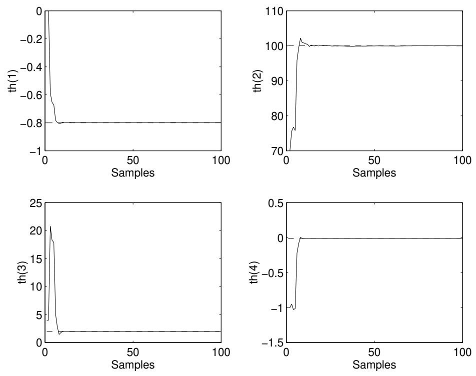

Figure 3: The estimated parameters.

◇

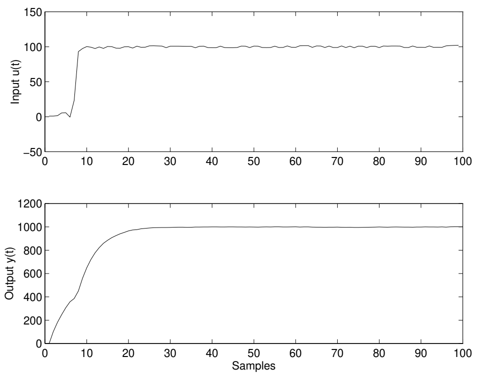

Figure 4: The system inputs and outputs.

2.4 Convergence analysis

The convergence properties of the extremum controller can be analysed by the Ordinary Differential Equation (ODE) approach of Ljung [Ljung 77].

Consider a system

$$S: \; y(t) = \varphi^T(t)\theta_0 + e(t) \tag{23}$$

and a model for the system is

$$M: \; y(t) = \varphi^T(t)\theta \tag{24}$$

the RELS estimator (17) can be rewritten as

$$\begin{array}{l} \hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1)) \\ P(t+1)^{-1} = P(t)^{-1} + \varphi(t)\varphi^T(t) \end{array} \tag{25}$$

For the analysis purposes, we will introduce the matrix $R(t)$

$$R(t) = \frac{1}{t}P(t)^{-1} \tag{26}$$

then the standard form of RLS will be taken as

$$\begin{array}{rcl} \hat{\theta}(t) & = & \hat{\theta}(t-1) + \frac{1}{t}R(t)^{-1}\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\ R(t) & = & R(t-1) + \frac{1}{t}[\varphi(t)\varphi^T(t) - R(t-1)] \end{array} \tag{27}$$

Clearly, the RLS algorithm can be cast in this form where $R(t)$ represents the Hessian matrix of the cost function. If we wish to carry through the transformation to an ODE, the system which is controlled by some adaptive feedback laws should give rise to closed loop stability. This means that if the closed loop system matrix is denoted by $A(\hat{\theta}(t))$ at time t, then $\hat{\theta}(t)$ must belong to the set

$$D_s = \{\theta | A(\theta) \text{ has all eigenvalues inside the unit circle}\} \tag{28}$$

after some finite time. This is a strong condition involving closed loop stability under the adaptive feedback law.

By given assumption of stability, let $\theta \in D_s$, and $\hat{\theta}(t)$ replaced by some nominal value θ. Then we define

$$\begin{array}{rcl} f(\theta) & = & \lim_{t\to\infty} E\{\varphi(t)[y(t) - \varphi^T(t)\theta]\} \\ G(\theta, R) & = & \lim_{t\to\infty} E[\varphi(t)\varphi^T(t) - R] \end{array} \tag{29}$$

and corresponding ODE is then given by

$$\begin{array}{rcl} \dfrac{d\theta}{d\tau} & = & R^{-1}f(\theta) \\ \dfrac{dR}{d\tau} & = & G(\theta, R) \end{array} \tag{30}$$

If it is assumed that $e(t)$ in the system is uncorrelated with $\varphi(t)$, i.e., $E[\varphi(t)e(t)] = 0$, and $y(t)$ is substituted by (23), the (29) can then be written as

$$\begin{array}{rcl} f(\theta) & = & G(\theta)(\theta_0 - \theta) \\ G(\theta, R) & = & G(\theta) - R \end{array} \tag{31}$$

where

$$G(\theta) = \lim_{t\to\infty} E[\varphi(t)\varphi^T(t)] \tag{32}$$

is a symmetric semi-positive definite matrix.

It can be proved that if $\hat{\theta}(t) \to \theta^*$ and $R \to R^*$, the local convergence point will then satisfy

$$\begin{aligned} f(\theta^*) &= G(\theta^*)(\theta_0 - \theta^*) = 0 \\ R^* &= G(\theta^*) \end{aligned} \tag{33}$$

If $G(\theta^*)$ is positive definite matrix, i.e.,

$$\lim_{t\to\infty} E[\varphi(t)\varphi^T(t)] > 0 \tag{34}$$

This implies that $\theta^* = \theta_0$. The positive definite $G(\theta^*)$ is a generalized persistent excitation condition. If condition holds, then

$$H(\theta^*) = (R^*)^{-1}\frac{d}{d\theta}f(\theta)|_{\theta=\theta_0} = -I \tag{35}$$

all of whose eigenvalues are at -1 in the left half-plane. Thus, the θ_0 is the only convergence point under the persistent excitation condition.

In adaptive extremum control algorithms, The persistently exciting of input signal is ensured by addition of the test perturbation signal.

3 Output nonlinearity

If the model consists of a dynamic linear part and static nonlinear part, and nonlinear part is placed after linear part, it is output nonlinearity. Maybe an output nonlinearity is in general more important than an input nonlinearity for a good description of a nonlinear system. A nonlinearity at output of a linear system is much more difficult to handle than one at input. The complexity of the problem will also depends on which of the variables in the process can be measured.

3.1 Models and basic extremum control laws

A process with output nonlinearity is illustrated in Figure 5, where $u(t)$ and $y(t)$ are the input and output signal, $z(t)$ is intermediate signal between linear part and nonlinear part.

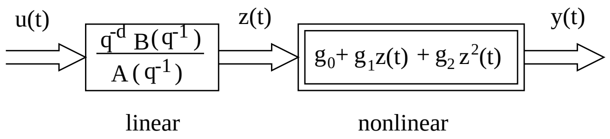

Figure 5: Process with input nonlinearity

For an output nonlinearity, the model of linear block can be written as

$$A(q^{-1})z(t) = q^{-d}B(q^{-1})u(t) \tag{36}$$

where

$$\begin{aligned} A(q^{-1}) = 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a} \\ B(q^{-1}) = b_0 + b_1q^{-1} + \cdots + b_{n_b}q^{-n_b} \end{aligned} \tag{37}$$

The model of the nonlinear block is given by

$$y(t) = g_0 + g_1z(t) + g_2z^2(t) \tag{38}$$

then the whole model will be

$$y(t) = g_0 + g_1\frac{B(q^{-1})}{A(q^{-1})}u(t-d) + g_2\left[\frac{B(q^{-1})}{A(q^{-1})}u(t-d)\right]^2 \tag{39}$$

This is a simple Wiener model. and it can be generalized by

$$y(t) = g_0 + \frac{B_1(q^{-1})}{A(q^{-1})}u(t-d) + \left[\frac{B_2(q^{-1})}{A(q^{-1})}u(t-d)\right]^2 \tag{40}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a} \\ B_1(q^{-1}) &= b_{10} + b_{11}q^{-1} + \cdots + b_{1n_{b1}}q^{-n_{b1}} \\ B_2(q^{-1}) &= b_{20} + b_{21}q^{-1} + \cdots + b_{2n_{b2}}q^{-n_{b2}} \end{aligned} \tag{41}$$

The model is called generalized Wiener model. It is easily seen that the Wiener type models are nonlinear in parameters. Most identification methods can not be used to estimate the parameters in this model because of the nonlinearity in parameters. If the intermediate signal is not measurable, it is then quite difficult to estimate the parameters of input-output model. Therefore a state-space model is taken into account

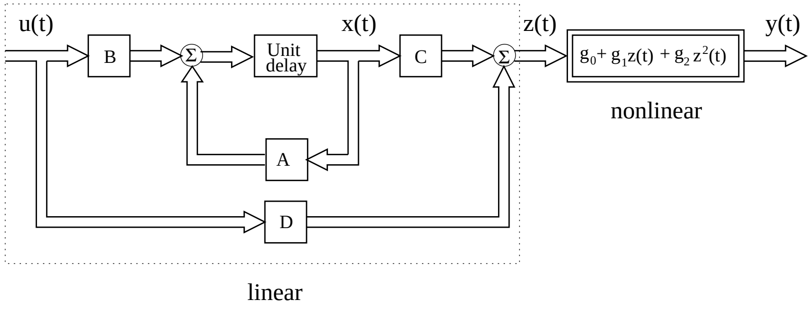

Figure 6: Process with input nonlinearity in state-space form

The process with output nonlinearity can also be represented by a state-space model, the block scheme of the state-space is illustrated in figure 6.

