Adaptive Extremum Control and Wind Turbine Control

Xin Ma

May 1997

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Ph.D. in engineering. The thesis has been prepared at the Institute of Mathematical Modelling (IMM), the Technical University of Denmark (DTU).

This thesis consists of two parts, i.e., adaptive extremum control and modelling and control of a wind turbine.

The first part of the thesis discusses design of adaptive extremum controllers for some processes which have the behaviour that the process should have as high efficiency as possible. The main contribution is the development of an adaptive extremum control algorithm based on the parameter estimation for the processes with output nonlinearity.

The second part of the thesis is concerned with modelling and control of a wind turbine. The investigation of control design is divided into below rated operation and above rated operation.

The behaviour of wind turbines in below rated operation belongs to the process discussed in the first part of the thesis, which can be considered as a connection between two parts.

Acknowledgements

It is pleasure for me take the opportunity to express my gratefulness to everybody who enabled me to perform the work presented in this thesis.

First, I wish to express my thanks to my supervisor Niels Kjøstad Poulsen (IMM) and Henrik Bindner (Risø National Laboratory) for their willing to be my supervisors and their interests in the project. Further, I thank them for reading the first version of the thesis and for giving helpful comments.

I would also like to thank the staff at the IMM, my friends Anca Daniella Hansen for enjoyable daily company, and Lars Henrik Hansen who has always time and patience with my questions. Especially I wish to thank the librarian Finn Kuno Christensen for his help to find literature sources.

I greatly appreciate the support by my parents and my parents-in-law, who have helped me to take care of my son during my study. I would also like to thank my son Johan Ziruo Ye for his understanding.

Finally, A special acknowledge is to my husband Tao Ye, whom I can never thank enough for his support and encouragement.

Lyngby, Denmark, May 1997.

Xin Ma

Summary

This thesis is divided into two parts, i.e., adaptive extremum control and modelling and control of a wind turbine.

The first part of the thesis deals with the design of adaptive extremum controllers for some processes which have the behaviour that process should have as high efficiency as possible.

Firstly, it is assumed that the nonlinear processes can be divided into a dynamic linear part and static nonlinear part. Consequently the processes with input nonlinearity and output nonlinearity are treated separately.

With the nonlinearity at the input it is easy to set up a model which is linear in parameters, and thus directly lends itself to parameter estimation and adaptive control. The extremum control law is derived based on static optimization of a performance function.

For a process with nonlinearity at output the intermediate signal between the linear part and nonlinear part plays an important role. If it can be

known, the only difference between the output nonlinearity and input nonlinearity is that its extremum control law will be determined through linear dynamics. If the intermediate signal is not available, the emphasis will be laid on the parameter estimation approaches. The EKF and RPEM methods are developed for state and parameter estimation for a nonlinear system.

The second part of the thesis discusses the aspects on modelling and control of a wind turbine.

Special attention is paid to mathematical modelling of wind turbines with emphasis on control design. The models have been validated by experimental data obtained from an existing wind turbine.

The effective wind speed experienced by the rotor of a wind turbine, which is often required by some control methods, is estimated by using a wind turbine as a wind measuring device.

The investigation of control design is divided into below rated operation and above rated operation. Below rated power, the aim of control is to extract maximum energy from the wind. The pitch angle of the rotor blades is fixed at its optimal value and turbine speed is adjusted to follow the changes in wind speed. Above rated power, the control design problem is to limit and smooth the output electrical power. The pitch control is investigated for both constant speed and variable speed wind turbines. The minimization of the turbine transient loads is focussed in both cases.

Resumé (in Danish)

Nærværende afhandling omhandler to relaterede emner, adaptive ekstremumsøgere samt modellering og styring af vindmøller.

Afhandlingens første del vedrører adaptive ekstremumsøgere. Målet er at introducere en styringsstrategi, der øger et systems effektivitet.

Først antages det at den ulineære proces består af en lineær dynamisk del samt en ulineær statisk del. Processer med ulineariteter i udgangen og i indgangen har deres specifikke egenskaber.

Hvis ulineariteterne er relateret til udgangen kan systemet let beskrives af en model der er lineær i parametrene og følgelig let indgå i en parameterestimation og adaptiv regulering. Den ekstremumsøgende styrestrategi er udledt ud fra statiske optimeringsmetoder.

Hvis ulineariteten er relateret til indgangen af systemet spiller det partielle signal mellem lineær og ulineær del en bestydelig rolle. Er dette signal kendt eller målt består den ekstremumsøgende styrestrategi af en optimering via

den lineære dynamik. Er dette signal ikke tilgængeligt må en estimationsbaseret metode anvendes. En EKF og en RPEM metode er udviklet for tilstands og parameterestimation.

Afhandlingens anden del vedrører modellering og styring af vindmøller.

I modelleringen er der fokuseret på anvendelsen af modellen til regulator design. Modellen er blevet valideret mod eksperimentelle data.

Den effektive vindhastighed er en modelleringsteknisk størrelse, som er et udtryk for vindens påvirkning af vindmøllen og anvendes ofte i fremkoblingsdelen for en styring. Der er i afhandlingen beskrevet og diskuteret metoder til estimation af den effektive vindhastighed med vindmøllen som egentligt målesystem.

Styring af en vindmølle består af to opgaver. Under nominel vindhastighed er målet at optimere den effekt der udnyttes fra vinden. Vindmøllens bladvinkel er fastlåst ved den optimale udnyttelsesgrad og omløbstallet justeres til at følge variationerne i vindhastigheden. Over nominel vindhastighed er målet at begrænse den optagne effekt til en præspecificeret størrelse. I afhandlingen behandles bladvinkelregulering med både konstant og variabelt omløbstal.

Parts of this thesis have previously been published in

- Xin Ma. *Modelling and Control of a Wind Turbine.* Master's thesis, No. 25/93, IMSOR, DTU.
- Xin Ma, Niels K. Poulsen and Henrik Bindner. *Modelling and Control of a Wind Turbine.* Technical report, IMM-rep-1994-27, IMM, DTU.
- Xin Ma, Niels K. Poulsen and Henrik Bindner. *Estimation of Wind Speed in Connection to a Wind Turbine.* Technical report, IMM-rep-1995-26, IMM, DTU.
- Xin Ma, Niels K. Poulsen and Henrik Bindner. *A Pitch Regulated Variable Speed Wind Turbine.* Technical report, IMM-rep-1995-27, IMM, DTU.
- Xin Ma and Niels K. Poulsen. *Adaptive Extremum Control.* Technical report, IMM-rep-1996-23, IMM, DTU.
- Xin Ma, Niels K. Poulsen and Henrik Bindner. *Estimation of Wind Speed in Connection to a Wind Turbine.* Accepted by the IASTED International Conference on Control, Mexico, May, 1997.
- Xin Ma, Niels K. Poulsen and Henrik Bindner. *Extremum Tracking Control of a Wind Turbine.* Technical report, to appear, IMM, DTU.

Contents

Part I

Adaptive Extremum Control

Glossary

Notations

A,B,C,D	Matrices in state-space model
F,H,M	Matrices in state-space model
$A(q^{-1})$	A-polynomial in operator q^{-1} for input-output model
$B(q^{-1})$	B-polynomial in operator q^{-1} for input-output model
$C(q^{-1})$	C-polynomial in operator q^{-1} for input-output model
$e(t)$	White noise
f	Nonlinear function
g	Gradient matrix
H	Hessian matrix
K	Kalman gain matrix
K_x	State update matrix
K_θ	Parameter update matrix
$P(t)$	$R(t)^{-1}$
$R(t)$	Hessian approximation in Gauss-Newton algorithm
$u(t)$	Input signal
$u_p(t)$	Perturbation signal
$V(\cdot)$	Criterion
$x(t)$	State vector
$y(t)$	Output signal
$w(t)$	Noise sequence
$W(t)$	Gradient of the estimated state
$\hat{y}(t)$	Predictor using running estimate

$z(t)$	Intermediate signal
$\epsilon(t)$	Prediction error
θ	Parameter vector
$\hat{\theta}(t)$	Recursive estimate of θ
θ_0	True value of θ
θ^*	Convergent point of θ
$\varphi(t)$	Vector formed from observed data
$\phi(t)$	Gradient of the prediction error
α	Step size
Λ	Covariance matrix of prediction error
$\hat{\Lambda}(t)$	Estimated covariance matrix of prediction error
Δ	Increment operator

Abbreviations

EKF	Extended Kalman Filter
LHP	Left Half Plane
ODE	Ordinary Differential Equation
PRBS	Pseudorandom Binary Sequence
RELS	Recursive Extended Least Squares
RLS	Recursive Least Squares
RML	Recursive Maximum Likelihood
RPEM	Recursive Prediction Error Method

Chapter 1

Introduction

In most control problems, the task of a regulator is to keep some variables at constant values, or to make them follow reference signals. In general, the system is assumed to be linear, and it is possible, in principle, to drive the output to any prescribed value. With such problems the reference values are often easily determined. It can be the desired altitude of an airplane, the predefined 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 variables 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 system, chemical and combustion processes. 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. Another example is ore-grinding, where the grinding efficiency will vary with the filling degree of the mill, which can be controlled through the incoming flow of raw material. The optimal point in maximizing efficiency may depend on the quality and composition of this raw material. For a wind turbine, the pitch angle of blades or the rotor speed of wind turbine is changed depending on the wind speed to give maximum output power. This is also an extremum control problem.

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 influence of disturbances.

A common assumption is that there is non dynamics in the system. This is called static system. In practice, this condition can be fulfilled by using a sufficiently large sampling interval. But the result may be a slow optimization. In many cases, however, static models may be adequate, and stochastic approximation methods can then be used for optimization to handle noise measurements.

Actually more common cases in practice are the input signal will influence the system behaviour at subsequent times, i.e., the performance has not settled at new steady-state value before the next measurement is taken. This is so called dynamic system. One of methods to handle this kind of system is to derive a nonlinear dynamic model.

[Blackman 1962] is a good introduction to the field of extremum control. The classification used in this paper is *perturbation systems*, *switching systems* and *self driving systems*. 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. 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.

A fourth class of methods that is not described in [Blackman 1962] has been developed later on. It is based on using a parameterized model in combining parameter identification and extremum control. The project presented here is based on this type of methods.

Since fewer ideas for extremum control have emerged since 60's. It is then still worth mentioning the three types of methods classified by [Blackman 196 The methods were later summarized by [Sternby 80b].

Perturbation methods

The basic idea of perturbation methods is to add a periodic test signal to the control signal, and observe its effect at the output. The task of an extremum controller in the perturbation methods is to keep the gradient of the nonlinearity at zero.

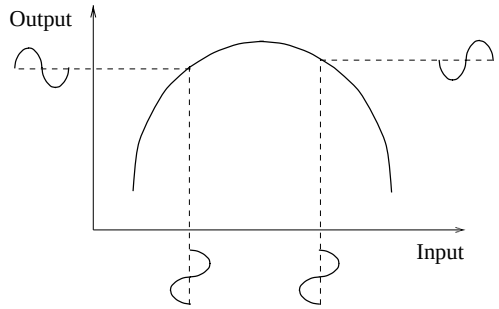

Figure 1.1. Effect of input test signal at the output for a static nonlinearity.

The effect of an input test signal at output of a static nonlinearity is illustrated in Figure 1.1. The most commonly used test signal form is sinusoid.

However, if the system contains dynamics, the dynamics will then introduce a phase lag θ in the test signal component of the output. The result of correlation will be multiplied by a factor $\cos\theta$. This gives a sign error in the correlation signal if $\theta > 90^0$. This situation can be avoided if a corresponding phase lag is introduced to the test signal before correlation. Another way is to use a perturbation signal of sufficiently low frequency. The phase lag will then be small, so that the dynamics can be neglected. But this give a long response time for the overall system.

With the perturbation signal technique, the correlating device must be given a certain amount of time to produce an accurate slope signal. During this time the control signal should be kept constant, so that the total input varies with test signal only.

The perturbation methods may be the oldest extremum control method. But it is also well suited to multi-input systems. In order to apply a gradient method in search for an extremum, the partial derivatives of the static response curve with respect to the different inputs are needed. This can be realized by using perturbation signals with different frequencies for each input.

Switching methods

Another basic idea for extremum control is the switching methods. The input is driven at constant speed in the same direction until no further improvement is registered. The drift direction is then reversed. Different algorithms of this type can be described in terms of their specific conditions for altering the direction of input changes. The control law is thus a set of switching conditions. The input may be changed continuously or in discrete steps. The second one is called stepping method.

In a continuous sweep method, the sweep direction is reversed when the output has decreased from its maximum value by a fixed amount Δ. The design parameters are then the sweep rate and the value of Δ. If the output is disturbed by noise, the method may give excessive switching unless the value of Δ is sufficiently increased. But it will on the other hand increase the hunting loss by increasing Δ. It is then necessary to compromise in choosing Δ. Filtering is another possibility for reducing the noise sensitivity. The problem is then that more dynamics are introduced into the system, and the hunting loss will again be increased. Unnecessary switching may also be caused by input dynamics. The switching conditions may be chosen

in many ways. Since the derivative of the output should be zero at an extremum point, it can be used to determine when to reverse the sweeping direction. If a threshold is introduced, The switching did not occur until the derivative was less than $-\Delta$ after passage of the maximum.

The stepping method gives the input signal

$$\Delta u(t+1) = \Delta u(t)\ \ sign(\Delta y(t)) \tag{1.1}$$

where $u(t)$ and $y(t)$ are input and output signal for a static system. With this control law, the closed-loop system will then end up with input oscillating a few steps around the extremum point. The method is also called hill-climbing method which is given in Appendix A. There are two design parameters to choose in such a system: the stepping period and step length. The method converges quickly by a large step size, but on the other hand it will cause a large deviation from the optimum. A variable step size might then be useful, but it will increase the complexity of the algorithm. The stepping period should be kept as small as possible in order to speed up the system. But when dynamics are included in the model, the easiest way to handle the system is to wait for the steady state between each input change. Measurement noise will introduce a risk of stepping in the wrong direction. The steady state deviations from the optimum will then be increased.

Self-driving method

In a self-driving system the available information at every instant is used to produce a control signal that will drive the system towards an optimum.

For a static system

$$y(t) = f(u(t))$$

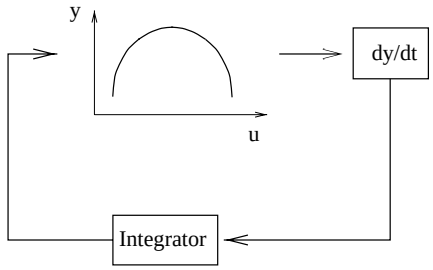

Figure 1.2. Self driving system

The first order derivative of the output is then used to drive the input via an integrator

$$u(t) = \int \dot{y}(t)dt \tag{1.2}$$

The system has to be started manually, since $\dot{y} = \dot{u} = 0$ is always a stationary point. If the process is started in the correct direction with $\dot{u} \neq 0$, it will continue in the same direction until $\dot{y} = 0$ and then stop. If the process is started in the wrong direction, it can be detected by observing the negative value of dy/dt. the problem can be handled by measuring du/dt, then $f'(u) = \frac{dy/dt}{du/dt}$ can be used in the control law instead of $\dot{y}$. Dynamics will introduce the further problem to the system. One way to compensate for the dynamics is to filter the input signal. This filter should be a good guess of the system dynamics, and a possible control law is

$$u(t) = \int k\frac{\dot{y}(t)}{\dot{u}^*(t)}dt \tag{1.3}$$

where u^* is filtered input signal.

Adaptive extremum control methods

Until now only little information, such as the output or the slope of the nonlinearity, is collected about the system. For the methods developed

later on, the control action is calculated from a model obtained by some kind of identification. The input may be chosen as the estimated extremum position. For this type of methods, each control action is preceded by an identification phase. Based on the estimates a control step is then taken, and cycle is repeated. This is so-called *adaptive extremum control* method. The idea is illustrated in Figure 1.3.

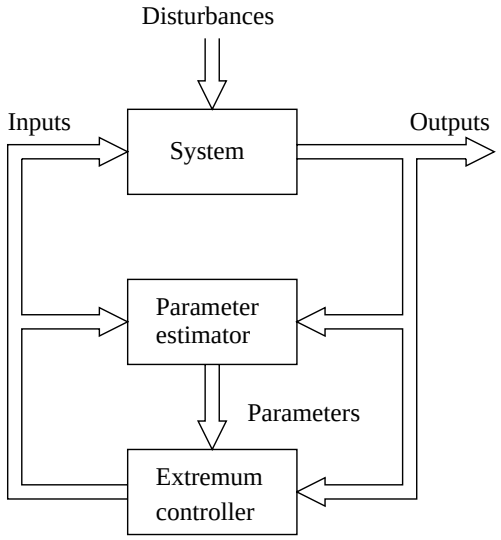

Figure 1.3. Block diagram of an adaptive extremum controller

The extremum control problem treated here will be assumed to have unknown nonlinearities or at least partly unknown nonlinearities. Any a priori knowledge about the system will be used in setting a model. This is especially true for nonlinear systems. It may provide possibilities for choosing a model structure that allows a good description of the nonlinear phenomena. To be able to use system identification it is of course desirable to have a model which is linear in its unknown parameters. With no such a priori

knowledge available, more general nonlinear models have to be used. In this way it may be possible to handle quite complicated, but partially known nonlinear systems.

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 or after the linear part, or between two linear parts. Different model structures have been illustrated in Figure 1.4 - 1.6. If the nonlinear part is placed before the linear part, it is called input nonlinearity. On the contrary, it is output nonlinearity if the linear part is followed by nonlinearity. The process with input nonlinearity behaves differently from the process with output nonlinearity. An output nonlinearity is much more difficult to handle than input nonlinearity. The complexity of the problem will also depends on which of the variables in the process can be measured. The process with the nonlinear part between two linear parts is not considered in the report.

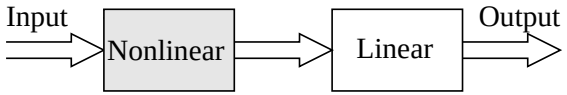

Figure 1.4. Process with input nonlinearity

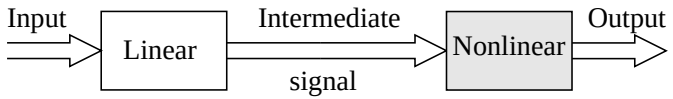

Figure 1.5. Process with output nonlinearity

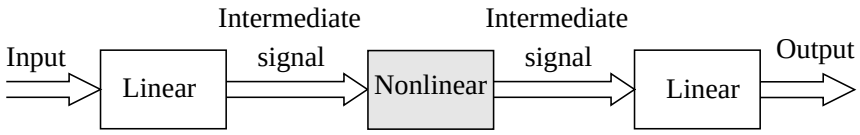

Figure 1.6. Process with nonlinear part between linear parts.

Only very few people have discussed what happened when there was an output nonlinearity. Most of them assumed that the intermediate signal was measurable. The problem will thus be simplified. However, more research is needed to find out how to handle the system in which the intermediate signal is not available, which is also an emphasis of this project.

With an input nonlinearity a so called Hammerstein model is usually obtained, which is a special case of the Uryson series. An output nonlinearity can, however, be viewed as a special case of the Volterra series. This choice of model structures will have a large influence on the behaviour of the model.

In case the model of the process dynamic is unknown, if the assumed structure is correct, an on-line parameter estimation method should be combined with the on-line extremum control method. This leads to the model-based adaptive extremum control method which generates the control action by making use of a model obtained by some kinds of recursive identification methods. The recursive extended least squares method (RELS) and recursive maximum likelihood method (RML) are normally good choices for estimation problems. The current best estimates are used to determine the new input value. By using a model and system identification, it is also possible to follow time variations in the process. For the control scheme in Figure (1.3), it may be necessary to superimpose a perturbation signal on the control signal to ensure identify ability of the parameters.

The adaptive control on the basis of the certainty equivalence principle was developed by [Keviczky and Haber 74] first. In the paper a generalized Hammerstein model was applied to derive the adaptive extremum control law, and they suggested that a possible way to avoid identification problem was to add an extra disturbance signal to the input. Another possibility to separate the extremum control into identification, optimization and feed forward phases was applied by [Bamberger and Isermann 78]. An analysis based on the differential equation approach of Ljung was made by [Sternby 78]. In this paper a self-tuning regulator was extended for extremum control of generalized Hammerstein model. It was shown by Sternby that the parameter estimates might well converge to some wrong values.

The rest of this part of thesis will be organized as follows.

Chapter two gives the model descriptions for nonlinear processes. It is assumed that the nonlinear system can be separated into linear dynamics and nonlinear dynamics. For a process with nonlinearity at input, a Hammerstein type model is obtained. While a Wiener type model is derived for a process with nonlinearity at output. Both input-output formulae and state-space models are given in the chapter. Some general nonlinear models are represented later on.

In chapter three an adaptive extremum control algorithm is derived for the discrete-time system with input nonlinearity and (partly) unknown dynamics. A Hammerstein type model can be used to describe the process with nonlinearity at input. Since the model is linear in parameters, the RELS or RML method can directly be used to estimate the parameters. The extremum control law based on the estimated model will then be derived. The

input signal to the process is chosen as the estimated position of the optimum with a superimposed perturbation signal which assures the persistent excitation of the process. Two examples are given to assess the performance of algorithm. The convergence properties of the algorithm are analysed by using the ODE approach.

Chapter four concerns the process with nonlinearity at output. The extremum control problem is in general much difficult to handle in this case. However, a special case is when the intermediate signal between the linear and nonlinear part can be measured. The problem will be simplified. The identification can be implemented for linear part and nonlinear part separately, and the extremum control law can then be derived based on static optimization of a performance function. When the intermediate signal is not measurable, the emphasis will give to the parameter identification, since the extremum control law relies heavily on the estimated model. The extended Kalman filter (EKF) method used as a joint parameter and state estimator is implemented for a nonlinear state-space model. The recursive prediction error method and the recursive line search prediction error method are derived for a nonlinear innovations model. The behaviour of the different estimation methods is investigated by simulation examples.

Chapter five gives summary and conclusions of the investigations in the previous chapters.

Chapter 2

Models of Nonlinear Dynamic Processes

As it is mentioned in chapter 1, the key point of extremum control is the basic assumption of a model which describes the performance function or the process dynamics. The most important feature is that the process is assumed to be nonlinear, and the biggest problem is then to choose a proper model structure, since the model-based extremum control methods generate the control action by making use of a model obtained by some kind of identification method.

The mathematical descriptions of the nonlinear systems have been the subject in many articles. Some frequently applied models are given by

[Haber and Keviczky 76] and [Vadstrup 85]. However, only discrete-time systems are considered in this thesis.

The static response curve relating input to output in extremum control systems is inherently nonlinear. A general description of a nonlinear discrete-time model is

$$y(t) = f(y(t-1), y(t-2), \cdots, u(t-d), u(t-d-1), \cdots, \theta, t) + \omega(t) \quad (2.1)$$

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

Nonlinear systems have many different types. For special classes of nonlinear systems where the linear dynamics and nonlinear dynamics can be separated, the methods worked out for the identification of linear discrete-time systems can be extended to the nonlinear systems. If we assume that the nonlinear part can be placed before or after the linear part, or between two linear parts, the block schemes for these three most well-known forms have been shown in chapter 1, the models with different properties can be obtained. If the nonlinear part is placed before the linear part, it is input nonlinearity. Otherwise it is output nonlinearity if the nonlinearity is at output.

The remainder of this chapter is organized as follows. Section 2.1 presents the nonlinear dynamic models in simple input-output relationships. Section 2.2 contains the state-space descriptions of nonlinear dynamic systems. Both process with input nonlinearity and process with output nonlinearity are represented. Particular interest in this chapter is given to the models

being linear in parameters, since identification methods can easily be implemented to estimate model parameters. A summary is given in section 2.3.

2.1 Models in input-output formulae

2.1.1 Nonlinearity at input

A block scheme of a process with input nonlinearity in input-output formulae is illustrated in Figure 2.1, where the process is divided into a nonlinear static part and a linear dynamic part in series. $u(t)$ and $y(t)$ are the input and output signals, $z(t)$ is an intermediate signal between the linear part and nonlinear part.

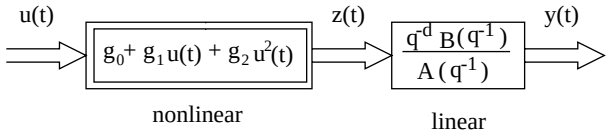

Figure 2.1. Process with input nonlinearity

The nonlinear block can be represented by a second order polynomial for the sake of simplicity, since a quadratic assumption is acceptable for extremum controllers operating close to the optimum point.

$$z(t) = g_0 + g_1 u(t) + g_2 u^2(t) \quad (2.2)$$

The linear block can be obtained by describing it as a linear difference equation

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})z(t) \tag{2.3}$$

where d is time delay, and

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b} \end{aligned} \tag{2.4}$$

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

Combining two models gives the representation for the whole process

$$\begin{aligned} A(q^{-1})y(t) &= q^{-d}B(q^{-1})[g_0 + g_1 u(t) + g_2 u^2(t)] \\ &= g_0\bar{B} + g_1 B(q^{-1})u(t-d) + g_2 B(q^{-1})u^2(t-d) \end{aligned} \tag{2.5}$$

where $\bar{B} = B(1)$. This is a simple Hammerstein model that can be extended to a generalized Hammerstein model

$$A(q^{-1})y(t) = b_{00} + B_1(q^{-1})u(t-d) + B_2(q^{-1})u^2(t-d) \tag{2.6}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B_1(q^{-1}) &= b_{10} + b_{11} q^{-1} + \cdots + b_{1n_{b1}} q^{-n_{b1}} \\ B_2(q^{-1}) &= b_{20} + b_{21} q^{-1} + \cdots + b_{2n_{b2}} q^{-n_{b2}} \end{aligned} \tag{2.7}$$

Thus we get a system equation which is linear in parameters and can directly be written in regressive form

$$y(t) = \varphi^T(t)\theta \tag{2.8}$$

where θ is the parameter vector, $\varphi(t)$ is a vector which includes previous input and output signals.

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

2.1.2 Nonlinearity at output

For the general description of the nonlinear model in input-output formula (2.1), if the process has the nonlinearity at output, the model can then be illustrated by Figure 2.2.

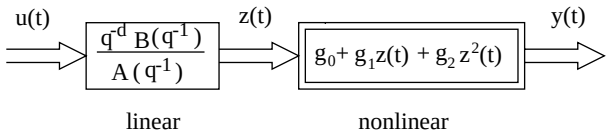

Figure 2.2. Process with output nonlinearity

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

$$A(q^{-1})z(t) = q^{-d}B(q^{-1})u(t) \tag{2.9}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b} \end{aligned} \tag{2.10}$$

The model of the nonlinear block is given by

$$y(t) = g_0 + g_1 z(t) + g_2 z^2(t) \tag{2.11}$$

The whole model can be obtained by inserting (2.9) into (2.11). This gives

$$y(t) = g_0 + g_1\frac{B(q^{-1})}{A(q^{-1})}u(t-d) + g_2\left[\frac{B(q^{-1})}{A(q^{-1})}u(t-d)\right]^2 \tag{2.12}$$

This is a simple Wiener model, and it can be extended to a generalized Wiener model

$$y(t) = g_0 + \frac{B_1(q^{-1})}{A(q^{-1})}u(t-d) + \left[\frac{B_2(q^{-1})}{A(q^{-1})}u(t-d)\right]^2 \tag{2.13}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a} \\ B_1(q^{-1}) &= b_{10} + b_{11}q^{-1} + \cdots + b_{1n_{b1}}q^{-n_{b1}} \\ B_2(q^{-1}) &= b_{20} + b_{21}q^{-1} + \cdots + b_{2n_{b2}}q^{-n_{b2}} \end{aligned} \tag{2.14}$$

It can be easily found that the Wiener type models are nonlinear in parameters. Most identification methods can not be used to estimate parameters of this type of models.

2.1.3 Some general nonlinear models

If the model of nonlinear block (2.2) for the process with input nonlinearity is extended to $k = 1, 2, \cdots l$, i.e.,

$$z(t) = \sum_{k=0}^{l} g_k u^k(t) \tag{2.15}$$

the model of the process with input nonlinearity (2.5) will then be modified by

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})\left[\sum_{k=0}^{l} g_k u^k(t)\right] = \sum_{k=0}^{l} g_k B(q^{-1})u^k(t-d) \tag{2.16}$$

The model can be rewritten as

$$A_1(q^{-1})y(t) = b_{00} + \sum_{k=1}^{l} B_k(q^{-1})u^k(t-d) \tag{2.17}$$

which can be generalized to a feedback Hammerstein model

$$A_1(q^{-1})y(t) = b_{00} + \sum_{k=1}^{l} B_k(q^{-1})u^k(t-d) + \sum_{k=2}^{j} A_k(q^{-1})y^k(t) \tag{2.18}$$

where

$$\begin{aligned} A_1(q^{-1}) &= 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a} \\ A_k(q^{-1}) &= a_{k1}q^{-1} + \cdots + a_{kn_{ak}}q^{-n_{ak}} \quad k = 2, \cdots j \\ B_k(q^{-1}) &= b_{k0} + b_{k1}q^{-1} + \cdots + b_{kn_{bk}}q^{-n_{bk}} \quad k = 1, \cdots l \end{aligned} \tag{2.19}$$

Furthermore, the above model is still a special case of the feedback Uryson model

$$A_1(q^{-1})y(t) = b_{00} + \sum_{k=1}^{l} B_k(q^{-1})f_k(u(t-d)) + \sum_{k=2}^{j} A_k(q^{-1})g_k(y(t)) \tag{2.20}$$

By including the terms which describe the interaction of $u(t-d)$ and $y(t)$, a general nonlinear model will be represented by

$$\begin{aligned} A_1(q^{-1})y(t) = b_{00} &+ \sum_{k=1}^{l} B_k(q^{-1})f_k(u(t-d)) + \sum_{k=2}^{j} A_k(q^{-1})g_k(y(t)) \\ &+ \sum_{k=1}^{m} D_k(q^{-1})h_k(u(t-d), y(t)) \end{aligned} \tag{2.21}$$

where the polynomials $A_k(q^{-1})$ and $B_k(q^{-1})$ are given by (2.19), and polynomial $D_k(q^{-1})$ is

$$D_k(q^{-1}) = d_{k1}q^{-1} + \cdots + d_{kn_{dk}}q^{-n_{dk}} \quad k = 1, \cdots m \tag{2.22}$$

In Uryson model (2.20), function f_k and g_k depend only on $u(t-d)$ and $y(t)$. If we wish to generalize it, another type of model, the Volterra model, can also be used to model the process with output nonlinearity. For the sake of simplicity, only model of second degree are presented

$$A_1(q^{-1})y(t) = g_0 + B_1(q^{-1})u(t) + \sum_{k=0}^{l} B_{2k}(q^{-1})u(t)u(t-k) + \sum_{k=0}^{j} A_{2k}(q^{-1})y(t)y(t-k) \tag{2.23}$$

where

$$\begin{aligned} A_1(q^{-1}) &= 1 + a_{11}q^{-1} + \cdots + a_{1n_{a1}}q^{-n_a} \\ B_1(q^{-1}) &= b_{10} + b_{11}q^{-1} + \cdots + b_{1n_{b1}}q^{-n_{b1}} \\ A_{2k}(q^{-1}) &= a_{2k1}q^{-1} + \cdots + a_{ak(j-k)}q^{-(j-k)} \\ B_{2k}(q^{-1}) &= b_{2k0} + b_{2k1}q^{-1} + \cdots + b_{2k(l-k)}q^{-(l-k)} \end{aligned} \tag{2.24}$$

This is called the second order feedback Volterra model.

2.2 Models in state-space formulae

2.2.1 Nonlinearity at input

A simple Hammerstein model can also be represented in state-space form. The description is illustrated in Figure 2.3.

The nonlinear block of the process can be represented by equation (2.2), and the linear block is

$$\begin{aligned} x(t+1) &= Ax(t) + Bz(t) \\ y(t) &= Cx(t) + Dz(t) \end{aligned} \tag{2.25}$$

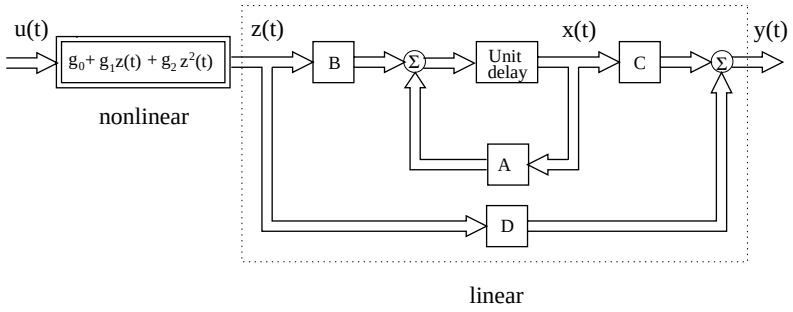

Figure 2.3. Process with input nonlinearity in state-space form

where $x(t)$ is a state vector. Matrices A, B, C and D are assumed to be time invariant. These two models would give a state-space form of the simple Hammerstein model

$$\begin{aligned} x(t+1) &= Ax(t) + B[g_0 + g_1u(t) + g_2u^2(t)] \\ y(t) &= Cx(t) + D[g_0 + g_1u(t) + g_2u^2(t)] \end{aligned} \tag{2.26}$$

which can be generalized to

$$\begin{aligned} x(t+1) &= Ax(t) + b_0 + B_1u(t) + B_2u^2(t) \\ y(t) &= Cx(t) + d_0 + D_1u(t) + D_2u^2(t) \end{aligned} \tag{2.27}$$

2.2.2 Nonlinearity at output

The process with output nonlinearity represented by a state-space model is illustrated in figure 2.4.

The model of the linear block is given by

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ z(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.28}$$

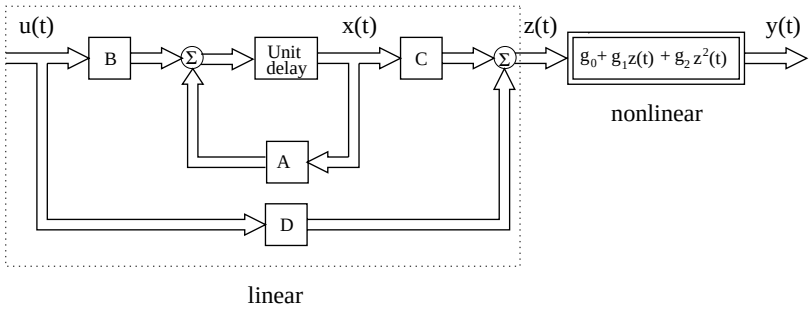

Figure 2.4. Process with output nonlinearity in state-space form

The model of nonlinear block has been given by equation (2.11), therefore we can attain the total model of nonlinear process by

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= g_0 + g_1[Cx(t) + Du(t)] + g_2[Cx(t) + Du(t)]^2 \end{aligned} \tag{2.29}$$

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

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= g_0 + C_1 x(t) + C_2 x^2(t) \end{aligned} \tag{2.30}$$

2.2.3 A general nonlinear state-space model

The state-space form for both process with input nonlinearity and process with output nonlinearity can be represented by a general nonlinear state-space model in equation (2.31) and (2.32), where θ is parameter vector. It is assumed that the noise sequences $\{w_\theta(t)\}$ and $\{e_\theta(t)\}$ are uncorrelated, i.e., the matrix $Q^{we} = 0$. Matrices $Q^w(\theta)$ and $Q^e(\theta)$ depend on θ in an

arbitrary way. It is also assumed that the matrix elements are differentiable with respect to θ.

$$\begin{aligned} x(t+1) &= f(x(t), u(t), \theta(t)) + w_\theta(t) \\ y(t) &= h(x(t), \theta(t)) + e_\theta(t) \end{aligned} \tag{2.31}$$

with

$$\begin{aligned} E\{w_\theta(t) w_\theta^T(s)\} &= Q^w(\theta)\delta_{ts} \\ E\{e_\theta(t) e_\theta^T(s)\} &= Q^e(\theta)\delta_{ts} \\ E\{w_\theta(t) e_\theta^T(s)\} &= Q^{we}(\theta)\delta_{ts} \\ E\{x(0) x^T(0)\} &= \Pi(\theta) \end{aligned} \tag{2.32}$$

2.3 Summary

To perform adaptive extremum control of nonlinear system, it is necessary to make assumptions about the structure of the process to be controlled. When the systems are unknown, they have to be identified. Most identification methods are based on the assumption that the model is linear in the parameters. One class of systems is obtained by dividing the process into a nonlinear static part and a linear dynamic part. Approximation theory can be used to derive different types of series expansion representations of nonlinear systems. The representations include Volterra, Wiener and Uryson series. These three expansions have one thing in common: they can be used

to model processes where linear dynamics is followed by a nonlinearity. The Uryson series can, however, also include nonlinearities at the input. A special case of Uryson series is represented by Hammerstein models, which have the advantage that the model is linear in the parameters.

Chapter 3

Input Nonlinearity

The idea of an extremum controller which combines a recursive estimation algorithm with a synthesis algorithm will be investigated in this chapter for a process with input nonlinearity. With the nonlinearity at the input it is easy to set up a model which is linear in the parameters, and thus directly lends itself to parameter estimation and adaptive control. The extremum control law is derived based on static optimization of a performance function. This is an important practical problem, since a number of industrial processes are such that their performance can be improved by adjusting plant variables so as to increase the efficiency of the processes.

In order to simplify the notations and analysis, only a special low order case of the model will be taken. The system considered is of a second order Hammerstein model with or without dynamics. The process with higher order dynamic model can be treated in the similar way.

Some optimization background considered to be of particular relevance to the extremum control problems is given in Appendix A. The algorithms include the hill-climbing method, gradient method, Newton and Gauss-Newton method. The extremum control law derived in this chapter can be considered as a direct implementation of the Newton iteration.

When the system dynamic is unknown, the model has to be identified. An identification method, e.g., the recursive extended least squares method, for the Hammerstein type models will be presented. Based on the current best estimated parameters, an on-line extremum control law is derived.

This chapter is organized as follows. Section 3.1 is concerned with adaptive extremum control for the process with input nonlinearity. Some examples are given to illustrate the performance of the algorithm in section 3.2. The convergence property of the algorithm is discussed in section 3.3. The summary and conclusions are given in section 3.4.

3.1 Adaptive extremum control for Hammerstein model

3.1.1 The modified Hammerstein model

In many investigations of extremum control systems it is assumed that the systems are static, i.e., input signal has only an instantaneous effect. This assumption can be justified if the time between the changes in the reference value is sufficiently long. A typical description of the static system is

$$y(t) = b_0 + B_1(q^{-1})u(t-d) + B_2(q^{-1})u^2(t-d) + \omega(t) \qquad (3.1)$$

If there are dynamics in the process, the input signal will influence the system behaviour at subsequent times. It means that the performance may not have settled at new steady-state value before the next measurement is taken. This will give an interaction in the control system. The correlation and interaction between different measurements of the performance will confuse the optimization routine. Therefore it is necessary to take the dynamics into consideration when doing the optimization. One possibility discussed above is to wait until the transients have vanished before the next change is made. Of course, this will increase the convergence time, especially if the process has long time constants. Another way around the problem is to base the optimization on nonlinear dynamic model [Åström and Wittenmark 89].

A nonlinear dynamic model of Hammerstein type for the process with input nonlinearity has been given in (2.6) in chapter 2. 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 (2.6) is modified by

$$A(q^{-1})y(t) = b_0 + B_1(q^{-1})u(t-d) + B_2(q^{-1})u^2(t-d) + C(q^{-1})e(t) \tag{3.2}$$

with

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

where $u(t)$, and $y(t)$ are the input and output signals, $e(t)$ is a white noise with zero mean normal distribution.

3.1.2 Extremum control law

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

$$J(u(t)) = E\{y(t+d)\} \tag{3.3}$$

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

by

$$\frac{dJ}{du} = 0 \tag{3.4}$$

Admissible control law may use all the information available, i.e., $u(t)$ may depend on $y(t)$ and all previous inputs and outputs.

Consider a system with input nonlinearity, which can be represented by the generalized Hammerstein model (3.2), the static characteristics of the process will be

$$\bar{A}y = b_0 + \bar{B}_1 u + \bar{B}_2 u^2 \tag{3.5}$$

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 input.

$$u(t) = -\frac{\bar{B}_1}{2\bar{B}_2} \tag{3.6}$$

and optimal value of the output is

$$y^* = \frac{b_0}{\bar{A}} - \frac{\bar{B}_1^2}{4\bar{A}\bar{B}_2} \tag{3.7}$$

The extremum controller is thus a constant gain and no feedback controller. In the adaptive case a constant gain controller will reduce the excitation of the process and identifiability may be lost.

The extremum control law can be interpreted as a direct implementation of Newton iteration

$$\begin{aligned} u(t+1) &= u(t) - \left[\frac{dy}{du} \Big/ \frac{d^2y}{du^2}\right]_{u(t)} \\ &= u(t) - \left[\frac{\bar{B}_1 + 2\bar{B}_2 u(t)}{2\bar{B}_2}\right] \\ &= -\frac{\bar{B}_1}{2\bar{B}_2} \end{aligned} \tag{3.8}$$

The equivalence of the self-tuning adjustment rule and Newton iteration can easily be seen.

3.1.3 Parameter estimation

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

If we take the case of $C(q^{-1}) \neq 1$, the model (3.2) can be written as

$$y(t) = \varphi^T(t)\theta + \epsilon(t) \tag{3.9}$$

where

$$\begin{aligned}\varphi(t) = [&-y(t-1), \cdots, -y(t-n_a), 1, u(t-d), \cdots, u(t-n_{b1}-d),\\ &u^2(t-d), \cdots, u^2(t-n_{b2}-d), \epsilon(t-1), \cdots, \epsilon(t-n_c)]^T\end{aligned} \tag{3.10}$$

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

and

$$\epsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1) \tag{3.11}$$

The estimates using RELS method are given by equations

$$\boxed{\begin{aligned}\hat{y}(t) &= \varphi^T(t)\hat{\theta}(t-1)\\ \hat{\theta}(t) &= \hat{\theta}(t-1) + k(t)[y(t) - \hat{y}(t)]\\ k(t) &= P(t)\varphi(t)[\lambda(t) + \varphi^T(t)P(t)\varphi(t)]^{-1}\\ P(t+1) &= [I - k(t)\varphi^T(t)]P(t)/\lambda(t)\end{aligned}} \tag{3.12}$$

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

It is then possible to make a direct adaptive extremum controller for the process discussed in this chapter. The estimated parameters may then be used in the control law (3.6) instead of the true parameter values

$$u(t) = -\frac{\hat{\bar{B}}_1(t)}{2\hat{\bar{B}}_2(t)} \tag{3.13}$$

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

This leads to a certainty equivalence controller.

3.1.4 The adaptive extremum control algorithm

The on-line parameter estimator combined with on-line extremum controller leads to an adaptive extremum controller which can be summarized as

Step 1: Apply input $u(t)$ to the nonlinear system and measure the output $y(t)$.

Step 2: Use RELS or RML algorithm to estimate parameters for the model (3.2).

Step 3: The 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. The problem has been discussed by [Sternby 78]. The report shows that direct application of the certainty equivalence principle to adaptive extremum control of a Hammerstein type model may cause identification problems. The estimates will converge to a certain hyper-surface in the parameter space, but in most cases to other than the true parameter values.