The state-space model of linear part is

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ z(t) &= Cz(t) + Du(t) \end{aligned} \tag{42}$$

where $x(t)$ is state vector, $z(t)$ is intermediate signal between input and output block. Matrices A, B, C and D are assumed to be time invariant.

The model of nonlinear part is given by (38). The model of the whole process can thus be derived

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= g_0 + g_1[Cx(t) + Du(t)] + g_2[Cx(t) + Du(t)]^2 \end{aligned} \tag{43}$$

If it is assumed that matrix $D = 0$, the model can be written by

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= G_0 + G_1 x(t) + x^T(t) G_2 x(t) \end{aligned} \tag{44}$$

where

$$\begin{aligned} G_0 &= g_0 \\ G_1 &= g_1 C \\ G_2 &= g_2 C^T C \end{aligned} \tag{45}$$

If the state noise and measurement noise are considered, and it is assumed that no measurement noise on the intermediate signal $z(t)$ if it can be measured, the model will then be

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= G_0 + G_1 x(t) + x^T(t) G_2 x(t) + e(t) \end{aligned} \tag{46}$$

where $w(t)$ and $e(t)$ are uncorrected noise sequences.

Model (46) can be represented by a general nonlinear state-space model

$$\begin{aligned} x(t+1) &= f(x(t), u(t), \theta(t)) + w_\theta(t) \\ y(t) &= h(x(t), \theta(t)) + e_\theta(t) \end{aligned} \tag{47}$$

with

$$\begin{aligned} E\{w_\theta(t) w_\theta^T(s)\} &= Q^w(\theta)\delta_{ts} \\ E\{e_\theta(t) e_\theta^T(s)\} &= Q^e(\theta)\delta_{ts} \\ E\{w_\theta(t) e_\theta^T(s)\} &= Q^{we}(\theta)\delta_{ts} \\ E\{x(0) x^T(0)\} &= \Pi(\theta) \end{aligned} \tag{48}$$

where θ is parameter vector. Since we have assumed that $w_\theta(t)$ and $e_\theta(t)$ are uncorrelated, the matrix $Q^{we} = 0$. Matrices $A(\theta)$, $B(\theta)$,$G_0(\theta)$, $G_1(\theta)$, $G_2(\theta)$ and $Q^w(\theta)$, $Q^e(\theta)$ depend on θ in an arbitrary way. It is assumed that the matrix elements are differentiable with respect to θ.

The extremum control objective is to make the steady-state response of the output at maximum or minimum. The control law can be found by optimizing the performance function.

For an input-output model, the static response between the input and output is given by

$$
\begin{aligned}
A(1)z &= B(1)u \\
y &= g_0 + g_1 z + g_2 z^2
\end{aligned} \tag{49}
$$

which gives the static output

$$
\begin{aligned}
y &= g_0 + g_1 \frac{B(1)}{A(1)} u + g_2 \left(\frac{A(1)}{B(1)} u\right)^2 \\
&= g_0 + g_1 \frac{B(1)}{A(1)} u + g_2 \frac{B^2(1)}{A^2(1)} u^2
\end{aligned} \tag{50}
$$

the extremum control law can be achieved by maximizing or minimizing the output of the static response

$$
u(t) = -\frac{g_1 A(1)}{2 g_2 B(1)} + u_p(t) \tag{51}
$$

where $u_p(t)$ is a perturbation signal.

If the parameters have to be estimated, then the control law will be modified by

$$
u(t) = -\frac{\hat{g}_1 \hat{A}(1)}{2 \hat{g}_2 \hat{B}(1)} + u_p(t) \tag{52}
$$

For the output nonlinearity, the complexity of the problem will also depend on which of the variables in the process can be measured. If it is possible to measure the intermediate signal $z(t)$ between the linear part and nonlinear part, which is assumed by most people, the complexity will be much simplified. But more search is needed to find out how to handle systems where the intermediate signal is not available. With the intermediate signal is not measurable, the problem is more difficult. These two situations will be discussed separately in the following sections.

3.2 The intermediate signal is measurable

Since the intermediate signal can be measured, it is also possible to do the system identification separately for the linear part and nonlinear part instead of estimate the performance function. An adaptive extremum controller could again be used to keep the output of the linear part around the estimated position of the extremum of the nonlinearity.

The RELS algorithm has been given in (17). The model of linear part used in parameter estimation is written as linear regression

$$
z(t) = \varphi_1^T(t)\theta_1 + \epsilon(t) \tag{53}
$$

with

$$
\begin{aligned}
\varphi_1(t) &= [-z(t-1), \cdots - z(t-n_a), u(t-d), \cdots u(t-d-n_b), \\
&\quad\ \epsilon(t-1), \cdots, \epsilon(t-n_c)]^T \\
\theta_1 &= [a_1, \cdots a_{na}, b_1, \cdots, b_{nb}, c_1, \cdots, c_{nc}]^T
\end{aligned} \tag{54}
$$

The model of nonlinear part is

$$
y(t) = \varphi_2^T(t)\theta_2 + \epsilon(t) \tag{55}
$$

where

$$
\begin{aligned}
\varphi_2(t) &= [1,\ z(t),\ z^2(t)]^T \\
\theta_2(t) &= [g_0,\ g_1,\ g_2]^T
\end{aligned} \tag{56}
$$

In this case, the only difference between the output nonlinearity and input nonlinearity is that its input is not determined directly, but through linear dynamics.

Example 3.1 *An unknown nonlinear dynamics*

Consider a system with linear part

$$
z(t) + az(t-1) = bu(t-1) + e_1(t) \tag{57}
$$

and nonlinear part

$$y(t) = g_0 + g_1 z(t) + g_2 z^2(t) + e_2(t) \tag{58}$$

where $a = -0.8$, $b = 1$, $g_0 = 100$, $g_1 = 2$ *and* $g_2 = -0.01$. $e_1(t)$ *and* $e_2(t)$ *are zero mean white noise with variance* $\sigma_{e1}^2 = 0.2^2$ *and* $\sigma_{e2}^2 = 0.01^2$. *The maximum attainable value of the system output is 200, the corresponding optimal input is 20.*

In this example we assume that both parameters in linear part and nonlinear part are unknown. Since we have assumed that the intermediate signal $z(t)$ *can be measured, the identification for linear dynamics and nonlinear dynamics can be done separately.*

The model used to estimate parameters in linear part is given in (53) - (54), the regression vector and parameter vector can be written as

$$\varphi_1(t) = [-z(t-1), \quad u(t-1)]^T$$

$$\hat{\theta}_1 = [\hat{a}, \quad \hat{b}]^T$$

and the model used to estimate parameters in nonlinear part is given in (55) - (56), which is written as

$$\varphi_2(t) = [1, \; z(t), \; z^2(t)]^T$$

$$\hat{\theta}_2 = [\hat{g}_0, \; \hat{g}_1, \; \hat{g}_2]^T$$

The extremum control law is identical in form with example 3.1, but parameters a *and* b *are replaced by estimated value* $\hat{a}$ *and* $\hat{b}$.

$$u(t) = -\frac{\hat{g}_1(t)(1 + \hat{a}(t))}{2\hat{g}_2(t)\hat{b}(t)} + u_p(t) \tag{59}$$

where the perturbation signal $u_p(t)$ *is PRBS signal, which will ensure the identifiability of parameter* θ_1. *The noise on the intermediate signal* $e_1(t)$ *will ensure the identifiability of parameter* θ_2.