In practice the adaptive extremum control algorithm will need some modifications. One way to avoid the identification problem is to add an extra

perturbation signal to the input. The test perturbation signal can be a zero mean random variable with small variance.

The extremum control law will thus be modified by

$$u(t) = -\frac{\hat{\bar{B}}_1(t)}{2\hat{\bar{B}}_2(t)} + u_p(t) \tag{3.14}$$

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

A block scheme of an adaptive extremum controller incorporated with a test perturbation signal is illustrated in Figure 3.1.

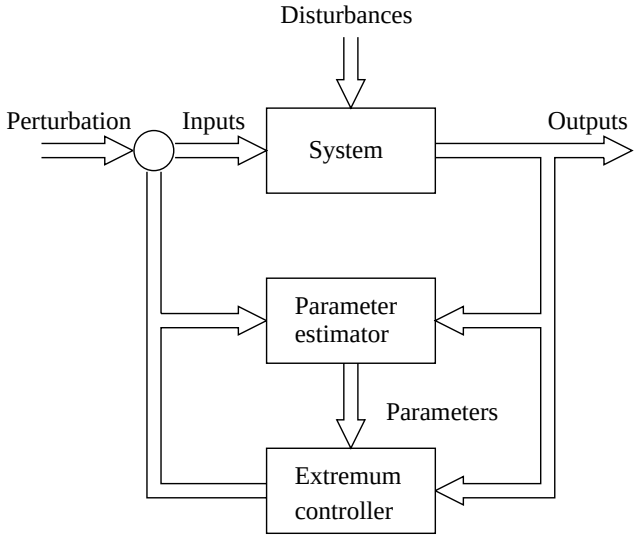

Figure 3.1. Block diagram of an adaptive extremum controller incorporating a test perturbation

Another problem encountered in the algorithm is that the controller is very sensitive to the $\hat{\bar{B}}_2(t)$. If it is close to zero, the controller will not work

well. Actually in most extremum application there will be some knowledge of the bounds on the parameters. The bounds can be used to constrain the extremum controller action in one of two ways. Either the estimated parameters can be checked against known bounds and constrained if necessary, or the extremum control action can be checked against bounds and constrained appropriately. Another updating formula based on stochastic approximation is given by [Sternby 78]

$$u(t+1) = u(t) - \gamma(t)[\hat{\bar{B}}_1(t) + 2\hat{\bar{B}}_2(t)u(t)] \tag{3.15}$$

where $\gamma(t)$ can be

$$\gamma(t) = \frac{1}{t} \tag{3.16}$$

or

$$\gamma(t+1)^{-1} = \gamma(t)^{-1} + 2\hat{\bar{B}}_2(t+1) \tag{3.17}$$

as [Keviczky and Haber 74] suggested. Inserting (3.13) into (3.15) gives $u(t+1) = u(t)$, and so it seems reasonable to believe that (3.13) and (3.15) will behave similarly if the estimates converge.

Since parameter b_0 is not included in the extremum control law, it is then not necessary to estimate it. An incremental form of the model (3.2) can be used to eliminate the constant coefficient b_0, but only if the model (3.2) can be modified by

$$A(q^{-1})y(t) = b_0 + B_1(q^{-1})u(t-d) + B_2(q^{-1})u^2(t-d) + C(q^{-1})\frac{e(t)}{\Delta} \tag{3.18}$$

where we assume that noise $e(t)$ is a drift disturbance, and ($\Delta = 1 - q^{-1}$) operator is an integral action to cancel the effect of the step output disturbances, $e(t)/\Delta$ can then be considered as a white noise with zero mean. The innovation form is thus given by

$$A(q^{-1})\Delta y(t) = B_1(q^{-1})\Delta u(t-d) + B_2(q^{-1})\Delta u^2(t-d) + C(q^{-1})e(t) \tag{3.19}$$

where

$$\begin{aligned} \Delta y(t) &= y(t) - y(t-1) \\ \Delta u(t) &= u(t) - u(t-1) \\ \Delta u^2(t) &= u^2(t) - u^2(t-1) \end{aligned} \tag{3.20}$$

3.2 Case studies

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

Example 3.1 *A nonlinear static system*

It is assumed that the model structure of the process is known and a nonlinear static system is considered in this example

$$y(t) = b_0 + b_1 u(t) + b_2 u^2(t) + e(t) \tag{3.21}$$

This corresponds to a process with input nonlinearity which is approximated by a second order polynomial. The parameters of the process are $b_0 = 100$, $b_1 = 2$ *and* $b_2 = -0.01$. *Further,* $e(t)$ *is a zero mean white noise with variance 1. The maximum attainable value of the output is* $y^* = 200$, *and optimal input value is* $u^* = 100$.

If it is assumed that the parameters of the model are unknown, then they have to be estimated. The linear regression used in estimation is

$$y(t) = \varphi^T(t)\hat{\theta} + \epsilon(t) \tag{3.22}$$

where the regression vector and parameter vector are

$$\begin{aligned} \varphi(t) &= [1,\ u(t),\ u^2(t)]^T \\ \hat{\theta} &= [\hat{b}_0,\ \hat{b}_1,\ \hat{b}_2]^T \end{aligned} \tag{3.23}$$

then the RLS method can be used to estimate the unknown parameters.

The extremum control law is the current estimate of the optimal point u^* incorporated with an extra disturbance signal $u_p(t)$

$$u(t) = -\frac{\hat{b}_1(t)}{2\hat{b}_2(t)} + u_p(t) \tag{3.24}$$

where a PRBS signal is selected as the perturbation signal $u_p(t)$ which ensures the persistent excitation of the input $u(t)$.

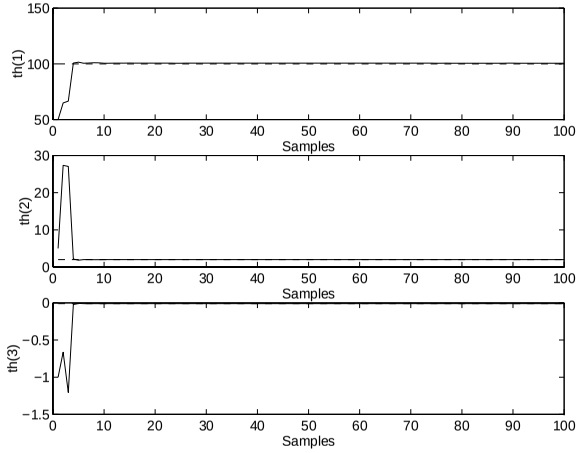

Figure 3.2. The estimated parameters.

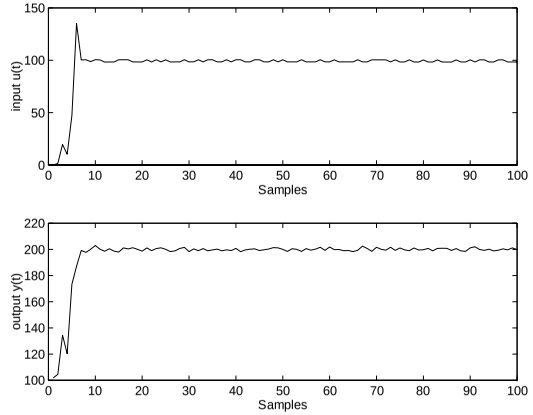

Figure 3.3. The system input and output.

The estimated parameters are shown in Figure 3.2, where $th(1) = \hat{b}_0$, $th(2) = \hat{b}_1$ and $th(3) = \hat{b}_2$. Since it is assumed to know that the process has a maximum point, the Hessian matrix must then be negative definite, i.e., $\hat{b}_2 < 0$. Therefore the initial value of $\hat{b}_2$ can be set to a negative value. The input and output of the system are shown in Figure 3.3.

The simulation shows that the estimated parameters converge very fast to the true values. The extremum controller achieves the optimal value of $y^* = 200$ very quickly and holds it there with minor perturbations which are caused by the test perturbation signal $u_p(t)$. ◇

Example 3.2 *A nonlinear dynamic system*

Now a nonlinear dynamic system is considered. The system is assumed to be described by a second-order Hammerstein model

$$y(t) + ay(t-1) = b_0 + b_1 u(t-1) + b_2 u^2(t-1) + e(t) \tag{3.25}$$

with parameter $a = -0.8$*, and parameters* b_0*,* b_1 *and* b_2 *have the same values as example 3.1. For this process , the maximum attainable value of the process output is* $y^* = 1000$ *and the optimal input is* $u^* = 100$*.*

If it all the parameters of the process are unknown, then the estimation algorithm has to be implemented. The linear regression (3.12) can be written by

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

with

$$\begin{aligned} \varphi(t) &= [-y(t-1),\ 1,\ u(t-1),\ u^2(t-1)]^T \\ \hat{\theta} &= [\hat{a},\ \hat{b}_0,\ \hat{b}_1,\ \hat{b}_2]^T \end{aligned} \tag{3.26}$$

The extremum control law is derived by maximizing the output of static response and the resulting controller has the same structure as the controller in example 3.1

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

and $u_p(t)$ *is a PRBS signal.*

Figure 3.4 and Figure 3.5 give the results of simulation. The estimated parameters are shown in Figure 3.4, where $th(1) = \hat{a}$*,* $th(2) = \hat{b}_0$*,* $th(3) = \hat{b}_1$ *and* $th(4) = \hat{b}_2$*. It can be seen that the estimated parameters converge fast, while output* $y(t)$ *achieves the maximum value after 20 steps. This*

is slower than the convergence speed of estimated parameters. The slower convergence speed is caused by the dynamics in the system. In short, after initial transient the adaptive extremum controller behaves as well as the controller in example 3.1.

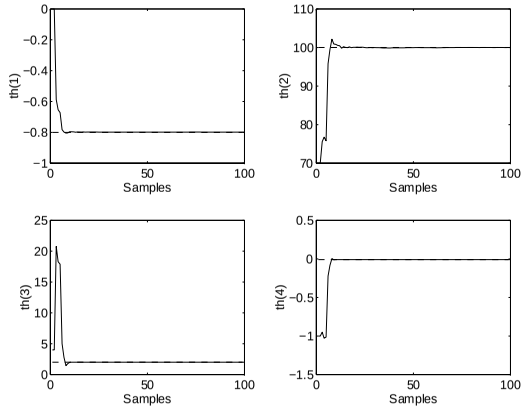

Figure 3.4. The estimated parameters.

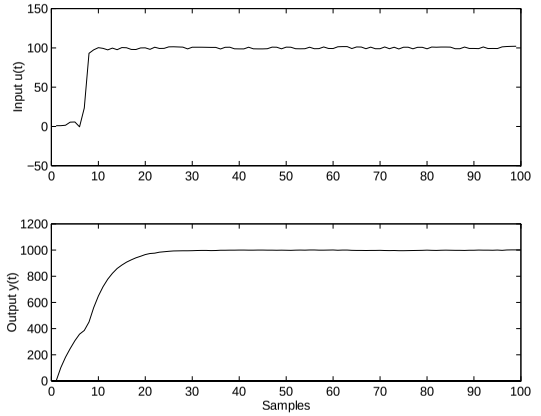

Figure 3.5. The system inputs and outputs.

◇

3.3 Convergence analysis

The convergence properties of the extremum controller can be analysed by the Ordinary Differential Equation (ODE) approach of Ljung [Ljung 77]. A brief summary of the approach is given in Appendix B.

Consider a system

$$S: \ y(t) = \varphi^T(t)\theta_0 + e(t) \tag{3.27}$$

and a model for the system is

$$M: \ y(t) = \varphi^T(t)\theta \tag{3.28}$$

the RLS estimator can be rewritten as

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + P(t)\varphi(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1)) \\ P(t+1)^{-1} &= P(t)^{-1} + \varphi(t)\varphi^T(t) \end{aligned} \tag{3.29}$$

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

$$R(t) = \frac{1}{t}P(t)^{-1} \tag{3.30}$$

the standard form of RLS will then be taken as

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + \tfrac{1}{t}R(t)^{-1}\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\ R(t) &= R(t-1) + \tfrac{1}{t}[\varphi(t)\varphi^T(t) - R(t-1)] \end{aligned} \tag{3.31}$$

In order to form the ODE, fix $\hat{\theta}(t)$ at some nominal value θ and perform the operations

$$\begin{aligned} f(\theta) &= \lim_{t\to\infty} E\{\varphi(t)[y(t) - \varphi^T(t)\theta]\} \\ G(\theta, R) &= \lim_{t\to\infty} E[\varphi(t)\varphi^T(t) - R] \end{aligned} \tag{3.32}$$

and the corresponding ODE is then given by

$$\begin{aligned} \frac{d\theta}{d\tau} &= R^{-1}f(\theta) \\ \frac{dR}{d\tau} &= G(\theta, R) \end{aligned} \tag{3.33}$$

If it is assumed that $e(t)$ in the system is uncorrelated with $\varphi(t)$, i.e., $E[\varphi(t)e(t)] = 0$, inserting (3.27) into (3.32), we have

$$\begin{aligned} f(\theta) &= G(\theta)(\theta_0 - \theta) \\ G(\theta, R) &= G(\theta) - R \end{aligned} \tag{3.34}$$

where

$$G(\theta) = \lim_{t\to\infty} E[\varphi(t)\varphi^T(t)] \tag{3.35}$$

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

$$\begin{aligned} f(\theta^*) &= G(\theta^*)(\theta_0 - \theta^*) = 0 \\ R^* &= G(\theta^*) \end{aligned} \tag{3.36}$$

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

$$\lim_{t\to\infty} E[\varphi(t)\varphi^T(t)] > 0 \tag{3.37}$$

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

$$H(\theta^*) = (R^*)^{-1}\frac{d}{d\theta}f(\theta)|_{\theta=\theta_0} = -I \tag{3.38}$$

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.

Example 3.3 *Convergence analysis*

We consider the most simple case of the nonlinear static system in Example 3.1, where the system is written as

$$y(t) = \varphi^T(t)\theta_{\mathbf{0}} + e(t) \tag{3.39}$$

where $\theta_{\mathbf{0}} = [b_0, b_1, b_2]^T$*, and model is given by*

$$y(t) = \theta_1 + \theta_2 u(t) + \theta_3 u^2(t) \tag{3.40}$$

The control law (3.24) is the estimate of extremum location $u_0 = -\theta_2/2\theta_3$*, to which is added a perturbation signal* $u_p(t)$*, then*

$$u(t) = \hat{u}_0(t) + u_p(t) \tag{3.41}$$

where $\hat{u}_0(t) = -\hat{\theta}_2/2\hat{\theta}_3$*. If the algorithm converges then* $\hat{u}_0(t) \to u_0$*, i.e.,*

$$u(t) = u_0 + u_p(t) \tag{3.42}$$

It is assumed that the perturbation signal $u_p(t)$ *is zero mean white noise with variance* σ_p^2*, i.e.,* $u_p(t) \in N(0, \sigma_p^2)$*, and furthermore it is assumed that* $u_p(t)$ *is independent of noise* $e(t)$ *in the process.*

By implementing the RLS estimator, if the estimates $\hat{\theta}(\mathbf{t}) \to \theta^*$*, we will have that the local convergence point of ODE satisfies*

$$f(\theta^*) = G(\theta^*)(\theta_{\mathbf{0}} - \theta^*) = 0$$

$$R^* = G(\theta)$$

It follows that $\theta_1 = b_0$*,* $\theta_2 = b_1$ *and* $\theta_3 = b_2$ *are the unique locally stable convergence point with the condition*

$$G(\theta) = \bar{E}\varphi(t)\varphi^T(t) > 0 \tag{3.43}$$

i.e., $G(\theta)$ *is positive definite and* $H(\theta^*)$ *matrix will have its eigenvalues at* -1*.*

Since

$$\begin{aligned}
\bar{E}\varphi(t)\varphi^T(t) &= \bar{E}\begin{bmatrix} 1 \\ u(t) \\ u^2(t) \end{bmatrix}[1 \;\; u(t) \;\; u^2(t)] \\
&= \bar{E}\begin{bmatrix} 1 \\ u_0 + u_p(t) \\ (u_0 + u_p(t))^2 \end{bmatrix}[1 \;\; u_0 + u_p(t) \;\; (u_0 + u_p(t))^2] \\
&= \begin{bmatrix} 1 & u_0 & u_0^2 + \sigma_p^2 \\ u_0 & u_0^2 + \sigma_p^2 & u_0^3 + 3u_0\sigma_p^2 \\ u_0^2 + \sigma_p^2 & u_0^3 + 3u_0\sigma_p^2 & u_0^4 + 6u_0^2\sigma_p^2 + 3\sigma^4 \end{bmatrix}
\end{aligned} \tag{3.44}$$

it can be proved that $det(\bar{E}\varphi(t)\varphi^T(t)) = 2\sigma_p^6 > 0$ *which implies that the test perturbation signal* $u_p(t)$ *ensures the persistent excitation. Without this condition the desired convergence properties are not guaranteed.* ◇

3.4 Summary

In this chapter an adaptive extremum control law is derived for the system which has input nonlinearity and (partly) unknown system dynamics.

An adaptive extremum controller is formed on a discrete time generalized Hammerstein model. By applying the Hammerstein model it is then possible to estimate the unknown parameters of the process. Furthermore, the extremum control law is derived in such a way that the extremum point

of the static characteristic should be chosen in every step. The optimal controller is given by the current estimate of extremum location achieved by an on-line recursive estimation algorithm. The identification problem is avoided by applying the persistently exciting input.

Two simulation examples have been presented to illustrate the performance of controller and estimator. The simulation results show the good convergence properties of the estimator, and processes achieve the extremum values very fast. The convergence properties of the algorithm have been analysed by using ODE approach.

Chapter 4

Output Nonlinearity

If the model consists of a dynamic linear part and static nonlinear part, and the linear dynamics is followed by a nonlinearity, it is called 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 depend on which of the variables in the process can be measured.

The models of the process with output nonlinearity have been given in chapter 2. Based on the models, an adaptive extremum control law will be derived in this chapter. Then the problem turns to two cases: whether the intermediate signal between the linear and nonlinear part can be measured or not. Most difficulties will arise in the situation when the intermediate signal is not measurable. Therefore we will give more attention to this case.

This chapter is organized as follows. The extremum control law is derived for the processes with output nonlinearity in section 4.1. Section 4.2 is concerned with the adaptive extremum control of process with output nonlinearity and the measured intermediated signal. The case for unmeasured intermediate signal is discussed in section 4.3 where the EKF and RPEM approach as parameter estimators will be applied to state-space models. A summary is given in section 4.4.

4.1 Basic extremum control law

For a process with output nonlinearity given by the model (2.9)-(2.11), if the noise terms $\xi(t)$ and $e(t)$ enter in the equations, the model will be modified by

$$A(q^{-1})z(t) = B(q^{-1})u(t-d) + C(q^{-1})\xi(t) \tag{4.1}$$

$$y(t) = g_0 + g_1 z(t) + g_2 z^2(t) + e(t) \tag{4.2}$$

where $\xi(t)$ and $e(t)$ are zero mean white noises. The model of the process consists of a nonlinear noise model.

The extremum control objective is to make the steady-state response of the output at maximum or minimum. The control law can be derived by optimizing the performance function (3.3) in section 3.1.

The steady-state response of the process with output nonlinearity given by model (4.1)-(4.2) is

$$\begin{aligned} \bar{A}z &= \bar{B}u \\ y &= g_0 + g_1 z + g_2 z^2 \end{aligned} \tag{4.3}$$

where $\bar{A} = A(1)$ and $\bar{B} = B(1)$. The above equations give the static output

$$\begin{aligned} y &= g_0 + g_1 \frac{\bar{B}}{\bar{A}} u + g_2 (\frac{\bar{A}}{\bar{B}} u)^2 \\ &= g_0 + g_1 \frac{\bar{B}}{\bar{A}} u + g_2 \frac{\bar{B}^2}{\bar{A}^2} u^2 \end{aligned} \tag{4.4}$$

Since only the mean value of the output is interested, the extremum control law can then be achieved by maximizing or minimizing the output of the static response, i.e.,

$$\frac{dy}{du} = 0 \tag{4.5}$$

which leads to

$$u(t) = -\frac{g_1 \bar{A}}{2 g_2 \bar{B}} + u_p(t) \tag{4.6}$$

and extremum point of the output is

$$y^* = g_0 - \frac{g_1^2}{4 g_2} \tag{4.7}$$

The optimal control law is constant and contains no feedback, therefore it is necessary to add a perturbation signal $u_p(t)$ to ensure the persistent excitation.

If process dynamic is unknown, the parameters have to be estimated, the control law will then be modified by

$$u(t) = -\frac{\hat{g}_1(t)\hat{\bar{A}}(t)}{2\hat{g}_2(t)\hat{\bar{B}}(t)} + u_p(t) \tag{4.8}$$

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 block and

nonlinear block, which is assumed by most people, the complexity will be significantly simplified. However, more search is needed to find out how to handle systems where the intermediate signal is not available. In this case, the problem is more difficult to handle. These two situations will be discussed separately in the following sections.

4.2 The intermediate signal is measurable

[Navarro and Zarrop 95] has given an adaptive extremum control algorithm by turning a controller to optimize a performance function

$$J(u) = E\{y(t+d)^2\} \tag{4.9}$$

where $y(t)$ is process output given by

$$A(q^{-1})y(t) = q^{-1}B(q^{-1})u(t) + e(t) \tag{4.10}$$

In this paper a performance function (4.9) rather than an input-output model (4.10) is estimated. This is actually a special case of the process with output nonlinearity and measurable intermediate signal. If the problem is compared to the model (4.1) - (4.2), $y(t)$ in (4.10) can thus be considered as an intermediate signal, and the model of the nonlinear block (4.2) will have the parameters $g_0 = 0$, $g_1 = 0$, and $g_2 = 1$.

Since the intermediate signal can be measured, it is then possible to do the system identification for the linear part and nonlinear part separately instead of estimate the performance function. The adaptive extremum control law (4.8) could then be used to keep the output of the process around the estimated position of the extremum.

The RELS algorithm has already been given in (3.12) in Chapter 3. The model of linear part used in parameter estimation is written as a linear regression

$$z(t) = \varphi_1^T(t)\theta_1 + \epsilon(t) \tag{4.11}$$

with

$$\begin{aligned} \varphi_1(t) &= [-z(t-1), \cdots - z(t-n_a), u(t-d), \cdots u(t-d-n_b), \\ &\quad \epsilon(t-1), \cdots, \epsilon(t-n_c)]^T \\ \theta_1 &= [a_1, \cdots a_{na}, b_1, \cdots, b_{nb}, c_1, \cdots, c_{nc}]^T \end{aligned} \tag{4.12}$$

The model of nonlinear part is

$$y(t) = \varphi_2^T(t)\theta_2 + \epsilon(t) \tag{4.13}$$

where

$$\begin{aligned} \varphi_2(t) &= [1,\ z(t),\ z^2(t)]^T \\ \theta_2(t) &= [g_0,\ g_1,\ g_2]^T \end{aligned} \tag{4.14}$$

If the intermediate signal between the linear part and nonlinear part can be known, the only difference between the output nonlinearity and input nonlinearity is that the input for the process with output nonlinearity is not determined directly, but through the linear dynamics. Some simple examples will be given to show the behaviour of adaptive extremum controller. The convergence properties of the algorithm can be analysed in the same way as it is given in previous section.

Example 4.1 *An unknown nonlinear dynamics*

Consider a system with linear part

$$z(t) + az(t-1) = bu(t-1) + e_1(t) \tag{4.15}$$

and nonlinear part

$$y(t) = g_0 + g_1 z(t) + g_2 z^2(t) + e_2(t) \tag{4.16}$$

where $a = -0.8$, $b = 1$, $g_0 = 100$, $g_1 = 2$ *and* $g_2 = -0.01$. *Further,* $e_1(t)$ *and* $e_2(t)$ *are zero mean white noise with variances* $\sigma^2_{e1} = 1^2$ *and* $\sigma^2_{e2} = 0.1^2$ *respectively. The maximum attainable value of the system output is 200, the corresponding optimal input is 20.*

In this example we assume that the linear dynamics of the process is known, i.e., parameters a and b are available. Only parameters g_0, g_1 *and* g_2 *for the nonlinear part are unknown. Therefore we have to estimate them. The RLS method can directly be used to estimate the parameters.*

Based on the estimated parameters, the extremum control law is given by

$$u(t) = -\frac{\hat{g}_1(t)(1+a)}{2\hat{g}_2(t)b} + u_p(t) \tag{4.17}$$

where $u_p(t)$ *is perturbation signal. The extremum control law is determined by both the linear dynamics and nonlinear dynamics. The perturbation signal is actually not necessary in this case, since the noise on the intermediate signal* $e_1(t)$ *acts as a perturbation signal and improves the identifiability.*

The estimated parameters are shown in Figure 4.1 where $th(1) = \hat{g}_0$, $th(2) = \hat{g}_1$ *and* $th(3) = \hat{g}_2$. *Figure 4.2 gives the input* $u(t)$, *output* $y(t)$ *and intermediate signal* $z(t)$. *The estimated parameters converge to the true value very fast. The output achieves the maximum value after 20 sec. Simulation shows good convergence property of adoptive extremum control algorithm.*

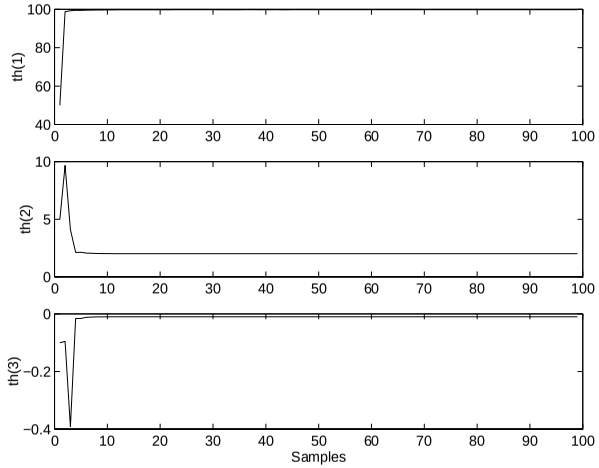

Figure 4.1. The estimated parameters.

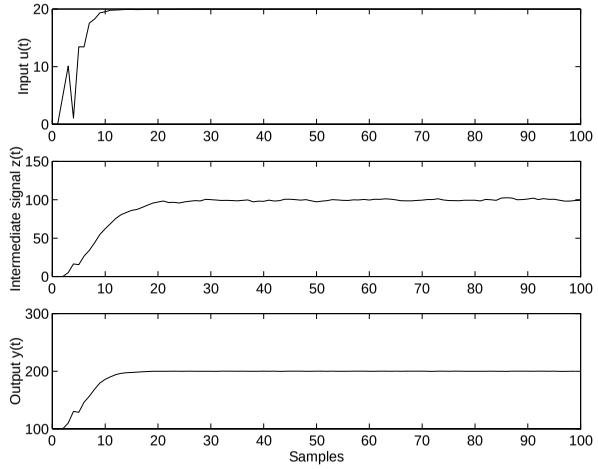

Figure 4.2. Input, output and intermediate signal

◇

Example 4.2 *Both linear dynamics and nonlinear dynamics are unknown*

Consider the process in example 4.1, if we assume that both parameters in linear part and nonlinear part are unknown, the identification for linear dynamics and nonlinear dynamics will then be implemented separately.

The model used to estimate parameters in linear part can be written in regression form (4.11) - (4.12), the regression vector and parameter vector are

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

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

and the model used to estimate parameters in nonlinear part is given in (4.13) - (4.14).

The extremum control law is identical in form with example 4.1, however parameters a and b are replaced by estimated values

$$u(t) = -\frac{\hat{g}_1(t)(1+\hat{a}(t))}{2\hat{g}_2(t)\hat{b}(t)} + u_p(t) \tag{4.18}$$

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

Simulations are shown in Figure 4.3 - Figure 4.5. The estimated parameters in linear part are given in Figure 4.3 where $th1(1) = \hat{a}$ and $th1(2) = \hat{b}$. The estimated parameters in nonlinear part are given Figure 4.4 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 4.5.

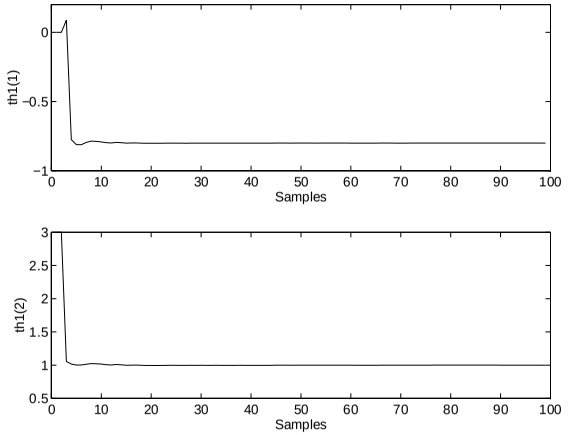

Figure 4.3. The estimated parameters.

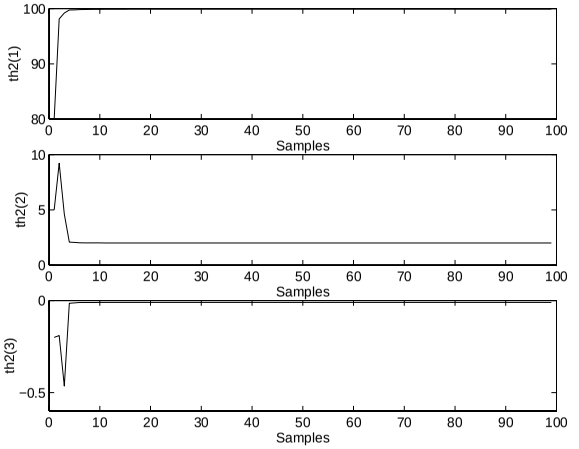

Figure 4.4. The estimated parameters.

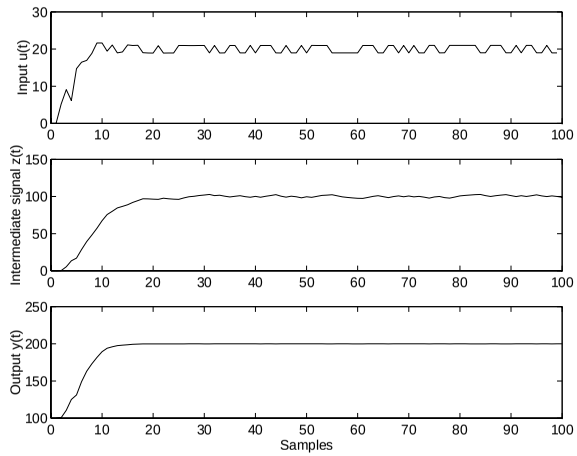

Figure 4.5. Input, output and intermediate signal

The simulation results show the good convergence properties of the estimator. The perturbations on the input signal are caused by $u_p(t)$. Since the estimated parameters converge to the true values very fast, the extremum controller behaves as well as the controller in example 4.1. ◇

4.3 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 by equation (2.12) that the model for the process with output nonlinearity will be nonlinear in parameters. 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 processes, 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, and the estimation problem can be solved by analyzing the parameter identifiability.

Different estimation algorithms 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 linear process, while these approaches will be investigated for a nonlinear system. 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.

A reliable process model is the basis for development of the model based control schemes. The extremum control strategy relies heavily on estimated model. Therefore the emphasis of this section will be given to the estimation problem which is also the most difficult part in extremum control algorithm.

4.3.1 The EKF as a parameter estimator for the nonlinear system

The algorithm

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 in [Ljung 79b], which is summarized in Appendix C.

The joint state and parameter estimation can of course be understood as a state estimation problem for a nonlinear system, where the unknown parameters are thought of random variables. Therefore it is a fairly natural thing to include the unknown parameters in the state vector. 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.

In this section the performance of the EKF algorithm used as a joint state and parameter estimation method for a nonlinear system rather than a linear system will be investigated.

First of all we suppose that the system dynamics can be described by a general nonlinear state-space model given in equation (2.31) - (2.32) in terms 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 (2.31) - (2.32) 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 $Q^w(\theta) = Q^w$, $Q^e(\theta) = Q^e$. It is also assumed that the noise sequences $\{w_\theta(t)\}$ and $\{e_\theta(t)\}$ are uncorrelated, $Q^{we}(\theta) = 0$.

The EKF approach given in Appendix C for a linear system to determine the unknown parameter vector θ will be modified for a nonlinear system (2.31) - (2.32). The unknown parameter vector θ in nonlinear model is obtained by extending the state vector x with the parameter vector θ

$$\underline{x}(t) = \begin{pmatrix} x(t) \\ \theta(t) \end{pmatrix} \tag{4.19}$$

the nonlinear state-space model will then be

$$\begin{aligned} \underline{x}(t+1) &= \underline{f}(\underline{x}(t), u(t)) + \begin{pmatrix} w(t) \\ 0 \end{pmatrix} \\ y(t) &= h(\underline{x}(t)) + e(t) \end{aligned} \tag{4.20}$$

where

$$\begin{aligned} \underline{f}(\underline{x}(t), u(t)) &= \begin{bmatrix} f(x(t), u(t), \theta(t)) \\ \theta(t) \end{bmatrix} \\ h(\underline{x}(t)) &= h(x(t), \theta(t)) \end{aligned} \tag{4.21}$$

The EKF algorithm for this joint state and parameter model is

$$\hat{\underline{x}}(t+1) = \underline{f}(\hat{\underline{x}}(t), u(t)) + K_{\underline{x}}(t)[y(t) - h(\hat{\underline{x}}(t))] \tag{4.22}$$

$$\begin{aligned} \hat{\underline{x}}(0) &= \hat{\underline{x}}_0 \\ K(t) &= [F(\hat{\underline{x}}(t), u(t))\bar{P}(t)H^T(\hat{\underline{x}}(t))] \\ &\quad \cdot [H(\hat{\underline{x}}(t))\bar{P}(t)H^T(\hat{\underline{x}}(t)) + Q^e]^{-1} \end{aligned} \tag{4.23}$$

$$\begin{aligned}\bar{P}(t+1) &= F(\underline{\hat{x}}(t),u(t))\bar{P}(t)F^T(\underline{\hat{x}}(t),u(t)) + \bar{Q}^w \\ &\quad - K(t)[H(\underline{\hat{x}}(t))\bar{P}(t)H^T(\underline{\hat{x}}(t)) + Q^e]K^T(t) \\ \bar{P}(0) &= \bar{P}_0\end{aligned} \tag{4.24}$$

where

$$\begin{aligned}F(\underline{\hat{x}}(t),u(t)) &= \frac{\partial}{\partial \underline{x}} f(\underline{x},u)\Big|_{\underline{x}=\underline{\hat{x}}(t)} \\ &= \begin{bmatrix} A(\hat{\theta}(t)) & M(\hat{\theta}(t),\hat{x}(t),u(t)) \\ 0 & I \end{bmatrix}\end{aligned} \tag{4.25}$$

$$\begin{aligned}H(\underline{\hat{x}}(t)) &= \frac{\partial}{\partial \underline{x}} h(\underline{x})\Big|_{\underline{x}=\underline{\hat{x}}(t)} \\ &= [C(\hat{\theta}(t)) \quad D(\hat{\theta}(t),\hat{x}(t))]\end{aligned} \tag{4.26}$$

$$\bar{Q}^w = \begin{bmatrix} Q^w & 0 \\ 0 & 0 \end{bmatrix} \tag{4.27}$$

$$\begin{aligned}\underline{\hat{x}}_0 &= \begin{bmatrix} \hat{x}_0 \\ \hat{\theta}_0 \end{bmatrix} \\ \bar{P}_0 &= \begin{pmatrix} \Pi(\hat{\theta}_0) & 0 \\ 0 & \Sigma_0 \end{pmatrix}\end{aligned} \tag{4.28}$$

and

$$\begin{aligned}A(\hat{x}(t),\hat{\theta}(t)) &= \frac{\partial}{\partial x} f(x(t),u(t),\hat{\theta}(t))\Big|_{x=\hat{x}(t)} (a\ \ n_x | n_x\ \ matrix) \\ M(\hat{x}(t),u(t),\hat{\theta}(t)) &= \frac{\partial}{\partial \theta} f(\hat{x}(t),u(t),\theta(t))\Big|_{\theta=\hat{\theta}(t)} (a\ \ n_x | n_\theta\ \ matrix) \\ C(\hat{x}(t),\hat{\theta}(t)) &= \frac{\partial}{\partial x} h(x(t),\hat{\theta}(t))\Big|_{x=\hat{x}(t)} (a\ \ n_y | n_x\ \ matrix) \\ D(\hat{x}(t),\hat{\theta}(t)) &= \frac{\partial}{\partial \theta} h(\hat{x}(t),\theta(t))\Big|_{\theta=\hat{\theta}(t)} (a\ \ n_y | n_\theta\ \ matrix)\end{aligned} \tag{4.29}$$

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

$$\begin{aligned}&\hat{x}(t+1) = f(\hat{x}(t),u(t),\hat{\theta}(t)) + K_x(t)[y(t) - h(\hat{x}(t),\hat{\theta}(t))] \\ &\hat{x}(0) = \hat{x}_0 \\ &\hat{\theta}(t+1) = \hat{\theta}(t) + K_\theta(t)[y(t) - h(\hat{x}(t),\hat{\theta}(t))] \\ &\hat{\theta}(0) = \hat{\theta}_0 \\ &K_x(t) = [A_tP_1(t)C_t^T + M_tP_2^T(t)C_t^T + A_tP_2(t)D_t^T + M_tP_3(t)D_t^T]S_t^{-1} \\ &K_\theta(t) = [P_2^T(t)C_t^T + P_3(t)D_t^T]S_t^{-1} \\ &S_t = C_tP_1(t)C_t^T + C_tP_2(t)D_t^T + D_tP_2^T(t)C_t^T + D_tP_3(t)D_t^T + Q^e \\ &P_1(t+1) = A_tP_1(t)A_t^T + A_tP_2(t)M_t^T + M_tP_2^T(t)A_t^T + M_tP_3(t)M_t^T \\ &\qquad\qquad - K_x(t)S_tK_x^T(t) + Q^w \\ &P_1(0) = \Pi(\hat{\theta}_0) \\ &P_2(t+1) = A_tP_2(t) + M_tP_3(t) - K_x(t)S_tK_\theta^T(t) \\ &P_2(0) = 0 \\ &P_3(t+1) = P_3(t) - K_\theta(t)S_tK_\theta^T(t) \\ &P_3(0) = \Sigma_0\end{aligned} \tag{4.30}$$

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

$$\begin{aligned} A_t &= A(\hat{x}(t), \hat{\theta}(t)) \\ C_t &= C(\hat{x}(t), \hat{\theta}(t)) \\ D_t &= D(\hat{x}(t), \hat{\theta}(t)) \\ M_t &= M(\hat{x}(t), u(t), \hat{\theta}(t)) \end{aligned} \tag{4.31}$$

It is a well-known fact that the behavior of the EKF as joint state and parameter estimator is sensitive to the 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 in [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.

Case study

A simulation example is given to test the performance of the EKF algorithm. The purpose of the example is to find the extremum point of the process with output nonlinearity and keep the output at this point. Since the dynamics of the process is unknown, and intermediate signal is not measurable, the EKF algorithm will be used as a joint state and parameter estimator. In the

example the model structure of the process is assumed to be in accordance with the true one.

Example 4.3 *For the model in example 4.1 in section 4.2, it is assumed that the dynamics of linear part is known, while the parameters in the model of nonlinear part have to be estimated. It is also assumed that the intermediate signal $z(t)$ is not able to measure.*

Let the model (4.15) - (4.16) be given by a state-space form

$$\begin{aligned} x(t+1) &= ax(t) + bu(t) + w(t) \\ y(t) &= h(\theta, x(t)) + e(t) \end{aligned} \tag{4.32}$$

and

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

with parameters $a = 0.8$, $b = 1$, $g_0 = 100$, $g_1 = 2$ 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 $\sigma_e^2 = 0.01^2$ respectively. The variances of measurement noise and the process noise in the model are chosen in accordance with those of the true system.

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

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

The extended Kalman filter algorithm (4.30) - (4.31) is applied on model (4.32) with

$$\begin{aligned} A_t &= a \\ M_t &= [0, \ 0, \ 0] \\ C_t &= \hat{g}_1(t) + 2\hat{g}_2(t)\hat{x}(t) \\ D_t &= [1, \ \hat{x}(t), \ \hat{x}^2(t)] \end{aligned} \tag{4.33}$$

The resulting algorithm will be

$$\hat{x}(t+1) = a\hat{x}(t) + bu(t) + K_x(t)[y(t) - h(\hat{\theta}(t), \hat{x}(t))] \tag{4.34}$$

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

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K_\theta(t)[y(t) - h(\hat{\theta}(t), \hat{x}(t))] \tag{4.35}$$

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

where

$$h(\hat{\theta}(t), \hat{x}(t)) = \hat{g}_0(t) + \hat{g}_1(t)\hat{x}(t) + \hat{g}_2(t)\hat{x}^2(t) \tag{4.36}$$

The state update matrix $K_x(t)$ and the parameter update matrix K_θ will be computed by applying algorithm (4.30).

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 4.6 - Figure 4.7. The estimated parameters are given in Figure 4.6, 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 achieves the extremum value by tuning the extremum control law.

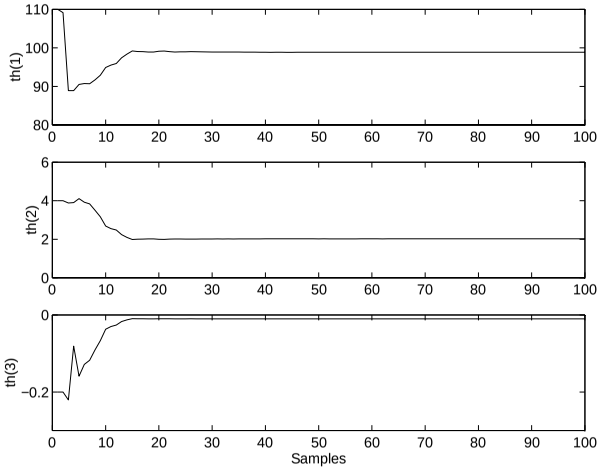

Figure 4.6. The estimated parameters.

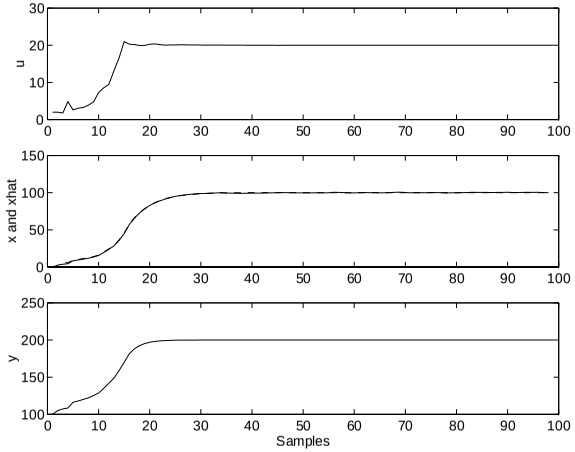

Figure 4.7. The input u and output y, state x (solid) and estimated state $\hat{x}$ (dashed).

The convergence of estimation 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 (4.30) 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 are available, the nonlinearity is weak and noise assumptions are close to the truth.

As it is stated in [Ljung 79b], the reason for the divergence of the 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 the steady-state Kalman gain does not depend on θ, we will have good convergence properties. The estimation problem is significantly simplified if the filter gain sensitivities to the parameters are neglected, i.e., $[dK_x/d\theta]\epsilon(t) = 0$. The estimation problem can then be separated into two estimators. The parameters were computed in a parameter estimator, and resulting model is used in a state estimator.

For a nonlinear system the algorithm is more sensitive. According to equations (C.37) - (C.39) in Appendix C, 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. However, for a nonlinear system, since the state update matrix $K_x(t)$ in the algorithm (4.30) 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 (4.33)). Therefore the 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 $\psi(t) = -d/d\theta\epsilon(t)$ has a tendency to be incorrect. ◇

4.3.2 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 which minimizes prediction error criterion will lead to a recursive prediction error method (RPEM). The RPEM is take into consideration, since Ljung 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 RPEM algorithm.

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

The algorithm

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

$$V(\theta) = \frac{1}{2} E\epsilon^T(t,\theta)\Lambda^{-1}\epsilon(t,\theta) \tag{4.37}$$

is given in Appendix D. In this section the method obtained for a linear system in Appendix D will be modified for a nonlinear system.

A general nonlinear state-space model is given by equations (2.31) - (2.32), whereas the RPEM will apply to a nonlinear innovations model, which is written as

$$\begin{aligned} x(t+1) &= f(x(t) + K_x(\theta(t))\epsilon(t), u(t), \theta(t)) \\ y(t) &= h(x(t), \theta(t)) + \epsilon(t) \end{aligned} \tag{4.38}$$

with

$$E\epsilon(t)\epsilon^T(t) = \Lambda\delta_{ts} \tag{4.39}$$

A very simple and natural feather is to make the recursion in the RPE algorithm 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 [Ljung 79b].

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

$$\hat{x}(t+1|t) = f(\hat{x}(t|t), u(t), \theta) \tag{4.40}$$

and date update will be

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + k_x(\theta)\epsilon(t+1) \tag{4.41}$$

the prediction error is

$$\epsilon(t+1) = y(t+1) - h(\hat{x}(t+1|t), \theta) \tag{4.42}$$

Differentiating (4.40) and (4.41) gives

$$\frac{d}{d\theta}\hat{x}(t+1|t) = A(\hat{x}(t|t),\theta)\frac{d}{d\theta}\hat{x}(t|t) + M(\hat{x}(t|t), u(t), \theta) \tag{4.43}$$

and

$$\begin{aligned} \frac{d}{d\theta}\hat{x}(t+1|t+1) = \frac{d}{d\theta}\hat{x}(t+1|t) - k_x(\theta)C(\hat{x}(t+1|t),\theta)\frac{d}{d\theta}\hat{x}(t+1|t) \\ - k_x(\theta)D(\hat{x}(t+1|t),\theta) + [\frac{d}{d\theta}k_x(\theta)]\epsilon(t+1) \end{aligned} \tag{4.44}$$

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

$$\begin{aligned} A(\hat{x}(t|t),\theta) &= \tfrac{\partial}{\partial x} f(x(t|t), u(t), \theta)\big|_{x=\hat{x}(t|t)} \\ M(\hat{x}(t|t), u(t), \theta) &= \tfrac{\partial}{\partial \theta} f(\hat{x}(t|t), u(t), \theta)\big|_{\theta=\theta} \\ C(\hat{x}(t+1|t),\theta) &= \tfrac{\partial}{\partial x} h(x(t+1|t),\theta)\big|_{x=\hat{x}(t+1|t)} \\ D(\hat{x}(t+1|t),\theta) &= \tfrac{\partial}{\partial \theta} h(\hat{x}(t+1|t),\theta)\big|_{\theta=\theta} \end{aligned} \tag{4.45}$$

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, (4.43) and (4.44) can be rewritten as

$$W(t+1|t) = A(\hat{x}(t|t),\theta)W(t|t) + M(\hat{x}(t|t), u(t), \theta) \tag{4.46}$$

$$\begin{aligned} W(t+1|t+1) &= [I - k_x(\theta)C(\hat{x}(t+1|t),\theta)]W(t+1|t) \\ &\quad - k_x(\theta)D(\hat{x}(t+1|t),\theta) + [\tfrac{d}{d\theta}k_x(\theta)]\epsilon(t+1) \end{aligned} \tag{4.47}$$

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

$$\psi(t+1) = C(\hat{x}(t+1|t),\theta)W(t+1|t) + D(\hat{x}(t+1|t),\theta) \tag{4.48}$$

In practice the algorithm (D.18) in Appendix D is not implemented in a straight forward way with matrix R^{-1} and $\hat{\Lambda}^{-1}$. The matrix inversion lemma is used to derive an equivalent form of the algorithm. Therefore we introduce

$$P(t) = \gamma(t)R^{-1}(t) \tag{4.49}$$

The RPE algorithm is summarized

$$\begin{aligned} &\hat{x}(t|t-1) = f(\hat{x}(t-1|t-1), u(t-1), \hat{\theta}(t-1)) \\ &W(t|t-1) = A_t W(t-1|t-1) + M_t \\ &\hat{y}(t) = h(\hat{x}(t|t-1), \hat{\theta}(t-1)) \\ &\epsilon(t) = y(t) - \hat{y}(t) \\ &\psi(t) = C_t W(t|t-1) + D_t \\ &\hat{\Lambda}(t) = \hat{\Lambda}(t-1) + \gamma(t)[\epsilon(t)\epsilon^T(t) - \hat{\Lambda}(t-1)] \\ &S(t) = \psi^T(t)P(t-1)\psi(t) + \lambda(t)\hat{\Lambda}(t) \\ &K_\theta(t) = P(t-1)\psi(t)S^{-1}(t) \\ &\hat{\theta}(t) = \hat{\theta}(t-1) + K_\theta(t)\epsilon(t) \\ &P(t) = [P(t-1) - K_\theta(t)S(t)K_\theta^T(t)]/\lambda(t) \\ &\hat{x}(t|t) = \hat{x}(t|t-1) + k_x(\hat{\theta}(t))\epsilon(t) \\ &W(t|t) = [I - k_x(\hat{\theta}(t))C_t)]W(t|t-1) - k_x(\hat{\theta}(t))D_t \\ &\qquad + [\frac{d}{d\theta}k_x(\hat{\theta}(t))]\epsilon(t) \end{aligned} \tag{4.50}$$

where

$$A_t = A(\hat{x}(t-1|t-1), \hat{\theta}(t-1))$$

$$M_t = M(\hat{x}(t-1|t-1), u(t-1), \hat{\theta}(t-1))$$

$$C_t = C(\hat{x}(t|t-1), \hat{\theta}(t-1))$$

$$D_t = D(\hat{x}(t|t-1), \hat{\theta}(t-1))$$

According to Ljung, the RPE algorithm should contain 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

$$\begin{aligned} D_s &= \{\theta | A(\theta) - K_x(\theta)C(\theta) \textit{ has all eigenvalues} \\ &\quad \textit{strictly inside the unit circle}\} \end{aligned} \tag{4.51}$$

and projection algorithm is given in Appendix D. For the convergence results, $\hat{\theta}(t) \in D_s$ (at least infinitely often).

Case studies

A simulation example is given to investigate the performance of the RPE algorithm.

Example 4.4 *We take the model from example 4.3 in section 4.3.1. In this example we will assume that we have not any knowledge about dynamics of the linear part, while the dynamics of the nonlinear part is known. The variances matrices of the noise Q^w and Q^e are also assumed to be unknown. For model (4.32) an innovations model is given by*

$$\begin{aligned} x(t+1) &= ax(t) + bu(t) + K_x\epsilon(t) \\ y(t) &= h(t, x(t)) + e(t) \end{aligned} \tag{4.52}$$

where

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

The estimated parameter vector θ will be

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

The two step RPEM (4.50) will be implemented for the innovations model with

$$A_t = \hat{a}(t-1)$$

$$C_t = g_1 + 2g_2\hat{x}(t|t-1)$$

$$D_t = [0, \ 0, \ 0]$$

$$M_t = [\hat{x}(t-1|t-1), \ u(t-1), \ 0]$$

The extremum control law based on the estimated model is

$$u(t) = -\frac{g_1(1+\hat{a})}{2g_2\hat{b}}$$

The simulation results are given in Figure 4.8 - 4.9. The estimated parameters are shown in Figure 4.8, where $th(1) = \hat{a}$, $th(2) = \hat{b}$ and $th(3) = \hat{k_x}$. The true values of parameters a and b are given by the dashed line in the figure. The estimates converge to the correct values very fast. The input, output and state are given in Figure 4.9. The output achieves the extremum value quickly. The algorithm shows a satisfactory performance.

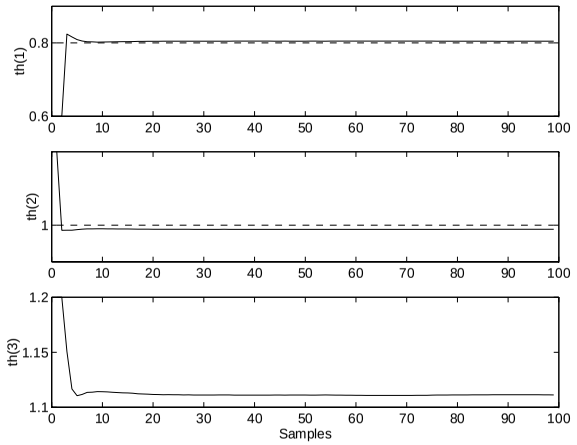

Figure 4.8. The estimated parameters.

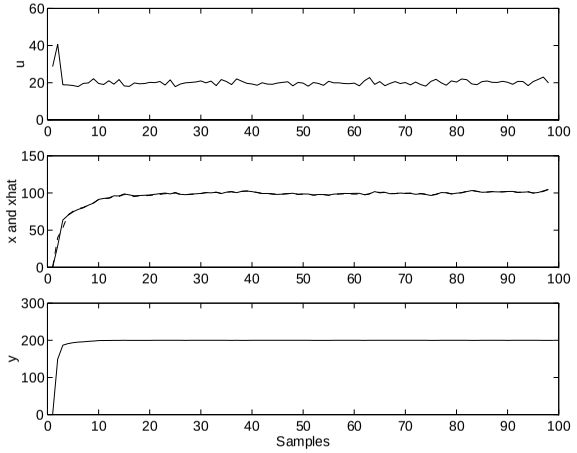

Figure 4.9. The input u and output y, state x (solid) and estimated state $\hat{x}$ (dashed)

Simulation experiences show that the RPEM algorithm in two steps gives a better transient response than the original RPEM. This is because the estimated sensitivity matrix ψ is counted in parameter estimates at the first step. The simulation will not be heavily influenced by the assumption of initial K_x. However, there is a little bias in the estimated parameters.

The bad assumptions of the initial parameter estimates and state estimates will make the simulation diverge, which means that good assumptions of initial values are required for a successful simulation.

In the innovations model (4.38), the steady-state Kalman gain, rather than the covariance matrices, is parameterized. This steady-state Kalman gain is used in the RPE algorithm all the time as a time-invariant predictor. 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. ◇

4.3.3 The modified recursive prediction error method

A modified recursive prediction error method based on a line search strategy was suggested by Ljungquist in [Ljungquist and Balchen 93]. The 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.

Line search technique

A line search method used to solve optimization problem has been in Appendix A.6. When the search direction has been determined for parameter estimator

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \alpha(t) K_\theta(t) \epsilon(t) \tag{4.53}$$

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

$$V_j(t) = \epsilon_j(t)^T \hat{\Lambda}^{-1}(t) \epsilon_j(t) + \Delta\hat{\theta}_j(t)^T P^{-1}(t) \Delta\hat{\theta}_j(t) \tag{4.54}$$

where

$$\Delta\hat{\theta}_j(t) = \hat{\theta}_j(t) - \hat{\theta}(t-1) \tag{4.55}$$

$\Lambda(t)$ and $P(t)$ are positive definite matrices.

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

$$\hat{\theta}_j(t) = \hat{\theta}(t-1) + \alpha_j(t) K_\theta(t) \epsilon(t) \tag{4.56}$$

where $\alpha_j(t)$ is the update stepsize. Once the search direction at each discrete sample time is determined, the value of criterion (4.54) is a function of the stepsize α alone. Then the optimal stepsize can be obtained by minimizing the criterion

$$\alpha_0(t) = \min_\alpha V(\alpha, t) \tag{4.57}$$

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 (4.54) 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, which has been given in Appendix A.6. The α_0 is obtained by minimizing an approximating polynomial of $V(\alpha, t)$ in α of degree two. Since an one sample criterion is used, the criterion can easily be computed for each stepsize.

The principle of the approach is illustrated in Figure 4.10.

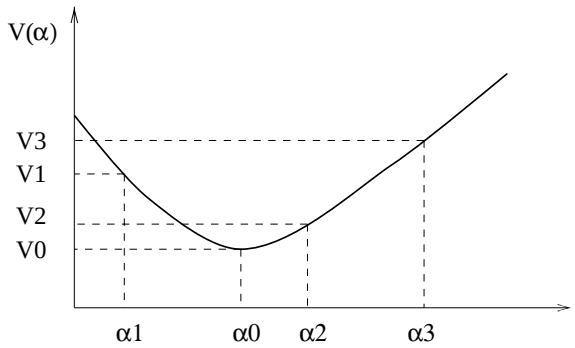

Figure 4.10. Optimal stepsize

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 (4.54), and the optimal α_0 can be obtained by

$$\alpha_0 = \frac{1}{2}\frac{(\alpha_2^2 - \alpha_3^2)V_1 + (\alpha_3^2 - \alpha_1^2)V_2 + (\alpha_1^2 - \alpha_2^2)V_3}{(\alpha_2 - \alpha_3)V_1 + (\alpha_3 - \alpha_1)V_2 + (\alpha_1 - \alpha_2)V3} \qquad (4.58)$$

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.

The modified RPEM

A recursive line search prediction error method is given 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 (4.54)
end
α_0 can be calculated by equation (4.58).

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.

Step 6 to 9 in the algorithm are used to recompute the state estimates with the most recent parameter estimates.

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

Case studies

Example 4.5 *As the model in example 4.4, the same parameters will be estimated by implementing recursive line search prediction error method.*

The simulation results are presented in Figure 4.11 - 4.12. The dashed lines in Figure 4.11 are the true parameter values of the process, the solid lines are the estimated parameters by implementing the line search RPEM, and point lines are the estimated parameters by using two step RPEM. The input, output and state given in Figure 4.12 are the results by implementing the line search RPEM.

When the initial assumptions of the estimated parameters and states are not good enough, the two step RPEM will give a biased estimation, however, the modified RPEM can give an unbiased estimation.

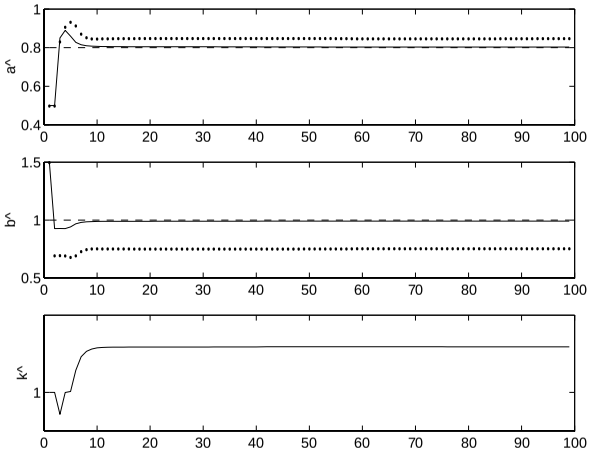

Figure 4.11. The estimated parameters.

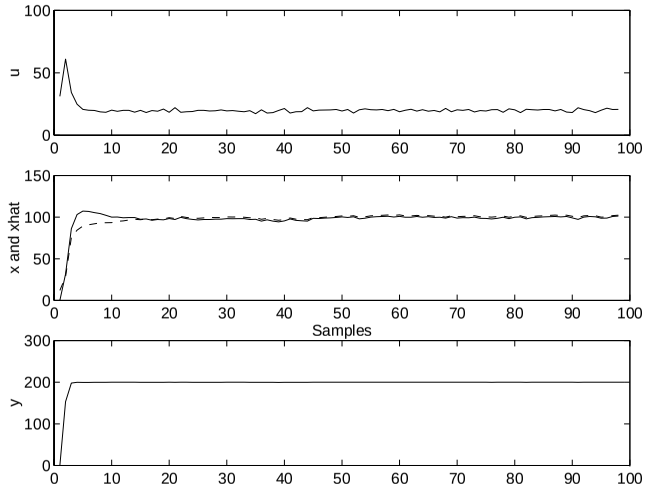

Figure 4.12. The input and output

The simulation indicates that 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. $\diamond$

4.4 Summary

When the intermediate signal between linear part and nonlinear part of the nonlinear model is measurable, the problem will be simplified. The identification can then be implemented for linear part and nonlinear part

separately. In this case the only difference between the output nonlinearity and input nonlinearity is that input for the process with output nonlinearity is not determined directly but though the linear dynamics.

When the intermediate signal between linear part and nonlinear part is not measurable, the parameter estimation becomes very complex. Since the extremum control law relies heavily on the estimated model, the estimation problem is then the key point of implementation of the control law.

Some of the conclusions reached in this chapter are based on well-established theory, others are based on simulation studies, since nonlinear systems are difficult to deal with strictly theoretically. Clearly, simulation studies do not prove that one strategy is in general better than another, but they can show that one approach is preferable to another for a class of problems.

For the system with known noise structure and weak nonlinearity, the EKF as a joint state and parameter estimator works well. However, for the system with unknown dynamics and unknown noise structure, the method may give biased estimates, and it does not seldom diverge if the initial estimates are not sufficiently good. The reason for divergence can be interpreted that the effect on the Kalman gain K_x of a change in θ is not taken into account. This lack of coupling between $K_x(t)$ and θ in the algorithm may lead to divergence of the estimates.

Example 4.3 simulates of the behaviour of the EKF estimator and extremum controller on a nonlinear system with known noise variances. The EKF algorithm and extremum control law work well in this case. However, the EKF algorithm as a state and parameter estimator is more sensitive when it is applied to a nonlinear system rather than a linear system. For

a nonlinear system, the state update matrix $K_x(t)$ in the algorithm is calculated by using a linearized model which is actually an approximation of the true model. The approximation will make the algorithm more sensitive. 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.

In order to improve the global convergence results, the RPEM is applied for an innovations model. The $[d/d\theta\bar{K}_x(\theta)]\epsilon(t)$ is included in the RPE algorithm. In example 4.4 a two step RPE algorithm is applied to an innovations model. The estimator and regulator work well if the initial assumption about estimated parameters and states are good enough. 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. However, for a nonlinear system K_x will vary rapidly as the inputs and the operational point change, at least when the estimation has not converged and extremum point has not been reached. 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.

Both the EKF algorithm as joint state and parameter estimation method and the RPEM can be successfully applied to a number of estimation problems.

A modified RPE algorithm based on line-search strategy is developed in section 4.3.3. In this algorithm the line-search performs in the computed search direction to find an optimal stepsize during each iteration. Example 4.5 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.

Chapter 5

Conclusions

Problems of extremum control have been frequently studied by a good number of authors. The reason may be the possibility to describe several interesting practical problems. There are, e.g., industrial processes where the object of control is to maximize some physical variable such as an efficiency. If there exists some optimal value for the control variable giving maximal value for the output, then the problem is of this type.

To perform adaptive extremum control of nonlinear systems, it is necessary to make assumptions about the structure of the process. One of class of systems is obtained by dividing the process into a nonlinear static part and linear dynamic part. Approximation theory is used to derive different types of series expansion representations of nonlinear systems. The representations include Volterra, Wiener and Uryson series. They can be used to model the processes where the linear dynamics is followed by a nonlinearity.

The Uryson series can also include the nonlinearity at the input. A special case of Uryson series is represented by Hammerstein models, which are common configurations considered in the extremum control systems. The state-space models can also be employed to describe nonlinear processes.

For a system with input nonlinearity, if the systems are unknown, they have to be identified. Most identification methods are based on the assumption that the model is linear in parameters. A Hammerstein model of the process with the input nonlinearity has this kind of property. An on-line recursive estimation algorithm, e.g., the recursive extended least squares (RELS) method is applied to identify the parameters of the model. The extremum control law is derived by maximizing or minimizing the static response of process output, which will keep the process operating at, or in the vicinity of, the extremum point of the performance function or process output despite of the influence of the disturbances. A test perturbation signal is necessary to ensure the parameter identifiability, which has been proved by analyzing convergence properties by using ODE approach. Two examples have been given to show the good convergence properties of the estimator and good behaviour of the extremum controller.

For the process with output nonlinearity, two different cases have been discussed according to that the intermediate signal can be measured or not. Different models and estimation methods will be used at different cases.

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

linear dynamics. The simulation examples show the good behaviour of the estimator and controller.

If the intermediate signal is not measurable, the problem becomes quite complex. Since the extremum control law depend heavily on the estimated model, the estimation problem becomes the key point of the adaptive extremum control algorithm.

First of all the extended Kalman filter as a joint state and parameter estimator is considered. A state-space model of the process is employed in order to implement the EKF algorithm. Simulations for a process with unknown nonlinear dynamics and known noise structure show the good convergence properties of the 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. The algorithm may converge only if the assumptions about the noise structure of the model are in accordance with those of the true system. Otherwise the estimates will be biased.

However, the EKF algorithm is more sensitive when it is applied to a nonlinear system. Since the state update matrix K_x is computed based on a linearized model that is actually an approximation of the true model. When the linearized model is applied, the steady-state Kalman depends also on the estimated states. The algorithm will then be more sensitive. The parameter estimates will be poor when the estimated states are not close to the true values. The poor model will make the estimation to diverge.

Ljung suggested the recursive prediction error method (RPEM) applied for an innovations model to improve the global convergence results. Simulation indicates that estimator and regulator work well if the initial assumptions about estimated parameters and states are good enough. The RPEM based on the innovations state-space model implies that the algorithm in principle is based on the assumption that the state updating matrix K_x is constant, at least asymptotically. However, for the nonlinear system K_x will change rapidly as the inputs and the operational point change.

A modified RPE algorithm based on line-search strategy is preferable for a nonlinear system. In this algorithm the line-search performs in the computed search direction to find an optimal stepsize during each iteration. Simulations have showed that the modified RPE algorithm has a better performance than the original RPE algorithm for a nonlinear process with unknown linear dynamic and noise structure. The increased robustness of the modified RPE is especially important when the estimation scheme is to be applied online.

The simulations indicate that both the EKF algorithm as joint state and parameter estimation method and the RPEM can be successfully applied to a number of estimation problems. It is consequently impossible to state that one algorithm is always preferable to another. The available information is very important to select the best method for a given application.

Bibliography

[Bamberger and Isermann 78] W. Bamberger and R. Isermann. Adaptive On-line Steady-State Optimization of Slow Dynamic Process. *Automatica*, Vol. 14, pp 223-230. 1978.

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

[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.

[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 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. 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.

[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.

[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 Nonlinear 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. *IEE Conference Publication,* No. 389, March, 1994.

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

Part II

Modelling and Control of A Wind Turbine

Glossary

Notations

$\mathbf{A_c}$,$\mathbf{B_c}$,$\mathbf{C_c}$,$\mathbf{D_c}$,$\mathbf{L_c}$	Matrices in continuous-time state-space model	
$\mathbf{A_d}$,$\mathbf{B_d}$,$\mathbf{C_d}$,$\mathbf{D_d}$	Matrices in discrete-time state-space model	
$\mathbf{A_{3p}}$,$\mathbf{B_{3p}}$,$\mathbf{C_{3p}}$	Matrices in state-space 3p-effect model	
$\mathbf{A_w}$,$\mathbf{B_w}$,$\mathbf{C_w}$	Matrices in state-space wind model	
$\mathbf{A_p}$,$\mathbf{B_p}$,$\mathbf{C_p}$,$\mathbf{D_p}$,$\mathbf{L_p}$	Matrices in state-space turbine model	
C_p	Power coefficient	
$C_{p,max}$	Maximum power coefficient	
C_t	Force coefficient	
D_g	Slope of torque-speed curve	[Nms/rad]
D_t	Tower damping	[kg/s]
$\mathbf{e}$	White noise vector	
f	Frequency	[Hz]
F_t	Axial force	[N]
h	Measuring height	[m]
h_t	Tower height	[m]
H_i	Height of convective boundary layer	[m]
J	Performance function	
J_r	Rotor and low speed shaft inertia	$[kgm^2]$
J_g	High speed shaft and generator inertia	$[kgm^2]$
k	Gain of wind speed model	
$\mathbf{K}$	Kalman gain	
k_{3p}	Gain in 3P-effect model	

k_t	Tower stiffness	$[kg/s^2]$
K_p	Proportional gain in PI regulator	$[^o/KW]$
K_s	Spring coefficient of the drive train	[Nm/rad]
K_β	Gain of pitch actuator	
l_t	Nacelle displacement	[m]
L	Monon-Obukhov length	[m]
$\mathbf{L}$	Feedback gain	
m_t	Mass of nacelle and rotor	[kg]
m_1	Phase number of generator stator	
n_{gear}	Gear ratio	
n_p	Number of pole pairs in generator	
p_1, p_2	Parameters of wind model	
P_e	Produced electrical power	[W]
P_r	Rotor power	[W]
$\mathbf{Q_y}$,$\mathbf{Q_u}$,$\mathbf{Q_i}$	Weighting matrices	
Q_1, Q_2	Weighting parameters	
r	Roughness length	[cm]
$\mathbf{r}$	Reference	
R	Rotor radius	[m]
R_1,R_2	Resistances of generator stator and rotor winding	
$\mathbf{R_1}$,$\mathbf{R_2}$	Variance matrices of state and measurement noise	
S	Slip	
T_{em}	Electromagnetic torque	[Nm]
T_g	Generator mechanical torque	[Nm]
$T_{g,ref}$	Generator torque reference	[Nm]
T_{hs}	Torque on the high speed shaft	[Nm]
T_i	Integration time in PI regulator	[s]
T_{ls}	Torque on the low speed shaft	[Nm]
T_n	3P-effect on torque	[Nm]

T_r	Rotor torque	[Nm]
T_s	Sampling time	[s]
T_r	Rotor aerodynamic torque	[Nm]
$\mathbf{u}$	Input vector	
v	Effective wind speed experienced by the rotor	[m/s]
$\hat{v}$	Estimated wind speed	[m/s]
v_m	Average wind speed	[m/s]
v_{wind}	Wind speed	[rad/s]
v_{*0}	Friction velocity	[m/s]
$\mathbf{w}$	State noise vector	
$\mathbf{x}$	State vector	
$\mathbf{x_i}$	Integral state	
$\mathbf{x_w}$	Wind model state vector	
$\mathbf{x_{3p}}$	3P-effect model state vector	
$\hat{\mathbf{x}}$	Estimated state vector	
$\bar{\mathbf{x}}$	Augmented state	
$\mathbf{y}$	Output vector	
β	Pitch angle	[deg]
β_{ref}	Reference pitch angle	[deg]
η	Generator efficiency	
η_{gear}	Efficiency coefficient of gearbox	
κ	Von Kármán constant	
λ	Tip speed ratio	
λ_{opt}	Optimal tip speed ratio	
ω	Frequency	[rad/s]
ω_0	Grid frequency	[rad/s]
ω_g	Angular velocity of high speed shaft and generator	[rad/s]

ω_r	Angular velocity of rotor	[rad/s]
$\omega_{r,ref}$	Rotor speed reference	[rad/s]
$\omega_{r,opt}$	Optimal rotor speed	[rad/s]
ω_t	Angular velocity of tower bending movement	[rad/s]
ρ	Air density	$[kgm^3]$
σ_p^2	Variance of point wind speed	
σ_v^2	Variance of effective wind speed	
τ_β	Time constant of pitch actuator	
θ_g	High speed and generator shaft angle	[rad]
θ_{ls}	Low speed shaft angle	[rad]
θ_r	Rotor shaft angle	[rad]
θ_t	Angle of tower bending	[deg]
θ_ϵ	Torsion of drive train	[rad]
ζ	Damping ratio in 3P-effect model	

Abbreviations

LQG	Linear Quadratic Gaussian control law
RDE	Riccati Difference Equation
CS	Constant Speed
VS	Variable Speed

Chapter 6

Introduction

Owing to current concern over the environment, there is much interest in wind power. In recent years wind turbine technology has undergone rapid development in response to the demands for increased use of renewable sources of energy. Using a wind turbine for production of electrical energy requires reliable operation. Especially at high wind speeds fluctuations from the wind result in large mechanical loads of the turbine. Therefore an active control system is often used to realize a long lifetime of turbine and produce high quality power or increase energy capture. The main goal of the project is to develop high performance control systems for a wind turbine.

6.1 A wind turbine

There are two basic configurations, i.e., the horizontal axis wind turbine and vertical axis wind turbine. The wind turbine focussed on here is a large scale horizontal axis wind power plant. The configuration of a wind turbine is depicted in Figure 6.1. The main components of the turbine are the tower, nacelle, rotor blades and hub. The nacelle contains the drive train and generator.

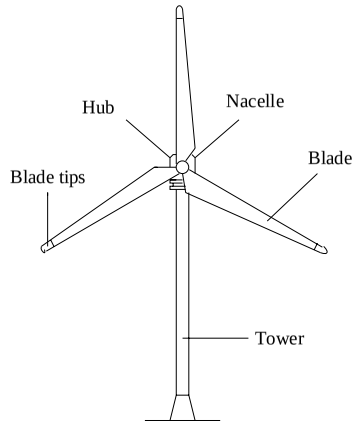

Figure 6.1. Wind turbine configurations

A WD34 wind turbine from *Vestas - Danish Wind Technology A/S* is chosen as an example for investigation of control systems. The WD34 is a large scale horizontal axis wind power plant with rated power of 400 KW. The plant has a three bladed rotor and an automatic yaw system to turn the rotor into the wind. The variable pitch capability is employed in a regulating fashion, with all blades acting unison. The turbine is connected to a large utility grid, a consequence of which is the lock of the rotational speed of the generator to the frequency of the grid, and hence the plant is operated

at nearly constant rotor speed. The most important data of the plant are listed in Appendix E.

6.2 The turbine control problem

Wind turbine control is an application area with an interesting set of problems for control engineering. A review of wind turbine control has been given by [Salle et al., 1990]. In control engineering terms, the wind turbine is a dynamic system excited by a disturbance input, the wind, and measurement noise.

The turbine is normally operated between a lower and upper limited wind speed. It can be started at a cut-in wind speed, and shut down at a cut-off wind speed. When the wind speed drops too low, the produced energy by the turbine is not enough to compensate for the consumed energy by the turbine, i.e., the turbine can not generate worthwhile quantities of power, the turbine is then stopped. When the wind speed is too high it is again stopped since it would be uneconomic to construct the turbine to be robust enough to operate in all wind speeds. As the wind speed increases, the energy available for capture increases as roughly the cube of the wind speed. The high wind speeds are not encountered frequently enough to make it economic to extract the energy available. A correspondingly high rating is required for power train. At a predetermined rated wind speed the power input to the wind turbine will have reached the limit for continuous operation. When the wind speed exceeds this level the excess power in the wind must be discarded by the rotor to prevent the turbine overloading. The power is maintained at its rated value until a maximum wind speed is reached when the turbine is

shut down [Leithead et al., 1991]. A typical power curve is shown in Figure 6.2.

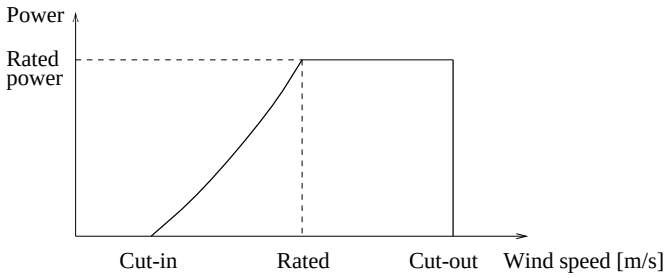

Figure 6.2. Typical power curve

The control design objectives are summarized as follows

- maximization of energy capture in partial load.
- limiting and smoothing of electrical power in full load.
- minimization of the turbine transient loads and thereby maximization of the turbine life in both partial and full load.

The control strategies are divided into below rated operation (partial load) and above rated operation (full load). The measurement of the wind speed can be used to determine whether the wind turbine is operating above or below the rated wind speed.

The turbine can be operated in one of two modes, fixed speed and variable speed. In fixed speed mode, the turbine is directly connected to the public grid and the rotor is constrained to nearly constant speed, see Figure 6.3. In variable speed mode, the generator is connected to the grid via a power conversion unit which might be a rectifier-inverter system and rotor speed is allowed to vary, see Figure 6.4.

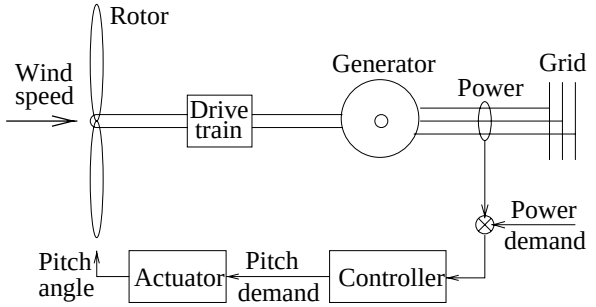

Figure 6.3. A constant speed wind turbine

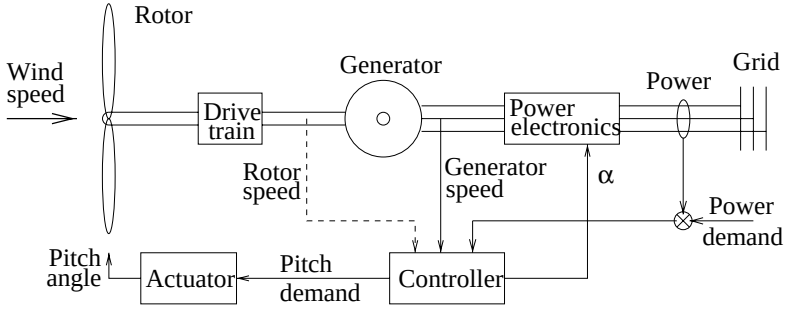

Figure 6.4. A variable speed wind turbine

There are two fundamental ways of controlling the operation of a wind turbine, either actively or passively.

Passive pitch control relies on the inherent mechanical properties of the turbine blades causing stalling at rated wind speed. The majority of wind turbines in Denmark are stall controlled. A stall controlled turbine is designed with a fixed pitch blade angle. Near rated wind speed the rotor blades are designed to stall to smooth power through their insensitivity to fluctuations in wind speed, but the wind turbine structures experience

greater mean thrust loads [Leithead et al., 1991]. A stalling rotor is self-regulating providing power limiting and good power quality without the need for a control system. The good power quality refers to the extent of rapid fluctuations in the generated power is small. The passive control will not be discussed in this report.

Active control can be achieved by controlling the blade angle (pitch), nacelle rotation (yaw), and turbine rotational speed.

The torque induced on the rotor by the wind depends on the pitch angle of the blades. Hence, the torque may be reduced by feathering and vice versa. Above rated wind speed, the pitch of the blades are continuously set to the angle of pitch at which rated power is generated. The adjustment of pitch angle is usually made in response to power measurement [Leithead et al., 1992]. Pitch control has so far been the dominating method for power control. Improvement in power quality and alleviation of fatigue damage can be achieved by continuously monitoring the wind turbine and altering the pitch angle of the blades accordingly. A control system with the ability to vary pitch in an active feedback control is required.

The variable speed concept for wind turbines is still relatively rarely implemented. In variable speed wind turbines the generator does not directly couple to the grid. Therefore the rotor is permitted to rotate at any speed by the power generation unit which might be a generator and a frequency converter combination. For variable speed operation there may be more than one control action. In this project the approach is to combine variable speed operation with a variable pitch capability in full load. The strategy is then to regulate the load on the wind turbine by adjusting the rotor speed and pitch angle to maintain power generation at rated values. In partial

load the pitch angle is fixed at optimal value, a controller without pitch action through the generator reaction torque will be implemented. The control strategy is then to maximize energy capture by adjusting the rotor speed to follow the wind speed variations.

On both constant and variable speed turbines electrical power is the most readily available measurement signal. In addition to electrical power measurement, the generator shaft speed will be measured for a variable speed turbine. In the case of partial load, a measurement of rotor speed may also be required for a variable speed plant.

6.3 Outline of the second part of the thesis

An outline of the second part of the thesis is given in this section.

Chapter 6 is an introduction to the wind turbine systems and control problems. The motivations behind the design decisions are explained.

Chapter 7 describes all the significant dynamic features encountered on a practical wind turbine, with emphasis on the use of such a model in the validation and investigation of control systems. The result is a general nonlinear mathematical model which is used for simulations. The model is validated by using the data from an existing WD34 wind turbine.

Chapter 8 contains simple linear models of wind turbines which will be used for design of controllers. Similar models for the constant speed turbine and

variable speed turbine are derived in this chapter. The design models are made as simple as possible while keeping all significant dynamics.

Chapter 9 deals with estimation of the wind speed in connection to a wind turbine. The Newton-Raphson method, Kalman filter method and extended Kalman filter method are investigated for both partial load and full load. The experimental data are used to test the estimation methods. The problem that might be caused by the methods is discussed at the end of chapter.

Chapter 10 is concerned with control of the wind turbines in full load. Both the constant speed and variable speed turbine are investigated. The active pitch control is the most popular control method in practical applications. The emphasis of the chapter is given to the LQG control method for a solely pitch controlled wind turbine, as well as a combined variable speed and pitch control wind turbine.

Chapter 11 is focussed on control of a variable speed wind turbine in partial load. Below rated wind speed the control strategies are to maximize the energy capture from the wind and minimize the transient loads. The pitch angle of rotor blades is fixed at optimal value, control without pitch action through the generator reaction torque will be implemented. The LQG control and tracking control methods are investigated and the trade-off has to be made between different control objectives. Some implementation considerations are given at the end of this chapter.

Chapter 12 gives a summary and conclusions of the second part of the thesis.

Chapter 7

Simulation Model of the Wind Turbine

7.1 Introduction

This chapter provides insight into the modelling of an entire wind turbine system. The motivation is to give the information about dynamics of a wind turbine, with emphasis upon the use of such a model in the investigation of control systems.

The major components of a turbine are the tower, rotor (the blades and hub), drive train and power generation unit. The driving train consists of the low-speed shaft, gearbox and high-speed shaft. The simulation model

should include contribution from each component, since each feature contributes to the final overall dynamic performance of the complete system. The model for each part may be simplified without significant reduction in the accuracy of representation. Furthermore, it is necessary to derive a wind model for the simulation of the wind turbine performance.

The first stage in the modelling process is to model the individual system component. Figure 7.1 illustrates the basic model structure of a wind turbine and the interactions between the different dynamic components in the model. Both constant speed and variable speed wind turbines are modelled. In the figure the generator reaction torque is only used for a variable speed wind turbine.

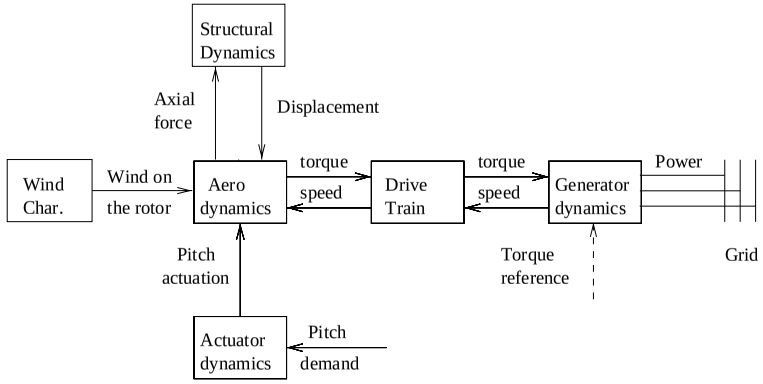

Figure 7.1. Basic structure of a wind turbine model

The dynamic behaviour of the model will be verified to demonstrate that the simulation approximates the reality. The wind turbine modelled is WD34. Since we can not get access to the experimental data for the variable speed wind turbine, only the constant speed turbine model will be validated.

7.2 Wind model

In order to understand the dynamic performance of a wind turbine, it is necessary to have the knowledge of wind characteristics.

The winds, in the macro-meteorological sense, are movements of air masses in the atmosphere. These large movements are generated primarily by differences in the temperature within the atmosphere which are due to differential solar heating [Freris, 1990]. The lower region of the atmosphere, is of interest for wind turbine operation. Movement of air in this region is influenced by frictional force, large obstructions on the surface of the earth and temperature gradient in the vertical direction.

The wind speed is described as a slowly varying average wind speed superimposed by a rapidly varying turbulent wind speed.

$$v_{wind} = v_m + \Delta v_{wind} \tag{7.1}$$

The average wind speed v_m is influenced by the weather and geographic conditions. It can be estimated by averaging the filtered point wind measurements through a low-pass filter, with a period of 10 minutes. The turbulent wind speed Δv_{wind} will be modelled in the frequency range of $8.3 \cdot 10^{-4} - 10$ Hz, see [Knudsen, 1989].

The direction of the wind will not be discussed, since the wind turbine considered in this report is designed for upwind operation. It is then assumed that the rotor of the turbine is aligned to the wind direction by active yawing control.

For a wind turbine, it is usual to measure the wind speed by an anemometer situated on the top of nacelle. The measured wind speed is called point wind speed. Although there is no such thing as the wind speed experienced by the wind turbine, since the rotor is subject to a spatially distributed wind field which varies in time, the turbine may be considered to experience an effective wind speed which, in some sense, is an average over the rotor disc. The high frequencies of the effective wind speed experienced by the rotor will be damped comparing with the frequencies of the point wind speed.

This section will define a model describing the behaviour of the effective turbulent wind speed on the rotor. First of all, a power spectrum for the point wind speed is chosen and simplified. Secondly, by filtering the power spectrum of the point wind speed with a filter function, the power spectral density function for the effective wind speed experienced by the rotor can be found. Finally, a linear model which has approximately the same spectrum will be obtained. In this report the turbulent wind speed is assumed constant over the whole rotor.

7.2.1 The point wind speed

The model of the wind speed at measurement point is

$$v_{wind} = v_m + v_p \tag{7.2}$$

where v_p is the turbulent point wind speed.

The turbulent point wind speed is described by power spectrum given by [Højstrup, 1982]

$$\frac{f \cdot S_p(f)}{v_{*0}^2} = \frac{0.5 f_i}{1 + 2.2 f_i^{\frac{5}{3}}} \cdot (\frac{h_i}{-L})^{\frac{2}{3}} + \frac{105 f_{ru}}{(1 + 33 f_{ru})^{\frac{5}{3}}} \cdot \frac{(1 - \frac{h}{h_i})^2}{(1 + 15\frac{h}{h_i})^{\frac{2}{3}}} \tag{7.3}$$

f_i and f_{ru} are defined as

$$f_i = \frac{f \cdot h_i}{v_m} \tag{7.4}$$

$$f_{ru} = \frac{\frac{f \cdot h}{v_m}}{1 + 15\frac{h}{h_i}} \tag{7.5}$$

List of Symbol

v_m	the mean wind speed [m/s]
v_{*0}	friction velocity [m/s]
h	measuring height [m]
h_i	height of convective boundary layer [m] $h_i = 1000\,\mathrm{m}$
L	Monon-Obukhov length [m]
f	frequency $f = \frac{\omega}{2\pi}$ [Hz]

The above spectrum consists of two parts. The first part corresponds to the thermal turbulence (low frequency) and the second part corresponds to the mechanical turbulence (high frequency).