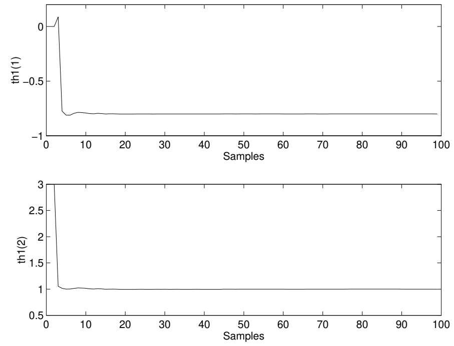

Figure 7: The estimated parameters.

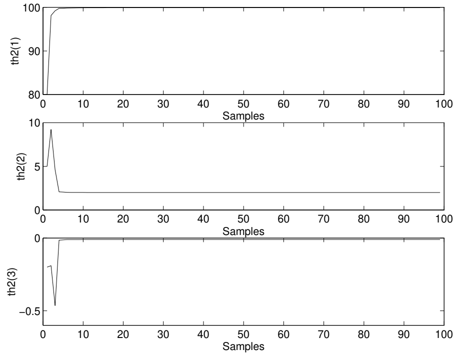

Figure 8: The estimated parameters.

Simulations are shown in Figure 7 - 9. The estimated parameters in linear part is given in Figure 7, where $th1(1) = \hat{a}$ *and* $th1(2) = \hat{b}$. *The estimated parameters in nonlinear part is given Figure 8, where* $th2(1) = \hat{g}_0$, $th2(2) = \hat{g}_1$ *and* $th2(3) = \hat{g}_3$. *The input* $u(t)$, *output* $y(t)$ *and intermediate signal* $z(t)$ *are given in Figure 9. The perturbation on the input signal comes from* $u_p(t)$. *Simulation shows good convergence property of adpative extremum control algorithm.*

$\diamond$

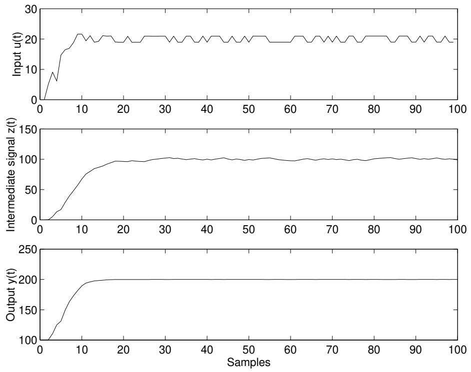

Figure 9: Input, output and intermediate signal

3.3 The intermediate signal is not measurable

If the intermediate signal $z(t)$ can not be measured, the problem will be very complex. It has been shown in section 3.1 that the input-output model will be nonlinear in parameters if $z(t)$ is unmeasurable. It is then very difficult to estimate parameters in this case. Therefore a state-space model is taken into account for state and parameter estimation.

For this type of process, sometimes no unique solution to the estimation problem exists, because only certain combinations of parameters are identifiable. The parameter identifiability restricts the number of parameters which can be estimated simultaneously. In this section it is assumed that the correct model structure is available.

Different estimation algorithm will be discussed in this section. The extended Kalman filter (EKF) method and recursive prediction error method (RPEM) seem to be desirable to the state and parameter estimation for a nonlinear process. A modified recursive prediction error method based on a line-search strategy is developed. The performance and robustness of different online state and parameter estimation strategies for the nonlinear state-space model will be investigated by simulation examples.

3.3.1 The EKF as a parameter estimator for the nonlinear system

The extended Kalman filter (EKF) is an approximate filter for nonlinear systems, based on first-order linearization. Its use for joint parameter and state estimation problem for linear systems with unknown parameters is well known and widely spread. The algorithm and convergence analysis have been given by Ljung [Ljung 79b].

It is a fairly natural thing to include the unknown parameters in the state vector when the unknown parameters are thought of random variables. Based on observations of other random variables that are correlated with parameters, we may infer information about their values. Since the unobserved state vector is assumed to be correlated with the output of the system, the value of the state vector can be estimated based on the observation of output.

A general nonlinear state-space model with output nonlinearity has been given by (47) - (48). We suppose that the true system dynamics can be described in term of a parameter vector θ, which is considered to be a random vector with a certain prior distribution. Matrices $Q^w(\theta)$ and $Q^e(\theta)$ in nonlinear model (47) - (48) are assumed to be independent of θ, but are chosen fixed in some ad hoc way in the remainder of this section. This corresponds to the fact that noise characteristics are independent of the state, i.e., $w_\theta(t) = w(t)$, $e_\theta(t) = e(t)$ and $Q^w(\theta) = Q^w$, $Q^e(\theta) = Q^e$, $Q^{we}(\theta) = 0$.

The unknown parameter vector θ in nonlinear model is obtained by extending the state vector x with the parameter vector θ

$$\underline{x}(t) = \begin{pmatrix} x(t) \\ \theta(t) \end{pmatrix} \tag{60}$$

Then the nonlinear state-space model will be

$$\begin{aligned} \underline{x}(t+1) &= \underline{f}(\underline{x}(t), u(t)) + \begin{pmatrix} w(t) \\ 0 \end{pmatrix} \\ y(t) &= h(\underline{x}(t)) + e(t) \end{aligned} \tag{61}$$

where

$$
\begin{aligned}
\underline{f}(\underline{x}(t), u(t)) &= \begin{bmatrix} f(x(t), u(t), \theta(t)) \\ \theta(t) \end{bmatrix} \\
h(\underline{x}(t)) &= h(x(t), \theta(t))
\end{aligned}
\tag{62}
$$

The EKF algorithm for this joint state and parameter model is

$$
\underline{\hat{x}}(t+1) = \underline{f}(\underline{\hat{x}}(t), u(t)) + K(t)[y(t) - h(\underline{\hat{x}}(t))] \tag{63}
$$

$$
\underline{\hat{x}}(0) = \underline{\hat{x}}_0
$$

$$
\begin{aligned}
K(t) &= [F(\underline{\hat{x}}(t), u(t))\bar{P}(t)H^T(\underline{\hat{x}}(t))] \\
&\quad \cdot [H(\underline{\hat{x}}(t))\bar{P}(t)H^T(\underline{\hat{x}}(t)) + Q^e]^{-1}
\end{aligned}
\tag{64}
$$

$$
\begin{aligned}
\bar{P}(t+1) &= F(\underline{\hat{x}}(t), u(t))\bar{P}(t)F^T(\underline{\hat{x}}(t), u(t)) + \bar{Q}^w \\
&\quad - K(t)[H(\underline{\hat{x}}(t))\bar{P}(t)H^T(\underline{\hat{x}}(t)) + Q^e]K^T(t)
\end{aligned}
\tag{65}
$$

$$
\bar{P}(0) = \bar{P}_0
$$

where

$$
\begin{aligned}
F(\underline{\hat{x}}(t), u(t)) &= \frac{\partial}{\partial \underline{x}} f(\underline{x}, u)\Big|_{\underline{x}=\underline{\hat{x}}(t)} \\
&= \begin{bmatrix} A(\hat{\theta}(t)) & M(\hat{\theta}(t), \hat{x}(t), u(t)) \\ 0 & I \end{bmatrix}
\end{aligned}
\tag{66}
$$

$$
\begin{aligned}
H(\underline{\hat{x}}(t)) &= \frac{\partial}{\partial \underline{x}} h(\underline{x})\Big|_{\underline{x}=\underline{\hat{x}}(t)} \\
&= [C(\hat{\theta}(t)) \quad D(\hat{\theta}(t), \hat{x}(t))]
\end{aligned}
\tag{67}
$$

$$
\bar{Q}^w = \begin{bmatrix} Q^w & 0 \\ 0 & 0 \end{bmatrix} \tag{68}
$$

$$
\begin{aligned}
\underline{\hat{x}}_0 &= \begin{bmatrix} \hat{x}_0 \\ \hat{\theta}_0 \end{bmatrix} \\
\bar{P}_0 &= \begin{pmatrix} \Pi(\hat{\theta}_0) & 0 \\ 0 & \Sigma_0 \end{pmatrix}
\end{aligned}
\tag{69}
$$

and

$$
\begin{aligned}
A(\hat{x}(t), \hat{\theta}(t)) &= \frac{\partial}{\partial x} f(x(t), u(t), \hat{\theta}(t))\Big|_{x=\hat{x}(t)} (a \;\; n_x|n_x \;\; matrix) \\
M(\hat{x}(t), u(t), \hat{\theta}(t)) &= \frac{\partial}{\partial \theta} f(\hat{x}(t), u(t), \theta(t))\Big|_{\theta=\hat{\theta}(t)} (a \;\; n_x|n_\theta \;\; matrix) \\
C(\hat{x}(t), \hat{\theta}(t)) &= \frac{\partial}{\partial x} h(x(t), \hat{\theta}(t))\Big|_{x=\hat{x}(t)} (a \;\; n_y|n_x \;\; matrix) \\
D(\hat{x}(t), \hat{\theta}(t)) &= \frac{\partial}{\partial \theta} h(\hat{x}(t), \theta(t))\Big|_{\theta=\hat{\theta}(t)} (a \;\; n_y|n_\theta \;\; matrix)
\end{aligned}
\tag{70}
$$