If the vertical temperature gradient is about $-10^oC/km$, the air mass will neither move up nor move down, this is called neutral condition. If the vertical temperature gradient is less than $-10^oC/km$, the air mass will move up, it is an unstable atmosphere; Otherwise it is a stable atmosphere if the gradient is more than $-10^oC/km$. The unstable, neutral and stable condition happen about 6%, 60% and 34% of time [Petersen et al., 1980]. Therefore we choose the neutral condition for the model. For a neutral

condition the first part of the model can be neglected, which means that the high frequency part is dominant. The above function can then be reduced to

$$\frac{f \cdot S_p(f)}{v_{*0}^2} = \frac{105 f_{ru}}{(1+33 f_{ru})^{\frac{5}{3}}} \cdot \frac{(1-\frac{h}{h_i})^2}{(1+15\frac{h}{h_i})^{\frac{2}{3}}} \tag{7.6}$$

In [Højstrup, 1982], the model variance is obtained by integration of (7.6) over positive frequencies only

$$\sigma_p^2 = \int_0^\infty S_p(f)df = \frac{105}{22} \frac{(1-\frac{h}{h_i})^2}{(1+15\frac{h}{h_i})^{\frac{2}{3}}} v_{*0}^2 \tag{7.7}$$

According to [Larsen et al., 1992], for a neutral condition

$$\frac{v_{*0}}{v_m} = \frac{\kappa}{ln(h/r_l)} \tag{7.8}$$

where κ is von Kármán constant (≈ 0.4), h is the measuring height, r_l is the roughness length. Table 7.1 shows roughness class and corresponding roughness length for different landscapes [Petersen et al., 1980]. A reasonable choice for the roughness length is between class 1 and 2.

Roughness class	Type of landscape	Roughness length r_l
0	Sea	1 mm
1	Open country	1 cm
2	Farmland	5 cm
3	Small town	30 cm

Table 7.1. Roughness class and roughness length for different landscape

By choosing the roughness length properly, it is then possible to achieve the spectral density function and the variance of the point wind speed by inserting the friction velocity v_{*0} into (7.6) and (7.7). It can be easily found

that the spectrum and variance of the point wind depend on the average wind speed.

7.2.2 The wind experienced by the rotor

It is impossible to measure the wind speed on the rotor, since the wind speed varies over the disc swept by the rotor. The wind experienced by the rotor can be considered as an average of the spatial turbulence, which will be modelled by dynamically filtering the point wind

$$S_{ef}(f) = S_p(f)F(f) \tag{7.9}$$

The power spectrum of the filter is given by [Knudsen, 1983]

$$F(f) = \frac{1}{(1+\frac{8\sqrt{\pi}}{3}\frac{R}{v_m}f)(1+4\sqrt{\pi}\frac{R}{v_m}f)} \tag{7.10}$$

where R is rotor radius.

7.2.3 The approximated effective wind speed

For the purpose of simulation, a linear model for the effective wind speed has to be derived.

The power spectrum of the effective wind speed $S_{ef}(f)$ can be approximated by the power spectral density function

$$\begin{aligned} \Phi_v(\omega) &= \frac{k^2}{(1+p_1^2\omega^2)(1+p_2^2\omega^2)}\Phi_e(\omega) \\ &= H(j\omega)\Phi_e(\omega)H(-j\omega) \end{aligned} \tag{7.11}$$

where $\Phi_e(\omega) = 1/(2\pi)$ is the spectral density function for zero mean white noise with unity variance. This leads to that the effective wind speed will be approximated by an asymptotically stable second order system with transfer function $H(S)$ [Østergaard, 1994]

$$H(S) = \frac{k}{(1+p_1S)(1+p_2S)} \tag{7.12}$$

the system is driven by white noise.

It should be pointed out that the spectra presented in section 7.2.1 and 7.2.2 integrate to only half the variance when the integration is carried out over positive frequencies only. This means that when the spectrum of the effective wind speed on the rotor is compared to the spectrum of the model $\Phi_v(\omega)$, it has been reduced by a factor 2.

The system DC-gain k is found by

$$\lim_{\omega\to 0}\Phi_v(\omega) = \lim_{\omega\to 0}\frac{1}{4\pi}S_{ef}(\frac{\omega}{2\pi})$$

and parameters p_1 and p_2 can be obtained by minimizing the performance function

$$J = \int_{\omega_1}^{\omega_2}[log(\Phi_v(\omega)) - log(\frac{1}{4\pi}S_{ef}(\frac{\omega}{2\pi}))]^2 d\omega$$

Frequency interval is chosen from $\omega_1 = 10^{-3}$ rad/sec to $\omega_2 = 10$ rad/sec, and optimization can be made by using MATLAB function FMINS.

The spectrum of the effective wind speed on the rotor and the spectrum (7.11) at average wind speed of $16m/s$ are plotted in Figure 7.2. The parameters achieved from optimization are plotted for different average wind speeds in Figure 7.3.

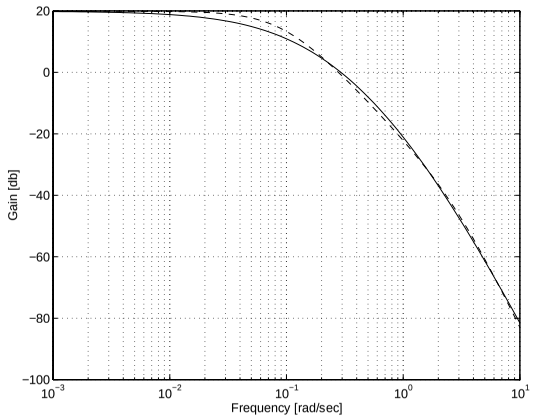

Figure 7.2. Spectrum of the effective wind speed $\frac{1}{4\pi}S_{ef}(\frac{\omega}{2\pi})$ (-) and the approximated effective wind speed $\Phi_v(\omega)$ (- -) at average wind speed of 16 m/s

The derived wind model is

$$\begin{aligned} v_{wind} &= v_m + \Delta v_{wind} \\ \Delta\ddot{v}_{wind} &= -\frac{p_1+p_2}{p_1p_2}\Delta\dot{v}_{wind} - \frac{1}{p_1p_2}\Delta v_{wind} + \frac{k}{p_1p_2}e \end{aligned} \tag{7.13}$$

The variance of effective turbulent wind speed is given by

$$\begin{aligned} \sigma_v^2 &= \int_{-\infty}^{\infty}\Phi_v(\omega)d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}H(j\omega)H(-j\omega)d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{k^2}{(1+p_1^2\omega^2)(1+p_2^2\omega^2)}d\omega \\ &= \frac{1}{2}\frac{k^2}{p_1+p_2} \end{aligned} \tag{7.14}$$

which has been plotted in Figure 7.4.

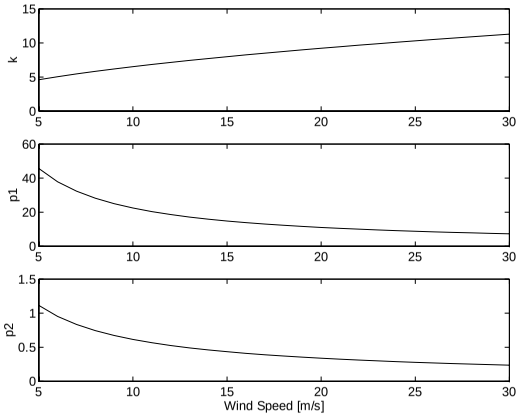

Figure 7.3. Parameters k, p_1 and p_2 at different average wind speed

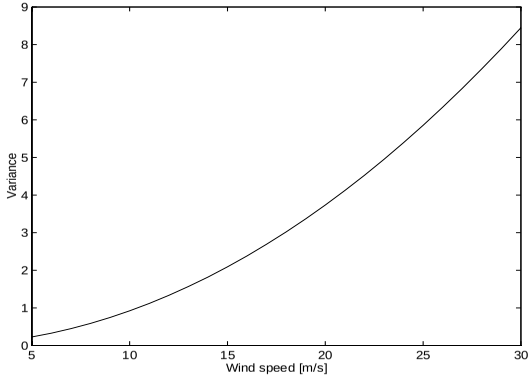

Figure 7.4. Variances of the effective wind speed at different average wind speeds

The effective wind speeds experienced by the rotor at average wind speeds of 7 m/s and 16 m/s are plotted in Figure 7.5. The wind sequences are obtained from the simulations.

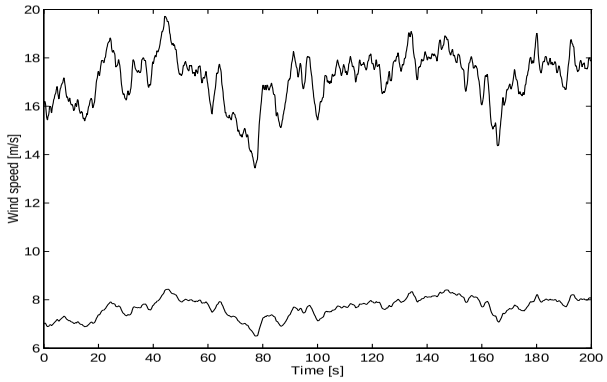

Figure 7.5. Simulated wind at the average wind speed of 7 m/s (dashed) and 16 m/s (solid)

However, the effective wind speed experienced by the rotor has to be modified by including the contribution from the tower motion, which will be explained in the following sections.

7.3 Aerodynamics

A wind turbine is a device for converting kinetic energy from the wind to electrical energy. The rotor blades of the turbine sweep through a complex three dimensional wind field which varies both in time and over the rotor disc.

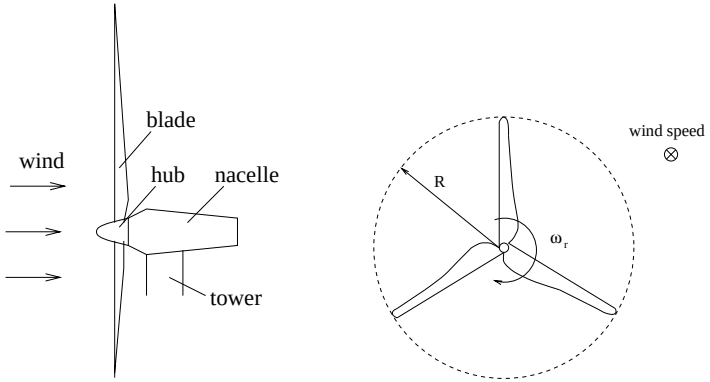

Figure 7.6. Side view and front view of rotor

Figure 7.6 is the side view and front view of rotor disc. The wind turbine is operated facing against the wind by active yawing system.

All wind turbines, whatever their design, extract the pressure energy in the following way. The turbine first causes the approaching wind to slow down gradually, which results in a rise in the static pressure. Across the turbine swept surface there is a drop in static pressure, such that, the air is below the atmospheric pressure level. As the air proceeds downstream, the pressure climbs back to the atmospheric value, causing a further slowing down of the wind. Thus, between the far upstream and far wake conditions, there is no change in static pressure, but a reduction in ordered kinetic energy

[Freris, 1990]. The principle of energy extraction process is illustrated in Figure 7.7.

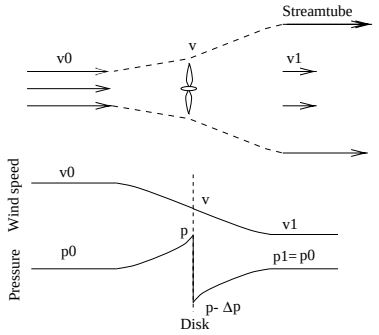

Figure 7.7. The principle of energy extraction process

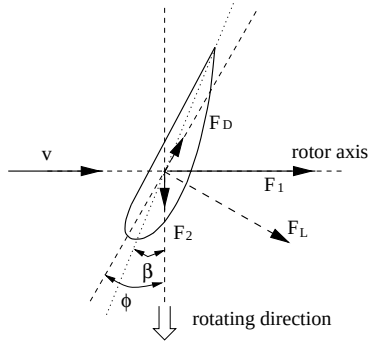

Figure 7.8. Aerodynamic forces at a rotor blade section

How much the extracted energy is converted into usable energy depends upon the particular turbine design. The change of the pressure induced an aerodynamic force which is usually separated into a lift F_L and a drag F_D component. The components F_L and F_D are in turn transformed into a pair of axial and tangential forces F_1 and F_2. Only the tangential component F_2 produces the driving torque around the rotor shaft. The axial force F_1 has

no driving effect but puts stress on the turbine, which leads to a flap-wise bending load on the rotor blades. Furthermore, the axial force is transferred to the hub leading to a thrust on the nacelle and a bending of the tower.

The aerodynamic part is defined as a transfer from the wind speed on the rotor, the pitch angle and angular velocity of the rotor to the rotor aerodynamic torque and axial force. The aerodynamics is modelled in a simple manner, where the rotor surface of wind turbine is assumed experiencing a uniform wind field developed in the previous section.

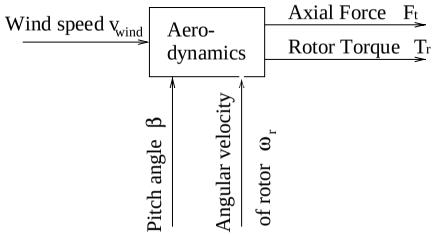

Figure 7.9. Aerodynamics

7.3.1 Aerodynamic power and torque

The rotor aerodynamic power can be calculated by [Andersen et al., 1980]

$$P_r = \frac{1}{2}\pi\rho R^2 v^3 C_p(\lambda,\beta) \tag{7.15}$$

where $\rho = 1.225\ kgm^3$ is air density, R is the rotor radius and v is the wind speed experienced by the rotor. The power coefficient $C_p(\lambda,\beta)$ is a turbine specific function defining the ability of the turbine to convert the kinetic energy of the wind to mechanical energy. $C_p(\lambda,\beta)$ is a nonlinear function

of the pitch angle β and tip speed ratio λ which is defined by

$$\lambda = \frac{v}{\omega_r R} \tag{7.16}$$

where ω_r is the angular velocity of rotor and $\omega_r \cdot R = v_{tip}$ is tip speed.

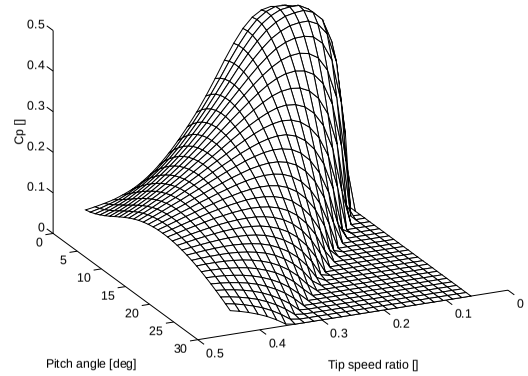

Figure 7.10. C_p surface for WD34 (Negative values are replaced by zeros)

C_p surface for WD34 calculated by Risø National Laboratory is plotted in Figure 7.10, where negative values are replaced by zeros. The C_p surface has a unique maximum value which is given by an optimal pitch angle and an optimal tip speed ratio. For the WD34 wind turbine, the maximum value of C_p is 0.4440, the optimal pitch angle is 0.5^o and the optimal tip speed ratio is 0.1357.

The rotor aerodynamic torque can be easily achieved by dividing the rotor power by the angular velocity of the rotor shaft

$$T_w = \frac{1}{2\omega_r}\pi\rho R^2 v^3 C_p(\lambda,\beta) \tag{7.17}$$

7.3.2 3p effect

As well as the disc averaged torque T_w, the rotor experiences various low speed torque harmonic associated with the rotational frequency (nP) of the blades. For a three bladed wind turbine, the most important disturbances are 3P and 1P peaks with the 3P peak much more pronounced than the 1P peak.

The spectral peaks are caused by the wind speed turbulence, the wind gradient and shadow of the tower. The wind speed depends on the altitude above ground, which causes the wind speed to be faster at top of the disc swept by the rotor than at the lowest part. This is the gradient effect. The tower shadow phenomenon is that when one of the blades sweeps in front of the supporting tower, the torque induced on the blade will be reduced. [Leithead et al., 1992] indicates that the 1P peak is predominantly deterministic and 3P peak is predominantly stochastic.

The 1P peak modelled by a simple sinusoid with amplitude A is given by [Wilkie et al., 1990]

$$T_{nP} = A\sin(\omega_r t) \tag{7.18}$$

The 1P-effect is not included in the report.

A suitable model of the 3P peak is given by a lightly damped second order function with white noise as input [Flensborg and Sørensen, 1995]

$$\frac{T_{3P}(s)}{e(s)} = \frac{k_{3P}T_w(3\omega_r)^2}{s^2 + 2\zeta(3\omega_r)s + (3\omega_r)^2} \tag{7.19}$$

The spectrum of the model has a peak at $3\omega_r$. The amplitude k_{3P} and damping ratio ζ are determined empirically by comparing the model output with the data sequences obtained from the existing WD34 wind turbine.

The total rotor torque experienced by the rotor is then modified by

$$T_r = T_w + T_{3P} \tag{7.20}$$

7.3.3 Axial force

The force, induced by the wind on the rotor, causes the bending movement of the tower in axial direction. The movement is depicted in Figure 7.11.

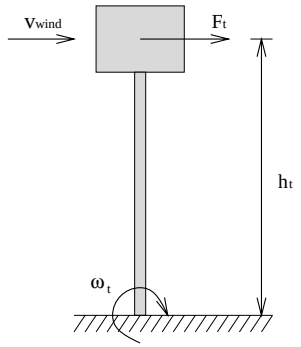

Figure 7.11. The bending movement of the tower

ω_t in the figure is the angular velocity of the bending movement and h_t is the height of the tower. The driving force F_t, assumed acting on the centre of the rotor, is given by [Andersen et al., 1980]

$$F_t = \frac{1}{2}\pi\rho R^2 v^2 C_t(\lambda,\beta) \tag{7.21}$$

where v is the effective wind speed experienced by the rotor, which differs from the wind in front of the rotor due to the tower motion. Therefore v has to be modified by considering the contribution of the tower movement.

$$v = v_{wind} - \omega_t h_t \tag{7.22}$$

The contribution of the tower movement is significant for high wind speeds.

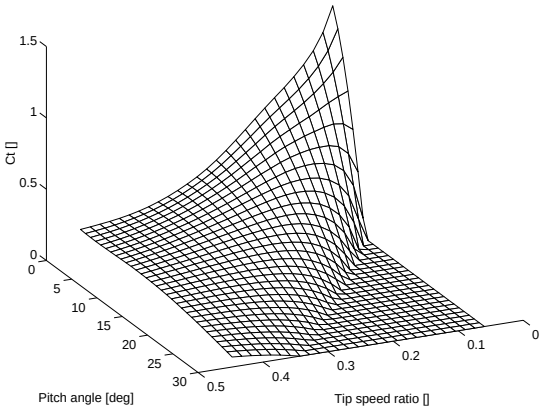

Figure 7.12. C_t surface for WD34 (Negative values are replaced by zeros)

The force coefficient $C_t(\lambda, \beta)$ is a strongly non-linear function of the pitch angle β and the tip speed ratio λ. The C_t surface for WD34 is plotted in Figure 7.12, where negative values are replaced by zeros.

7.4 Structural dynamics

The axial force on the rotor causes the tower to bend. This may cause the fatigue damage of turbines. The tower bending dynamics have an influence on the stability of control loops and have thus to be taken into account in the design of controller, i.e., the controller will be designed so as not to excite the tower bending.

The dynamic behaviour of the tower is modelled in a simple way. The one degree of freedom modelled in the tower is an axial deflection of the nacelle. The tower can be considered as a rigid body with a spring and a damper, as illustrated in Figure 7.13

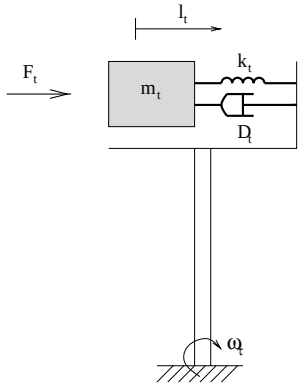

Figure 7.13. Structural dynamics

The tower model is given by [Bongers et al., 1990]

$$m_t\ddot{l}_t + D_t\dot{l}_t + k_t l_t = F_t \tag{7.23}$$

where l_t is tower displacement. We also have the relation $l_t = \theta_t h_t$ and $\dot{\theta}_t = \omega_t$. θ_t is the angle of tower bending movement.

The structure dynamics can then be rewritten as

$$m_t h_t\ddot{\theta}_t + D_t h_t\dot{\theta}_t + k_t h_t\theta_t = F_t \tag{7.24}$$

The resonance frequency of the tower motion is 6.9 rad/s for the WD34 wind turbine.

The list of symbol is given below

List of Symbol

l_t	nacelle displacement	[m]
m_t	mass of nacelle and rotor	[kg]
k_t	tower stiffness	[kg/s^2]
D_t	tower damping	[kg/s]
F_t	axial force	[N]
θ_t	angle of tower movement	[deg]
ω_t	angular velocity of tower movement	[rad/sec]
h_t	height of tower	[m]

7.5 Drive train

The drive train converts the input aerodynamic torque on the rotor into the torque on the low-speed shaft which is scaled down through the gearbox and then induce a torque on the high speed shaft. The drive train transmission system has been illustrated in Figure 7.14.

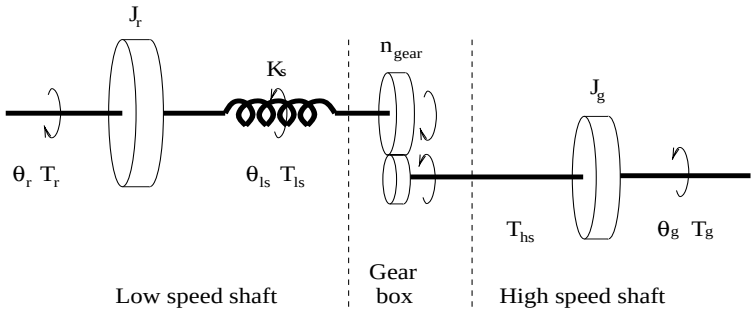

Figure 7.14. Drive train dynamics

J_r in the figure represents the inertia of the rotor, low speed shaft and gearbox. The inertia of high speed shaft and generator is represented by J_g.

The stiffness of blades, hub, main shaft and gearbox has been transformed as a total stiffness on the low speed shaft K_s.

The transmission from rotor torque T_r to generator mechanical torque T_g is described by the following equations.

The dynamic equation for the rotor and low speed shaft is

$$J_r\ddot{\theta}_r = T_r - T_{ls} \tag{7.25}$$

and

$$T_{ls} = K_s(\theta_r - \theta_{ls}) \tag{7.26}$$

The torque and shaft rotation are transmitted through the gearbox to induce the torque of high speed shaft

$$T_{hs} = \eta_{gear}\frac{T_{ls}}{n_{gear}} \tag{7.27}$$

where η_{gear} is the efficiency of the gearbox. The shaft angle or the generator shaft angle will be

$$\theta_g = n_{gear}\theta_{ls} \tag{7.28}$$

The dynamic equation for the high speed shaft is

$$J_g\ddot{\theta}_g = T_{hs} - T_g \tag{7.29}$$

The model equations of drive train can be reformulated in terms of the angular velocities

$$J_r\dot{\omega}_r = T_r - T_{ls} \tag{7.30}$$

$$J_g\dot{\omega}_g = \eta_{gear}\frac{T_{ls}}{n_{gear}} - T_g \tag{7.31}$$

and

$$T_{ls} = K_s \theta_\epsilon \tag{7.32}$$

and

$$\dot{\theta}_\epsilon = \omega_r - \frac{\omega_g}{n_{gear}} \tag{7.33}$$

The notations are given by

θ_r	rotor shaft angle [rad]
θ_g	high speed and generator shaft angle [rad]
θ_{ls}	low speed shaft angle [rad]
ω_r	angular velocity of rotor [rad/sec]
ω_g	angular velocity of high speed shaft and generator [rad/sec]
J_r	rotor and low speed shaft inertia [kgm^2]
J_g	high speed shaft and generator inertia [kgm^2]
T_r	rotor torque [Nm]
T_{ls}	torque on the low speed shaft [Nm]
T_{hs}	torque on the high speed shaft [Nm]
T_g	generator mechanical torque [Nm]
K_s	spring coefficient of the drive train [Nm/rad]
n_{gear}	gearbox ratio [-]
η_{gear}	efficiency coefficient of gearbox [-]

7.6 Generator model

There are two possible types of power generation units. If the generator is connected directly to the grid, the angular velocity of the turbine rotor is locked to the grid frequency. A machine of this type is a constant speed wind turbine. If the generator is connected to the grid via power conversion equipment, the angular velocity of rotor will be independent of the grid frequency. A machine of this type is a variable speed wind turbine.

The WD34 wind turbine has a 400 KW asynchronous generator. The rotor in the generator has three pole pairs. The three phase stator winding is connected directly to the grid with synchronous frequency ω_0. The generator converts the mechanical power from high speed shaft into electrical power. The difference between the high speed shaft torque and the generator reaction torque T_g induces a high speed shaft velocity ω_g by driving the generator mechanical load. The high speed shaft velocity is converted into electrical power through the action of the electrical load and the generator reaction torque is fed back to the drive train to balance the driving loads [Wilkie et al., 1990].

7.6.1 Constant speed power generation unit

The main components of an asynchronous machine are a revolving rotor and a fixed stator. An asynchronous machine operation is based on the principle of the electro-magnetic interaction between the rotating magnetic field created by the three phase current supplied to the stator winding and current induced in the rotor winding. The rotor can run either in the same direction as the field, or in the opposite direction with. The rotor speed is ω_g and the rotating speed of magnetic field is $\frac{\omega_0}{n_p}$. The difference between these two speed is defined by the slip

$$s = \frac{\frac{\omega_0}{n_p} - \omega_g}{\frac{\omega_0}{n_p}} \tag{7.34}$$

where n_p is is the number of pole pairs.

According to [Kostenko and Piotrovsky, 1969], an asynchronous machine can act as a generator, a motor or an electrical brake. If the speed is negative relative to the magnetic field, i.e., $\omega_g < 0$ and $S > 1$, the machine will use the power from field to stop the rotor and therefore acts as an electrical brake. If the rotor speed is positive and less than the field speed, i.e., $0 < \omega_g < \frac{\omega_0}{n_p}$ and $0 < S < 1$, the machine will act as a motor and try to change the rotor speed to a constant value. If the rotor speed is higher than the field speed, i.e., $\omega_g > \frac{\omega_0}{n_p}$ and $S < 0$, the machine will be a generator and produce power.

When the asynchronous machine operates as a generator, a mechanical torque T_g is delivered to the generator shaft. Due to different losses only part of this torque, the electromagnetic torque T_{em} is obtained and converted into electric energy. The electromagnetic torque as a function of the slip is given by

$$T_{em} = \frac{m_1 U_1^2 \frac{R_2}{S}}{\frac{\omega_0}{n_p}[(R_1 + \frac{R_2}{S})^2 + X^2]} \tag{7.35}$$

where R_1 and R_2 are the resistances of the stator winding and rotor winding, X is the total inductive reactance of the stator winding and rotor winding. Suppose the stator of the induction machine is connected to a circuit with a given voltage U_1, and the phase number of the stator is m_1. The torque-slip curve with $U_1 = const.$ is drawn in Figure 7.15.

When the machine is operating close to nominal values, the electromagnetic torque can be modelled by a constant torque-speed curve slope, i.e., the curve in Figure 7.15 can be approximated to the dashed line.

As a generator, the delivered mechanical power from high speed shaft to the generator is $T_g \omega_g$. Due to the different losses, only part of this power is

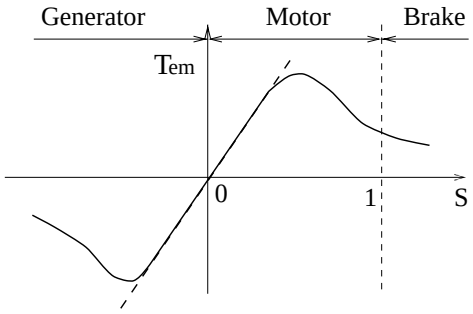

Figure 7.15. Torque-slip curve

transformed to electrical power. If the efficiency of the generator is η, the produced electrical power will be

$$P_e = \eta \, T_g \omega_g \tag{7.36}$$

If the losses in conversion of the electromagnetic power P_{em} into electrical power are neglected, the produced electrical power will be

$$P_e = T_{em} \omega_g \tag{7.37}$$

which means that $T_g = T_{em}/\eta$, and T_g can thus be modelled by

$$T_g \approx D_g(\omega_g - \frac{\omega_0}{n_p}) \tag{7.38}$$

with

$$D_g = \frac{P_{e,0}}{\eta \omega_{g,0}(\omega_{g,0} - \frac{\omega_0}{n_p})} \tag{7.39}$$

where $P_{e,0}$ is rated value of produced electrical power and $\omega_{g,0}$ is nominal generator speed. Using this model, the asynchronous generator acts like a viscous damper. For an asynchronous generator, parameter D_g is normally large, which means that the generator speed has a very stiff dynamic connection to the synchronous speed.

The value of D_g gives a very stiff dynamics, it is vital to the dynamics of the system. A compensation method given by [Schmidtbauer, 1994] is to modify D_g with use of feedback. D_g will be decreased to $D_g/(1 + K_c/n_p)$, if we use proportional feedback K_c of the generator torque. The damping of the system will be increased, if the feedback gain K_c is increased.

However, if the generator is equipped with a power conversion equipment, the feedback of the rotor velocity or position will be eliminated. This means that the converter makes the generator torque independent of the system dynamics since we get $D_g = 0$.

7.6.2 Variable speed power generator unit

By connecting a frequency converter between the generator and the grid, the coupling between the rotational speed and frequency of the grid can be eliminated, i.e., the generator speed will be independent of the grid frequency. By control of the firing angle of the frequency converter it is possible to control the electrical torque in the generator. The converter allows the turbine to be run at variable speed and makes the torque control in the generator possible and thereby a reduction of the stress on the drive train and gearbox. Figure 7.16 shows the configuration of a frequency converter.

The frequency converter is used to transform the constant frequency and constant voltage of the grid to variable frequency and voltage on the generator side. The main components are an AC/DC converter, a DC-link and a DC/AC converter. When power is flowing from the generator, the AC/DC converter acts like a rectifier, and the DC/AC converter acts like an inverter. The DC-link can be used to attenuate voltage fluctuations. More

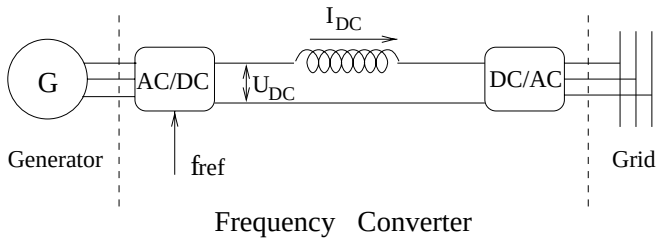

Figure 7.16. The main components of the frequency converter

details of a frequency converter are given by [Blåbjeg and Petersen, 1994] and [Tsiolis, 1994].

The fundamental dynamics of the frequency converter are very complex and nonlinear, but considerably faster than the fundamental drive train dynamics and therefore can be neglected in the modelling. This means that the generator torque will be equal to its reference value

$$T_g = T_{g,ref} \tag{7.40}$$

7.7 Pitch actuator

The pitch actuator consists of a mechanical and hydraulic system which is used to turn the blades of the wind turbine along their longitudinal axis. By varying the pitch angle, the aerodynamic torque input to the rotor is altered and hence the output power.

Because the inertia of the blades is large and the actuator should not consume a great deal of power, the actuator has limited capabilities. Its dynamics are non-linear with saturation limits on both pitch angle and pitch

rate. The actuator dynamics are depicted in Figure 7.17. When the pitch angle and pitch rate are less than the saturation limits, the pitch dynamics exhibits linear behaviour.

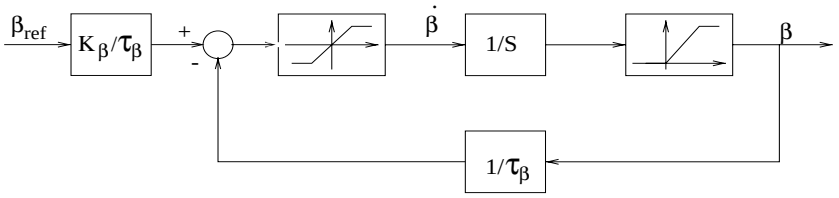

Figure 7.17. Pitch actuator

The actuator model describes the dynamic behaviour between the pitch demand from the pitch controller to the actuation of this demand. The actuator can be modelled as a first order system

$$G(S) = \frac{\beta(S)}{\beta_{ref}(S)} = \frac{K_\beta}{1 + \tau_\beta S} \tag{7.41}$$

where β_{ref} is reference control input and β is actuator output (pitch angle). The model can be rewritten as

$$\dot{\beta} = -\frac{1}{\tau_\beta}\beta + \frac{K_\beta}{\tau_\beta}\beta_{ref} \tag{7.42}$$

According to data from the WD34 wind turbine, the saturation level of the pitch angle is $-2^0 - 87^0$, and the saturation level of pitch rate is $\pm 10^0/s$. These limits should not be reached during the normal operation in order to avoid not only the fatigue damage and wear of the pitch actuator, but also the loss of performance.

7.8 An entire model

The most significant dynamics of the wind turbine have been modelled with emphasis on control design. All dynamic components of the model are given in section 7.2 - 7.7. An entire nonlinear simulation model of the wind turbine can then be derived by connecting the individual sub-models. The interconnections between the different dynamic components have been shown in Figure 7.1.

The simulation model is implemented in SIMULINK, which will then be validated using measurements from the WD34 wind turbine in next section.

7.9 Validation of model

The validation will be mainly based on a comparison between output power from simulation model and data obtained from an existing WD34 wind turbine.

The data sequences from the WD34 wind turbine given by Risø National Laboratory are collected in an open loop experiment, in which the reference signal to the pitch system was altered continuously as a square signal, and pitch angle, output electrical power and wind speed are measured. The collected data are obtained in an experiment of 20 minutes with a sample rate of 32 Hz. A part of data is plotted in Figure 7.18, where the pitch angle is a square signal with a period of 40 seconds. The wind speed is measured by an anemometer situated on the top of nacelle.

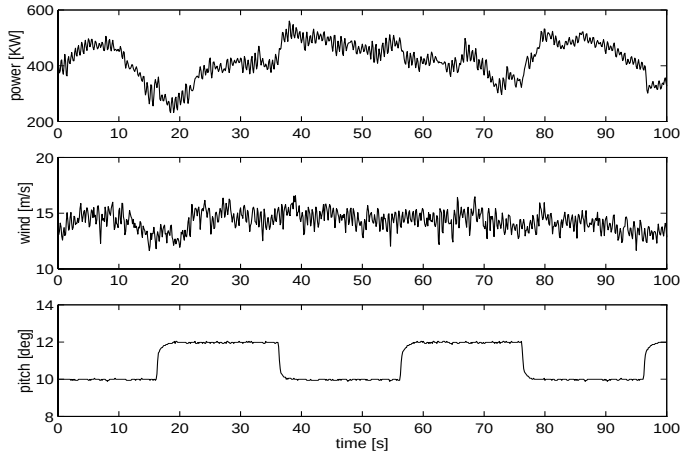

Figure 7.18. Data obtained from the WD34 wind turbine

The operating point of the experiment is given by the average values of the obtained measurements

$$\begin{aligned} v_m &= 13.38\ m/s \\ \beta_0 &= 11^o \\ \omega_{r,0} &= 3.68\ rad/s \end{aligned} \tag{7.43}$$

where $\omega_{r,0}$ is the nominal rotor speed of the turbine.

However, as already explained in section 7.3, the wind speed is slowed down behind the rotor swept surface by the rotor. When the anemometer is situated on the top of nacelle behind the rotor swept surface as it is in the experiment, the wind speed at measurement point is lower than the wind speed at a point in front of the rotor.

The block scheme of the validation of the turbine model is depicted in Figure 7.19. The simulation model developed in the previous sections is

tested by using filtered wind measurements as disturbance input and pitch angle measurements as control input. The output power from the simulation model will be compared with power measurements from the existing WD34 wind turbine.

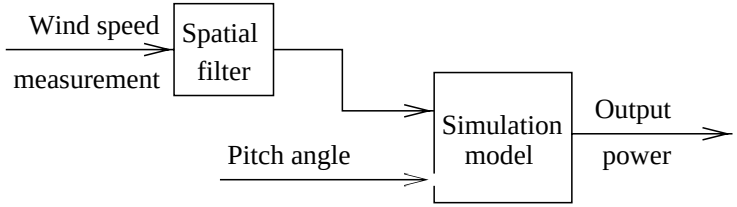

Figure 7.19. Validation of the wind turbine model

The measurements of the wind speed are the point wind speed, which are strongly influenced by the turbine. To correct for wind speed discrepancy and blade shadow, the wind speed experienced by the rotor will be constructed by filtering the point wind speed, which will be used as a disturbance input to the turbine model. The power spectrum of the spatial filter is given by (7.10), which can be approximated by a first order low pass filter

$$H(s) = \frac{1}{1 + 1.4715s} \tag{7.44}$$

The filter transfer function is derived at the operating point (7.43).

7.9.1 Validation of T_{3P} model

First of all, the parameters of the T_{3P} model have to be determined by the data sequences. The 3P-effect depends significantly on the specific wind turbine and site. This makes it necessary to determine the parameters of the model based on validation of the 3P-effect on the power measurements

from the turbine. It can be shown by the power measurements in Figure 7.20 that the 3P variations in the output power are about 10% of the power output.

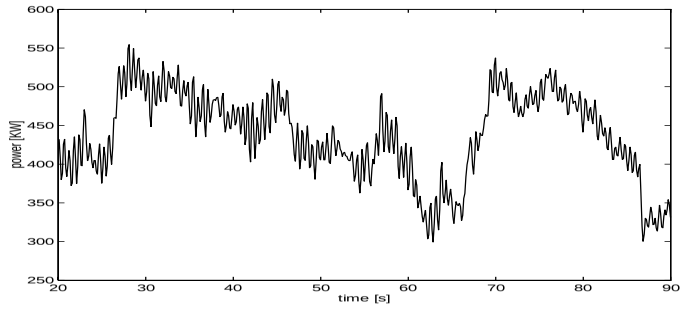

Figure 7.20. Power measurements of the WD34 wind turbine

The amplitude k_{3P} and damping ratio ζ in the model (7.19) will be determined in such a way that the amplitude of variations caused by 3P effect will reach 10% of the output power. The k_{3P} and ζ are mutually dependent. The power spectrum of the simulation output and power measurements obtained from the WD34 turbine are given in Figure 7.21. The good agreement of the resonance peak frequencies has been shown in the figure. The resonance peak of 3P-effect is at 11 rad/s which is equal to $3\omega_{r,0}$.

7.9.2 Validation results

The output electrical powers from the simulation model and the existing WD34 wind turbine are shown in Figure 7.22. The average value of the power measurements from WD34 is $\bar{P}_{WD34} = 358.61$ KW. However, the average value of the simulation output is $\bar{P}_{model} = 261.39$ KW. The difference

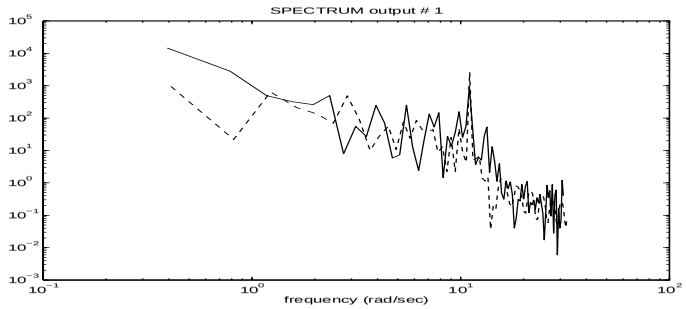

Figure 7.21. Power spectrum for power measurements from WD34 wind turbine (solid) and simulation model (dashed).

is $\Delta P = \bar{P}_{WD34} - \bar{P}_{model} = 97.22KW$. In Figure 7.22 the output power from simulation model is superimposed by the difference ΔP.

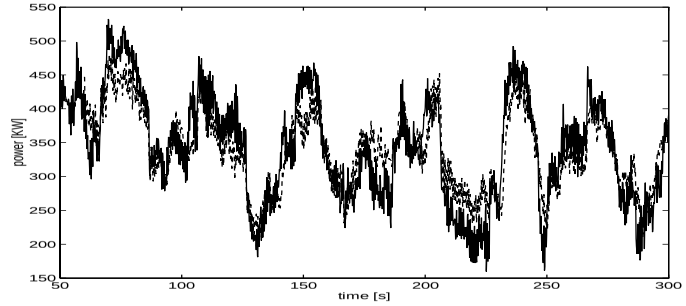

Figure 7.22. Output power from simulation model (solid) and power measurements from the WD34 wind turbine (dashed)

The difference ΔP is caused by the lower wind speed used as the disturbance input to the simulation model. The measurements of the wind speed is lower than the wind speed at a point just in front of rotor because of the influence of the turbine.

Except for the difference of the average values, Figure 7.22 shows a good agreement between the simulation data and measurements.

7.9.3 Another experiment

Another experiment is carried out by the Risø National Laboratory, in which the measurement point of the wind speed is 68 m (2×rotor diameter) in front of the rotor plane at the same height as the hub and pitch angle is fixed at 1^o. The data are collected with a sample rate of 25 Hz, and the average value of the measured wind speed is 6.44 m/s.

At the measurement point, the wind speed measured can be considered as the measurements without a disturbance of the presence of the wind turbine. However, the distance between the measurement point and the rotor disc leads to a time delay between the wind measurements and the wind experienced by the rotor. The time delay varies all the time, which depends also on the wind speed. Furthermore, the correlation between the measured wind and the wind experienced by the rotor is influenced by the distance. As the distance increases, the high frequency components in the wind speed change significantly.

In order to compare the data sequence from the simulation model and the WD34 wind turbine, the time delay is taken as the average value of 11 s at the mean wind speed of 6.44 m/s. The spatial filter used to filter the point wind can be approximated by

$$H(s) = \frac{1}{1 + 3.0613s} \tag{7.45}$$

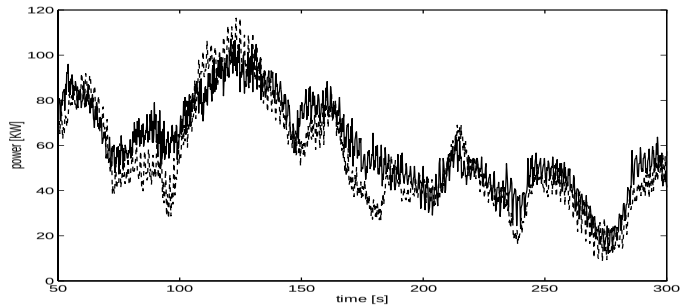

Figure 7.23. Output power from simulation model (dashed) and power measurements from WD34 wind turbine (solid)

The data sequences of output power from the simulation model and the experimental data are given in Figure 7.23, where the time delay is eliminated. Figure 7.23 shows a good agreement between the model output and plant measurements without discrepancy of the mean values.

The experiment shows also significant contents of both 1P and 2P-effect due to the unbalanced rotor. However, the 1P and 2P-effect will not be taken into account in the report.

7.10 Simulation of the uncontrolled wind turbine

The dynamic models of each component of the wind turbine have been derived in previous sections, the simulation can then be carried out for an uncontrolled constant speed wind turbine.

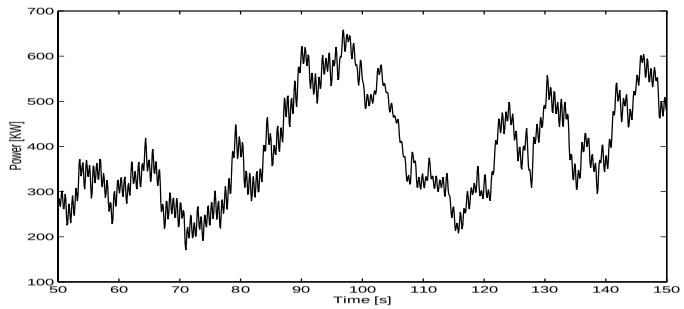

Figure 7.24. Output electrical power [KW] of the uncontrolled wind turbine. The wind speed for the upper plot is at average value of 16 m/s, and the wind speed for the lower plot is at average value of 7 m/s.

For the wind sequence with average value of 16 m/s given in Figure 7.5, the output power corresponding to this wind sequence is shown in Figure 7.24. The key parameters from the open-loop simulation of the constant speed wind turbine are given in Table 7.2. The results will be used for comparing the uncontrolled wind turbine with controlled wind turbine to show the performance of controllers. The results are achieved from the simulation of 200 s.

mean(P_e) [KW]	mean(T_g) [Nm]	mean(ω_g) [rad/s]	max($h_t\omega_t$) [cm]
408.05	4010.19	105.79	2.35
SD(P_e) [KW]	SD(T_g) [Nm]	SD(ω_g) [rad/s]	max($h_t\theta_t$) [cm]
123.25	1199.20	0.32	3.72

Table 7.2. Statistics obtained while simulating the uncontrolled wind turbine. Mean wind speed is 16 m/s.

7.11 Summary

All the significant dynamic features encountered on a practical wind turbine have been included in this chapter with an emphasis on control design. The result is a nonlinear model based on the individual sub-models. Both the constant speed turbine and variable speed wind turbine are modelled. The simulation model is derived by combining the analytic methods and simulation studies. The model of the constant speed turbine is validated by comparing the output power from a WD34 simulation model with data obtained from an existing WD34 wind turbine. The validation shows a good agreement between the simulation model and existing wind turbine.

Chapter 8

Design Model of the Wind Turbine

An appropriate model of system behaviour is the heart of control design. Although a nonlinear model is required for the simulation, a simple linear model is preferred for control design purpose.

An effective control algorithm must reflect the plant dynamic characteristics as well as the anticipated working environment. Hence, the control problem is conveniently divided into two time scales corresponding to slow mean wind speed changes and rapid turbulent wind speed variations. The mean speeds are treated as steady state operating points.

The design model is therefore split into two parts: one part describes the operating point and another part describes the dynamics of the system by a linear state-space model. The operating point determines the output when the system is at an equilibrium. The linear state-space model describes the deviation from the operating point when the system is excited. The control model corresponds to a linearization about the operating point.

To facilitate control system design, the models are required to be as simple as possible while retaining all significant dynamic components which include the aerodynamics, the actuator dynamics, the drive train and generator dynamics. The wind model and 3P-effect will also be included as noise models. The structural dynamics is not included in design models, since it has been shown by simulation that no significant improvement can be achieved by including the tower model. However, by noting position of resonance, the tower motion will be suppressed if it is necessary.

This chapter is organized as follows. Section 8.1 gives a linear state-space model of the turbine, in which linear models of aerodynamic torque, drive train and generator dynamics will be derived first. Then the models are represented for below rated operation and above rated operation separately. Section 8.2 describes the noise models which include the wind model and T_{3P} model. A composite model is given in section 8.3. The sample rates of digital controllers will be discussed in section 8.4, and discrete-time model will be formulated. A summary is given in 8.5.

8.1 Linear state-space models of the plant

8.1.1 Aerodynamic torque

The aerodynamic torque T_r given by (7.17) and (7.20) in section 7.3.1 is a nonlinear function of the wind speed, pitch angle and angular velocity of rotor

$$T_r = f(v,\ \beta,\ \omega_r) \tag{8.1}$$

The nonlinear aerodynamic torque can be linearized around an operating point

$$T_r = T_{w,0} + \Delta T_w + \Delta T_{3P} \tag{8.2}$$

where

$$\Delta T_w = \left.\frac{\partial T_w}{\partial v}\right|_{op} \Delta v + \left.\frac{\partial T_w}{\partial \beta}\right|_{op} \Delta\beta + \left.\frac{\partial T_w}{\partial \omega_r}\right|_{op} \Delta\omega_r \tag{8.3}$$

and

$$\begin{aligned} \Delta v &= v - v_m \\ \Delta\beta &= \beta - \beta_0 \\ \Delta\omega_r &= \omega_r - \omega_{r,0} \end{aligned} \tag{8.4}$$

where v_m, β_0 $\omega_{r,0}$ and $T_{w,0}$ define a steady-state operating point. Δv, $\Delta\beta$, $\Delta\omega_r$, ΔT_w and ΔT_{3P} are only small changes from the steady-state operating point. $\partial T_w/\partial v$, $\partial T_w/\partial\beta$ and $\partial T_w/\partial\omega_r$ are the partial derivatives of aerodynamic torque with respect to the wind speed, pitch angle and rotor speed at the operating point about which the system is linearized. The derivatives can be calculated from C_p curve according to the operating

point

$$\begin{aligned}\frac{\partial T_w}{\partial v} &= \frac{3}{2\omega_r}\pi\rho R^2 v^2 C_p + \frac{1}{2\omega_r^2}\pi\rho R v^3 \frac{\partial C_p}{\partial \lambda} \\ \frac{\partial T_w}{\partial \beta} &= \frac{1}{2\omega_r}\pi\rho R^2 v^3 \frac{\partial C_p}{\partial \beta} \\ \frac{\partial T_w}{\partial \omega_r} &= -\frac{1}{2\omega_r^2}\pi\rho R^2 v^3 C_p - \frac{1}{2\omega_r^3}\pi\rho R v^4 \frac{\partial C_p}{\partial \lambda}\end{aligned} \tag{8.5}$$

$\partial C_p/\partial\lambda$ and $\partial C_p/\partial\beta$ can be numerically calculated from C_p surface.

8.1.2 Drive train and generator

The drive train dynamics can be represented by

$$\begin{aligned}\Delta\dot{\theta}_\epsilon &= \Delta\omega_r - \frac{1}{n_{gear}}\Delta\omega_g \\ \Delta\dot{\omega}_r &= -\frac{K_s}{J_r}\Delta\theta_\epsilon + \frac{1}{J_r}\Delta T_r \\ \Delta\dot{\omega}_g &= \frac{\eta_{gear}K_s}{n_{gear}J_g}\Delta\theta_\epsilon - \frac{1}{J_g}\Delta T_g\end{aligned} \tag{8.6}$$

For a constant speed wind turbine, the generator reaction torque is modelled by $\Delta T_g = D_g\Delta\omega_g$, and it is equal to generator torque reference for a variable speed wind turbine, i.e., $\Delta T_g = \Delta T_{g,ref}$.

8.1.3 Above the rated wind speed

As already stated in Chapter 6, when the wind speed exceeds the rated value, the objectives of the control are to maintain the power at its rated

value and prevent turbine overloading. Both constant speed and variable speed turbine are investigated for this purpose.

Since the structural dynamics is not included in design models, the wind speed experienced by the rotor can then be approximated by $\Delta v = \Delta v_{wind}$. If pitch system dynamics is included in the model and pitch reference is taken as a control input, a state-space model of the constant speed wind turbine will be represented

$$\begin{bmatrix}\Delta\dot{\theta}_\epsilon \\ \Delta\dot{\omega}_r \\ \Delta\dot{\omega}_g \\ \Delta\dot{\beta}\end{bmatrix} = \begin{bmatrix} 0 & 1 & -\frac{1}{n_{gear}} & 0 \\ -\frac{K_s}{J_r} & \frac{1}{J_r}\frac{\partial T_w}{\partial\omega_r} & 0 & \frac{1}{J_r}\frac{\partial T_w}{\partial\beta} \\ \frac{\eta_{gear}K_s}{n_{gear}J_g} & 0 & -\frac{D_g}{J_g} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_\beta}\end{bmatrix}\begin{bmatrix}\Delta\theta_\epsilon \\ \Delta\omega_r \\ \Delta\omega_g \\ \Delta\beta\end{bmatrix}$$

$$+\begin{bmatrix}0 \\ 0 \\ 0 \\ \frac{K_\beta}{\tau_\beta}\end{bmatrix}\Delta\beta_{ref} + \begin{bmatrix}0 \\ \frac{1}{J_r}\frac{\partial T_w}{\partial v} \\ 0 \\ 0\end{bmatrix}\Delta v_{wind} + \begin{bmatrix}0 \\ \frac{1}{J_r} \\ 0 \\ 0\end{bmatrix}\Delta T_{3P} \tag{8.7}$$

For a constant speed wind turbine, the output electrical power can be approximated by $P_e = \eta\, T_g(1-S)\frac{\omega_0}{n_p}$, where S is the slip defined by (7.34) at operating point, the output equation of the state-space model will then be

$$\Delta P_e = \begin{bmatrix} 0 & 0 & \frac{\eta(1-S)\omega_0}{n_p}D_g & 0 \end{bmatrix}\begin{bmatrix}\Delta\theta_\epsilon \\ \Delta\omega_r \\ \Delta\omega_g \\ \Delta\beta\end{bmatrix} \tag{8.8}$$

A variable speed wind turbine has two inputs, i.e., the pitch reference and generator torque reference. A state-space model for a variable speed turbine

is defined by

$$\begin{bmatrix} \Delta\dot{\theta}_\epsilon \\ \Delta\dot{\omega}_r \\ \Delta\dot{\omega}_g \\ \Delta\dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & -\frac{1}{n_{gear}} & 0 \\ -\frac{K_s}{J_r} & \frac{1}{J_r}\frac{\partial T_w}{\partial \omega_r} & 0 & \frac{1}{J_r}\frac{\partial T_w}{\partial \beta} \\ \frac{\eta_{gear} K_s}{n_{gear} J_g} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_\beta} \end{bmatrix} \begin{bmatrix} \Delta\theta_\epsilon \\ \Delta\omega_r \\ \Delta\omega_g \\ \Delta\beta \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -\frac{1}{J_g} \\ \frac{K_\beta}{\tau_\beta} & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta_{ref} \\ \Delta T_{g,ref} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J_r}\frac{\partial T_w}{\partial v} \\ 0 \\ 0 \end{bmatrix} \Delta v_{wind} + \begin{bmatrix} 0 \\ \frac{1}{J_r} \\ 0 \\ 0 \end{bmatrix} \Delta T_{3P} \quad (8.9)$$

For a wind turbine, in addition to power measurement, a measurement of generator shaft speed may also be made. The generator torque can easily be derived from these two measurements. The angular velocity of generator shaft ω_g and the generator reaction torque will then be chosen as outputs for a variable speed wind turbine [Leithead and Connor, 1994].

$$\begin{bmatrix} \Delta\omega_g \\ \Delta T_g \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta_\epsilon \\ \Delta\omega_r \\ \Delta\omega_g \\ \Delta\beta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\beta_{ref} \\ \Delta T_{g,ref} \end{bmatrix} \quad (8.10)$$

A generalized representation of the state-space model of the plant is given by

$$\begin{aligned} \dot{\mathbf{x}}_\mathbf{p} &= \mathbf{A_p x_p} + \mathbf{B_p u} + \mathbf{L_{p1}} \Delta v_{wind} + \mathbf{L_{p2}} \Delta T_{3P} \\ \mathbf{y} &= \mathbf{C_p x_p} + \mathbf{D_p u} \end{aligned} \quad (8.11)$$

where $\mathbf{x_p} = [\Delta\theta_\epsilon, \Delta\omega_r, \Delta\omega_g, \Delta\beta]^T$, $\mathbf{u} = \Delta\beta_{ref}$, $\mathbf{y} = \Delta P_e$ and $\mathbf{D_P} = \mathbf{0}$ for a constant speed plant, and $\mathbf{u} = [\Delta\beta_{ref},\ \Delta T_{g,ref}]^T$, $\mathbf{y} = [\Delta\omega_g,\ \Delta T_g]^T$ for a variable speed plant.

A bode diagram for a constant speed model is given in Figure 8.1, where the operating point of the turbine is

$$\begin{aligned} v_m &= 16\ m/s \\ \beta_0 &= 12.34^o \\ \omega_{r,0} &= 3.68\ rad/s \\ P_{e,0} &= 400\ KW \end{aligned} \quad (8.12)$$

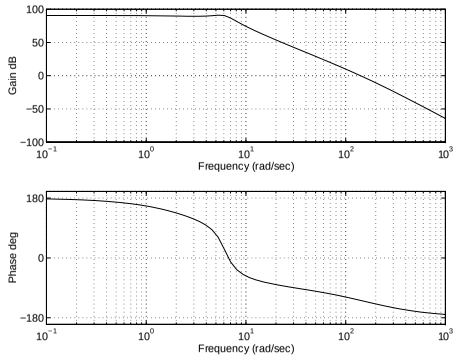

Figure 8.1. Bode plot for a constant speed wind turbine model: from reference pitch angle $\Delta\beta_{ref}$ to output electrical power ΔP_e.

At this operating point, the eigenvalues of the constant speed turbine model are given in Table 8.1. The eigenvalues of the drive train cause the resonance peak of 6.13 rad/s in the bode plot.

Eigenvalue	subsystem
-1.3167 ± 5.9913i	drive train
-169.2580	generator
-2.8571	actuator

Table 8.1. Eigenvalues for the model of a constant speed turbine

Bode diagrams for a variable speed wind turbine model is given in Figure 8.2 and Figure 8.3 at the operating point (8.12).

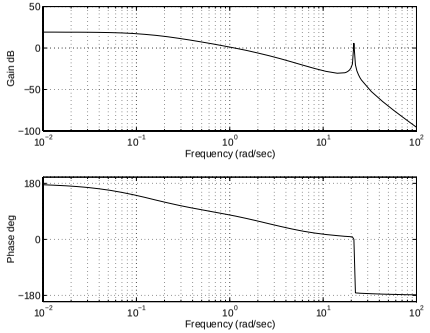

Figure 8.2. Bode plot for a variable speed wind turbine model: from reference pitch angle $\Delta\beta_{ref}$ to generator shaft speed $\Delta\omega_g$.

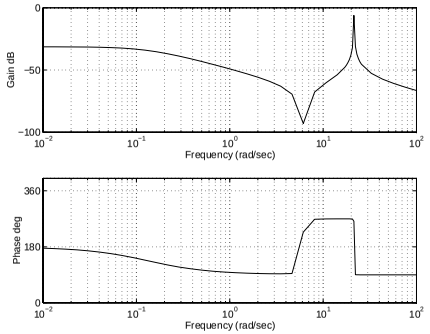

Figure 8.3. Bode plot for a variable speed wind turbine model: from generator reference torque $\Delta T_{g,ref}$ to generator shaft speed $\Delta\omega_g$.

The resonance peak is at frequency

$$\omega_2 = \sqrt{K_s \left(\frac{1}{J_r} + \frac{\eta_{gear}}{n_{gera}^2 J_g} \right)} \tag{8.13}$$

which is equal to 21.4 rad/s. The drive train exhibits a lightly damped resonance which is due to the weak connection between the rotor and grid.

The frequency function from generator torque reference to generator speed has an anti-resonance

$$\omega_1 = \sqrt{\frac{K_s}{J_r}} \tag{8.14}$$

which is equal to 6.06 rad/s.

8.1.4 Below the rated wind speed

Below rated wind speed, the objectives of the control are to maximize the captured energy and minimize the dynamic loads.

For a constant speed wind turbine, the pitch angle of the rotor blades is fixed at its optimal value and no control action is performed.

With a variable speed wind turbine, optimal energy is achieved by keeping the tip-speed ratio at its optimal value. The turbine must then track the variations of the wind speed, which demands large variations of torque and speed. The model of 3p-effect will not be included in the design model in this case.

Below rated wind speed, the generator reference torque is varied to regulate the generator speed and the pitch angle is fixed at its optimal value. It is then not necessary to include pitch dynamics in the design model. Therefore the plant is a single-input single-output system. The model can be described

by

$$\begin{bmatrix} \Delta\dot{\theta}_\epsilon \\ \Delta\dot{\omega}_r \\ \Delta\dot{\omega}_g \end{bmatrix} = \begin{bmatrix} 0 & 1 & -\frac{1}{n_{gear}} \\ -\frac{K_s}{J_r} & \frac{1}{J_r}\frac{\partial T_w}{\partial \omega_r} & 0 \\ \frac{\eta_{gear} K_s}{n_{gear} J_g} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta_\epsilon \\ \Delta\omega_r \\ \Delta\omega_g \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J_g} \end{bmatrix} \Delta T_{g,ref} + \begin{bmatrix} 0 \\ \frac{1}{J_r}\frac{\partial T_w}{\partial v} \\ 0 \end{bmatrix} \Delta v_{wind} \tag{8.15}$$

and output equations are

$$\begin{bmatrix} \Delta\omega_g \\ \Delta T_g \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta_\epsilon \\ \Delta\omega_r \\ \Delta\omega_g \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta T_{g,ref} \tag{8.16}$$

The model can be represented by (8.11) with $\mathbf{x_p} = [\Delta\theta_\epsilon,\ \Delta\omega_r,\ \Delta\omega_g]^T$, $\mathbf{u} = \Delta T_{g,ref}$, $\mathbf{y} = [\Delta\omega_g,\ \Delta T_g]^T$ and $\mathbf{L_{p2}} = \mathbf{0}$.

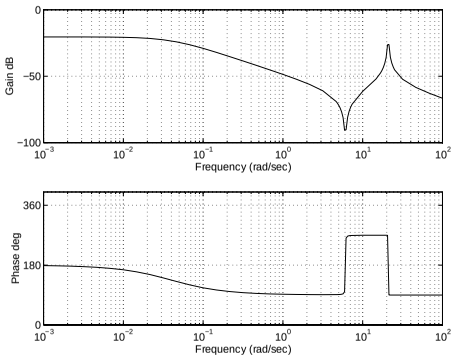

Figure 8.4. Bode plot for a variable speed wind turbine model: from generator reference torque $\Delta T_{g,ref}$ to generator shaft speed $\Delta\omega_g$.

Bode diagram is given for a frequency function from generator torque reference to generator speed in Figure 8.4 at an operating point

$$\begin{aligned} v_m &= 7\ m/s \\ \beta_{opt} &= 0.5^o \\ \omega_{r,0} &= 3.04\ rad/s \end{aligned} \tag{8.17}$$

8.2 Noise models

The wind model can be represented by a second order state-space model

$$\begin{bmatrix} \Delta\dot{v}_{wind} \\ \Delta\ddot{v}_{wind} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{p_1 p_2} & -\frac{p_1+p_2}{p_1 p_2} \end{bmatrix} \begin{bmatrix} \Delta v_{wind} \\ \Delta\dot{v}_{wind} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k}{p_1 p_2} \end{bmatrix} e_w$$

$$\Delta v_{wind} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta v_{wind} \\ \Delta\dot{v}_{wind} \end{bmatrix} \tag{8.18}$$

where e_w is a zero mean white noise with unit variance. If the state is chosen as $\mathbf{x_w} = [\Delta v_{wind}\ \Delta\dot{v}_{wind}]^T$, the above model can be written as a noise model

$$\begin{aligned} \dot{\mathbf{x}}_\mathbf{w} &= \mathbf{A_w x_w} + \mathbf{B_w} e_w \\ \Delta v_{wind} &= \mathbf{C_w x_w} \end{aligned} \tag{8.19}$$

In a design model of 3P-effect, the angular velocity of rotor and rotor torque should be considered constant, i.e., $\omega_r = \omega_{r,0}$ and $T_w = T_{w,0}$

$$\frac{\Delta T_{3P}(s)}{e_{3P}(s)} = \frac{k_{3P} T_{w,0} (3\omega_{r,0})^2}{s^2 + 2\zeta(3\omega_{r,0})s + (3\omega_{r,0})^2} \tag{8.20}$$

the 3P-effect will then be modelled by state-space equations

$$\begin{bmatrix} \Delta\dot{T}_{3P} \\ \Delta\ddot{T}_{3P} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(3\omega_{r,0})^2 & -2\zeta(3\omega_{r,0}) \end{bmatrix} \begin{bmatrix} \Delta T_{3P} \\ \Delta\dot{T}_{3P} \end{bmatrix} + \begin{bmatrix} 0 \\ k_{3P} T_{w,0} (3\omega_{r,0})^2 \end{bmatrix} e_{3P}$$

$$\Delta T_{3P} = [\ 1\ \ 0\ ] \begin{bmatrix} \Delta T_{3P} \\ \Delta \dot{T}_{3P} \end{bmatrix} \tag{8.21}$$

which in turn can be represented by

$$\begin{aligned} \dot{\mathbf{x}}_{\mathbf{3P}} &= \mathbf{A_{3P}x_{3P}} + \mathbf{B_{3P}}e_{3P} \\ \Delta T_{3P} &= \mathbf{C_{3P}x_{3P}} \end{aligned} \tag{8.22}$$

where e_{3P} is a zero mean white noise with unit variance, and it is uncorrelated with noise sequence e_w. The ΔT_{3P} model can only be included in the design model, when the angular velocity of rotor can be considered constant. Since there are substantial variations in the rotor speed when a variable speed wind turbine is employed below the rated wind speed, the ΔT_{3P} model will not be included in the design model in this case.

8.3 A composite model

A composite model can be built in which the plant model (8.11) and noise models (8.19) and (8.22) are unified into one single state-space model

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}}_{\mathbf{p}}(\mathbf{t}) \\ \dot{\mathbf{x}}_{\mathbf{w}}(\mathbf{t}) \\ \dot{\mathbf{x}}_{\mathbf{3P}}(\mathbf{t}) \end{bmatrix} &= \begin{bmatrix} \mathbf{A_p} & \mathbf{L_{p1}C_w} & \mathbf{L_{p2}C_{3P}} \\ \mathbf{0} & \mathbf{A_w} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_{3P}} \end{bmatrix} \begin{bmatrix} \mathbf{x_p}(\mathbf{t}) \\ \mathbf{x_w}(\mathbf{t}) \\ \mathbf{x_{3P}}(\mathbf{t}) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{B_p} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(\mathbf{t}) + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{B_w} & \mathbf{0} \\ \mathbf{0} & \mathbf{B_{3P}} \end{bmatrix} \begin{bmatrix} e_w(t) \\ e_{3P}(t) \end{bmatrix} \\ \mathbf{y}(\mathbf{t}) &= [\ \mathbf{C_p}\ \ \mathbf{0}\ \ \mathbf{0}\ ] \begin{bmatrix} \mathbf{x_p}(\mathbf{t}) \\ \mathbf{x_w}(\mathbf{t}) \\ \mathbf{x_{3P}}(\mathbf{t}) \end{bmatrix} + \mathbf{D_p u}(\mathbf{t}) \end{aligned} \tag{8.23}$$

Below the rated wind speed, the model of 3P-effect will be omitted in the design model.

If we choose $\mathbf{x(t)} = [\mathbf{x_p(t)}\ \mathbf{x_w(t)}\ \mathbf{x_{3P}(t)}]^{\mathbf{T}}$ and $\mathbf{e(t)} = [e_w(t)\ e_{3P}(t)]^T$, the composite model can be rewritten as

$$\begin{aligned} \dot{\mathbf{x}}(\mathbf{t}) &= \mathbf{A_c x}(\mathbf{t}) + \mathbf{B_c u}(\mathbf{t}) + \mathbf{L_c e}(\mathbf{t}) \\ \mathbf{y}(\mathbf{t}) &= \mathbf{C_c x}(\mathbf{t}) + \mathbf{D_c u}(\mathbf{t}) \end{aligned} \tag{8.24}$$

The main advantage of state-space model is that a priori knowledge about the system in a natural way can be included in the model. Since we have a well-known physical structure and physical model of the drive train system, the model should be well suited for the controller design purpose.

8.4 Discrete time model

A digital controller is normally implemented, which samples the plant output and calculates a control input which is fixed during the sample period. In this section the sample rate will be discussed and the discrete-time model will be formulated.

Generally, the performance of a digital controller improves with increasing sample rate, but cost may also be increased with faster sampling. The selection of the best sample rate for a digital control system is a compromise. The sampling theorem states that in order to reconstruct an unknown band-limited continuous signal from samples of that signal, the sample rate must be at least twice as fast as the highest frequency contained in the unknown signal. In practice, the signal will never be band limited, however, some

frequency contents of the signal can be considered insignificant above a certain frequency.

Based on the sampling theorem, the sample rate denoted by f_s must be at least twice the required closed-loop bandwidth of the system f_b, i.e.,

$$\frac{f_s}{f_b} > 2 \tag{8.25}$$

This is a fundamental lower bound on the sample rate.

However, [Franklin et al., 1990] indicates that the theoretical lower bound would be judged far too slow for an acceptable time response. He suggests that the desired sampling rate should be

$$6 \leq \frac{f_s}{f_b} \leq 40 \tag{8.26}$$

For the constant speed WD34 wind turbine, the drive train resonance frequency of 6.13 rad/s ($\approx 1Hz$) is dominated frequency, a sample rate of 10 Hz in the current used PI-controller has been chosen by *Vestas*. This sample rated will also be used in LQG controller for the constant speed pitch controlled wind turbine in order to make an unbiased comparison. However, for a variable speed pitch controlled wind turbine, due to a very lightly damped resonance at frequency of 21.4 rad/s (3.4Hz), a sample rate of 33.33 Hz (the sample time is 0.03s) is chosen to attenuate the drive train resonance.

Then a continuous model can be transformed to a discrete-time model

$$\begin{aligned} \mathbf{x(t+1)} &= \mathbf{A_d x(t) + B_d u(t) + w(t)} \\ \mathbf{y(t)} &= \mathbf{C_d x(t) + D_d u(t)} \end{aligned} \tag{8.27}$$

8.5 Summary

The design models for wind turbines have been derived in this chapter. In the case of full load, models for a constant speed turbine and a variable speed turbine are formulated separately. In partial load, only a variable speed turbine with fixed pitch is modelled. Both the wind model and the 3P-effect are considered as noise models. The T_{3P} model is included only in full load, in which there are not significant variations in the rotor speed. The sample rates for digital control systems are determined and a generalized discrete-time state-space model is derived.

Chapter 9

Estimation of The Wind Speed

Using a wind turbine for production of electrical energy requires reliable and effective operation, an active control system is often considered to realize it. Some control methods require the knowledge of the wind speed experienced by the turbine. Since the nonlinearity of the process from the aerodynamics of the turbine depends significantly on the wind speed, it seems that the wind speed is vital to the behaviour of closed-loop system.

Although there is no such thing as the wind speed experienced by a wind turbine, since the rotor is subject to a spatially distributed wind field which varies in time, the turbine may be considered to experience an effective wind speed which in some sense is an average over the rotor disc. This makes

a direct measurement of effective wind speed impossible. However, it is also impossible to predict the true wind speed passing through the rotor disc by measuring it by an anemometer, because the discrepancy between the measured wind speed and true speed on the rotor is considerably large. Even if it were possible, the presence of the turbine disturbs the wind speed and the measurement would need to be made at separate site where the wind speed correlated almost exactly with the wind speed experienced by the turbine. Hence direct measurement of wind speed is of limited value in regulating the wind turbine [Leithead et al., 1990]. The idea is then to estimate the effective wind speed experienced by the rotor by using a wind turbine as a wind measuring device.

Below the rated wind speed, the wind turbine will be adjusted to capture as much as possible of the energy from the wind. If a tracking controller is employed to keep the wind turbine running at optimal tip speed ratio where the turbine can achieve maximum power, the estimated wind speed is often used to determine the optimal control action. In this case the accurate estimated wind speed is a prerequisite for an appropriate control system design. Above the rated wind speed, a controller can be used to smooth the power generated by the wind turbine. If a gain scheduled LQG controller is applied to compensate for nonlinearities of the process, the estimated wind speed is often chosen as a scheduling variable to determine the operating point where the controller can be calibrated each sample.

An input and output signal from the turbine are required by an estimator, no matter which estimation method is used, see Figure 9.1. The plant considered here is a WD34 constant speed wind turbine with PI pitch control above rated wind speed and fixed pitch angle below rated wind speed. The

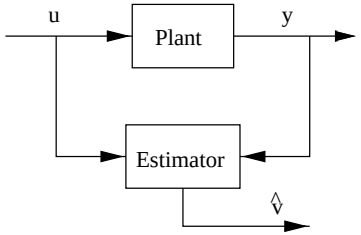

Figure 9.1. Block scheme of estimator

input signal of the turbine is pitch demand and the output measurement is electrical power.

In the following sections the Newton-Raphson method, Kalman filter method and extended Kalman filter method will be developed for wind speed estimation based on the model of the wind turbine. The performance of the algorithms will be investigated by simulations. An identical wind sequence is used for all simulations. Afterwards, some experimental data are used to test the estimation approaches. An interesting discussion concerning wind speed estimation is given at the end of chapter.

9.1 The Newton-Raphson method

The Newton type methods have been given in Appendix A. The reason to use the Newton-Raphson method is that the method has particularly fast convergence properties in most cases
[Dennis and Schnabel, 1983].

The wind speed estimator is achieved by minimizing a cost function $J(t,v)$

$$\hat{v}(t) = \underset{v}{argmin} J(t,v) \tag{9.1}$$

and the cost function is given by

$$J(t,v) = (P_r(t) - f(v))^2 \tag{9.2}$$

where $P_r(t)$ is a measurement of rotor power at time t, which is assumed that to be known, and $f(v)$ is given by

$$f(v) = \frac{1}{2}\pi\rho R^2 v^3 C_p(\lambda,\beta) \tag{9.3}$$

The problem is equivalent to finding the solution to (9.4) [Poulsen, 1985]

$$I(t,v) = P_r(t) - \frac{1}{2}\pi\rho R^2 v^3 C_p(\lambda,\beta) = 0 \tag{9.4}$$

The iteration form of the estimator is

$$\bar{v}_{n+1} = \bar{v}_n - H_n^{-1} g_n \tag{9.5}$$

where $\bar{v}_n$ is the result of n iterations, H_n and g_n can be obtained by

$$g_n = I(t,v)|_{\bar{v}_n} = P_r(t) - \frac{1}{2}\pi\rho R^2 \bar{v}_n^3 C_p(\bar{\lambda}_n,\beta) \tag{9.6}$$

$$H_n = \nabla_v I(t,v)|_{\bar{v}_n} = \left[-\frac{3}{2}\pi\rho R^2 v^2 C_p(\lambda,\beta) - \frac{1}{2}\pi\rho R^2 v^3 \frac{\partial C_p}{\partial v}\right]_{\bar{v}_n} \tag{9.7}$$

where

$$\frac{\partial C_p}{\partial v} = \frac{\partial C_p}{\partial \lambda}\frac{\partial \lambda}{\partial v} = \frac{1}{\omega_r R}\frac{\partial C_p}{\partial \lambda} \tag{9.8}$$

$\partial C_p/\partial\lambda$ can be calculated from C_p curve. By inserting (9.8) into (9.7), the H_n is

$$H_n = -\frac{3}{2}\pi\rho R^2 \bar{v}_n^2 C_p(\bar{\lambda}_n,\beta) - \frac{1}{2\omega_r}\pi\rho R \bar{v}_n^3 \frac{\partial C_p}{\partial \bar{\lambda}_n} \tag{9.9}$$

where

$$\bar{\lambda}_n = \frac{\bar{v}_n}{\omega_r R} \tag{9.10}$$

At sample time t, the measurement is $P_r(t)$, the iteration will start from $\bar{v}_0 = \hat{v}(t-1)$ and continue until

$$I(t,\bar{v}_{n_{stop}}) = (P_r(t) - f(\bar{v}_{n_{stop}})) < c \tag{9.11}$$

where the value of c is small.

The estimate of wind speed at time t is

$$\hat{v}(t) = \bar{v}_{n_{stop}} \tag{9.12}$$

The precondition to use the Newton-Raphson method for estimating the wind speed is the measurable rotor power $P(t)$. However, this condition can not be fulfilled. The rotor power has then to be approximated by the static relation

$$P_r = \frac{P_e}{\eta_{gear}\cdot\eta} \tag{9.13}$$

where $\eta_{gear}\cdot\eta$ is the coefficient of the energy loss in the system.

The simulations are carried out at average wind speed of 16 m/s (above rated) and 7 m/s (below rated). For above rated operation, the wind turbine is pitch regulated to limit the power output. For below rated operation, the pitch angle of the rotor blades is fixed at 1^o and no control action is performed. In both cases the power measurement and pitch demand are taken as inputs to the estimator.

The effective wind speed and estimated wind speed, as well as the estimation error are given in Figure 9.2 at average wind speed of 16 m/s and Figure 9.3

at average wind speed of 7 m/s. The mean value and the standard deviation of estimation error from the simulation are given in Table 9.1.

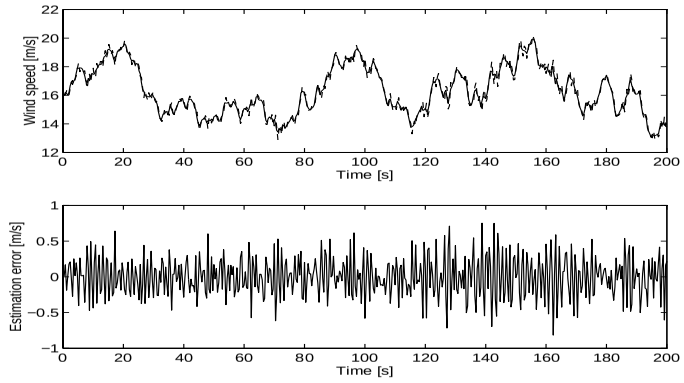

Figure 9.2. The upper shows the effective wind speed v (solid) and the estimate of the wind speed $\hat{v}$ (dashed). The lower plot shows the estimation error

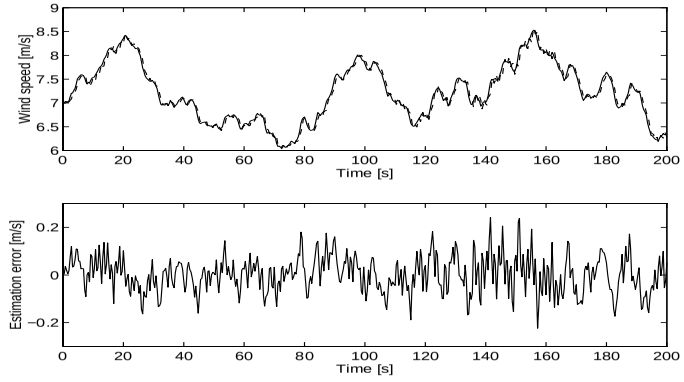

Figure 9.3. The upper shows the effective wind speed v (solid) and the estimate of the wind speed $\hat{v}$ (dashed). The lower plot shows the estimation error

$e = v - \hat{v}$	Above rated wind speed	below rated wind speed
mean value	-0.0020	0.0029
Standard deviation	0.2530	0.0709

Table 9.1. Simulation results

9.2 The Kalman filter method

The Kalman filter is the best linear estimate in a least squares sense, [Kwakernaak and Siven, 1972]. In a general nonlinear case, a very common approach to handle the state estimator is based on linearization of nonlinear functions around a nominal trajectory. For the linear models derived in chapter 8 the Kalman filter provides the exact solutions for the filtering problem in different cases (above or below the rated power, constant or variable speed turbine).

An optimal estimator is sought to minimize the criterion $E\{(\mathbf{x(t)} - \hat{\mathbf{x}}\mathbf{(t)})^T \cdot (\mathbf{x(t)} - \hat{\mathbf{x}}\mathbf{(t)})\}$ for the discrete-time system

$$\begin{aligned} \mathbf{x(t+1)} &= \mathbf{A_d x(t)} + \mathbf{B_d u(t)} + \mathbf{w(t)} \\ \mathbf{y(t)} &= \mathbf{C_d x(t)} + \mathbf{D_d u(t)} + \mathbf{e(t)} \end{aligned} \tag{9.14}$$

where $\mathbf{w(t)}$ and $\mathbf{e(t)}$ are mutually independent white noise sequence with zero mean and covariance

$$E\left\{ \begin{pmatrix} \mathbf{w(t)} \\ \mathbf{e(t)} \end{pmatrix} (\mathbf{w(t)} \quad \mathbf{e(t)}) \right\} = \begin{pmatrix} \mathbf{R_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R_2} \end{pmatrix} \tag{9.15}$$

The Kalman filter method has been already given in Appendix B. If the operating point is fixed, the matrices $\mathbf{A_d}$, $\mathbf{B_d}$, $\mathbf{C_d}$ and $\mathbf{D_d}$ can be taken as constant matrices, the Kalman gain will then converge to a constant gain as t goes to infinity.

$$\begin{aligned} \mathbf{K}_\infty &= \mathbf{A_d}\mathbf{\Sigma}_\infty \mathbf{C_d}^T(\mathbf{C_d}\mathbf{\Sigma}_\infty \mathbf{C_d}^T + \mathbf{R_2})^{-1} \\ \mathbf{\Sigma}_\infty &= \mathbf{A_d}\mathbf{\Sigma}_\infty \mathbf{A_d}^T + \mathbf{R_1} - \mathbf{K}_\infty(\mathbf{C_d}\mathbf{\Sigma}_\infty \mathbf{C_d}^\mathbf{T} + \mathbf{R_2})\mathbf{K}_\infty \end{aligned} \tag{9.16}$$

The state estimator will then be computed from the current estimate plus the new data via

$$\mathbf{\hat{x}(t+1)} = \mathbf{A_d\hat{x}(t)} + \mathbf{B_d u(t)} + \mathbf{K}_\infty(\mathbf{y(t)} - \mathbf{C_d\hat{x}(t)} - \mathbf{D_d u(t)}) \tag{9.17}$$

The constant Kalman gain can be precalculated. The off-line calculation offers a computational advantage.

The performance of the Kalman filter method is assessed by simulations. The Kalman estimator is based on a linear composite model unifying the plant model (8.7) - (8.8), the wind model (8.19) and the model of 3P-effect (8.22). Above rated operation the linear model is derived at nominal operating point (8.12). Below rated operation the model is linearized at operating point

$$\begin{aligned} v_m &= 7\ m/s \\ \beta_0 &= 1^o \\ \omega_{r,0} &= 3.68\ rad/s \end{aligned} \tag{9.18}$$

The effective wind speed and estimated wind speed, as well as estimation error are given in Figure 9.4 at average wind speed of 16 m/s and Figure 9.5 at average wind speed of 7 m/s. In order to make a comparison, the same wind sequences are used for all simulations in this chapter. Table 9.2 gives the mean value and the standard deviation of estimation error from the simulation.

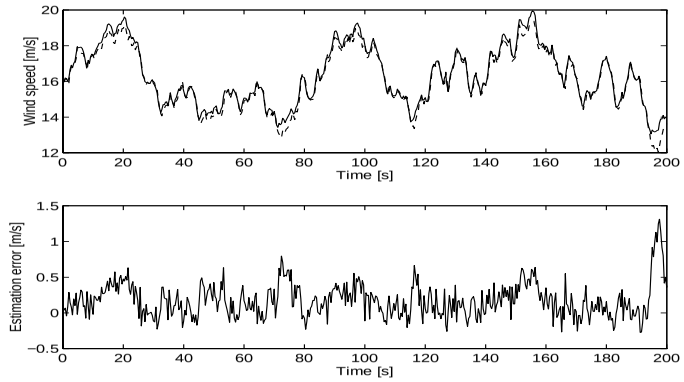

Figure 9.4. The upper shows the effective wind speed v (solid) and the estimate of the wind speed $\hat{v}$ (dashed). The lower plot shows the estimation error

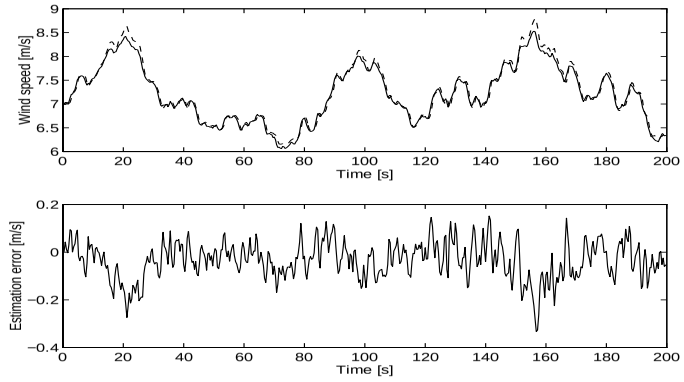

Figure 9.5. The upper shows the effective wind speed v (solid) and the estimate of the wind speed $\hat{v}$ (dashed). The lower plot shows the estimation error

$e = v - \hat{v}$	Above rated wind speed	below rated wind speed
mean value	0.1723	-0.0343
Standard deviation	0.2258	0.0736

Table 9.2. Simulation results

It can be clearly seen from Figure 9.4 and Figure 9.5 that the estimation error increases when the wind speed deviates from the nominal wind speed. The reason is that the linear model used by the Kalman filter design is fixed at nominal operating point. The difference between the linear model and actual system increases as the wind leaves the nominal point. Hence, the use of the extended Kalman filter may be necessary for estimation of the wind speed.

9.3 The extended Kalman filter method

A linearized plant representation is typically employed for estimator and controller design. However, a wind turbine is a nonlinear system, in particular, the aerodynamic behaviour is highly nonlinear. Because the role of the rotor in converting wind energy into mechanical energy is a central one, the nonlinear aerodynamic of the rotor exerts a substantial influence on the characteristics of the whole system. [Leith and Leithead, 1997]. Since it is known how the dynamics of the process change with the operating conditions, it is possible to change the parameters of the estimator by monitoring the operating condition of the process.

A better choice for linearization trajectory is to use the current estimate of the state. Linearizing the model about it at every sampling time and applying the Kalman filter to the resulting linearized model yields the algorithm known as *the extended Kalman filter* [Gelb, 1974]. The extended Kalman filter can be considered as an optimal nonlinear estimation algorithm for nonlinear systems, which has already been given in Appendix B.

The investigation of the extended Kalman filter method used for wind speed estimation will be carried out separately for below rated operation and above rated operation.

9.3.1 Above rated wind speed

The aerodynamic torque T_r depends nonlinearly on both the effective wind speed v and the pitch angle β for a constant speed turbine. Since a PI regulator is applied to the system for the purpose of power limitation, it can be assumed that the nominal operating point of the system is around the power output of 400 KW. The contour level of 400 KW is drawn in Figure 9.6.

For each wind speed above rated wind speed, the rated aerodynamic torque $T_{w,0}$ is attained at a unique pitch angle, i.e., for a constant aerodynamic torque, the pitch angle β is a function of the wind speed v. The nonlinearity of the model depends then on the effective wind speed only. This means that the nonlinear aerodynamic torque can be linearized by monitoring the current estimate of the wind speed at every sampling time.