The algorithm can be separated into two steps if we write matrix $K_{\underline{x}}(t)$ and $\bar{P}$ as

$$
K_{\underline{x}}(t) = \begin{bmatrix} K_x(t) \\ K_\theta(t) \end{bmatrix}
$$

$$
\bar{P}(t) = \begin{bmatrix} P_1(t) & P_2(t) \\ P_2^T(t) & P_3(t) \end{bmatrix}
$$

Parameters can then be computed in a parameter estimator, and the resulting model is used in a state estimator. The algorithm is summarized as follows

$$
\begin{aligned}
&\hat{x}(t+1) = f(\hat{x}(t), u(t), \hat{\theta}(t)) + K_x(t)[y(t) - h(\hat{x}(t), \hat{\theta}(t))] \\
&\hat{x}(0) = \hat{x}_0 \\
&\hat{\theta}(t+1) = \hat{\theta}(t) + K_\theta(t)[y(t) - h(\hat{x}(t), \hat{\theta}(t))] \\
&\hat{\theta}(0) = \hat{\theta}_0 \\
&K_x(t) = [A_t P_1(t) C_t^T + M_t P_2^T(t) C_t^T + A_t P_2(t) D_t^T \\
&\qquad\qquad + M_t P_3(t) D_t^T + Q^{we}] S_t^{-1} \\
&K_\theta(t) = [P_2^T(t) C_t^T + P_3(t) D_t^T] S_t^{-1} \\
&S_t = C_t P_1(t) C_t^T + C_t P_2(t) D_t^T + D_t P_2^T(t) C_t^T + D_t P_3(t) D_t^T + Q^e \\
&P_1(t+1) = A_t P_1(t) A_t^T + A_t P_2(t) M_t^T + M_t P_2^T(t) A_t^T + M_t P_3(t) M_t^T \\
&\qquad\qquad - K_x(t) S_t K_x^T(t) + Q^w \\
&P_1(0) = \Pi(\hat{\theta}_0) \\
&P_2(t+1) = A_t P_2(t) + M_t P_3(t) - K_x(t) S_t K_\theta^T(t) \\
&P_2(0) = 0 \\
&P_3(t+1) = P_3(t) - K_\theta(t) S_t K_\theta^T(t) \\
&P_3(0) = \Sigma_0
\end{aligned}
\tag{71}
$$

and $\hat{\theta}_0$, $\hat{x}_0$ and Σ_0 represent some *a priori* information about parameter vector θ and state vector x.

In the algorithm

$$\begin{aligned} A_t &= A(\hat{x}(t), \hat{\theta}(t)) \\ C_t &= C(\hat{x}(t), \hat{\theta}(t)) \\ D_t &= D(\hat{x}(t), \hat{\theta}(t)) \\ M_t &= M(\hat{x}(t), u(t), \hat{\theta}(t)) \end{aligned} \tag{72}$$

It is a well-known fact that the behavior of EKF as joint state and parameter estimator is sensitive to a priori statistics assumptions, and divergence may occur if the initial state and parameter estimates are not sufficiently good. The algorithm may converge only if the assumption about the noise structure of the model are in accordance with those of the true system. Otherwise the estimates will be biased. Ljung [Ljung 79b] has proved that the lack of coupling between state update matrix $K_x(t)$ and parameter vector θ, i.e., $[dK_x/d\theta]\epsilon(t) = 0$ is assumed in the algorithm, will lead to divergence of the estimation.

Example 3.2 *Let the model (57) - (58) in example 3.1 be given by a state-space form*

$$\begin{aligned} x(t+1) &= ax(t) + bu(t) + w(t) \\ y(t) &= h(\theta, x(t)) + e(t) \end{aligned} \tag{73}$$

and

$$h(\theta, x(t)) = g_0 + g_1 x(t) + g_2 x^2(t)$$

where $a = 0.8$, $b = 1$, $g_0 = 100$, $g_1 = 2$ *and* $g_2 = -0.01$. $w(t)$ *and* $e(t)$ *are uncorrelated zero mean white noise with variance* $\sigma_w^2 = 0.2^2$ *and* $\sigma_e^2 = 0.01^2$.

It is assumed that the dynamics of linear part is known, but the dynamics of nonlinear part has to be estimated. The variances of measurement noise and the process noise are chosen in accordance with those of the true system. It is also assumed that the intermediate signal $z(t)$ *is not able to measure.*

Only the parameters in output equation has to be estimated, the estimated parameter vector θ *is*

$$\theta = [g_0, \quad g_1, \quad g_2]^T$$

The extended Kalman filter algorithm (71) - (72) is applied on model (73) with

$$\begin{aligned} A_t &= a \\ M_t &= [0, \quad 0, \quad 0] \\ C_t &= \hat{g}_1(t) + 2\hat{g}_2(t)\hat{x}(t) \\ D_t &= [1, \quad \hat{x}(t), \quad \hat{x}^2(t)] \end{aligned} \tag{74}$$

The resulting algorithm will be

$$\hat{x}(t+1) = a\hat{x}(t) + bu(t) + K_x(t)[y(t) - h(\hat{\theta}(t), \hat{x}(t))] \tag{75}$$

$$\hat{x}(0) = \hat{x}_0$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K_\theta(t)[y(t) - h(\hat{\theta}(t), \hat{x}(t))] \tag{76}$$

$$\hat{\theta}(0) = \hat{\theta}_0$$

where

$$h(\hat{\theta}(t), \hat{x}(t)) = \hat{g}_0(t) + \hat{g}_1(t)\hat{x}(t) + \hat{g}_2(t)\hat{x}^2(t) \tag{77}$$

The state update matrix $K_x(t)$ *and the parameter update matrix* K_θ *can be computed by using (71).*

The extremum control law based on the estimated model will be

$$u(t) = -\frac{\hat{g}_1(t)(1+a)}{2\hat{g}_2(t)b}$$

Simulation results are given in Figure 10 - Figure 11. The estimates parameters are given in Figure 10, where $th(1) = \hat{g}_0$, $th(2) = \hat{g}_1$ *and* $th(3) = \hat{g}_2$. *simulation shows that the estimated parameters converge to the correct values, and system output achieve the extremum value by tuning the extremum control law.*

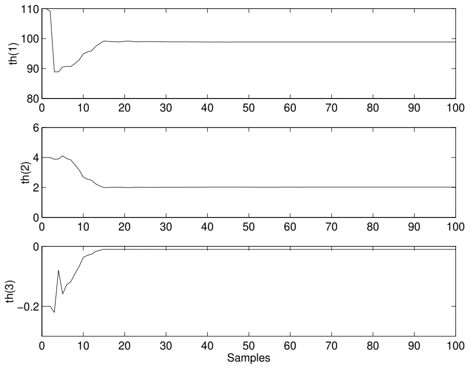

Figure 10: The estimated parameters.

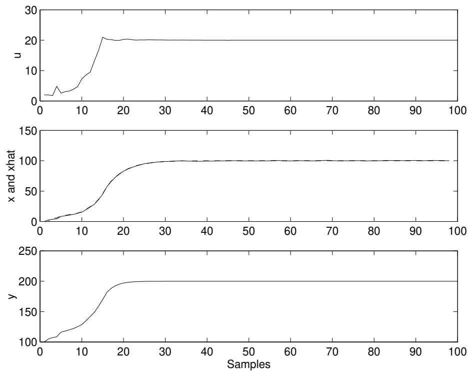

Figure 11: The estimated parameters.