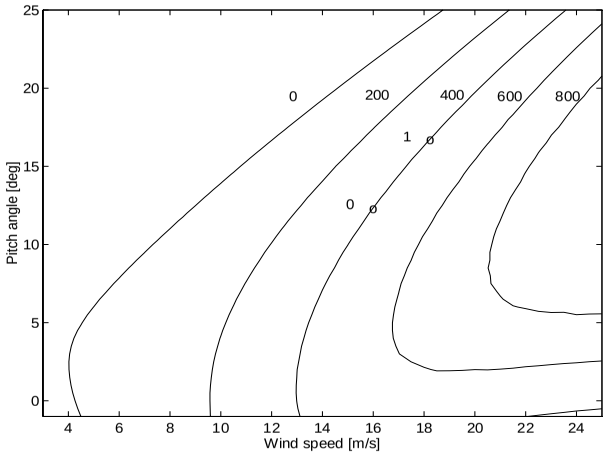

Figure 9.6. Contour plot of output electrical power $P_e = [0, 200, 400, 600, 800]$ KW for the constant speed WD34 wind turbine.

In Figure 9.6, it is assumed that the nominal operating point for the system is at point 0 defined by v_m, β_0 and $\omega_{r,0}$ and working point for rotor is around point 1 defined by v_1, β_1 and $\omega_{r,0}$. The linearized aerodynamic torque at nominal operating point 0 has been given by equations (8.2) - (8.4), and the linearization of rotor torque about point 1 is given by

$$T_r = T_{w,1} + \Delta T_{w,1} + \Delta T_{3P} \tag{9.19}$$

and

$$\Delta T_{w,1} = \left.\frac{\partial T_w}{\partial v}\right|_{v_1} \Delta v' + \left.\frac{\partial T_w}{\partial \beta}\right|_{v_1} \Delta \beta' + \left.\frac{\partial T_w}{\partial \omega_r}\right|_{v_1} \Delta \omega_r' \tag{9.20}$$

$$\begin{aligned} \Delta v' &= v - v_1 = \Delta v - \Delta v_1 \\ \Delta \beta' &= \beta - \beta_1 = \Delta \beta - \Delta \beta_1 \\ \Delta \omega_r' &= \omega_r - \omega_{r,0} = \Delta \omega_r \end{aligned} \tag{9.21}$$

where $\Delta v_1 = v_1 - v_m$ and $\Delta \beta_1 = \beta_1 - \beta_0$. The stationary rotor torque on the contour level of 400 KW is constant, i.e., $T_{w,1} = T_{w,0}$. The small deviation of aerodynamic torque ΔT_w can be rewritten as

$$\Delta T_{w,1} = \left.\frac{\partial T_w}{\partial v}\right|_{v_1} \Delta v + \left.\frac{\partial T_w}{\partial \beta}\right|_{v_1} \Delta \beta + \left.\frac{\partial T_w}{\partial \omega_r}\right|_{v_1} \Delta \omega_r - d_{T_w}(1) \tag{9.22}$$

where

$$d_{T_w}(1) = \left.\frac{\partial T_w}{\partial v}\right|_{v_1} \Delta v_1 + \left.\frac{\partial T_w}{\partial \beta}\right|_{v_1} \Delta \beta_1 \tag{9.23}$$

The partial derivatives $\partial T_w/\partial v$, $\partial T_w/\partial \beta$ and $\partial T_w/\partial \omega_r$ of the aerodynamic torque with respect to the wind speed, pitch angle and rotor speed can be searched by the estimated wind speed $\hat{v}(t)$ at sampling time t. The scheduling table for working point is illustrated in Figure 9.7, in which the parameters are calculated by assuming that the output power produced by the turbine is 400 KW.

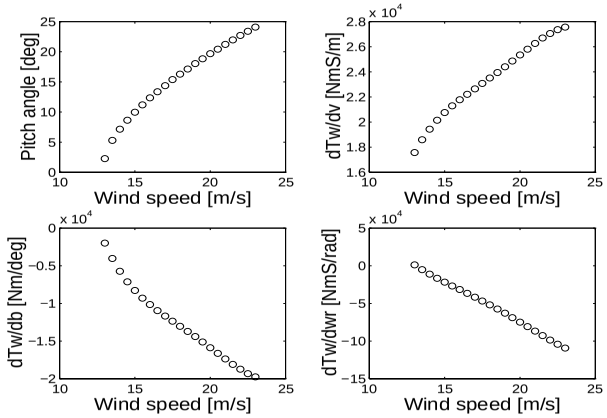

Figure 9.7. Schedule table of operating point to produce 400 KW electrical power. ($dTw/dv = \partial T_w/\partial v$, $dTw/db = \partial T_w/\partial \beta$ and $dTw/dwr = \partial T_w/\partial \omega_r$)

The wind model is also changed with the average wind speed. The parameters in the wind model can be determined by monitoring the average wind speed. If the mean wind speed is assumed constant, it is only the aerodynamics that will be compensated. The state-space linear model will be represented by

$$\begin{aligned} \dot{\mathbf{x}}(\mathbf{t}) &= \mathbf{A_c}(\hat{v}(t))\mathbf{x}(\mathbf{t}) + \mathbf{B_c}\mathbf{u}(\mathbf{t}) + \mathbf{H_c}\mathbf{d}(\mathbf{t}) + \mathbf{L_c}\mathbf{e_1}(\mathbf{t}) \\ \mathbf{y}(\mathbf{t}) &= \mathbf{C_c}\mathbf{x}(\mathbf{t}) + \mathbf{e_2}(\mathbf{t}) \end{aligned} \tag{9.24}$$

where the state is $\mathbf{x(t)} = [\Delta\theta_\epsilon, \Delta\omega_r, \Delta\omega_g, \Delta\beta]^T$, and

$$\mathbf{H_c} = \begin{bmatrix} 0 \\ -\frac{1}{J_r} \\ 0 \\ 0 \end{bmatrix} \tag{9.25}$$

$$\mathbf{d(t)} = \mathbf{d_{T_w}}(\hat{v}(t), t) \tag{9.26}$$

The so-called extended Kalman estimator based on the discrete-time linear state-space model at sampling time t can be described by

$$\begin{aligned} \hat{\mathbf{x}}(\mathbf{t+1}) &= \mathbf{A_d}(\hat{v}(t))\hat{\mathbf{x}}(\mathbf{t}) + \mathbf{B_d}\mathbf{u}(\mathbf{t}) + \mathbf{H_d}\mathbf{d}(\mathbf{t}) + \mathbf{K}(\mathbf{t})[\mathbf{y}(\mathbf{t}) - \mathbf{C_d}\mathbf{x}(\mathbf{t})] \\ \mathbf{K}(\mathbf{t}) &= \mathbf{A_d}(\hat{v}(t))\mathbf{\Sigma}(\mathbf{t})\mathbf{C_d}^T[\mathbf{C_d}\mathbf{\Sigma}(\mathbf{t})\mathbf{C_d}^T + \mathbf{R_2}]^{-1} \\ \mathbf{\Sigma}(\mathbf{t}) &= \mathbf{A_d}(\hat{v}(t))\mathbf{\Sigma}(\mathbf{t})\mathbf{A_d}^T(\hat{v}(t)) + \mathbf{R_1} \\ &\quad -\mathbf{K}(\mathbf{t})(\mathbf{C_d}\mathbf{\Sigma}(\mathbf{t})\mathbf{C_d}^T + \mathbf{R_2})\mathbf{K}^T(\mathbf{t}) \end{aligned} \tag{9.27}$$

where $\mathbf{R_1}$ and $\mathbf{R_2}$ are variance metrics of the state noise and the measurement noise.

The simulation is performed at average wind speed of 16 m/s. The estimated wind speed is a scheduling variable to adjust the model parameters. Hence the parameters of the Kalman estimator can be changed by monitoring the operating point.

The effective wind speed, the estimated wind speed and the estimation error are given in Figure 9.8. The mean value and standard deviation are 0.085 and 0.1236 respectively. It is obvious that the estimation error will not be increased when the wind speed is far from the nominal wind speed.

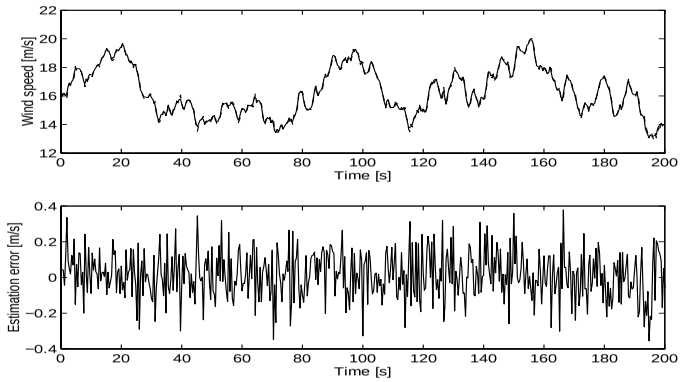

Figure 9.8. The upper shows the effective wind speed v (solid) and the estimate of the wind speed $\hat{v}$ (dashed). The lower plot shows the estimation error

9.3.2 Below rated wind speed

Below rated wind speed, if the wind speed can be estimated, it can be used as a scheduling variable to determine the pitch angle reference for a constant speed wind turbine, when the turbine is designed in such a way that the aerodynamic torque is sensitive to the pitch angle. It can also be used to determine the optimal rotor angular velocity for a variable speed wind turbine to achieve maximum power. Since the existing WD34 wind turbine is designed for a fixed pitch angle below rated wind speed, which means that no control action is available, the estimator will be implemented for a

fixed pitch WD34 wind turbine. The similar methods can be developed for a variable pitch or a variable speed wind turbine.

The aerodynamic torque (9.19) at the working point 1 can be written by

$$T_r = T_{w,0} + \Delta T_{w,1} + \Delta T_{3P} \tag{9.28}$$

where

$$\Delta T_{w,1} = \left.\frac{\partial T_w}{\partial v}\right|_{v_1} \Delta v + \left.\frac{\partial T_w}{\partial \omega_r}\right|_{v_1} \Delta\omega_r - d_{T_w}(1) \tag{9.29}$$

where

$$d_{T_w}(1) = \left.\frac{\partial T_w}{\partial v}\right|_{v_1} \Delta v_1 - (T_{w,1} - T_{w,0}) \tag{9.30}$$

The parameters $T_{w,1}$, $\partial T_w/\partial v$, and $\partial T_w/\partial \omega_r$ can be found by searching scheduling table by the estimated wind speed. The scheduling table is shown in Figure 9.9.

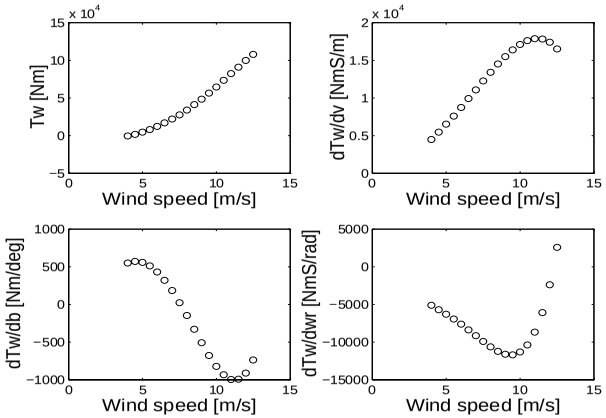

Figure 9.9. Schedule table of working point. ($dTw/dv = \partial T_w/\partial v$, $dTw/db = \partial T_w/\partial \beta$ and $dTw/dwr = \partial T_w/\partial \omega_r$.)

The state-space model and the extended Kalman filter algorithm can be written in the same way as equations (9.24)-(9.26) and (9.27).

The simulation is performed at the average wind speed of 7 m/s. The estimation results are given in Figure 9.10. The mean value and standard deviation of the estimation error are 0.0025 and 0.0302 respectively.

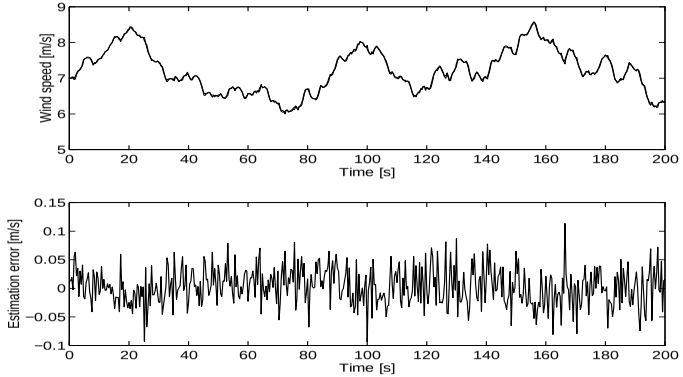

Figure 9.10. The upper shows the effective wind speed v (solid) and the estimate of the wind speed $\hat{v}$ (dashed). The lower plot shows the estimation error

9.4 A comparison

A comparison of the three estimation methods are divided into two operational modes, i.e., above rated wind speed and below rated wind speed. The results that were obtained in previous sections by the three different estimation methods are listed in Table 9.3 at average wind speed of 16 m/s and Table 9.4 at average wind speed of 7 m/s.

$e = v - \hat{v}$	Newton	Kalman	Extended Kalman
mean value	-0.0020	0.1723	0.0142
Standard deviation	0.2530	0.2258	0.1236

Table 9.3. At average wind speed of 16 m/s.

$e = v - \hat{v}$	Newton	Kalman	Extended Kalman
mean value	-0.0029	-0.0343	0.0025
Standard deviation	0.0709	0.0736	0.0302

Table 9.4. At average wind speed of 7 m/s.

The Newton-Raphson method with the assumption $P_r = P_e/\zeta$ provides a small mean value of estimation error in both cases, but it gives a large variance. The reason is that the rotor power P_r is not measurable and the assumption $P_r = P_e/\zeta$ leads to the estimation error. However, the Newton-Raphson method is the most simple method. It is not based on the model of the whole system and it is not necessary to know the average wind speed.

The Kalman filter method gives both large variance and mean value of estimation error in the three methods, because the linear model used in the Kalman filter design is fixed at nominal operating point. It can be clearly seen from the simulations, the estimation error is increased when the wind speed deviates from the nominal wind speed, where the difference between the linear model and actual system increases. The Kalman filter algorithm yields less filtering accuracy than the extended Kalman filter algorithm,

but it offers computational advantage. The Kalman gain can be calculated off-line.

The extended Kalman filter is a special kind of open-loop adaptation or change of estimator parameters, which gives the best performance in the three methods. The method achieves the smallest variances in both cases. The extended Kalman filter method has the advantage that the linear model used in the Kalman filter design can be calibrated for operating point by monitoring the estimated wind speed. Therefore the changes in process gain can be considered by the estimator, and the estimation error will not be increased when the wind speed deviates from the nominal wind speed.

Both the Kalman filter method and the extended Kalman filter method require a good model of the system. Even though the extended Kalman filter gives the best performance, because of the computation time required by the method and limited available resources, the Newton and Kalman filter methods are still good choices for implementation on a real turbine.

9.5 Test

The data sequences from the WD34 wind turbine were collected in an open-loop experiment, in which the pitch angle of rotor blades was fixed at 1^o, the wind speed and power output were measured. The wind was measured by an anemometer which is at 68m in front of the turbine, at the same height as the hub.

A comparison of the experimental data and the estimated data is very difficult. First of all, the measured wind speed has to be filtered to obtain the wind speed experienced by the rotor. The filter function is an approximation. Secondly, the distance between the measurement point and rotor of the turbine may cause a time delay and discrepancy between the measured wind speed and the true wind speed at the wind turbine, especially the high frequency components in the wind. Thirdly, the power measurements are corrupted by 1P, 2P and 3P-fluctuations, which have to be filtered out. The model of 3P-effect is not included in the design model in this test.

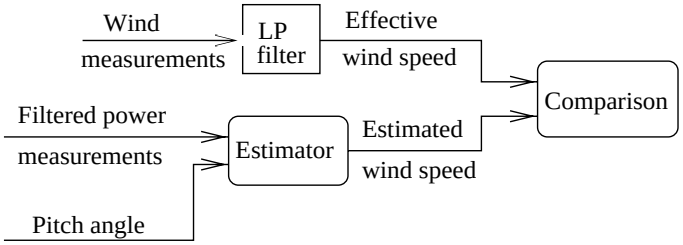

Figure 9.11. Block scheme of the comparison

The block scheme of the comparison of the experimental result and estimated wind speed is illustrated in Figure 9.11. The effective wind speed experienced by the turbine is derived by dynamically filtering the measurements of the point wind speed. The filter function was given in (7.45). The pitch angle and power measurements will be used as an input and output signal to the estimator. Since there are evident contents of 1P, 2P and 3P-effect in the power measurements, which depend significantly on the specific wind turbine and site, the power measurements have to be filtered by band-stop filters with bandwidth of $0.15 \times (n\omega_{r,0})$ and a fourth order Butterworth low-pass filter with cut-off frequency of 8 rad/s to discard high frequency

components. The block scheme of prefiltering the power measurements is shown in Figure 9.12.

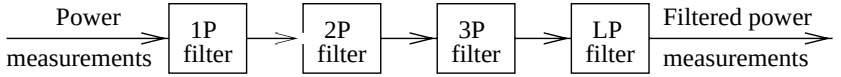

Figure 9.12. Prefiltering of power measurements

The test is carried out by using the Kalman filter method. The simulation result is plotted in Figure 9.13. The estimated wind speed given in the figure seems to agree with the filtered measurements. The time delay between the estimated wind speed and filtered wind measurements can be obviously seen. This is caused by the distance between the measurement point and the turbine. The two data sequences deviate in some areas, it may caused by the distance or the precision of the filter function.

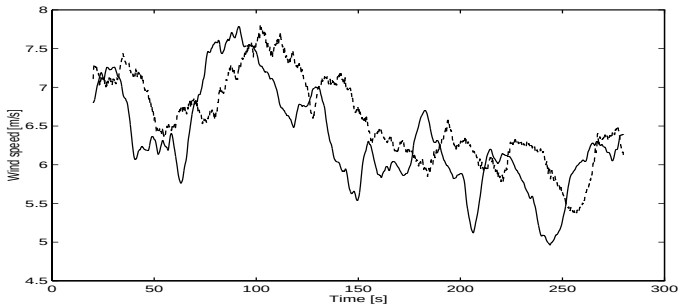

Figure 9.13. The filtered measurements v (solid) and the estimate of the wind speed $\hat{v}$ (dashed)

9.6 Discussion

It has been mentioned by [Connor and Leithead, 1994] and [Ekelund, 1997], that the wind speed cannot be uniquely determined, since the aerodynamics are non-linear and the relationship between the wind speed and the aerodynamic power is non-unique: There may be more than one wind speed for a given aerodynamic power. Hence, the wind speed is not observable in the nonlinear model. In practice, this becomes a problem when the tip-speed ratio is close to the point where the aerodynamic torque has a maximum with respect to wind speed.

Figure 9.14 shows a $P_r - v$ curve for the WD34 constant speed wind turbine at the pitch angle of 1^o. Figure 9.15 gives the $P_r - v$ curves for the WD34 constant speed wind turbine at pitch angle of $5^o, 10^o, 15^o, 20^o$.

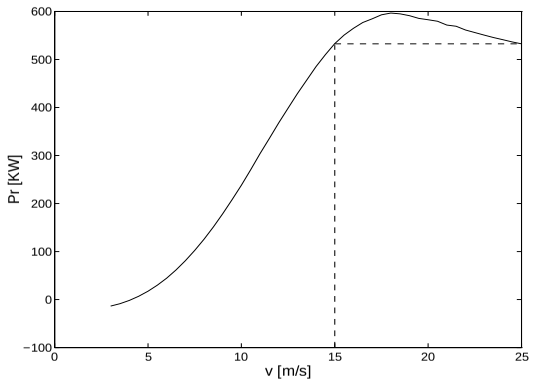

Figure 9.14. A $P_r - v$ curve for the WD34 constant speed wind turbine when the pitch angle is 1^o.

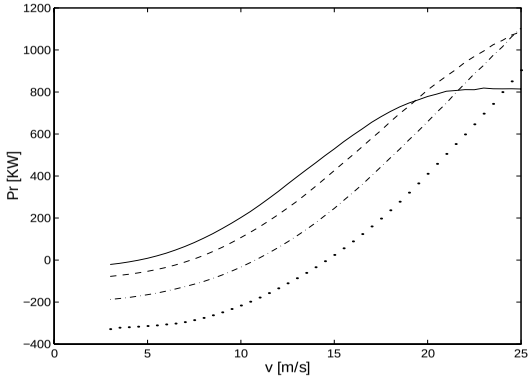

Figure 9.15. $P_r - v$ curves for the WD34 constant speed wind turbine with different pitch angles. (— $\beta = 5^o$; - - $\beta = 10^o$; -· $\beta = 15^o$; ·· $\beta = 20^o$)

It can be found that the quadratic part of the curve in Figure 9.14 appears at high wind speed ($v > 15$ m/s). The wind speed where the aerodynamic power has maximum power is 18 m/s. The estimation problem will occur when the wind speed is close to 18 m/s. However, since the wind turbine is equipped with active pitch controller, the operating point of the turbine will be moved to the curves in Figure 9.15, which have a unique wind speed for a given aerodynamic power. In short, for pitch controlled wind turbines the estimation problem will not occur because they are always operating in the non-stall region.

By investigation, most wind turbines produced by *Vestas* have the similar $P_r - v$ curves with the little differences in wind speed where the aerodynamic power has maximum value.

In practice, the turbine will seldom go into the area where the peak of aerodynamic power occurs if the switching from below rated wind speed to

above rated wind speed or verse vice are properly designed. In order to increase the robustness of the controller, the operating point closed to the switching point can be fixed for the controller, and gain scheduling with the estimated wind speed as a scheduling variable will only be implemented when the pitch is larger than 5^o.

As it has been suggested by [Ekelund, 1997], an alternative approach is to define the operational strategy in the torque-rotor speed plane. Any control strategy is equivalent to a curve in this plane. Only an estimate of aerodynamic torque is required rather than an estimate of wind speed.

9.7 Summary

Three different estimation methods have been presented in this section. An analysis and comparison of three methods are made based on the simulation results. The estimation methods are tested by experimental data, and the estimated wind speed shows a good agreement with the filtered measurements of the point wind speed. The results indicate that estimation of the wind speed is a useful method for the controller design, if it requires the knowledge of the wind speed. However, the problem may occur at switching area, which require carefully design of the switching from above rated wind speed to below rated wind speed or verse vice.

Chapter 10

Control Above Rated Power

The emphasis of this chapter is given to the design of control systems above rated power. Design of control systems for a constant speed and variable speed wind turbine is treated separately.

The control objectives for wind turbines outlined in the introduction need to be clarified more precisely. The load transients experienced by the wind turbine are of two types. Firstly, the wind, as a disturbance input to the turbine, results in large oscillations in drive train torque and electrical power. The control system should attempt to reduce these to a minimum and reduce the stress on the power train components. Variations in loads cannot be completely eliminated but reducing their magnitude is desirable. To

what extent these benefits are accomplished depends on the control system and the characteristics of the plant. Secondly, there are variations in the structure loads. If the control system operates perfectly, the turbine structure states will track their steady state values. Of course, the control system does not perform perfectly and turbine experiences additional transient structural loads, the controller should attempt to reduce these to a minimum. In addition, the wind turbine is subject to 3P-effect due to rotational sampling of the wind field. The controller should not cause these to be aggravated.

The control system must increase rejection of the disturbances caused by wind speed variations, and also reject slow external disturbances such as changes in mean wind speed by reducing the steady state errors. At the same time, the controller should reduce the extent of control action and prevent too high a control demand [Leithead et al., 1992]. A trade-off has to be made between different objectives.

The issue as to whether structural resonances should be avoided is not considered here and so no avoiding action is incorporated in the operational strategy. However, the tower motion will be noted in simulations.

Pitch control

The wind turbine is subject to stochastic varying loads which are induced by wind speed variations both in time and over the disc swept by the rotor. The torque induced on the rotor by the wind is dependent on the pitch angle of the blades. This makes it possible to control the energy absorbed by the turbine by changing the pitch angle of the blades.

In constant speed wind turbines, the rotor directly drives the grid-connected generator and hence must rotate at an approximate fixed multiple of the grid frequency. The constant speed turbine is characterized by stiff system dynamics, with a large rotor inertia. When the wind speed rises above rated, the control task for constant speed pitch-regulated turbines is to vary the pitch angle of the blades suitably to regulate the power output to the rated value, while minimizing the load transients and thereby reducing fatigue.

However, the task of power smoothing requires fast variations of the pitch angle of blades which result in large mechanical loads on the blades and reduced life time of turbines. Therefore the control design should be based on a trade-off between the power smoothing and pitch movements.

A constant speed wind turbine is a single input and single output system.

Combined variable speed and pitch control

In variable speed wind turbines, the generator is decoupled from the grid by the power electronics and rotor may rotate at any speed. A variable speed machine has two possible means of control. Power electronics controlling the electrical link to the grid, provides a variable speed effect. The pitch mechanism, if available, alters the effective rotor aerodynamic efficiency.

The usual philosophy for designing a control system for above rated operation is to use a fast control action on generator reference torque and a slower control action on pitch angle to minimize actuator activity.
[Leithead and Connor, 1994]

Above rated wind speed, a variable speed wind turbine is a two input and two output system with interaction between the two control actions.

Different control methods

Since the control system has often been restricted to PI control in practical applications, and it has also been employed by the WD34 wind turbine, it is then necessary to give a short description of the active pitch PI control in order to make comparisons with other control methods. The LQG control method for a solely pitch controlled wind turbine as well as a combined variable speed and pitch controlled wind turbine will be investigated in this chapter. A gain scheduling is introduced to the control scheme to compensate for the nonlinear rotor aerodynamics.

In order to make comparisons, an identical wind sequence at average wind speed of 16 m/s given in Figure 7.5 is used for simulations. The operating point is given in (8.12) in chapter 8.

This chapter is organized as follows. Section 10.1 gives a short description of an available PI pitch controller for the constant speed WD34 wind turbine. Section 10.2 discusses the LQG pitch control for the constant speed wind turbine. A combined variable speed and pitch control system is investigated in section 10.3. A summary of the different methods is given in section 10.4.

10.1 PI pitch control

The PI control is the most commonly used control method for medium and large scale wind turbines of today. Hence, an available PI pitch controller for a constant speed turbine will be described in this section.

A PI control loop on electrical power is the standard control configuration, which is illustrated in Figure 10.1. The basic problem is then to make the

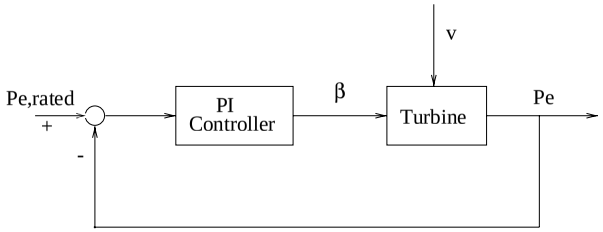

Figure 10.1. The PI control for the wind turbine

output power to follow the set point. The electrical power is measured and compared to the desired value (the value of rated power) with error used to adjust the plant through a PI-controller.

The controller used in the WD34 wind turbine is a digital PI-controller which is implemented as a bilinear transformation of a conventional analogue PI-controller

$$G(z) = K_p(1 + \frac{1}{ST_i}) \tag{10.1}$$

with

$$S = \frac{2}{T_s}\left(\frac{z-1}{z+1}\right)$$

where K_p is proportional gain, T_i is integration time and T_s is sampling time. The sampling time and integration time of the PI controller used by the existing WD34 wind turbine are 0.1 sec and 0.06 sec respectively. The proportional gain is chosen by simulation to constrain the extent of pitch action.

$$\begin{aligned} K_p &= -2.9 \cdot 10^{-6}\ ^o/KW \\ T_i &= 0.06\ s \end{aligned} \tag{10.2}$$

The step responses of the output power P_e and pitch angle β obtained by the PI-controller are given in Figure 10.2. The step input to the turbine

is the wind speed which changes from 16 m/s to 17 m/s at 1 sec. It can be seen that the output power is suppressed by the PI pitch regulator to follow the power reference. However, the overshoot of the output power is large, which is almost of same amplitude as the uncontrolled system. The 3P-effect is not included in simulation model in order to show the steady state responses.

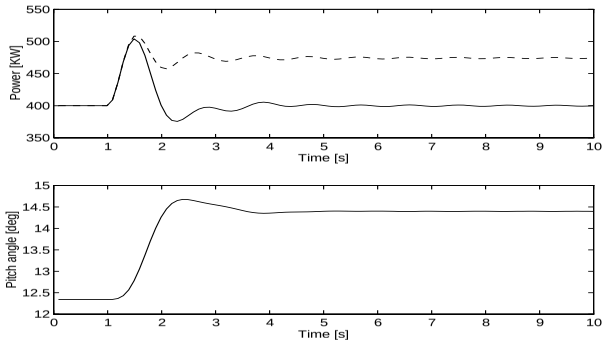

Figure 10.2. Step responses of output power and pitch angle with PI-controller. The dashed line in the upper plot is open-loop response of output power.

The bode plots of the transfer function from the wind to the output power for the controlled and uncontrolled wind turbine are give in Figure 10.3. The figure shows that the PI-controller attenuates the variations in the electrical power caused by the wind variations up to about 1.7 rad/s. However, between 1.7 rad/s and 7.5 rad/s the variations are increased, which is undesirable. It can be found that the 3P resonance at the frequency of 11 rad/s is neither attenuated nor amplified, but 1P resonance at frequency of 3.685 rad/s is amplified. There is a slight increase in amplitude at the structural resonance frequency of 6.9 rad/s. Simulation shows that these slight increase is not harmful.

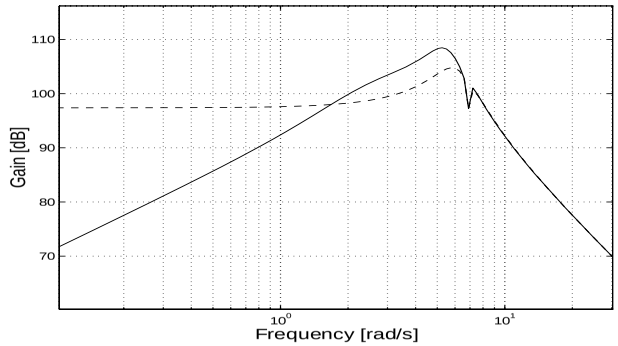

Figure 10.3. Bode plots of the transfer function from Δv to ΔP_e. The dashed line is for the open-loop system and the solid line is for the closed loop system with PI-controller.

The PI-controller is furthermore designed with a nonlinear gain scheduling. The nonlinearity reduces the gain with increasing pitch to compensate for the increase in gain through the system.

To obtain a more realistic evaluation of the controller the full non-linear stochastic model of the wind turbine is employed for simulation. The PI-regulator given in (10.1) - (10.1) is implemented for power limitation. The results are shown in Figure 10.4, in which a comparison between the power output from the open-loop system and the closed-loop system shows the performance of the PI-regulator. The pitch angle of blades is also plotted in the figure.

It is obvious that the influence of the disturbances from the wind on the output power is reduced by the controller, i.e., the controller enhances the disturbance rejection properties of the turbine and smooth the generated power.

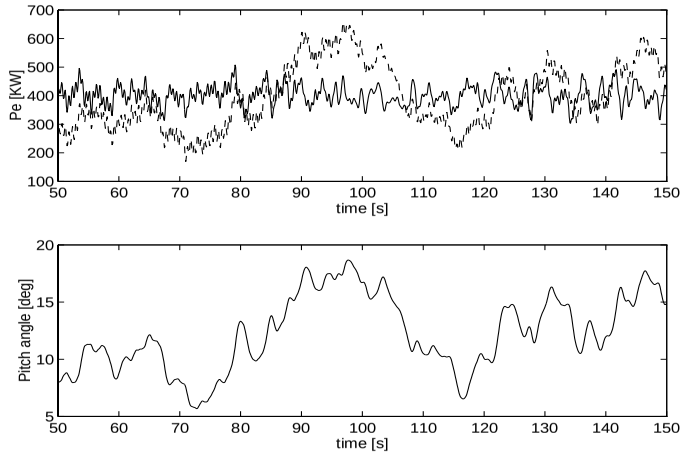

Figure 10.4. Simulation of the wind turbine with a PI-controller. The upper plot shows the power output from the uncontrolled (dashed) and controlled (solid) wind turbine. The lower plot shows the pitch angle of the blades.

The key parameters from the simulation of 200 sec are given in Table 10.1. The output power and generator reaction torque are used as an indicator of the performance of closed-loop system. The pitch angle and pitch rate are given to show the pitch performance since there are strict limits on them in order to prevent the control action from being too strong. The speed and displacement of tower bending movement are given to see if the controller excites the tower bending.

By comparing the parameters in Table 10.1 with Table 7.2 obtained from an open-loop simulation, it can be found that the standard deviation of generator reaction torque has been decreased significantly by the PI-controller, i.e., the load transients throughout the turbine have been alleviated and smooth power can thus be achieved. It can also be found that the PI-controller does not give the significant increases in the tower bending movements.

mean(P_e) [KW]	mean(T_g) [Nm]	mean(ω_g) [rad/s]	max($h_t\omega_t$) [cm/s]
399.46	3928.83	105.77	2.59
SD(P_e) [KW]	SD(T_g) [Nm]	SD(ω_g) [rad/s]	max($h_t\theta_t$) [cm]
34.92	340.20	0.09	3.65
max(β) [deg]	min(β) [deg]	SD(β)	max($\dot{\beta}$) [deg/s]
19.65	3.88	3.55	5.04

Table 10.1. Statistics obtained while simulating the PI pitch control system. Mean wind speed is 16 m/s.

However, it has been mentioned by [Leithead, 1989a] that the PI-controller may not be entirely satisfactory because it is unable to add damping.

10.2 LQG pitch control

LQG optimal control law

Linear Quadratic Gaussian (LQG) optimal control law is firmly established based on state-space description of systems. The main characteristics of LQG are optimization in terms of quadratic performance criteria and incorporation of the Kalman optimal estimation theory [Bitmead et al., 1990]. Figure 10.5 depicts the interconnection of the plant, the observer and the control law for a pitch regulated wind turbine.

The quadratic cost function which we seek to minimize is given by

$$\mathbf{J}(\mathbf{N},\mathbf{t}) = E\left\{\sum_{j=0}^{N-1}\{Q_y(\Delta P_e(t+j))^2 + Q_u(\Delta\beta_{ref}(t+j))^2\}\right\} \qquad (10.3)$$

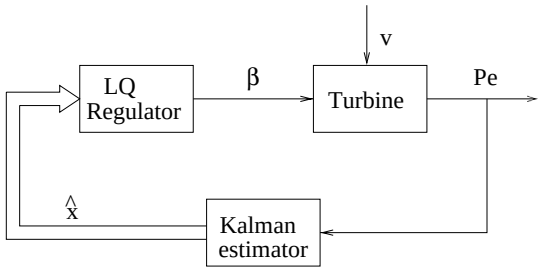

Figure 10.5. The LQG control for the wind turbine

where E denotes expectation used with respect to the stochastic uncertainties, and Q_y and Q_u are non-negative definite scalars. The choice of the weighting parameters depends on the relative importance of the output power quality or pitch variations. The trade-off can be made between the output performance and control effect. Since the limitation of pitch rate is easily reached, the weighting matrices should be chosen to prevent too high a control demand.

A suitable model for a constant speed wind turbine has been given in Chapter 8. An additional state will be included to deal with stationary non-zero errors

$$\mathbf{x_i(t+1)} = \mathbf{x_i(t)} + (\mathbf{r(t)} - \mathbf{y(t)}) \tag{10.4}$$

This is an integral state. The $\mathbf{r(t)}$ in the equation is an externally prescribed reference signal. Since the linear state-space model describes the deviations from a nominal operating point, then the reference $\mathbf{r(t)}$ is zero and the augmented discrete-time state-space model will be

$$\begin{aligned} \begin{bmatrix} \mathbf{x(t+1)} \\ \mathbf{x_i(t+1)} \end{bmatrix} &= \begin{bmatrix} \mathbf{A_d} & \mathbf{0} \\ -\mathbf{C_d} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x(t)} \\ \mathbf{x_i(t)} \end{bmatrix} + \begin{bmatrix} \mathbf{B_d} \\ \mathbf{0} \end{bmatrix} \mathbf{u(t)} + \mathbf{w(t)} \\ \mathbf{y(t)} &= [\ \mathbf{C_d}\ \ \mathbf{0}\ ] \begin{bmatrix} \mathbf{x(t)} \\ \mathbf{x_i(t)} \end{bmatrix} + \mathbf{e(t)} \end{aligned} \tag{10.5}$$

and

$$\mathbf{x_i(t+1)} = [\ \mathbf{0}\ \ \mathbf{I}\ ] \begin{bmatrix} \mathbf{x(t)} \\ \mathbf{x_i(t)} \end{bmatrix} \tag{10.6}$$

where $\mathbf{y} = \Delta P_e$ and $\mathbf{u} = \Delta\beta_{ref}$. Denoting $\bar{\mathbf{x}}(\mathbf{t}) = [\mathbf{x(t)}\ \ \mathbf{x_i(t)}]^{\mathbf{T}}$ and $\mathbf{C_i} = [\ \mathbf{0}\ \ \mathbf{I}\ ]$, the augmented model will be represented by

$$\begin{aligned} \bar{\mathbf{x}}(\mathbf{t+1}) &= \bar{\mathbf{A}}_{\mathbf{d}}\bar{\mathbf{x}}(\mathbf{t}) + \bar{\mathbf{B}}_{\mathbf{d}}\mathbf{u(t)} + \mathbf{w(t)} \\ \mathbf{y(t)} &= \bar{\mathbf{C}}_{\mathbf{d}}\bar{\mathbf{x}}(\mathbf{t}) + \bar{\mathbf{D}}_{\mathbf{d}}\mathbf{u(t)} + \mathbf{e(t)} \\ \mathbf{x_i(t+1)} &= \mathbf{C_i}\bar{\mathbf{x}}(\mathbf{t}) \end{aligned} \tag{10.7}$$

with $\bar{\mathbf{D}}_{\mathbf{d}} = \mathbf{0}$ for a constant speed turbine. The cost function (10.3) will then be modified by

$$\begin{aligned} \mathbf{J(N, x(t))} &= E\{\sum_{j=0}^{N-1} \{Q_y(\Delta P_e(t+j))^2 + Q_u(\Delta\beta_{ref}(t+j))^2 \\ &\quad + \mathbf{x_i^T(t+j)Q_i x_i(t+j)}\}\} \\ &= E\left\{\sum_{j=0}^{N-1} \{\bar{\mathbf{x}}^{\mathbf{T}}\mathbf{(t+j)Q_x}\bar{\mathbf{x}}\mathbf{(t+j)} + Q_u(\Delta\beta_{ref}(t+j))^2\}\right\} \end{aligned} \tag{10.8}$$

where $\mathbf{Q_x} = \bar{\mathbf{C}}_{\mathbf{d}}^{\mathbf{T}} Q_y \bar{\mathbf{C}}_{\mathbf{d}} + \mathbf{C_i^T Q_i C_i}$. The solution of the LQ optimal control problem can be derived by iterating the Riccati Difference Equation (RDE)

$$\begin{aligned} \mathbf{L_j} &= (\bar{\mathbf{B}}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P_j}\bar{\mathbf{B}}_{\mathbf{d}} + Q_u)^{-1}\bar{\mathbf{B}}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P_j}\bar{\mathbf{A}}_{\mathbf{d}} \\ \mathbf{P_{j+1}} &= \bar{\mathbf{A}}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P_j}\bar{\mathbf{A}}_{\mathbf{d}} - \bar{\mathbf{A}}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P_j}\bar{\mathbf{B}}_{\mathbf{d}}(\bar{\mathbf{B}}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P_j}\bar{\mathbf{B}}_{\mathbf{d}} + Q_u)^{-1}\bar{\mathbf{B}}_{\mathbf{d}}^{\mathbf{T}}\mathbf{P_j}\bar{\mathbf{A}}_{\mathbf{d}} + \mathbf{Q_x} \\ &\qquad j = 0, \cdots N-1 \end{aligned} \tag{10.9}$$

and the feedback control law is

$$\mathbf{u(t)} = -\bar{\mathbf{L}}_{\mathbf{N-1}}\bar{\mathbf{x}}\mathbf{(t)} \tag{10.10}$$

Since the state is not directly available for measurement, the approach is to construct an observer and interconnect the control law with the estimated

state

$$\mathbf{u(t)} = -[\mathbf{L} \quad \mathbf{L_i}] \begin{bmatrix} \hat{\mathbf{x}}(\mathbf{t}) \\ \hat{\mathbf{x}}_\mathbf{i}(\mathbf{t}) \end{bmatrix} \tag{10.11}$$

The Kalman estimation method given in section 9.2 can be used to estimate the state.

The control strategy with different weights on the the pitch reference is illustrated by the step responses in Figure 10.6. The step input to the turbine is wind speed which changes from 16 m/s to 17 m/s at 1 sec.

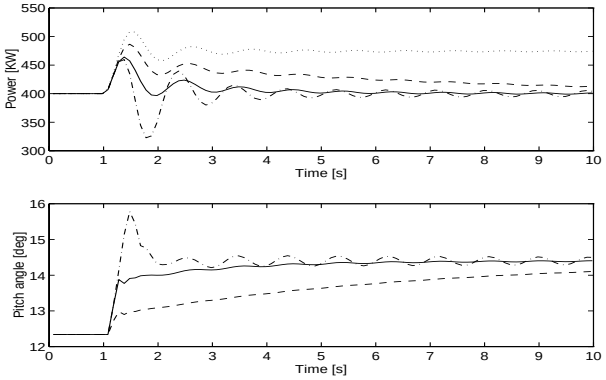

Figure 10.6. Step responses of output power P_e and pitch angle β obtained by LQG controller with different weight on pitch reference. $Q_y = 1$, $Q_u = 4.3 \cdot 10^{11}$ (—), $20 \cdot 10^{11}$ (- -), 10^{10} (-·). The dotted line in the upper plot is the power output in open-loop.

With high weight on the control action the turbine follows the power reference slowly (dashed line). However, the large control action excites the tower bending movements and leads to the vibrations on pitch control action and turbine dynamics (dash-dot line). In practice, the large pitch action is not allowed to avoid heavy mechanical loads. A compromise is made

between power smoothing and load alleviation by choosing $Q_y = 1$ and $Q_u = 4.3 \cdot 10^{11}$. The step response with these weighting parameters is given by the solid line. It can be found that the output power has been suppressed and the amplitude of overshoot is decreased by the LQG regulator compared to the PI-controller with the similar pitch action.

Gain scheduling

An effective control algorithm must reflect both the plant dynamic characteristics as well as the anticipated working environment. The control problem is therefore divided into two time scales corresponding to slow mean wind speed changes and rapid turbulent wind speed variations. Changes in mean wind speed cause the mean pitch angle of the blade to alter to maintain the mean power output at its rated value. These mean pitch positions are treated as steady state operating points, which can be determined by searching scheduling table 9.7 with the average wind speed obtained by filtering the measured point wind speed through a low-pass filter with a period of 10 minutes.