The convergence requires the condition that the initial values of the estimated parameters should be inside a convergence area (not too far from the true values), and the initial variance matrix of the estimated parameters $P_3(0)$ in the algorithm (71) - (72) can not be assumed too large in order to bound the estimated parameters at first several steps, otherwise the eigenvalues of matrix $A_t - K_x C_t$ will be far from the unit circle. This will cause the divergence of the process. Good initial state estimates are also important for the convergence properties. The increased number of parameters to be

estimated frequently results in reduced robustness.

The EKF as a joint state and parameter estimator works well when a good model and reasonable parameter estimates is available, the nonlinearity is weak and noise assumptions are close to the truth.

As is stated in [Ljung 79a], the reason for divergence of EKF as a joint parameter and state estimation for a linear system is that the effect on the Kalman gain K_x of a change in θ is not taken care of. For cases where steady-state Kalman gain does not depend on θ, we will have good convergence properties.

If the model parameter converges to some constant= θ, it means that P_2 and P_3 would tend to zero, the steady-state can then be found

$$\bar{P}_1(\theta) = A(\theta)\bar{P}_1(\theta)A^T(\theta) + Q^w - \bar{K}_x(\theta)\bar{S}(\theta)\bar{K}_x^T(\theta) \quad (78)$$

$$\bar{S}(\theta) = C(\theta)\bar{P}_1(\theta)C^T(\theta) + Q^e \quad (79)$$

$$\bar{K}_x(\theta) = [A(\theta)\bar{P}_1(\theta)C^T(\theta) + Q^{we}]\bar{S}^{-1}(\theta) \quad (80)$$

It is easy to see that the steady state Kalman gain matrix of a linear system is independent of state variables which appear only in M_t and D_t matrices. For a nonlinear system, since the state update matrix $K_x(t)$ in the algorithm (71) - (72) is calculated by using a linearized model, which is actually an approximation of the true model. Parameters of the linearized model depend not only on the estimated parameters but also on the estimated states (e.g., matrix C_t in (74)). Therefore a steady-state Kalman gain matrix of a nonlinear system depends on estimated state vector. This will cause the algorithm more sensitive. The model parameter estimates will be poor when the estimated states are not close to the true values. The poor model parameter estimates will make the algorithm to diverge, since $[dK_x/d\theta]\epsilon(t)$ is not accounted for in the algorithm, and sensitivity matrix $\psi(t)$ has a tendency to be incorrect.

$\diamond$

3.3.2 The RPEM applied to the innovations model

It is of course unrealistic to assume that the noise structure is known, while the dynamics are unknown. Therefore, if the noise characteristic of the model is chosen ad hoc, then the system parameter estimates will in general be biased [Ljung 79b]. Ljung suggested a Newton-type stochastic gradient algorithm in his report.

A recursive Gauss-Newton algorithm algorithm which minimizes prediction error criterion will lead to a recursive prediction error method (RPEM). The RPEM is take into consideration, since Ljung [Ljung 79b] has proved that the RPEM applied to the joint state and parameter estimation problem in a stationary innovations state-space model has better convergence properties than the corresponding EKF. The coupling between update state matrix $K_x(t)$ and parameter vector θ is included in the algorithm.

The recursive prediction error method and convergence analysis for a linear system is given by Ljung [Ljung and Söderström 83]. Then the question is whether this result holds for a nonlinear system.

A recursive algorithm derived for estimation of model parameters by minimizing a prediction error criterion

$$V(\theta) = \frac{1}{2} E\epsilon^T(t,\theta)\Lambda^{-1}\epsilon(t,\theta) \tag{81}$$

A model for a nonlinear system is given by equation (47) - (48), whereas RPEM will apply to a nonlinear innovations model, which is written as

$$\begin{aligned} x(t+1) &= f(x(t),u(t),\theta(t)) + K_x(\theta(t))\epsilon(t) \\ y(t) &= h(x(t),\theta(t)) + \epsilon(t) \end{aligned} \tag{82}$$

with

$$E\epsilon(t)\epsilon^T(t) = \Lambda\delta_{ts}$$

and

$$\begin{aligned} Q^w(\theta) &= K_x(\theta)\Lambda K_x^T(\theta) \\ Q^e(\theta) &= \Lambda \\ Q^{we}(\theta) &= K_x(\theta)\Lambda \end{aligned}$$

In the above model the steady-state Kalman gain is explicitly parameterized. Since only the Kalman gain is important for the algorithm, it then seems reasonable to parameterized the steady-state Kalman gain rather than the covariance matrices.

The recursive prediction error algorithm applied to the innovations model (82) can be summarized by

$$\begin{aligned} &\hat{y}(t) = h(\hat{x}(t),\hat{\theta}(t)) \\ &\epsilon(t) = y(t) - \hat{y}(t) \\ &\hat{\Lambda}(t) = \hat{\Lambda}(t-1) + \gamma(t)[\epsilon(t)\epsilon^T(t) - \hat{\Lambda}(t-1)] \\ &S(t) = \psi^T(t)P(t-1)\psi(t) + \lambda(t)\hat{\Lambda}(t) \\ &K_\theta(t) = P(t-1)\psi(t)S^{-1}(t) \\ &\hat{\theta}(t) = \hat{\theta}(t-1) + K_\theta(t)\epsilon(t) \\ &P(t) = [P(t-1) - K_\theta(t)S(t)K_\theta^T(t)]/\lambda(t) \\ &\hat{x}(t+1) = f(\hat{x}(t),u(t),\hat{\theta}(t)) + K_x\epsilon(t) \\ &W(t+1) = [A_t - K_xC_t]W(t) + \bar{M}_t - K_xD_t \\ &\psi(t+1) = [C_tW(t+1) + D(\hat{\theta}(t),\hat{x}(t+1))]^T \end{aligned} \tag{83}$$

where

$$\begin{aligned} A_t &= A(\hat{x}(t),\hat{\theta}(t)) \\ C_t &= C(\hat{x}(t),\hat{\theta}(t)) \\ D_t &= D(\hat{x}(t),\hat{\theta}(t)) \\ \bar{M}_t &= \bar{M}(\hat{x}(t),u(t),\epsilon(t),\hat{\theta}(t)) \\ K_x &= K_x(\hat{\theta}(t)) \end{aligned}$$

Matrices $A(\hat{x}(t),\hat{\theta}(t))$, $C(\hat{x}(t),\hat{\theta}(t))$ and $D(\hat{x}(t),\hat{\theta}(t))$ are defined in equation (70), but matrix $\bar{M}(\hat{x}(t),u(t),\epsilon(t),\hat{\theta}(t))$ has to be modified by including

$K_x(\theta)\epsilon(t)$

$$\bar{M}(\hat{x}(t), u(t), \epsilon(t), \hat{\theta}(t)) = \frac{\partial}{\partial\theta} f(\hat{x}(t), u(t), \theta(t))\Big|_{\theta=\hat{\theta}(t)} + \frac{\partial}{\partial\theta} K_x(\theta)\epsilon(t)\Big|_{\theta=\hat{\theta}(t)} \tag{84}$$

In the above algorithm, $\lambda(t) = \gamma(t-1)\frac{1-\gamma(t)}{\gamma(t)}$.

According to Ljung, a RPE algorithm should contains a projection into the stability region. This is used to ensure the stability of the algorithm. The stability region for the predictor is given by

$$D_s = \{\theta | A(\theta) - K_x(\theta)C(\theta) \textit{ has all eigenvalues strictly inside the unit circle}\} \tag{85}$$

For the convergence results, $\hat{\theta}(t) \in D_s$ (at least infinitely often).

A very simple and natural feather is to make the recursion in RPE algorithm (83) in two steps as a measurement update and a time update and to make a relinearization in between. Such feathers may have a major influence on transient behavior and convergence rate of the algorithm, but they will not effect the convergence results.

If θ is a correct description of the system, the time update in two step RPE algorithm will be

$$\hat{x}(t+1|t) = f(\hat{x}(t|t), u(t), \theta) \tag{86}$$

and date update will be

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + k_x(\theta)\epsilon(t+1) \tag{87}$$

the prediction error is

$$\epsilon(t+1) = y(t+1) - h(\hat{x}(t+1|t), \theta) \tag{88}$$

Differentiating (86) and (87) gives

$$\frac{d}{d\theta}\hat{x}(t+1|t) = A(\hat{x}(t|t), \theta)\frac{d}{d\theta}\hat{x}(t|t) + M(\hat{x}(t|t), u(t), \theta) \tag{89}$$

and

$$\begin{aligned} \frac{d}{d\theta}\hat{x}(t+1|t+1) &= \frac{d}{d\theta}\hat{x}(t+1|t) - k_x(\theta)C(\hat{x}(t+1|t), \theta)\frac{d}{d\theta}\hat{x}(t+1|t) \\ &\quad - k_x(\theta)D(\hat{x}(t+1|t), \theta) + [\frac{d}{d\theta}k_x(\theta)]\epsilon(t+1) \end{aligned} \tag{90}$$

matrices $A(\hat{x}(t|t), \theta)$, $M(\hat{x}(t|t), u(t), \theta)$ and $C(\hat{x}(t|t), \theta)$ are defined by

$$\begin{aligned} A(\hat{x}(t|t), \theta) &= \tfrac{\partial}{\partial x} f(x(t|t), u(t), \theta)\big|_{x=\hat{x}(t|t)} \\ M(\hat{x}(t|t), u(t), \theta) &= \tfrac{\partial}{\partial \theta} f(\hat{x}(t|t), u(t), \theta)\big|_{\theta=\theta} \\ C(\hat{x}(t+1|t), \theta) &= \tfrac{\partial}{\partial x} h(x(t+1|t), \theta)\big|_{x=\hat{x}(t+1|t)} \\ D(\hat{x}(t+1|t), \theta) &= \tfrac{\partial}{\partial \theta} h(\hat{x}(t+1|t), \theta)\big|_{\theta=\theta} \end{aligned} \tag{91}$$

Let $W(t+1|t)$ be matrix $d/d\theta\hat{x}(t+1|t)$ and $W(t+1|t+1)$ be matrix $d/d\theta\hat{x}(t+1|t+1)$. respectively, (89) and (90) can be rewritten as

$$W(t+1|t) = A(\hat{x}(t|t), \theta)W(t|t) + M(\hat{x}(t|t), u(t), \theta) \tag{92}$$

$$\begin{aligned} W(t+1|t+1) &= [I - k_x(\theta)C(\hat{x}(t+1|t), \theta)]W(t+1|t) \\ &\quad - k_x(\theta)D(\hat{x}(t+1|t), \theta) + [\tfrac{d}{d\theta}k_x(\theta)]\epsilon(t+1) \end{aligned} \tag{93}$$

and $\psi(t+1) = -d/d\theta\epsilon(t+1)$ will be

$$\psi(t+1) = C(\hat{x}(t+1|t), \theta)W(t+1|t) + D(\hat{x}(t+1|t), \theta) \tag{94}$$

The RPE algorithm in two steps is summarized

$$\begin{aligned} &\hat{x}(t|t-1) = f(\hat{x}(t-1|t-1), u(t-1), \hat{\theta}(t-1)) \\ &W(t|t-1) = A_t W(t-1|t-1) + M_t \\ &\hat{y}(t) = h(\hat{x}(t|t-1), \hat{\theta}(t-1)) \\ &\epsilon(t) = y(t) - \hat{y}(t) \\ &\psi(t) = C_t W(t|t-1) + D_t \\ &\hat{\Lambda}(t) = \hat{\Lambda}(t-1) + \gamma(t)[\epsilon(t)\epsilon^T(t) - \hat{\Lambda}(t-1)] \\ &S(t) = \psi^T(t)P(t-1)\psi(t) + \lambda(t)\hat{\Lambda}(t) \\ &K_\theta(t) = P(t-1)\psi(t)S^{-1}(t) \end{aligned} \tag{95}$$

$$
\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + K_\theta(t)\epsilon(t) \\
P(t) &= [P(t-1) - K_\theta(t)S(t)K_\theta^T(t)]/\lambda(t) \\
\hat{x}(t|t) &= \hat{x}(t|t-1) + k_x(\hat{\theta}(t))\epsilon(t) \\
W(t|t) &= [I - k_x(\hat{\theta}(t))C_t)]W(t|t-1) - k_x(\hat{\theta}(t))D_t \\
&\quad + [\frac{d}{d\theta}k_x(\hat{\theta}(t))]\epsilon(t)
\end{aligned}
$$

where

$$
\begin{aligned}
A_t &= A(\hat{x}(t-1|t-1), \hat{\theta}(t-1)) \\
M_t &= M(\hat{x}(t-1|t-1), u(t-1), \hat{\theta}(t-1)) \\
C_t &= C(\hat{x}(t|t-1), \hat{\theta}(t-1)) \\
D_t &= D(\hat{x}(t|t-1), \hat{\theta}(t-1))
\end{aligned}
$$

3.4 The modified recursive prediction error method

A modified recursive prediction error method based on line search strategy was suggested by Ljungquist [Ljungquist and Balchen 93]. Line search techniques are widely used to solve optimization problems. In this algorithm the line search performs in the computed search direction to find an optimal stepsize during each iteration. Due to the nonlinearities a line search strategy is used to avoid divergence when the search direction is profitable.

When the search direction has been determined for parameter estimator

$$\theta(t) = \hat{\theta}(t-1) + \alpha(t)K_\theta(t)\epsilon(t) \tag{96}$$

we have to decide how long the step in this direction should be.

The optimal parameter updating step size, which is found by performing the line search approach, can be computed by minimizing the one-sample criterion

$$V_j(t) = \epsilon_j(t)^T\hat{\Lambda}^{-1}(t)\epsilon_j(t) + \Delta\theta_j(t)^T P^{-1}(t)\Delta\theta_j(t) \tag{97}$$

where

$$\Delta\theta_j(t) = \theta_j(t) - \hat{\theta}(t-1) \tag{98}$$

$\Lambda(t)$ and $P(t)$ is positive definite matrix.

During the line search, a new parameter vector is computed according to

$$\theta_j(t) = \hat{\theta}(t-1) + \alpha_j(t)K_\theta(t)\epsilon(t) \tag{99}$$

where $\alpha_j(t)$ is the update stepsize. Once the search direction at each discrete sample time is determined, the value of criterion (97) is a function of the stepsize α alone. Then the optimal stepsize can be obtained by minimizing the criterion

$$\alpha_0(t) = \min_\alpha V(\alpha, t) \tag{100}$$

α_0 is optimal stepsize, which results in the smallest criterion for a given search direction. Different stepsize will result in different parameters and different prediction error.

In general, minimum point of the cost function (97) can not be solved by analytically, therefore a numerical method for estimating a value α_0 must be used. A quadratic interpolation method is chosen to estimate the optimal step size α_0. The α_0 is obtained by minimizing an approximating polynomial of $V(\alpha, t)$ in α of degree two. Since a one sample criterion is used, the criterion can easily be computed for each stepsize.

The principle of the approach is illustrated in Figure 12.

Let α_1, α_2 and α_3 be given distinct values of α, the $V(\alpha_1)$, $V(\alpha_2)$ and $V(\alpha_3)$ can be calculated according to criterion (97), and the optimal α_0 can be obtained by

$$\alpha_0 = \frac{1}{2}\frac{(\alpha_2^2 - \alpha_3^2)V_1 + (\alpha_3^2 - \alpha_1^2)V_2 + (\alpha_1^2 - \alpha_2^2)V_3}{(\alpha_2 - \alpha_3)V_1 + (\alpha_3 - \alpha_1)V_2 + (\alpha_1 - \alpha_2)V3} \tag{101}$$

where $Vj = V(\alpha_j)(j = 1, 2, 3)$.

A line search strategy, which ensures that new parameter vector $\hat{\theta}(t)$ does not result in a large criterion than the value obtained from previous vector $V(\alpha = 0)$, can handle the nonlinear estimation problem.

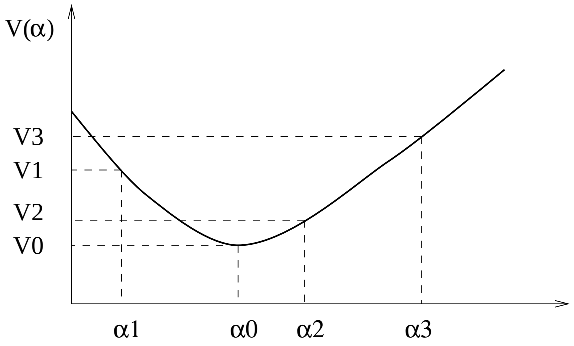

Figure 12: Optimal stepsize

A recursive line search prediction error method is summarized as follows.