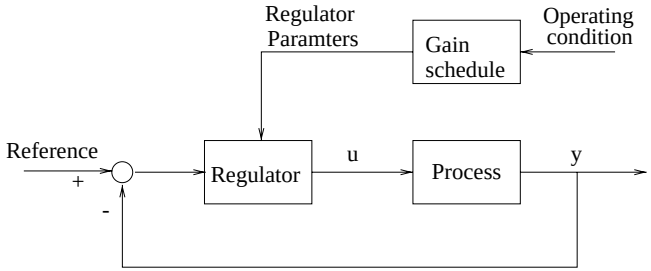

Figure 10.7. Block diagram of a gain scheduling system

Secondly, the aerodynamic behaviour is highly nonlinear. When a linear representation of rotor aerodynamic torque is employed, the partial derivatives $\partial T_w/\partial v$, $\partial T_w/\partial \beta$ and $\partial T_w/\partial \omega_r$ of the aerodynamic torque with respect

to wind speed, pitch angle and rotor speed are actually time-varying and depend on the operating conditions. Since we have known how the rotor aerodynamics change with the operating conditions of the process, it is then possible to change the parameters of the controller by monitoring the operating conditions. This is called gain scheduling. The principle of gain scheduling is illustrated in Figure 10.7. The gain scheduling has a linear regulator whose parameters are changed as a function of operating conditions in an open-loop fashion.

For a wind turbine, the estimated wind speed or the average wind speed can be used as scheduling variables to monitor the operating conditions of the process. The scheduling table is plotted in Figure 9.7 in Chapter 9. The parameters $\partial T_w/\partial v$, $\partial T_w/\partial \beta$ and $\partial T_w/\partial \omega_r$ in the linear aerodynamic torque model can be found by searching scheduling table with the estimated wind speed or the mean wind speed at each sample. This means that the linear model used in estimator and regulator is calibrated for each operating point and the nonlinear aerodynamics can then be compensated.

Simulation results

Simulation is carried out at the same operating point as the PI-controller, and the same wind sequence is used for the simulation. The simulation results can thus be compared with the results obtained by using the PI-controller. In order to make comparisons, the similar control actions are required and maximum value of the pitch rate is limited to about $5^o/s$ for both the PI and the LQG control. The LQG controller with weighting parameters $Q_y = 1$ and $Q_u = 4.3 \cdot 10^{11}$ is performed, which gives the similar pitch action as the PI-controller does. The gain scheduling is not implemented in this example.

The simulation results are shown in Figure 10.8, and key parameters from the simulation are given in Table 10.2.

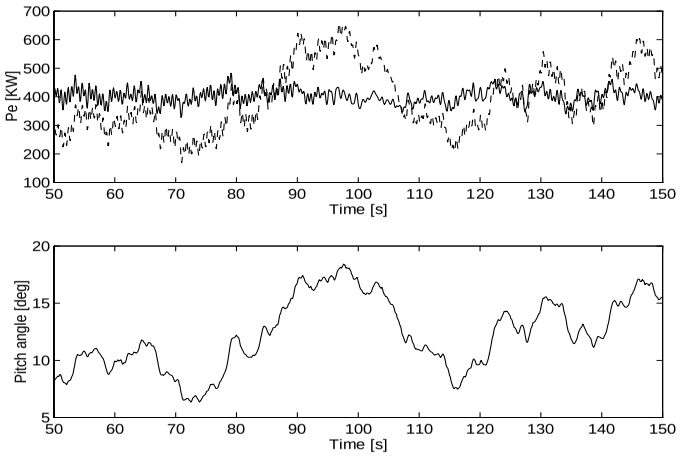

Figure 10.8. Simulation of the wind turbine with a LQG-controller. The upper plot shows the power output from the uncontrolled (dashed) and controlled (solid) wind turbine. The lower plot shows the pitch angle of the blades.

mean(P_e) [KW]	mean(T_g) [Nm]	mean(ω_g) [rad/s]	max($h_t\omega_t$) [cm]
399.16	3927.07	105.77	3.74
SD(P_e) [KW]	SD(T_g) [Nm]	SD(ω_g) [rad/s]	max($h_t\theta_t$) [cm]
24.75	241.11	0.06	3.45
max(β) [deg]	min(β) [deg]	SD(β)	max(β) [deg/s]
19.43	5.03	3.34	5.05

Table 10.2. Statistics obtained while simulating the LQG pitch control system. Mean wind speed is 16 m/s.

It can be found that the standard deviation of output power is reduced by the LQG controller compared to the PI-controller. This corresponds to the decrease of the variations in generator reaction torque. By keeping the generator torque variations small, the shaft loads are reduced and thereby the turbine life will be increased. As the PI-controller, the LQG controller does not give the significant increases in tower bending movements.

10.3 Combined variable speed and pitch control

The purpose of this section is to investigate the control system of a variable speed wind turbine which employs pitch regulation and generator reaction torque regulation in high wind speeds.

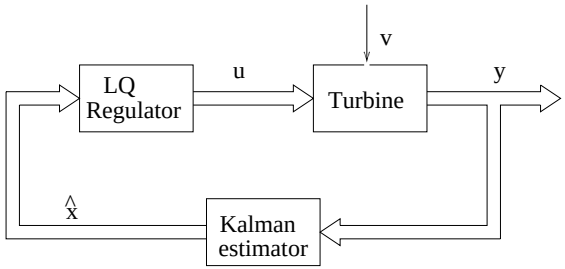

Figure 10.9. The LQG control for the wind turbine

Variable speed pitch regulation presents a multivariable control problem as an additional generator torque control loop is added to the existing pitch control loop. The ability to vary the rotor speed of a wind turbine increases the operational flexibility. Figure 10.9 shows the block scheme of combined

variable speed and pitch control of a wind turbine by implementing the LQG control method.

The dynamics of the generator and frequency converter have been discussed in section 7.6. The linear design model for a variable speed wind turbine given in Chapter 8 has the pitch reference and generator torque reference as inputs and the angular velocity and reaction torque of the generator as outputs

$$\mathbf{u} = \begin{bmatrix} \Delta\beta_{ref} \\ \Delta T_{g,ref} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} \Delta\omega_g \\ \Delta T_g \end{bmatrix} \tag{10.12}$$

The model is augmented by including integral actions to give zero steady-state errors. The model can be represented by (10.7). The performance criterion is speed wind turbine

$$\begin{aligned} \mathbf{J(N, x(t))} &= E\{\sum_{j=0}^{N-1}\{\mathbf{y^T(t+j)Q_y y(t+j)} \\ &\quad \mathbf{+u^T(t+j)Q_u u(t+j) + x_i^T(t+j)Q_i x_i(t+j)}\}\} \qquad (10.13) \\ &= E\{\sum_{j=0}^{N-1}\{\mathbf{\bar{x}^T(t+j)Q_x \bar{x}(t+j) + 2\bar{x}^T(t+j)Q_{xu} u(t+j)} \\ &\quad \mathbf{+u^T(t+j)Q'_u u(t+j)}\}\} \qquad (10.14) \end{aligned}$$

where matrices $\mathbf{Q_x} = \mathbf{\bar{C}_d^T Q_y \bar{C}_d + C_i^T Q_i C_i}$, $\mathbf{Q_{xu}} = \mathbf{\bar{C}_d^T Q_y \bar{D}_d}$ and $\mathbf{Q'_u} = \mathbf{\bar{D}_d^T Q_y \bar{D}_d + Q_u}$, and

$$\mathbf{Q_y} = \begin{bmatrix} q_{y1} & 0 \\ 0 & q_{y2} \end{bmatrix} \quad \mathbf{Q_u} = \begin{bmatrix} q_{u1} & 0 \\ 0 & q_{u2} \end{bmatrix} \quad \mathbf{Q_i} = \begin{bmatrix} q_{i1} & 0 \\ 0 & q_{i2} \end{bmatrix} \tag{10.15}$$

The choice of the weighting matrices $\mathbf{Q_y}$ and $\mathbf{Q_u}$ depend on the relative importance of the output power quality or input restrictions. q_{u1} and q_{u2} in matrix $\mathbf{Q_u}$ can be chosen such that the the pitch position and pitch rate should be restricted and the rotor speed will be remained within predefined limitation. $\mathbf{Q_i}$ will be chosen to punish the stationary error.

The step responses of the output power P_e, pitch position β and angular velocity of generator ω_g obtained by variable speed pitch control are given in Figure 10.10. The weighting matrices used in simulation is

$$\mathbf{Q_y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{Q_u} = \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{Q_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{10.16}$$

Different weights ($a = 10^4$ and $2 \cdot 10^5$) on the pitch action are chosen to show the effect of pitch action. The step input to the turbine is wind speed which changes from 16 m/s to 17 m/s at 1 sec.

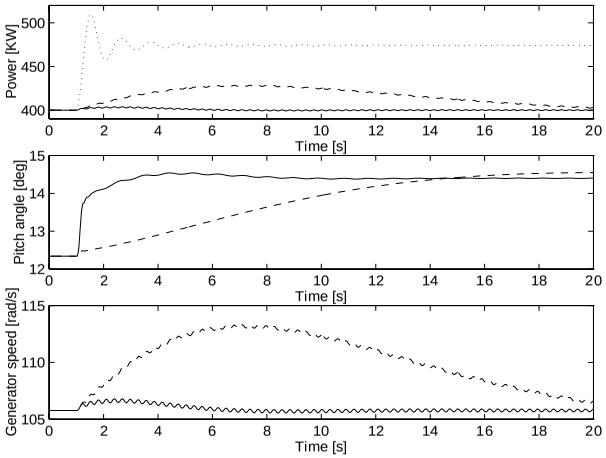

Figure 10.10. Step responses of output power, pitch angle and generator speed obtained by variable speed pitch control with different weight on pitch action. $q_{u1} = 10^4$ (—), $2 \cdot 10^5$ (- -). The dotted line in the upper plot is step response of output power in open-loop.

The step responses show that both control actions make contributions to the output performance. The output power is suppressed strongly by the variable speed operation and tracking performance is greatly improved by

the extra control action. With lower weight on the pitch action (the solid line), the rotor speed variations are prevented from being too large, and overshoot of the power output has been eliminated. With higher weight on pitch action (the dashed line), the pitch responses slowly, and the extent of rotor variations is large. The oscillations in the generator speed are caused by lightly damped drive train.

The performance of the variable speed control system is assessed by simulations. The wind sequence used to drive the system is the same as it is used to test the PI- and LQG-controller at a mean of 16 m/s. The combined variable speed and pitch control is performed by the LQG control law with weighting matrices

$$\mathbf{Q_y} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{Q_u} = \begin{bmatrix} 10^4 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{Q_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{10.17}$$

The results of the simulation are shown in Figure 10.11. The corresponding statistic results from simulation of 200 sec are given in Table 10.3.

Figure 10.11 shows a slow control action on the pitch angle and a fast control action on the generator reaction torque. The pitch control action keeps the actuator activity within acceptable limits and prevents the rotor speed variations becoming to large. Here the maximum value of the pitch rate is limited to about $5^o/s$ in order to compare with the performance of the controller for the constant speed wind turbine. The maximum deviation of generator shaft speed from its nominal value is 3.4% of nominal. The rotor speed of the turbine is actually restricted to a narrow band.

The statistic results indicate that significant reductions in standard deviations of the output power and generator reaction torque are given by the variable speed wind turbine compared to the constant speed wind turbine.

It means that the rapid fluctuations in output power which lead to the fatigue of the turbine components have been smoothed.

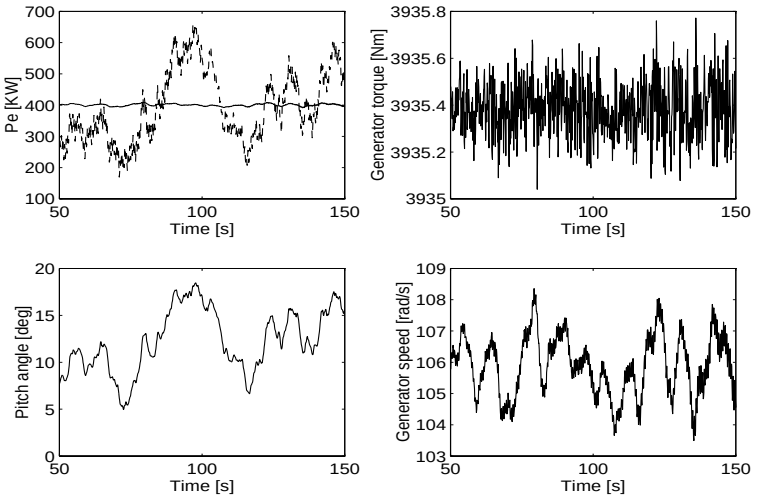

Figure 10.11. Simulation of the variable speed wind turbine.

mean(P_e) [KW]	mean(T_g) [Nm]	mean(ω_g) [rad/s]	max($h_t\omega_t$) [cm]
399.73	3935.38	105.70	4.18
SD(P_e) [KW]	SD(T_g) [Nm]	SD(ω_g) [rad/s]	max($h_t\theta_t$) [cm]
3.97	0.1143	1.05	4.09
max(β) [deg]	min(β) [deg]	SD(β)	max($\dot{\beta}$) [deg/s]
19.59	0.01	3.77	5.04

Table 10.3. Statistics obtained from simulating the variable speed pitch control wind turbine. Mean wind speed is 16 m/s.

The research studies reveal that the fast torque control has been used to alleviate the transient. When the wind speed rises above rated, the varying

rotor speed will alleviate the drive train torque changes. The slower pitch control has been used to minimize the long term drifts in wind speed and keep the rotor speed excursions within limits.

Due to the indirect grid connection, the drive train is lightly damped which may result in mechanical vibrations. The vibrations can be seen from the small oscillations in generator speed in Figure 10.11, which consequently induce the fluctuations on the rotor speed.

10.4 Summary

The most important results obtained from the simulations of the wind turbine with different controllers are illustrated in Figure 10.12. The values are plotted relative to the constant speed PI pitch controlled (CS PI) wind turbine.

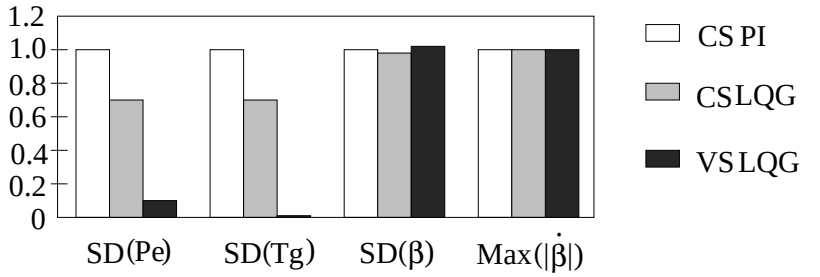

Figure 10.12. Chart for comparison (The comparison is relative to the CS PI-controlled wind turbine.

From the investigations of the constant speed and variable speed wind turbine with different control methods, the following conclusions can be made.

The PI pitch control is still a favoured means of regulation for constant speed wind turbines because of the simplicity of the method.

The LQG pitch control achieves a significant reduction in the extent of the transients of drive train load and hence smoothes the power. The investigations indicate that the LQG control is more appropriate than the PI control. However, the simulation results are based on an accurate model which may not always be obtained in practice. For the LQG control method, a state estimator must be employed in any practical implementation.

For a variable speed wind turbine, the system to be controlled is multivariable with inputs being the reference generator torque and pitch demand and outputs being the generator reaction torque and generator speed. Considerable improved performance is provided by performing the LQG variable speed pitch control. The ability to operate at variable rotor speed effectively adds compliance to the power train dynamics of the wind turbine and achieves a more gentle pitch motion regime. The pitch action in variable speed operation is used to prevent the rotor speed variations from being too large. The rotor speed is actually free to vary within a small interval.

The combined variable speed pitch control can also be realized by using the PI control method. Two control loops with the pitch reference and generator torque reference as inputs will be included in the control scheme. The PI multivariable control is not discussed in this thesis.

Another advantage of the variable speed operation is that rotor speed can be adjusted to reduce noise levels and hence make the wind turbine more environmentally agreeable, because the noise level produced by wind turbine is a function of the rotor speed.

The benefits of using variable speed wind turbine must be offset against the disadvantages: the more expensive generator system due to the introduction of power conversion device and reduced reliability of the variable speed electrical machinery. However, it is anticipated that the cost of power conversion device will decrease as technology advances.

As wind speed rises, for a fixed controller the standard deviation of the power output also rises due to the increased level of turbulence. It is therefore attractive to increase the controller activity by applying gain scheduling method which gives improved disturbance rejection. The gain scheduling method is in an open-loop fashion, the possibly occurred steady state errors of operating points can be eliminated by the integral actions.

Chapter 11

Control Below Rated Power

It was mentioned previously that the wind turbine operation was divided into two regions, i.e., above rated wind speed and below rated wind speed. Since a wind turbine spends most of its life operating in wind speeds less than rated, it seems more important to investigate the role and effects of control systems in partial load.

Below rated wind speed the control strategy is to maximize energy capture from the wind and minimize the turbine transient loads. In partial load it is the aerodynamic power that should be maximized, since the energy is captured by the rotor. However, the overall production of electrical energy is improved, since the captured energy is only temporarily stored in rotor.

The rotor characteristics of a wind turbine are summarized by the C_p surface, where the power coefficient $C_p(\lambda, \beta)$ defining the ability of the turbine to convert the kinetic energy of the wind to mechanical energy captured by the rotor is a nonlinear function of pitch angle β and tip speed ratio λ. In order to maximize the energy capture, these two variables should be kept at their optimal values to achieve the maximum value of C_p. The pitch angle can be easily fixed at the optimal value. However, since the tip speed ratio defined by $\lambda = v/(\omega_r R)$ is affected by the wind speed experienced by the turbine and the rotor speed, the rotor speed is the only way which can be regulated to achieve the optimal λ. Hence, a variable speed turbine is required to track the variations of wind speed and hence maximize the aerodynamic power.

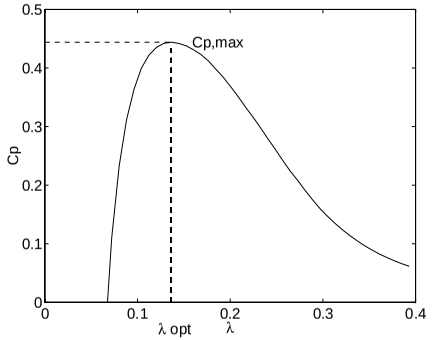

Figure 11.1. $Cp(\lambda)$-curve at pitch angle of 0.5^o

A $C_p(\lambda, \beta_0)$-curve of the WD34 wind turbine at optimal pitch angle of $\beta_0 = 0.5^o$ is shown in Figure 11.1. The rotor develops maximum mechanical power only at specific tip-speed ratio λ_{opt}. If the rotor speed is held constant, the operating point varies on the $C_p(\lambda)$-curve from cut-in to rated wind speed, resulting in reduced efficiency. On the other hand, if rotor speed is allowed to vary, it can track the wind velocity to keep the tip-speed ratio at its optimum value.

Since the sensitivity of the aerodynamic torque to the change of the pitch angle is lower at low wind speeds, no control action on the pitch angle will then be required normally.

With the pitch angle at its optimal value, the control objective is to maximize the power coefficient at all wind speeds by keeping the tip speed at its optimal value. On the other hand, the control objective is also optimized with respect to the minimization of the turbine transient loads. Obviously, the two objectives mentioned above contradict each other. In order to keep the tip speed ratio constant, the rotor speed must exactly follow the wind turbulence, which requires large variations in rotor speed, consequently, increases the transient loads. Therefore a trade-off must be made between two objectives. How close it is desirable to track the optimal tip speed ratio is related to the permissible loads. Below rated power a variable speed wind turbine is a single input single output system.

However, the energy output of the wind turbine can be increased by variable speed operation depending on wind velocity, since the constant speed wind turbine is normally operated more effectively for a single wind speed. The increase in energy capture for a turbine is drawn in Figure 11.2. The hatched area is the increased energy capture by a variable speed turbine. It has been pointed out by [Ernst and Leonhard, 1988] that the annual energy production of wind turbines increased by variable speed operation is $5\% - 10\%$ compared with the output of constant speed turbines.

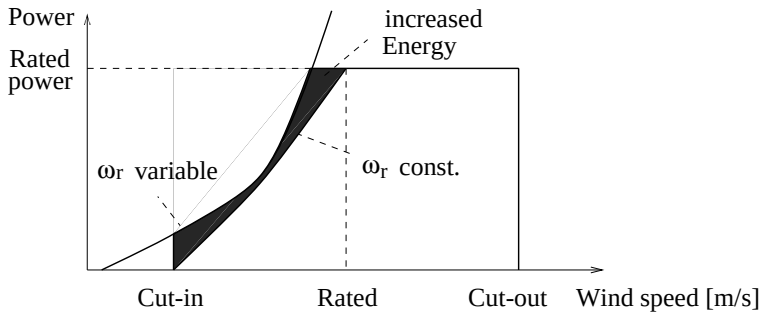

Figure 11.2. Energy capture capability.

It should be noted that the increased energy is in the aerodynamic power. The frequency converter will introduce some losses. However, it will not be considered in this project.

In this chapter two control schemes will be investigated, i.e., the LQG speed control and the tracking control, which are given in section 11.1 and 11.2 respectively. Some implementation considerations are given in section 11.3. A summary is made in section 11.4.

11.1 LQG speed control

Control without pitch action through the generator reaction torque can be implemented for below rated operation. The turbine in this case is caused to track a predefined optimal tip speed ratio as close as possible. The aim of control is to cause the wind turbine to extract energy from the wind as efficiently as possible, and meanwhile reduce the transient loads.

Obviously, there is a contradiction between two control objectives and it is necessary to choose a suitable compromise. A trade-off can be made by the LQG control method. The two objectives are taken into account by minimizing the performance function

$$J(N,t) = E\left\{\sum_{j=0}^{N-1}\{Q_1\Delta P_r(t+j) + Q_2(\Delta T_{g,ref}(t+j))^2\}\right\} \tag{11.1}$$

where ΔP_r is the power loss due to not keeping the optimal tip speed ratio. $\Delta T_{g,ref}$ is the deviation of generator torque reference from its optimal value. The trade-off between two control objectives can be realized by suitably choosing the parameters Q_1 and Q_2. If the scalar weighting parameter Q_2 is relatively small, the controller will emphasis on the maximization of power capture and will use large torque variations to realize it. With relatively high value of Q_2, the reduction of load variations becomes more important.

When the pitch angle is fixed at its optimal value, the aerodynamic power loss can be written as

$$\Delta P_r = \frac{1}{2}\pi\rho R^2 v^3 \Delta C_p \tag{11.2}$$

and

$$\Delta C_p = C_{p,max}(\lambda_{opt}) - C_p(\lambda) \tag{11.3}$$

The $C_p(\lambda)$ curve can be approximated by a second order polynomial when λ is close to the optimal [Ekelund, 1994a]

$$C_p(\lambda) = C_{p,max} - a_0(\lambda - \lambda_{opt})^2 \tag{11.4}$$

where a_0 is sensitivity of the quadratic curve, λ_{opt} is the optimal tip-speed ratio where C_p has maximum value, and $\lambda - \lambda_{opt} = \Delta\lambda$ is the tracking error. With this assumption the power loss will be a function of $\Delta\lambda$

$$\Delta P_r = \frac{1}{2}\pi\rho R^2 v^3 a_0(\Delta\lambda)^2 \tag{11.5}$$

This implies that the power loss is actually determined by the deviation of tip-speed ratio $\Delta\lambda$ from its optimal value. The performance function (11.1) can then be modified by

$$J(N,t) = E\left\{\sum_{j=0}^{N-1}\{Q_1(\Delta\lambda(t+j))^2 + Q_2(\Delta T_{g,ref}(t+j))^2\}\right\} \tag{11.6}$$

Since the tip speed ratio can be linearized at an operating point

$$\Delta\lambda = \left.\frac{\partial\lambda}{\partial\omega_r}\right|_{op}\Delta\omega_r + \left.\frac{\partial\lambda}{\partial v}\right|_{op}\Delta v = \mathbf{Hx} \tag{11.7}$$

where

$$\mathbf{H} = [\,0\ \ \frac{\partial\lambda}{\partial\omega_r}\ \ 0\ \ 0\ \ \frac{\partial\lambda}{\partial v}\,]$$

$$\mathbf{x} = [\,\Delta\theta_\epsilon\ \Delta\omega_r\ \Delta\omega_g\ \Delta\dot{v}\ \Delta v\,]^T$$

which leads to the criterion

$$\mathbf{J(N,x(t))} = E\left\{\sum_{j=0}^{N-1}\{\mathbf{x^T(t+j)Q_x x(t+j)} + Q_2(\Delta T_{g,ref}(t+j))^2\}\right\} \tag{11.8}$$

where $\mathbf{Q_x} = \mathbf{H^T}Q_1\mathbf{H}$. $\mathbf{x}$ is the state vector of the composite state-space model in partial load. The model has been given in Chapter 8.

The LQ control law is implemented by combining the Kalman filter to estimate the state vector

$$\mathbf{u(t)} = \mathbf{-L\hat{x}(t)} \tag{11.9}$$

The step responses of generator torque T_g and tip speed ratio λ with different values of weighting parameter Q_2 are given in Figure 11.3 and Figure 11.4 respectively, where the wind speed changes from 9 m/s to 10 m/s at 1 sec. The 3P-effect is not included in simulation model for the step responses. The figures show that a large control action provides a small steady-state error of tip speed ratio. The optimal value of tip speed ratio is 0.1357.

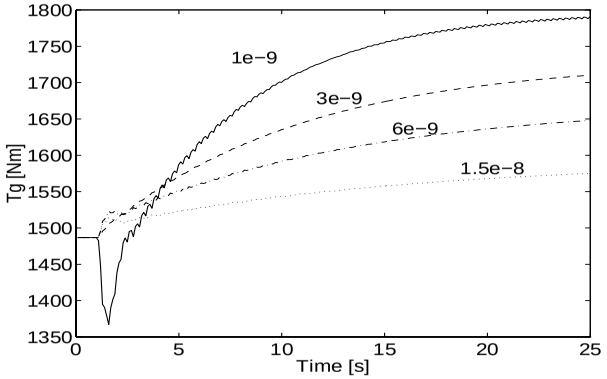

Figure 11.3. The step response of generator torque T_g with different values of Q_2 while $Q_1 = 1$

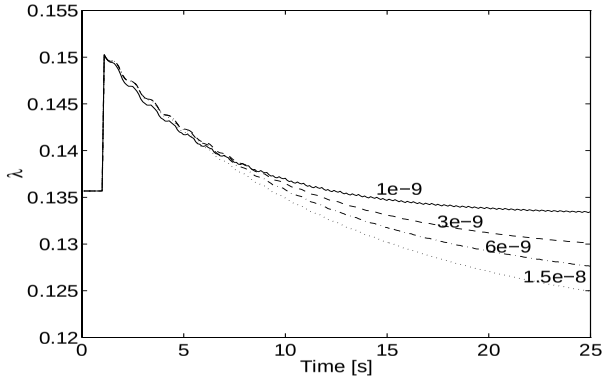

Figure 11.4. The step response of tip speed ratio λ with different values of Q_2 while $Q_1 = 1$

With a lower weighting parameter Q_2 the control action will make the turbine to accelerate fast to track the wind speed, which leads to that the generator torque will be decreased first and use the increased power of the wind to speed up the turbine. Therefore a part of the aerodynamic power is stored in the turbine inertia and will be transferred to the generator when the wind speed decreases [Ekelund, 1994a].

Gain scheduling is also necessary for below rated operation. However, since the wind speed variations are lower compared to the above rated operation, the mean wind speed can be taken as a scheduling variable to determine the operating point.

The performance of the LQG controller is evaluated by simulations. The wind sequence used to drive the system in simulations is at average value of 7 m/s, which is shown in Figure 11.5.

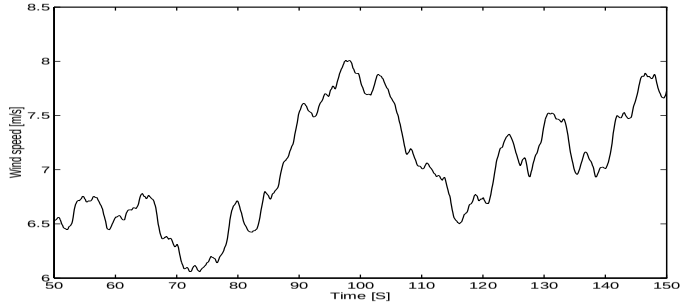

Figure 11.5. Wind sequence at average value of 7m/s

The weighting parameters of the LQG controller are $Q_1 = 1$ and $Q_2 = 1.5e-8$. Figure 11.6 shows the output electrical power P_e, C_p value, generator torque T_g and generator speed ω_g from the closed-loop system. The

corresponding statistic results from simulation of 200 sec are given in Table 11.1. Both the results of the variable speed (VS) wind turbine with LQG control and constant speed (CS) wind turbine are presented. For a constant speed wind turbine the pitch angle is fixed at its optimal value and no control action is available.

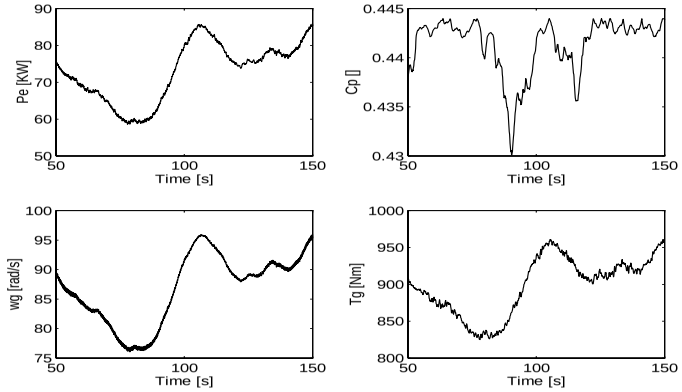

Figure 11.6. Simulation with the LQG controller in partial load.

	mean(T_g) [Nm]	mean(ω_g) [rad/s]	mean(P_e) [KW]
CS	797.11	105.90	78.74
VS	934.32	92.47	80.92
	SD(T_g) [Nm]	SD(ω_g) [rad/s]	mean(P_r) [KW]
CS	218.12	0.32	91.17
VS	53.16	8.11	93.74
	mean(C_p)	max($h_t\omega_t$) [cm]	max($h_t\theta_t$) [cm]
CS	0.4246	0.44	2.36
VS	0.4399	0.42	2.38

Table 11.1. Statistics obtained from simulation of the wind turbine with the LQG controller in partial load. Mean wind speed is 7 m/s.

The simulation results indicate the energy increased by variable speed operation is about 2.8% compared to the output of the constant speed turbine. However, even with the little increase of energy, the mean value of Cp is rather close to its optimal value of 0.4440. The reason is that the mean value of C_p for the constant speed turbine at this operating point is 0.4246, which means there is not so much extra energy that can be captured by variable speed operation. However, the energy output of the wind turbines can be increased by variable-speed operation depending on wind velocity. For the WD34 constant speed wind turbine, when the wind speed is about 8.5 m/s, the tip-speed ratio is rather close to its optimal value. The increased energy by variable speed operation near this wind speed is very little compared to the constant speed wind turbine. However, the captured energy will be increased when the wind speed leaves this value.

	mean(T_g) [Nm]	mean(ω_g) [rad/s]	mean(P_e) [KW]
CS	454.20	105.94	44.89
VS	695.16	78.74	51.28
	SD(T_g) [Nm]	SD(ω_g) [rad/s]	mean(P_r) [KW]
CS	138.37	0.37	51.88
VS	48.98	5.81	59.36
	mean(C_p)	max($h_t\omega_t$) [cm]	max($h_t\theta_t$) [cm]
CS	0.3784	0.30	1.89
VS	0.4409	0.28	1.70

Table 11.2. Statistics obtained from simulation of the wind turbine with the LQG controller in partial load. Mean wind speed is 6 m/s.

Another simulation is carried out at the average wind speed of 6 m/s and the same weighting parameters of the LQG controller are used. Some key results from simulations of the constant speed and variable speed wind turbine with the same wind sequence are given in Table 11.2. It can be easily calculated

that at this operating point the increased energy obtained by variable speed operation is about 14.4 % compared to the constant speed turbine.

With variable speed, the torque transients are smoothed out. The turbine moment of inertia acts as a low-pass filter. However, with large inertia, the angular velocity of rotor can not follow the wind speed variation quickly, which will lead to a tracking error.

Figure 11.6 shows the fast oscillations in generator torque which is caused by the disturbances from the tower shadow. The oscillations induce the mechanical vibrations on the rotor shaft.

11.2 Tracking control

In this section the optimal control problem will be transformed to a tracking problem, in which the optimal tip speed ratio will be maintained by tracking the rotor speed reference.

In order to determine the optimal tip speed ratio λ_{opt}, the wind speed experienced by the turbine is required to be known. However, since a direct measurement of the effective wind speed is not possible, the wind speed must be estimated from measurements made on the wind turbine itself. The operating point is then determined from this estimate. The problem of wind speed estimation has already been discussed in Chapter 9. Hence, a controller can be implemented to track the λ_{opt} according to the estimated wind speed.

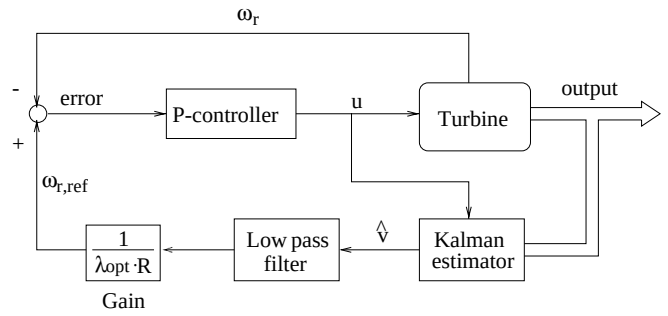

Figure 11.7. The tracking controller

A tracking controller scheme with the Kalman estimator is presented in Figure 11.7. Using the estimated wind speed, the rotor speed reference can be found based on the knowledge of the optimal tip speed ratio

$$\omega_{r,ref} = \frac{\hat{v}}{\lambda_{opt} R} \tag{11.10}$$

In addition to the generator shaft speed and the electrical power measurements, the measurements of the rotor speed are required. The tracking error, between the rotor speed measurement and rotor speed reference, is an input to a simple proportional controller. A low-pass filter is used to eliminate the high frequency disturbances from the wind gusts and tower shadow.

The design parameters for the tracking controller depicted in Figure 11.7 are the gain of the proportional controller and the cut-off frequency of the low-pass filter, which determine how close the controller can track the optimal tip speed ratio and make the trade-off between the energy capture and dynamic loads.

An alternative method is based on the observation that tracking optimal tip-speed ratio can be reformulated as a problem in the torque-speed plane. The possibility is to use a measured quantity of the drive train, normally the rotor speed as an indirect measure of the instantaneous wind speed. [Ekelund, 1994b], [Novak and Ekelund, 1994] and [Novak et al., 1995] use the generator speed and torque. The controller scheme is given in Figure 11.8.

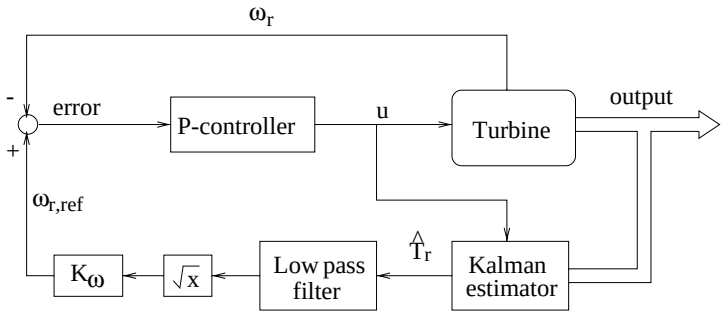

Figure 11.8. The tracking controller

In Figure 11.8, the aerodynamic torque, rather than the wind speed, is estimated using the linearized aerodynamic torque expression. Since we have the relation for the aerodynamic torque

$$T_{r,max} = \frac{1}{2}\pi\rho R^5 \lambda_{opt}^3 C_{p,max} \omega_{r,opt}^2 \tag{11.11}$$

The rotor speed reference can then be calculated from

$$\omega_{r,ref} = \sqrt{\frac{1}{\frac{1}{2}\pi\rho R^5 \lambda_{opt}^3 C_{p,max}}}\sqrt{\hat{T}_r} = K_\omega \sqrt{\hat{T}_r} \tag{11.12}$$

The reference signal will be compared with the measured one. The error signal will be taken as an input to the proportional controller which drive

the turbine to track speed reference. The method introduce the approximation that the rotor speed is maintained at its optimal value without any deviation.

In both schemes, if it is difficult to measure rotor speed, it can be approximated based on the measurement of the generator speed

$$\omega_r = \frac{\omega_g}{n_{gear}} \tag{11.13}$$

The step responses of the generator torque and tip speed ratio with different gains of the proportional controller for the first control scheme (Figure 11.7) are given in Figure 11.9 and Figure 11.10, where the time constant of the low-pass filter is set to 0 sec. The driving wind speed increases from 9 m/s to 10 m/s at 1 second.

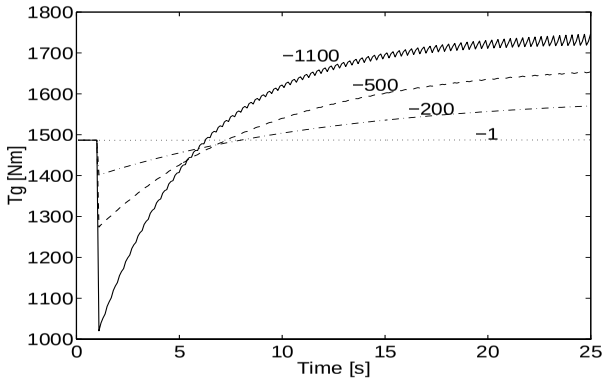

Figure 11.9. The step response of generator torque T_g with different gain of proportional controller.

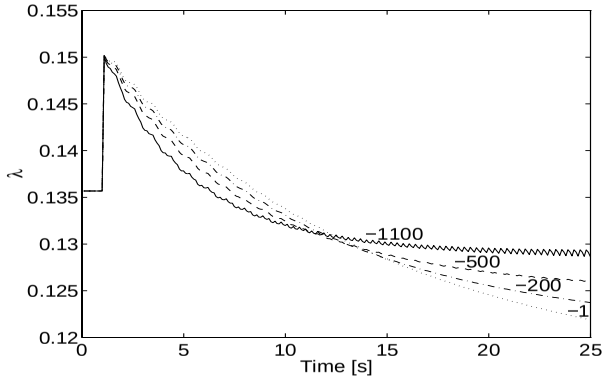

Figure 11.10. The step response of tip speed ratio λ with different gain of proportional controller.

The step responses show that the tracking controller has the similar performance as the LQG controller has. As the gain of proportional controller increases, the steady-state error will be decreased. However, the best value of the tip speed ratio achieved by the tracking controller is lower than it is achieved by the LQG controller. With the largest proportional gain of the controller (-1100), the performance has begun to diverge. The large penalty on the power loss will reduce the stability of the control system.

The original nonlinear system is simulated using both control schemes in which the low-pass filter is a second order Butterworth low-pass filter. The two tracking controllers show the similar performance. Figure 11.11 is the simulation results obtained by the first control scheme, where the gain of proportional controller is -100 and the cut-off frequency of the second order Butterworth low-pass filter is 3 rad/s. The simulations use the same wind sequence as the simulations in previous section.

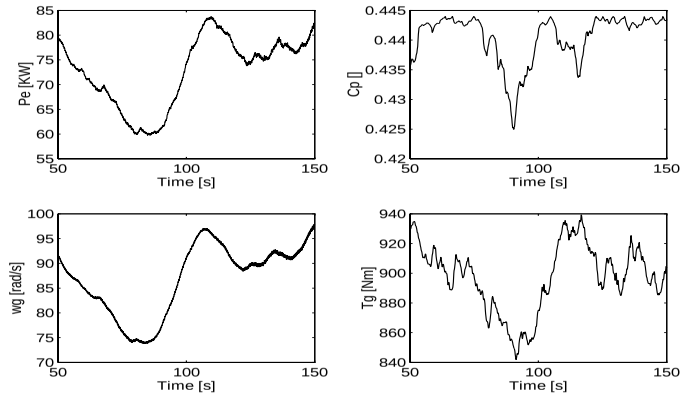

Figure 11.11. Simulation with the tracking controller in partial load.

	mean(T_g) [Nm]	mean(ω_g) [rad/s]	mean(P_e) [KW]
6 m/s	672.20	80.57	50.58
7 m/s	913.60	94.09	80.30
	SD(T_g) [Nm]	SD(ω_g) [rad/s]	mean(P_r) [KW]
6 m/s	23.61	7.75	59.03
7 m/s	29.47	10.07	93.26
	mean(C_p)	max($h_t\omega_t$) [cm]	max($h_t\theta_t$) [cm]
6 m/s	0.4383	0.27	1.74
7 m/s	0.4376	0.42	2.43

Table 11.3. Statistics obtained from simulation of the wind turbine with the tracking controller in partial load. Mean wind speed is 7 m/s.

The output power P_e, generator speed ω_g, generator reaction torque T_g and C_p value obtained from the simulation are given in Figure 11.11. Some key parameters from the simulation are given in Table 11.3 which shows that the tracking controller has the similar performance as the LQG controller. The captured power by variable speed operation is increased 2.3% at average wind speed of 7 m/s and 13.8% at average wind speed of 6 m/s, which is little less than it is achieved by the LQG controller with similar control action. However, the tracking controller seems more effective to filter out the fast variations in the generator torque reference by the low-pass filter in the control scheme, which consequently reduces the transient loads.

11.3 Implementation of control system

Since the wind frequently varies from below to above rated wind speed and vice versa, a fundamental requirement for the wind turbine is a smooth changes between the two control schemes. This is the most important implementation issue encountered when developing controllers for wind turbines.

For a pitch regulated constant speed wind turbine, below a certain rated wind speed, the generated power is less than the turbine rating and no control action is required. When the wind speed rises above rated, the power output is regulated at the rated power value by adjusting the pitch angle of the rotor blades. Hence it is necessary to start up and shut down the controller automatically as the wind speed fluctuates. Consequently, the design of switching from below rated to above or vice versa should be treated with some care to avoid prolong transient and to minimize the loads on the wind turbine.

The transients associated with the wind turbine controller start-up may be attributed to some form of windup within the controller when operating below rated. Therefore start-up strategy for reducing the switching transient is considered within anti-windup context.

When the wind speed falls below rated, the actuator will be saturated, the control feedback loop will be broken and plant will no longer respond to the controller. However, the pure integrator in the controller, introduced to induce the correct steady state error behaviour, continues to be driven by the control error and its output increases in magnitude or windup. Consequently, when the controller returns to unsaturated operation, the transient will be increased which can lead to a loss of performance and instability.

A common switching approach to the controller is simply to freeze controller integral action when below rated operation is detected, i.e., when the pitch angle demand falls below a specified threshold value, the accumulation sum in the integral calculation is suspended. This approach is implemented in this project and prevents the pure integrator in the controller from adopting an inappropriate state during below-rated operation. However, it has been indicated by [Leith and Leithead, 1997] that the approach is not always effective in preventing large start-up transients and the poor performance will occur when there is low-frequency dynamics in the controller.

[Leithead et al., 1992] proposes an alternative start-up technique whereby a minor feedback loop is introduced within the controller which switches in to permit the controller to continue operating below-rated wind speed. The minor feedback loop mimics the action of the physical wind turbine through the inclusion of a transfer function which models its dynamics.

The dynamics of the wind turbine drive train introduce a lag between changes in the wind speed and corresponding changes in power output, i.e., when the wind speed rises above rated, there is a lag before power output responds and control action is resumed. [Leith and Leithead, 1997]

indicates that the switching performance might be improved by anticipating the transient from below-rated to above rated power generation and thereby compensating for the dynamics of the drive train. It was suggested to achieve the predictive action by including a filter with suitable phase lead combined with the minor feedback loop.