1. $\hat{x}(t|t-1) = f(\hat{x}(t-1|t-1), u(t-1), \hat{\theta}(t-1))$
   $W(t|t-1) = A_t W(t-1|t-1) + M_t$

2. $\hat{y}(t) = h(\hat{x}(t|t-1), u(t), \hat{\theta}(t-1))$
   $\epsilon(t) = y(t) - \hat{y}(t)$
   $\psi(t) = C_t W(t|t-1) + D_t$

3. $\hat{\Lambda}(t) = \hat{\Lambda}(t-1) + \gamma(t)[\epsilon(t)\epsilon^T(t) - \hat{\Lambda}(t-1)]$
   $S(t) = \psi^T(t)P(t-1)\psi(t) + \lambda(t)\hat{\Lambda}(t)$
   $K_\theta(t) = P(t-1)\psi(t)S^{-1}(t)$

4. line search algorithm to find optimal stepsize α_0.
   Choose $\alpha_1 = 0$, $\alpha_2 = 0.4$ and $\alpha_3 = 0.7$,
   For $j = 1:3$
   $\hat{\theta}_j(t) = \hat{\theta}(t-1) + \alpha_j(t)K_\theta(t)\epsilon(t)$
   $\Delta\hat{\theta}_j(t) = \hat{\theta}_j(t) - \hat{\theta}(t-1)$
   $\hat{x}_j(t-1|t-1) = \hat{x}(t-1|t-2) + k_x(\hat{\theta}_j(t))\epsilon(t-1)$
   $\hat{x}_j(t|t-1) = f(\hat{x}_j(t-1|t-1), u(t-1), \hat{\theta}_j(t))$
   $\epsilon_j(t) = y(t) - h(\hat{x}_j(t|t-1), u(t), \hat{\theta}_j(t))$
   Compute $V_j(t)$ according to (97)

   end
   α_0 can be calculated by equation (101).

5. $\hat{\theta}(t) = \hat{\theta}(t-1) + \alpha_0(t)K_\theta(t)\epsilon(t)$

6. $P(t) = [P(t-1) - K_\theta(t)S(t)K_\theta^T(t)]/\lambda(t)$

7. $\hat{x}(t|t) = \hat{x}(t|t-1) + k_x(\hat{\theta}(t))\epsilon(t)$
   $W(t|t) = [I - k_x(\hat{\theta}(t))C_t)]W(t|t-1) - k_x(\hat{\theta}(t))D_t$
   $+[\frac{d}{d\theta}k_x(\hat{\theta}(t))]\epsilon(t)$

8. $t = t+1$ and go to 1.

The main difference between a recursive line search prediction error method and ordinary RPEM is that a line search is performed in the computed search direction and that a regularized estimation criterion is used. The algorithm increases robustness and flexibility compared to ordinary RPEM, whereas the computing requirements are also increased compare to RPEM.

Example 3.3 *We take the model from example 3.2. It is assumed that we have not any knowledge about dynamics of the linear part, but dynamics of the nonlinear part is known. The variances matrices of the noise Q^w and Q^e are also assumed unknown. For model (73) an innovations model is given by*

$$\begin{aligned} x(t+1) &= ax(t) + bu(t) + K_x\epsilon(t) \\ y(t) &= h(t, x(t)) + e(t) \end{aligned} \tag{102}$$

where

$$h(t, x(t)) = g_0 + g_1x(t) + g_2x^2(t)$$

The estimated parameter vector θ will be

$$\theta = [a,\ \ b,\ \ k_x]^T$$

For the purpose of comparison, the two step prediction error method and the modified prediction error method with line search are implemented to estimate the parameters of the process. The extremum control law is then given based on the estimated model.

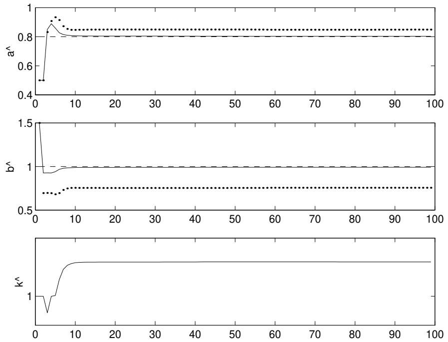

Figure 13: The estimated parameters.

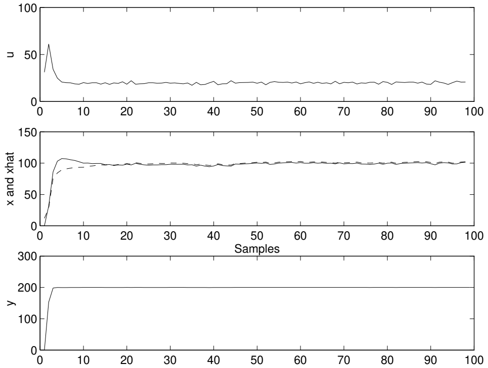

Figure 14: The estimated parameters.

The simulation results are presented in Figure 13 - 14. The dashed line presents the true parameter values of the system, the solid line is the estimated parameters by implementing the line search RPEM, and point line is the estimated parameters by using two step RPEM.

When the initial assumptions of the estimated parameters and states are not good enough, the two step RPEM will give a biased estimation, but the modified RPEM can give an unbiased estimation. The reason of biased estimation of two step RPEM is that the steady-state Kalman gain is parameterized. This actually implies that the algorithm in principle is based on the assumption that the state update matrix K_x is constant, or at least asymptotically constant. For nonlinear systems, K_x will vary rapidly as the input and the operational point change. Then if K_x is poor, the subsequent $W(t+1)$ and $\psi(t+1)$ will have a tendency to be incorrect. The estimator will be biased or diverge.

The modified RPEM performs better than the original RPEM. The robustness of the modified algorithm is higher, since too long and too short updating steps during transients are avoided. But the modified RPEM will increase the complexity and computational costs. Anyway the example indicates that the recursive line search prediction error method is preferable for a nonlinear system with nonlinearity at output. ◇

4 Conclusions

For a system with input nonlinearity, a Hammerstein model of the process with the input nonlinearity is linear in parameters. An on-line recursive estimation algorithm, e.g., recursive extended least squares (RELS) method or recursive maximum likelihood (RML) method can then be applied to identify the parameters of the model. The extremum control law is derived based on the estimated model. The behavior of the extremum controller has been shown by simulation examples, which keeps the process operating at, or in the vicinity of, the extremum point of the process output. A test perturbation signal is necessary to ensure the parameter identifiability, which has been proved by analyzing convergence properties by using ODE approach. Good convergence property of the estimator has been achieved by algorithm.

For the output nonlinearity two different cases have been discussed according to that the intermediate signal can be measured or not.

If the intermediate signal can be measured, The RELS algorithm is used to estimate the parameters for linear part and nonlinear part separately. The only difference between the output nonlinearity and input nonlinearity here is that its extremum control law will not be determined directly, but through linear dynamics.

If the intermediate signal is not measurable, the problem becomes quite complicated. Since the extremum control law depend heavily on the estimated model, the estimation problem becomes very important.

First of all a state-space model of the process is employed in order to implement EKF algorithm. Example 3.2 give the simulation for a process with unknown nonlinear dynamics, known linear dynamics and variances of the noises. Simulations show a good convergence properties of EKF algorithm if the assumed initial values of state and parameter estimates are good enough.

It is a well-known fact that the behavior of EKF as joint state and parameter estimator is sensitive to a priori statistics assumptions, and divergence may occur if the initial state and parameter estimates are not sufficiently good. Ljung [Ljung 79b] has proved that the lack of coupling between state update matrix $K_x(t)$ and parameter vector θ, i.e., $[dK_x/d\theta]\epsilon(t)$ is not included in the algorithm, will lead to divergence of the estimation.

For a nonlinear system, the EKF as a joint state and parameter estimator works well only when a good model and reasonable parameter estimates are available, the nonlinearity is weak and noise assumptions are close to the truth. Since the steady-state Kalman gain is achieved based on the linearized model which is actually an approximation of the true model. The linearized model will also lead to that the steady-state Kalman gain depends on estimated states. These will cause the algorithm more sensitive.

The improved convergence property of the recursive prediction error method (RPEM) for an innovations model has been proved by [Ljung 79b]. The

RPEM based on the innovations state-space model means that the algorithm in principle is based on the assumption that the state updating matrix K_x is constant, at least asymptotically. For a nonlinear system K_x will vary rapidly as the inputs and the operational point change. The RPEM may diverge when the estimate of K_x is poor.

A modified RPE algorithm based on line-search strategy is given in section 3.4. In this algorithm the line-search perform in the precomputed search direction to find an optimal stepsize during each iteration. Example 3.3 shows that the modified RPE algorithm gives a better performance than the original RPE algorithm. The robustness of modified algorithm is increased, since too long and too short steps are avoided.

References

[Bamberger and Isermann 78] W. Bamberger and R. Isermann. *Adaptive On-line Steady-State Optimization of Slow Dynamic Process.* Automatica, Vol. 14, pp223-230. 1978.

[Blackman 1962] P. F. Blackman. *Extremum-seeking Regulators* An Exposition of Adaptive Control, Pergamon Press, 1962.

[Clausen 87] Asger B. Clausen. *Adaptiv Ekstremumssøger.* Eksamensprojekt nr. 17/87, 1987, Institute of Mathematical Modelling, Technical University of Denmark.

[Dennis and Schnabel 83] John E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Inc., Englewood, Cliffs, New Jersey. USA.

[Frandsen et al. 95] Poul E. Frandsen, Kristian Jonasson and Ole Tingleff. *Unconstrained Optimization* Notes to the course *Algorithms for non-linear Optimization.* Institute of Mathematical Modelling, Technical University of Denmark. 1995.

[Gelb 74] Arthur Gelb *Applied Optimal Estimation* The MIT press, Massachusetts Institute of Technology, England, Cambridge, Massachusetts and London, 1974.

[Golden and Ydstie 89] Melinda P.Golden and B.Erik Ydstie. *Adaptive Extremum Control Using Approximate Process Models.* AIChE Journal, July, 1989, Vol. 35, No. 7.

[Goodwin and Sin 84] Graham C. Goodwin and Kwai S. Sin. *Adaptive Filtering Prediction and Control.* Prentice Hall. 1984.

[Haber and Keviczky 76] R. Haber and L. Keviczky. *Identification of Nonlinear Dynamic Systems.* Proc. of 4th IFAC Symposium on Identification and System Parameter Estimation, Tbilisi, USSR. Sept, 1976.

[Jacobs and Langden 70] O.L.R.Jacobs and S.M.Langdon. *An Optimal Extremal Control System.* Automatic, Vol. 6, pp. 297-301. 1970.

[Jazwinski 70] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York. 1970.

[Keviczky and Haber 74] L. Keviczky and R. Haber. *Adaptive Dual Extremum Control by Hammerstein Model.* Proc. IFAC Symposium on Stochastic Control, pp 333-341. Budapest, 1974.

[Keviczky 78] L. K. Keviczky. *Nonlinear Dynamic Identification of a Cement Mill to be Optimized.* Proc. of 4th IFAC Symposium on Identification and System Parameter Estimation, Tbilisi, USSR. Sept, 1976.

[Keviczky et al. 79] L.Keviczky, I.Vajk and J.Hetthéssy. *A Self-Tuning Extremal Controller for the Generalized Hammerstein Model* Proc. IFAC Symposium on Identification. pp. 1147-1151, 1979. Darmstadt.

[Ljung 77] Lennart Ljung. *Analysis of Recursive Stochastic Algorithms.* IEEE Trans. on Automatic Control, vol. AC-22, no. 4, Aug. 1977.

[Ljung 79a] Lennart Ljung. *The Extended Kalman Filter as a Parameter Estimator for Linear Systems.* Report, Department of Electrical Engineering, Linköpings University. Linköpings, Sweden. 1977.

[Ljung 79b] Lennart Ljung. *Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems.* IEEE Trans. on Automatic Control, vol. AC-24, no. 1, Feb. 1979.

[Ljung and Söderström 83] Lennart Ljung and Torsten Söderstrøm. *Theory and Practice of Recursive Identification.* The MIT Press, 1983.

[Ljung 87] Lennart Ljung. *System Identification: Theory for the User.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1987.

[Ljungquist 90] Dag Ljungquist. *Online Estimation in Nonlinear State-space Models with Application to Catalytic Cracking.* Dr.Ing.Thesis. Norwegian Institute of Technology. 1990.

[Ljungquist and Balchen 93] Dag Ljungquist and Jens G. Balchen. *Recursive Prediction Error Methods for Online Estimation in Nonlinear State-Space models* Proc. of the 32nd Conf. on Decision and Control, Texas. IEEE. Dec. 1993.

[Madsen 95] Kaj Madsen. *Minimering af Kvadratsummer* Hæfte 38. Institute of Mathematical Modelling, Technical University of Denmark. 1995.

[Madsen and Holst 96] Henrik Madsen and Jan Holst. *Modelling Non-linear and Non-stationary Time series.* Institute of Mathematical Modelling, Technical University of Denmark. 1996.

[Melgaard 94] Henrik Melgaard. *Identification of Physical Models.* IMM-PHD-1994-4, Institute of Mathematical Modelling, Technical University of Denmark. 1994.

[Navarro and Zarrop 95] Luis B. Navarro and Martin B. Zarrop. *Extremum Control Approach to Stochastic Control.* Proceedings of 3rd European Control Conference. Rome, Italy, September, 1995.

[Nørgaard 96] Peter Magnus Nørgaard. *System Identification and Control with Neural Networks.* Department of Automation, Technical University of Denmark. 1996.

[Poulsen 95] Niels K. Poulsen. *Stokastisk Adaptiv Regulering* Institute of Mathematical Modelling, Technical University of Denmark. 1995.

[Sternby 78] Jan Sternby. *Analysis of an Extremum Controller for Hammerstein Models.* Report, Department of Automatic Control, Lund University, 1978.

[Sternby 80a] Jan Sternby. *Adaptive Control of Extremum Systems.* Proceedings of an International Symposium on Adaptive Control, Bochum, March, 1980.

[Sternby 80b] Jan Sternby. *Extremum Control Systems-An Area for Adaptive control?* Proceedings of the Joint Automatic Control Conference, Vol. 1, WA2-A. 1980.

[Söderström and Stoica 89] Torsten Söderström and Petre Stoica. *System Identification.* Prentice Hall, 1989.

[Vadstrup 85] Pierré Vadstrup. *Ikke-lineære Systemer.* Intern Rapport, Servolaboratoriet, DTH, December, 1985.

[Vadstrup 88] Pierré Vadstrup. *Adaptiv Regulering af Ikke-lineære Systemer.* Licentiatafhandling, Servolaboratoriet, DTH. Danmark. December, 1988.

[Wellstead and Scotson 90] P.E.Wellstead and P.G.Scotson. *Self-tuning Extremum Control.* IEE Proceedings, Vol. 137, Pt. D, No. 3, May, 1990.

[Wellstead and Zarrop 91] P.E.Wellstead and Martin B.Zarrop. *Self-tuning Systems: Control and Signal Processing.* John Wiley & Sons Ltd, 1991.

[Wittenmark 93] Björn Wittenmark. *Adaptive Control of a Stochastic Non-linear System: An Example.* International Journal of Adaptive Control and Signal Processing, Vol. 7, 327-337, 1993.

[Wolfe 78] Michael A. Wolfe. *Numerical Methods for Unconstrained Optimization.* Van Nostrand Reinhold Company, Great Britain. 1978.

[Zarrop and Rommens 93] Martin B.Zarrop and M.J.J.J.Rommens. *Convergence of a Multi-input Adaptive Extremum Controller.* IEE Proceedings-D, Vol. 140, No. 2, March 1993.

[Zarrop 94] Martin B.Zarrop. *Self-tuning Extremum Control with Constraints.* Conference Publication, No. 389, IEE, March, 1994.

[Åström and Wittenmark 89] Karl J. Åström and Björn Wittenmark, *Adaptive Control.* Addison-Wesley, Reading, MA, 1989.