For a pitch regulated variable speed wind turbine, switching from speed control via reference torque in below rated wind speed to speed control via pitch demand and generator torque control via reference torque in above wind speed, results in control errors having discontinuous. The various controller terms must be implemented in such a way that these discontinuous do not affect the performance of the system.

[Leithead and Connor, 1994] gives a switching approach to the controller for a variable speed pitch regulated wind turbine based on the minor feedback loop. Below rated wind speed, the generator torque reference acts on the error in rotor speed via the controller. The minor feedback is operational and pitch demand is set to a predefined optimal value. Above rated wind speed, the minor of feedback is inoperative, both pitch angle and torque reference act on the error in power via the controller. Switching between below and above rated wind speed is activated by pitch demand changing sign.

The topic of switching between different control schemes is not discussed in this thesis.

11.4 Summary

There are two fundamental types of wind turbines, namely, constant speed wind turbines and variable speed wind turbines. In partial load it is not feasible to achieve maximum power with a constant speed wind turbine, since the constant speed wind turbine makes it impossible to control the tip speed ratio. A straightforward method of regulating the turbine below rated power is by active feedback control based on the rotor speed.

The ability to vary the rotor speed of a wind turbine offers several advantages below rated wind speed. Firstly, the captured energy from the wind depends on the tip-speed ratio λ, the variable speed wind turbines can be used to regulate the rotor speed so that λ will attain (or be in the neighborhood of) its optimal value and hence optimize energy capture. Secondly, a variable speed turbine offers the additional power train compliance and associated loads alleviation.

The LQG controller and the tracking controller seem to be performing similarly. However, if the main concern is focussed on power maximization, the LQG control scheme has a better performance than the tracking control scheme. On the other hand, the tracking controller is more effective to minimize the load transients. Since both controllers are based on a linearized model at a specific working point, a gain scheduling is required to compensate for nonlinearity in the process and the changes of operating point.

However, the LQG and tracking controller lead to maximum power output only when the controller gain is correctly adjusted, i.e., the optimum tip-speed ratio, the maximum value of C_p-curve and turbine dynamics are known. Some of these parameters may change during operation, which will lead to power loss.

An another drawback is the poor dynamic response. Due to the large inertia of the rotor, the angular speed of rotor cannot follow fast wind speed changes. As a consequence, the mean C_p-value drops slightly below its maximum. It has been pointed out by W. E. Leithead in [Leithead, 1989b] that the dynamics of a wind turbine prevent the full benefits of variable speed operation being realized and the control action cannot operate sufficiently fast to cope with fast changes in wind speed. The effect of turbulence and dynamics is to displace the operating state of the turbine from the $C_{p,max}$ curve. As the displacement of the state from $C_{p.max}$ curve increases the energy capture decreases. The extent of this reduction is dependent on the dynamics of the wind turbine, the quality of the control system and degree of turbulence of the wind.

Chapter 12

Summary and Conclusions

The aim of this project is to investigate the effects of different control systems for a wind turbine. The investigations are divided into below rated operation and above rated operation. It is clear that the wind turbine dynamics and control objectives change largely between different modes of operation.

A nonlinear wind turbine model based upon the existing WD34 wind turbine is developed first. All significant dynamic features encountered on a practical wind turbine have been included in the model with emphasis on the use of such a model in the evaluation and investigation of control systems. The model has been validated by experimental data obtained from the WD34 wind turbine. The validation results show a good agreement between experimental data and simulation output. The nonlinear wind turbine model is used for simulation purpose.

Linear wind turbine models are derived for design of control systems. rated operation, the active pitch control is implemented for a constant speed wind turbine to limit the power and minimize the load transients. However, a pitch regulate variable speed wind turbine has two control variables, namely pitch angle and generator torque reference, resulting a multivariable control problem. The pitch demands varied to regulate the generator speed and the reference torque is varied to regulate the drive train torques. For below rated operation, control without pitch action through the generator reaction torque is implemented. The turbine in this case is a single input single output system. In both operation modes the design models are linearized at a specific operating point. The nonlinearity in the turbine leads to that the models depend on the operating point significantly.

The estimation of the wind speed is discussed before the investigation of the control systems, since some control methods may need the knowledge of the wind speed. Three different estimation methods have been developed to estimate the wind speed. The investigation shows that the estimation of wind speed is a useful method for control design. Care must be taken when implementing the estimation approaches since the wind speed cannot be uniquely determined in some operating area.

In partial load, the aim of control is to extract maximum energy from wind and meanwhile reduce the transient loads. Since the energy is captured by the turbine, it is natural to optimize the performance criterion with regard to the aerodynamic power. In order to achieve the maximum C_p, the tip speed ratio should be kept at its optimal value. Therefore the turbine speed must follow the changes in wind speed, which requires large variations in the control signal and consequently increases the transient loads.

If the main concern is to minimize shaft torque variations, a turbine with a large inertia is prefered, since the aerodynamic torque variation has less influence on the turbine speed in this case. On the other hand it is better with small inertia when high power production is prefered, because it demands less effect to change the turbine speed in order to keep the tip speed ratio constant.

In full load, the PI pitch control, as well as the LQG pitch control and LQG variable speed pitch control are investigated. The simulation results show that the variable speed pitch control provides the best performance.

Variable speed operation of horizontal axis wind turbines has several potential advantages, of which two frequently mentioned ones are the additional energy capture below rated wind speed and the additional power train compliance and associated load alleviation above rated wind speed. The ability to operate at varying rotor speed effectively adds compliance to the power train dynamics of the wind turbine. Another reason for using variable speed operation is that the rotor speed can be adjusted to reduce noise levels and make the wind turbine more environmentally agreeable [Leithead and Connor, 1994].

These benefits of using variable speed turbine must be assessed against the disadvantages: the power generation system is more expensive due to the introduction of complex power conversion equipment. Another disadvantage is that the efficiency and reliability of the variable speed turbine and associated control system are reduced.

The performance of variable speed wind turbines is strongly dependent on the the quality of the control system through its ability to shape the dynamic

response of the turbine to wind speed turbulence. It has been indicated by [Leithead, 1989b] that the performance of variable speed wind turbine is to an extent reduced by the interaction of the wind speed turbulence with the system dynamics. Below rated wind speed the additional energy capture is inevitably reduced from that which might be expected from fully exploiting the variable speed capability. Above rated wind speed, when the regulation is generator reaction torque alone, the load alleviation which might be expected is again reduced. When regulation is by pitch action in addition to generator torque, improved performance can be expected.

The most important implementation issue encountered for developing wind turbine controllers is smooth change between the two operation modes. The most simple switching approach is to freeze the integral action in the controller when below rated operation is detected.

Bibliography

[Andersen et al., 1980] Andersen, P. S., Krabbe, U., Lundsager, P., and Petersen, H. (1980). Basismateriale for beregning af propelvinmølle. Technical report, Risø National Laboratory. Risø-M-2153.

[Bitmead et al., 1990] Bitmead, R. R., Gevers, M., and Werts, V. (1990). *Adaptive Optimal Control*. Prentice-Hall.

[Blåbjeg and Petersen, 1994] Blåbjeg, F. and Petersen, J. K. (1994). Asynkronmotoren og dens styringer. *Teknisk nyt special 1994*.

[Bongers et al., 1990] Bongers, P. M. M., Bierbooms, W., Dijkstra, S., and van Holten, T. (1990). An integrated dynamic model of a flexible wind turbine. Technical report, Delft University of Technology, Holland.

[Connor and Leithead, 1994] Connor, B. and Leithead, W. E. (1994). Control strategies for variable speed stall regulated wind turbines. In *5th European Wind Energy Association Conference and Exhibition*, Greece.

[Dennis and Schnabel, 1983] Dennis, J. E. and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Inc.

[Ekelund, 1994a] Ekelund, T. (1994a). Control of variable speed wind turbines in a broad range of wind speeds. Master's thesis, Control Engineering Laboratory, Chalmers University of Technology, Göteborg, Sweden.

[Ekelund, 1994b] Ekelund, T. (1994b). Speed control of wind turbines in the stall region. In *The Proceedings of the third IEEE Conference on Control Applications*, volume Vol. 1, pages pp 227–232, Glasgow.

[Ekelund, 1997] Ekelund, T. (1997). *Modeling and Linear Quadratic Optimal Control of Wind Turbines*. PhD thesis, Control Engineering Laboratory, Chalmers University of Technology, Göteborg, Sweden.

[Ernst and Leonhard, 1988] Ernst, J. and Leonhard, W. (1988). Optimization of the energy output of variable speed wind turbine. In *Wind Power 85' San Franisco Proceeding*, volume Vol. 1, pages p. 183–188.

[Flensborg and Sørensen, 1995] Flensborg, J. and Sørensen, J. B. (1995). Combined variable speed and pitch control for wind turbines. Master's thesis, Institute of Electronic System, Aalborg University.

[Franklin et al., 1990] Franklin, G. F., Powell, J. D., and Powell, M. L. (1990). *Digital Control of Dynamic Systems*. Addison-Wesley Publishing Company.

[Freris, 1990] Freris, L. L. (1990). *Wind Energy Conversion Systems*. Prentice Hall International (UK) Ltd.

[Gelb, 1974] Gelb, A. (1974). *Applied Optimal Estimation*. Massachusetts Institute of Technology.

[Højstrup, 1982] Højstrup, J. (1982). Velocity spectra in the unstable planetary boundary layer. *Journal of the Atmospheric Sciences*, Vol. 39(No. 10).

[Knudsen, 1983] Knudsen, T. (1983). Regulering af vindmøller. Master's thesis, IMSOR, DTH. Eksamensprojekt nr. 13/83.

[Knudsen, 1989] Knudsen, T. (1989). *Start/Stop Strategier for Vind-Diesel Systemer*. PhD thesis, IMSOR, DTH. Licentiatafhandling nr. 53.

[Kostenko and Piotrovsky, 1969] Kostenko, M. and Piotrovsky, L. (1969). *Electrical Machines, Alternating Current Machines*, volume 2. MIR Publishers, Moscow.

[Kwakernaak and Siven, 1972] Kwakernaak, H. and Siven, R. (1972). *Linear Optimal Control System*. Wiley-interscience, John Wiley & Sons, Inc., USA.

[Larsen et al., 1992] Larsen, S. E., Courtney, M., Hansen, F. A., Højstrup, J., and Jensen, N. O. (1992). Turbulence intensity and power spectra 70m above the water surface of the great belt. Technical report, Risø National Laboratory. Risø-M-2898(EN).

[Leith and Leithead, 1997] Leith, D. J. and Leithead, W. E. (1997). Implementation of wind turbine controllers. *Int. J. Control*, Vol. 66(No. 3).

[Leithead, 1989a] Leithead, W. E. (1989a). Control systems for wind turbines. *Wind Engineering*, Vol. 13(No. 6).

[Leithead, 1989b] Leithead, W. E. (1989b). Variable speed operation - does it have any advantages? *Wind Engineering*, Vol. 13(No. 6).

[Leithead and Connor, 1994] Leithead, W. E. and Connor, B. (1994). Control of a variable speed wind turbine with induction generator. In *International Conference on Control 94, Conference publication No. 389, IEE 1994*.

[Leithead et al., 1990] Leithead, W. E., Salle, S. A. D. L., and Reardon, D. (1990). Wind turbine control objectives and design. In *European Community Wind Energy Conference*, Madrid, Spain.

[Leithead et al., 1991] Leithead, W. E., Salle, S. A. D. L., and Reardon, D. (1991). Pole and objectives of control for wind turbines. *IEE Proceeding-C*, Vol. 138(No. 2).

[Leithead et al., 1992] Leithead, W. E., Salle, S. A. D. L., and Reardon, D. (1992). Classical control of active regulation of constant speed horizontal axis wind turbines. *Int. Journal of Control*, Vol. 55(No. 4):845–876.

[Novak and Ekelund, 1994] Novak, P. and Ekelund, T. (1994). Modeling, identification and control of a variable-speed hawt. In *5th European Wind Energy Association Conference and Exhibition*, Greece.

[Novak et al., 1995] Novak, P., Ekelund, T., Jovik, I., and Schmidtbauer, B. (1995). Modelling and control of variable-speed wind turbine drive-system dynamics. *IEEE Control Systems*, Vol. 15(no. 4):p. 28–38.

[Østergaard, 1994] Østergaard, P. (1994). Pitchregulering af en vindmølle. Master's thesis, IMM, DTU. IMM-EKS-1994-27.

[Petersen et al., 1980] Petersen, E. L., Troen, I., and Frandsen, S. (1980). Vindatlas for danmark. Technical report, Risø National Laboratory, Denmark.

[Poulsen, 1985] Poulsen, N. K. (1985). *Robust Self Tuning Controllers*. PhD thesis, IMSOR, DTU, Denmark.

[Salle et al., 1990] Salle, S. A. D. L., Reardon, D., W.E.Leithead, and Grimble, M. J. (1990). Review of wind turbine control. *Int. Journal of Control*, Vol. 52:p. 1295–111310.

[Schmidtbauer, 1994] Schmidtbauer, B. (1994). Basdynamik och kompenseringsmetoder för vindturbin/generator. Technical report, Control Engineering Laboratory, Göteborg, Sweden. ISSN 0280-8927.

[Tsiolis, 1994] Tsiolis, S. (1994). Asynkrongenerator och frekvensomriktare för drift av vindkraftverk med variabelt varvtal. Technical report, School of Electrical and Computer Engineering, Chalmers University of Technology, Göteborg, Sweden.

[Wilkie et al., 1990] Wilkie, J., Leithead, W. E., and Anderson, C. (1990). Modelling of wind turbines by simple models. *Wind Engineering*, Vol. 14(No. 4).

Appendix A

Optimization Background

Some background on the performance optimization which prepares the way for adaptive extremum control has been given here. The unconstrained optimization problems can be solved by a great deal of numerical methods. In this appendix the methods considered to be of particular relevance to the extremum control problems will be presented, which include the hill-climbing method, gradient method, Newton and Gauss-Newton method. As a complement of these methods the line search technique is introduced. An introduction of the unconstrained optimization has been given by [Frandsen et al. 95] and [Dennis and Schnabel, 1983].

A.1 Searching for an extremum

For a real function we want to find an argument vector x that corresponds to a minimal function value:

$$x^* = \arg\min_x f(x) \qquad \text{(A.1)}$$
$$f: \quad \mathbf{R^n} \rightarrow \mathbf{R}$$

i.e., $f(x^*) \leq f(x)$. $\forall x \in \mathbf{R^n}$. This is called unconstrained minimization problem. The function f is called the objective function and vector x^* with this property is called a global minimum point for f, or a global minimizer.

The ideal situation for optimization computation is that the objective function has a unique minimizer, i.e., the global minimizer. But in some cases the objective function will have more than one minimizer, unfortunately it is sometime very difficult to develop methods which can find a global convergence point. Instead, we will find a vector x^*, if $\epsilon > 0$ exists, then

$$\|x - x^*\| < \epsilon \;\Rightarrow f(x^*) \leq f(x) \qquad \text{(A.2)}$$

such a vector x^* is called a local minimum point for f. A local minimum for f is an argument vector giving the smallest function value inside a certain region.

The second order Taylor series for an objective function $f(x)$ in a neighbourhood around x is

$$f(x+h) = f(x) + h^T f'(x) + \frac{1}{2} h^T f''(x) h + O(\|h\|^3) \qquad \text{(A.3)}$$

where the gradient is defined by

$$g(x) = f'(x) \qquad \text{(A.4)}$$

and the second order derivative matrix, the so-called Hessian matrix, is defined by

$$H(x) = f''(x) \qquad \text{(A.5)}$$

A sufficient condition for $x = x^*$ being a local minimum of $f(x)$ is

$$g(x^*) = 0 \qquad H(x^*) > 0 \qquad \text{(A.6)}$$

Correspondingly, if the Hessian matrix is negative definite

$$g(x^*) = 0 \qquad H(x^*) < 0 \qquad \text{(A.7)}$$

$x = x^*$ will be a local maximum of $f(x)$.

All numerical methods for unconstrained optimization described in this appendix are iterative. That is to say, an initial estimate x_0 of a minimizer x^* of the objective function f is given, and a sequence $\{x_k\}$ of estimates of x^* is generated. This sequence will, under certain conditions on x_0 and on f, converge to x^*. This type of methods generally takes the form

$$x_{k+1} = x_k + \alpha_k h_k \qquad \text{(A.8)}$$

where x_k is current iteration, h_k is current search direction and α_k is step size.

As a ideal condition, a stopping rule for the iteration (A.8) is that the current error is small enough

$$\|x_k - x^*\| < \delta_1 \tag{A.9}$$

However, it cannot be used in real application, since x^* is not known. Instead we have to use an approximation to this condition

$$\|x_{k+1} - x_k\| < \epsilon_1 \tag{A.10}$$

Another rule is the current value of $f(x)$ is near enough to the minimal value $f(x^*)$. Using the same approximation, we have

$$f(x_k) - f(x_{k+1}) < \epsilon_2 \tag{A.11}$$

Since $g(x_k) \to 0$, if x_k converge to x^* for $k \to \infty$, the stopping criterion can also be

$$\|g(x_k)\| < \epsilon_3 \tag{A.12}$$

A.2 Hill-climbing algorithm

If we assume that a dynamic system is described by a one-dimensional simple form

$$f(x(t)) = y^* - a_0(x(t) - x^*)^2 \tag{A.13}$$

where f can be taken as a performance function or the system output. y^* is the maximum attainable value of f, x^* is the optimal value of adjustable input $x(t)$ which maximizes or minimizes the nonlinear function. The relation between f and the $x(t)$ is assumed to be quadratic, and a_0 is sensitivity

of the quadratic curve. The parameters y^*, x^* and a_0 which characterize the performance function or the system dynamics are normally unknown. The quadratic curve is illustrated in Figure A.1.

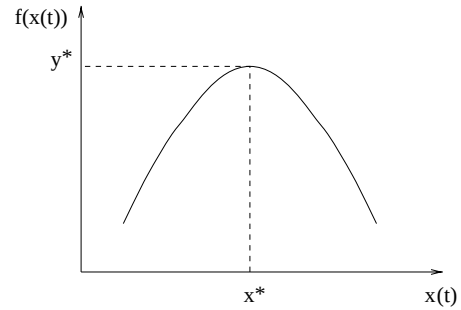

Figure A.1. Performance function for a single parameter extremum problem

A quadratic assumption of f is important, since it is in most cases acceptable for extremum controllers operating close to the optimum point.

The simplest hill-climbing algorithm is to move from an initial position to the optimum point by fixed length steps. The principle of the algorithm is shown in Figure A.2.

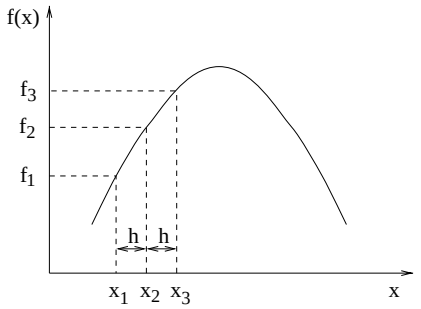

Figure A.2. Hill-climbing algorithms

where $f_i = f(x_i)$, $i = 1, 2, 3$. From position $x(t) = x_1$, a movement $x_2 = x_1 + h$ is made, the values f_1 and f_2 are compared by calculating performance function (A.13) which is assumed known. If f_2 is large than f_1, $x_2 = x_1 + h$ is taken as the value of $x(t+1)$, otherwise $x_2 = x_1 - h$ is adopted. The process is continued until the extremum is reached. The algorithm can be considered as a special case of algorithm (A.8), where the step size is h and the search direction is ± 1.

Since the input signal will move backward and forward either side of the optimum eventually, $x(t)$ will lie in a neighbourhood of the optimal value x^* at last. If we decrease the step length h, $x(t)$ will be closer to the optimal value, but the convergence speed will be slow down. If we increase the step length h, the output will have a substantial distance between the extremum, although the convergence speed is fast.

The algorithm can make steps in the wrong direction because of the presence of the noise in the system.

A.3 Gradient method

The gradient algorithm or the steepest descent algorithm allows the hill-climbing rate to be adjusted according to the size of the gradient. At each iteration the search direction is modified along the opposite direction of the gradient, i.e., $h_k = -f'(x_k)$, which is called the steepest decent direction. In this case the adjustment rule from x_{k+1} to x_k is given by

$$x_{k+1} = x_k - \alpha_k g(x_k) \tag{A.14}$$

where $g(x_k)$ is the gradient defined in (A.4) and $\alpha_k > 0$ is a gain parameter which controls the convergence rate. The gradient method relies on a first order approximation to the objective function in a neighbouring point x_{k+1} to x_k

$$\begin{aligned} f(x_{k+1}) &= f(x_k) + (x_{k+1} - x_k)^T g(x_k) + O(\|x_{k+1} - x_k\|^2) \\ &= f(x_k) - \alpha_k g(x_k)^T g(x_k) + O(\alpha_k^2) \end{aligned} \tag{A.15}$$

If the step size $\alpha_k > 0$ is adequately small, the first two terms in the function will dominate over the last, i.e., the third term is insignificant. It is then always possible to obtain a reduction of the objective function

$$f(x_{k+1}) < f(x_k)$$

A constant step size is quite often selected, whereas a number of methods can be applied to select the step size α_k, since the convergence rate relies on it. A line search method given in section A.6 will in general give the most rapid convergence.

The advantage of the gradient algorithm is that the size of the adjustment at each step is dependent upon the size of the gradient. The adjustment step is large when the algorithm is far from the extremum and reduces when the extremum is close. But the algorithm requires a knowledge of the gradient function and it might be sensitive to the noise. The gradient method has linear convergence (convergence of order 1), i.e.

$$\|e_{k+1}\| < c_1 \|e_k\| \ \textit{with}\ c_1 < 1,\ x_k\ \textit{close to}\ x^* \tag{A.16}$$

where $e_k = x_k - x^*$. The gradient method is much faster than the fixed step length methods.

A.4 The Newton method

Whereas the gradient method relies on a first order approximation to the objective function, the newton method is determined as a minimizer of a second order expansion of the objective function at current iteration x_k

$$f(x) = f(x_k) + (x - x_k)^T f'(x_k) + \frac{1}{2}(x - x_k)^T f''(x_k)(x - x_k) \qquad \text{(A.17)}$$

The idea is now at iteration x_k to minimize the objective function $f(x)$. If the Hessian $H(x_k)$ is positive definite, $f(x)$ will have a unique minimizer at a point where $f'(x) = 0$, i.e.

$$g(x_k) + H(x_k)(x - x_k) = 0$$

which leads to

$$x_{k+1} = x_k - H(x_k)^{-1} g(x_k) \qquad \text{(A.18)}$$

This is also a special case of general algorithm (A.8), when the step size $\alpha_k = 1$ and the search direction $h_k = -H(x_k)^{-1} g(x_k)$ which is called a Newton direction.

A maximum point will be found if $H(x_k)$ is negative definite for all k.

In case $H(x_k)$ stays positive definite for all k and if starting point is sufficient close to a minimizer, the method will usually converge very rapidly towards to minimum point. More precisely, the Newton method converges

quadratically toward x^*, i.e.,

$$\|e_{k+1}\| < c_2 \|e_k\|^2, \ x_k \ close \ to \ x^* \qquad \text{(A.19)}$$

The disadvantage of the Newton optimization algorithm is that it requires a knowledge of both the gradient and second order derivative of the performance function. These may be difficult to find even though they are known to exist. But the most severe drawback is the method lack of global convergence. When x_k is far from the solution, $f''(x_k)$ may not be positive definite. In this case the iteration may converge to a saddle point or a maximizer since the iteration is identical to the one used for solving $f'(x) = 0$. Any stationary point of f is a solution to the system.

The first modification is the Newton method with line search, therefore the iteration is modified by

$$\boxed{x_{k+1} = x_k - \alpha_k H(x_k)^{-1} g(x_k)} \qquad \text{(A.20)}$$

where the step size α_k is determined by line search algorithm which will be given in section A.6.

Another idea is to take the advantage of the safe convergence properties of the gradient method whenever the Newton method gets into trouble. The gradient method is selected, when x_k is far from the convergence point where the Hessian may not be positive definite. The Newton method is selected, when x_k is close to the convergence point where the gradient method

converges slowly. The quadratic convergence can be obtained when the sequence gets close to x^*.

A much more appealing modification of the original Newton method is often referred as the Damped Newton method where we combine the gradient method and the Newton method in the following way. The gradient method and Newton method are shown together here

Gradient method	**Newton method**
Solve $\mathrm{I}h_k = -f'(x)$	Solve $f''(x_k)h_k = -f'(x_k)$
$x_{k+1} = x_k + \alpha_k h_k$	$x_{k+1} = x_k + \alpha_k h_k$

where I is the identity matrix.

The idea of a Damped Newton method is to combine the two methods by adding a multiple of the identity matrix to $f''(x_k)$.

Damped Newton iteration

Solve $(f''(x_k) + \upsilon \mathrm{I})h_k = -f'(x_k)$, $\upsilon \geq 0$
Adjust υ
If $x_k + h_k$ acceptable then $x_{k+1} = x_k + h_k$.

It is easily seen this type of method is a compromise between the gradient method and Newton method. In case $f''(x_k + \upsilon \mathrm{I})$ is not safely positive definite, υ will be increased. If υ is large, h_k will be close to the gradient direction, whereas a small υ yields a nearly Newton direction. If υ is updated in each iteration, the method is called Levenberg-Maquardt method. The details for choosing υ is given by [Frandsen et al. 95] and [Dennis and Schnabel, 1983].

A.5 Gauss-Newton method

The so-called Gauss-Newton method is an iterative method to minimize the criterion

$$V(x) = \sum_{t=1}^{m} (f(t,x))^2 \tag{A.21}$$

We assume that f can be approximated by a first order Taylor series, which is given by

$$\bar{f}(t,x) = f(t,x_k) + (x - x_k)^T f'(t,x_k) \tag{A.22}$$

the approximated criterion is then

$$L(x) = \sum_{t=1}^{m} (\bar{f}(t,x))^2 \tag{A.23}$$

so we have $V(x) \approx L(x)$

The gradient at x_k in the Newton method is

$$g(x_k) = L'(x_k) = \sum_{t=1}^{m} 2\bar{f}(t,x_k) f'(t,x_k) \tag{A.24}$$

and Hessian is

$$R(x_k) = \sum_{t=1}^{m} 2f'(t, x_k) f'(t, x_k) \tag{A.25}$$

$R(x_k)$ is called the *Gauss-Newton Hessian* and it is obviously positive semidefinite. The Gauss-Newton iteration is derived as a minimizer of $L(x)$

$$x_{k+1} = x_k - \alpha_k R(x_k)^{-1} g(x_k) \tag{A.26}$$

In practice the iteration should be complemented with a line-search for determining the step size α_k. It is clear that the Newton and Gauss-Newton methods are identical, when the second order derivative can be neglected.

The Gauss-Newton has the attractive property that it requires only a first order derivative. The local convergence of the method is in general only linear (first order), which is theoretically slower than the Newton method. But experiences show that it is often faster in practice. In particular when it is used far from the minimum. Another problem in the Gauss-Newton method is that the $R(x_k)$ may be singular or ill-conditioning. It can be solving by adding a small diagonal matrix to the Hessian.

A.6 Line search technique

In some cases the gradient method or Newton method will be complemented by a line search, because the algorithm is derived from a first order or second order approximation to the objective function. The approximation

will usually be valid only in a certain neighborhood around the current iteration. The full step $\alpha_k = 1$ might bring the new iteration to a location that is far from the point predicted by the approximation.

The idea of line search methods is: when a gradient direction or a Newton direction has been determined, we have to decide how long the step in this direction should be. A frequently used method for determining step size α_k is to estimate a local minimizer of $f(x_k + \alpha h_k)$ regarded as a function of α

$$f(x_k + \alpha_k h_k) = \min_{\alpha > 0} f(x_k + \alpha h_k) \tag{A.27}$$

Introduce the notation

$$\varphi(\alpha) = f(x_k + \alpha h_k) \tag{A.28}$$

then one method of determining α_k is to estimate a local minimizer of φ, so that α_k satisfies

$$\varphi'(\alpha) = 0 \tag{A.29}$$

In general, (A.29) is a nonlinear equation and difficult to solve analytically. Therefore some numerical methods for estimating a value of α which satisfies (A.27) must be used. Most line searches are iterative procedures which are terminated when the current estimate of α_k satisfies a given set of convergence criteria. Line searches are therefore usually not exact.

The methods which are used frequently for estimating α_k are based upon the idea of approximating φ defined by (A.28) by a polynomial in α of degree two

or three, and determining the minimizer of the polynomial approximation analytically. The approximation of φ by polynomials of degrees two and three is called quadratic interpolation and cubic interpolation respectively. Only the quadratic interpolation is given here.

Quadratic interpolation

Let a_1, a_2 and a_3 be given distinct values of α, the corresponding φ is $\varphi(a_1)$, $\varphi(a_2)$ and $\varphi(a_3)$. The quadratic polynomial Φ for which

$$\Phi(a_i) = \varphi(a_i) \quad (i = 1, 2, 3) \tag{A.30}$$

then the so-called Lagrange interpolating polynomial of degree two is given by

$$\Phi(\alpha) = \frac{(\alpha - a_2)(\alpha - a_3)}{(a_1 - a_2)(a_1 - a_3)}\varphi_1 + \frac{(\alpha - a_1)(\alpha - a_3)}{(a_2 - a_1)(a_2 - a_3)}\varphi_2 + \frac{(\alpha - a_1)(\alpha - a_2)}{(a_3 - a_1)(a_3 - a_2)}\varphi_3 \tag{A.31}$$

where $\varphi_i = \varphi(a_i)(i = 1, 2, 3)$. Differentiating Φ, and solving the linear equation

$$\Phi'(\alpha) = 0 \tag{A.32}$$

we obtain for the critical point $\hat{\alpha}$ of φ,

$$\hat{\alpha} = \frac{1}{2}\frac{(a_2^2 - a_3^2)\varphi_1 + (a_3^2 - a_1^2)\varphi_2 + (a_1^2 - a_2^2)\varphi_3}{(a_2 - a_3)\varphi_1 + (a_3 - a_1)\varphi_2 + (a_1 - a_2)\varphi_3} \tag{A.33}$$

Differentiating Φ a second time, $\Phi''(\hat{\alpha}) > 0$ if and only if

$$\frac{(a_2 - a_3)\varphi_1 + (a_3 - a_1)\varphi_2 + (a_1 - a_2)\varphi_3}{(a_2 - a_3)(a_3 - a_1)(a_1 - a_2)} < 0 \tag{A.34}$$

Hence $\hat{\alpha}$ is a minimizer of Φ if and only if above condition holds.

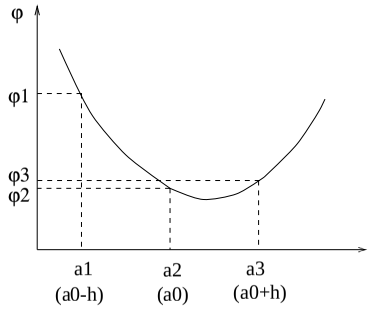

Figure A.3. Quadratic interpolation

In particular, if $a_1 = a_0 - h$, $a_2 = a_0$, and $a_3 = a_0 + h$, where $h > 0$ is given, and a_0 is a value of α, then

$$\hat{\alpha} = a_0 - \frac{h(\varphi_3 - \varphi_1)}{2(\varphi_3 - 2\varphi_2 + \varphi_1)} \tag{A.35}$$

and $\hat{\alpha}$ is a minimizer of Φ if and only if

$$\varphi_3 - 2\varphi_2 + \varphi_1 > 0 \tag{A.36}$$

It is intuitively obvious that if φ is twice continuously differentiable in a neighbourhood of a minimizer α^*, and a_1, a_2, a_3 are sufficiently close to α^*, then $\hat{\alpha}$ is likely to be a close approximation to α^*. The above line search procedures only require the evaluation of φ, not the first derivative of it.

In practice, the above method comprises a bracketing section, an interpolating section and a refining section. In the bracketing section, three distinct values $a_1 < a_2 < a_3$ of α are found such that the minimizer α^* of φ lies between a_1 and a_3. In the interpolation section, $\hat{\alpha}$ is computed from (A.33) if (A.34) is satisfied. In the refining section, a new set of interpolating points

is selected from a_1, a_2, a_3 and $\hat{\alpha}$, if $\hat{\alpha}$ is not a sufficiently good estimate of α^*, and interpolation section is re-entered. The interpolating and refining section are used repeatedly until $\hat{\alpha}$ is regarded as a sufficiently good estimate of α^*. Details of the bracketing section and the refining section are given by [Wolfe 78].

A.7 Summary

This appendix gives an introduction of unconstrained optimization techniques as a background knowledge to the extremum control problem. In the chapter the hill-climbing method, gradient method, Newton and Gauss-Newton method are introduced. The gradient method or Newton type methods can be incorporated with line search.

The gradient method has a linear final convergence, which is slower than the Newton method that has a second order convergence. The size of adjustment in the gradient method at each iteration is dependent upon the size of the gradient. The adjustment is large and the algorithm converges fast when x_k is far from the convergence point. The adjustment is small and the algorithm converges slowly when x_k is close to the convergence point.

The Newton method is simple and clear and thus easy to implement, the method converges quadratically from good starting guess if $f''(x^*)$ is positive definite. But the Newton method lacks of global convergence for many problems. It requires second order derivative of f, besides $f''(x_k)$ may not be positive. The Newton method can be modified by combining the line search where the step size α_k will be determined at each step. For a special

criterion, the Gauss-Newton method can guarantee the Hessian is positive semidefinite.

The gradient method and Newton method can be combined and used for different situations. The gradient method is selected, when x_k is far from the x^*. The Newton method is taken, when x_k is close to the x^*. The combined algorithm take the advantage of the safe convergence property of the gradient method and quadratic convergence property of the Newton method.

By implementing the line search algorithm, the step size α_k usually produces a satisfactory decrease in the value of f at each iteration when a search direction is determined. However, the line search is computationally expensive.

Appendix B

Convergence Analysis for Recursive Algorithms

In this section a general approach to the analysis of the asymptotic behaviour of recursive algorithms is described. In effect, the convergence analysis is reduced to stability analysis of a deterministic Ordinary Differential Equation (ODE). The approach used for analysis is developed by Ljung, see [Ljung 77] and [Wellstead and Zarrop 91]. Here we just give a brief summarization.

Convergence properties can be stated as

1. The system is stable.
2. The model parameter estimates converge.
3. The estimates converge to correct values.

The convergence property 1 corresponds to system identifiability and is essential for achieving the correct control objective. The property 3 corresponds to parameter identifiability and it is meaningful if the model is structurally consistent with the controlled system.

B.1 Basic ideas of ODE approach

It has been assumed that the system can be represented by a linear difference equation and that the estimation model is identical in form but with unknown coefficient values.

The ODE method will give us a means for studying the recursive algorithm when it is close to a convergence point. There are two basic steps in the ODE approach. One concerns a time compression so that the asymptotic behaviour of the algorithm can be examined on a reasonable time scale. The second step is the averaging out of stochastic and time-varying elements of recursive algorithms, leaving what is effectively the average behaviour of the parameter estimates.

The first step can be explained by considering a deterministic scalar difference equation

$$x(t) = x(t-1) + \gamma(t) f(x(t-1)) \tag{B.1}$$

where the positive scalar gain sequence $\{\gamma(t)\}$ satisfies

$$\lim_{t\to\infty} \gamma(t) = 0 \tag{B.2}$$

If $f(\cdot)$ is a smooth function and $\{x(t)\}$ is bounded, then (B.2) implies that $x(t)$ and $f(x(t))$ change slowly for large t. This assumption means that we

can approximate the equation

$$x(t+s) = x(t) + \sum_{k=t+1}^{t+s} \gamma(k) f(x(k-1)) \tag{B.3}$$

by

$$x(t+s) \simeq x(t) + \left[\sum_{k=t+1}^{t+s} \gamma(k)\right] f(x(t)) \tag{B.4}$$

If $f(x(t))$ is changing slowly, we can assume that it is constant over the interval t to $t+s$, when s is not too large. Now we introduce a new compressed time scale

$$\tau = \sum_{k=1}^{t} \gamma(k) \tag{B.5}$$

under certain conditions that $\tau = 0$ when $t = 0$ and τ tends to monotonically to infinity with t

$$\sum_{k=1}^{\infty} \gamma(k) = \infty \tag{B.6}$$

This is a requirement that the gain sequence does not converge to zero too rapidly and is also a standard requirement in recursive estimators when the true parameters are known to be constant. Now we can transform the equation (B.4) to the new time scale τ and write

$$x^D(\tau) = x(t) \tag{B.7}$$

$$\tau + \Delta\tau = \sum_{k=1}^{t+s} \gamma(k) \tag{B.8}$$

the rescaled difference equation will then be

$$x^D(\tau + \Delta\tau) = x^D(\tau) + \Delta\tau f(x^D(\tau)) \tag{B.9}$$

which in turn can be written as a differential equation in the τ time scale

$$\frac{d}{d\tau}x^D(\tau) = f(x^D(\tau)) \tag{B.10}$$

in the limit as s ($\Delta\tau$) goes to zero. This is an ODE. The asymptotic behaviour of original difference equation (B.1) is close to the trajectories of the ODE (B.10). The ODE is mainly a tool for examining local convergence.

It is possible to generalize the simple scalar recursion by making the function $f(x(t))$ in equation (B.1) dependent upon a further variable $\varphi(t)$. The equation (B.1) can be replaced by

$$x(t) = x(t-1) + \gamma(t)Q(t, x(t-1), \varphi(t)) \tag{B.11}$$

We shall refer to $x(\cdot)$ as the estimates and they could be the current estimates of some unknown parameter vector. The vector $\varphi(t)$ is an observation obtained at time t, and these are the objects that cause $x(t-1)$ to be updated to take new information into account. The observations are in general functions of the previous estimates $x(\cdot)$ and of a sequence of random vectors $e(\cdot)$. This means that the observation is a random variable, which may be affected by previous estimates. The following structure for the generation of $\varphi(t)$ will be used

$$\varphi(t) = A(x(t-1))\varphi(t-1) + B(x(t-1))e(t) \tag{B.12}$$

Let

$$D_s = \{x | A(x) \text{ has all eigenvalues strictly inside the unit circle}\} \tag{B.13}$$

and for each $x \in D_s$, we define a asymptotic stationary process

$$\bar{\varphi}(t, x) = A(x)\bar{\varphi}(t-1, x) + B(x)e(t) \qquad \bar{\varphi}(0, x) = 0 \tag{B.14}$$

and furthermore we define the function

$$f(x) = \lim_{t\to\infty} EQ(t, x, \bar{\varphi}(t, x)) \tag{B.15}$$

The expectation operation smoothes the influence of random components which may disguise the convergence characteristics of the recursive algorithm. The equation (B.11) can then be again written as ordinary differential equation (B.10).

If $f(x(t))$ is continuously differentiable with respect to x in a neighborhood of a stationary point x^* and derivatives converge uniformly in this neighborhood as t tends to infinity. Then

$$f(x^*) = 0 \tag{B.16}$$

and

$$H(x^*) = \frac{d}{dx}f(x)\bigg|_{x=x^*} \text{ has all eigenvalues in the LHP} \tag{B.17}$$

The matrix $H(x^*)$ defines the linear differential equation obtained from (B.10) by linearization around x^*. The above equations state that the algorithm (B.1) or (B.11) can converge only to stable stationary points of the differential equation (B.10).

Equations (B.16) and (B.17) are necessary conditions for the local convergence of recursion (B.1) to x^*. These conditions are not sufficient and therefore do not guarantee convergence. The equations (B.16) and (B.17) are normally used to check the possible local convergence points.

B.2 General results of ODE

Now consider a linear-in-parameters model

$$y(t) = \varphi^T(t)\theta \tag{B.18}$$

where θ is a vector of parameters. It is reasonable to choose the θ to minimize the variance of equation error, i.e.

$$\min_{\theta}\{V(\theta) = E[(y(t) - \varphi^T(t)\theta)^2]\} \tag{B.19}$$

A general recursive algorithm can be written as

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + \gamma(t)R(t)^{-1}Q(\hat{\theta}(t-1), \varphi(t)) \\ R(t) &= R(t-1) + \gamma(t)F(\hat{\theta}(t-1), R(t-1), \varphi(t)) \end{aligned} \tag{B.20}$$

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 law 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

$$D_s = \{\theta | A(\theta) \text{ has all eigenvalues inside the unit circle}\} \tag{B.21}$$

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

By given the assumption of stability, let $\theta \in D_s$, and let $\bar{\varphi}(t, \theta)$ denote $\varphi(t)$ with $\hat{\theta}(t)$ replaced by θ, we define

$$\begin{aligned} f(\theta) &= \lim_{t\to\infty} E[Q(\theta, \bar{\varphi}(t, \theta))] \\ G(\theta, R) &= \lim_{t\to\infty} E[F(\theta, R, \bar{\varphi}(t, \theta))] \end{aligned} \tag{B.22}$$

the corresponding ODE is then given by

$$\begin{aligned} \frac{d}{d\tau}\theta(\tau) &= R^{-1}(\tau)f(\theta(\tau)) \\ \frac{d}{d\tau}R(\tau) &= G(\theta(\tau), R) \end{aligned} \tag{B.23}$$

where τ is given by equation (B.5). The function $f(\cdot)$ represents the "corrective force". The positive definite matrix R^{-1} only modifies the direction of correction.

The trajectories of the ODE (B.23) are the asymptotic paths of the estimates generated by (B.20).

If $\hat{\theta}(t) \to \theta^*$ and $R(t) \to R^*$ $(R^* > 0)$ as $t \to \infty$(with a probability greater than one), then

$$\begin{aligned} f(\theta^*) &= 0 \\ G(\theta^*, R^*) &= 0 \end{aligned} \tag{B.24}$$

and the matrix

$$H(\theta^*) = (R^*)^{-1}\frac{d}{d\theta}f(\theta)|_{\theta=\theta^*} \tag{B.25}$$

must have all its eigenvalues in the left half-plane (including the imaginary axis). The result states that θ^* is a local stable, stationary point of ODE. Such points are the only possible convergence points of the estimation algorithms.

B.3 Local convergence of recursive least square algorithm

If $Q(\hat{\theta}(t-1), \varphi(t))$ and $F(\hat{\theta}(t-1), R(t-1), \varphi(t))$ in the general recursive algorithm (B.20) are defined by

$$\begin{aligned} Q(\hat{\theta}(t-1), \varphi(t)) &= \varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\ F(\hat{\theta}(t-1), R(t-1), \varphi(t)) &= \varphi(t)\varphi^T(t) - R(t-1) \end{aligned} \tag{B.26}$$

and choose $\gamma(t) = 1/t$, the RLS estimator can then be obtained

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + 1/tR(t)^{-1}\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\ R(t) &= R(t-1) + 1/t[\varphi(t)\varphi^T(t) - R(t-1)] \end{aligned} \tag{B.27}$$

where $\varphi(t)$ is a function of the input/output data available at time $t-1$. Both $y(t)$ and $\varphi(t)$ are generated by the system and controller.

In order to form ODE, $\hat{\theta}$ is fixed at some nominal value θ, and the matrices $f(\theta)$ and $G(\theta, R)$ in the ODE will be

$$\begin{aligned} f(\theta) &= \lim_{t\to\infty} E\{\varphi(t)[y(t) - \varphi^T(t)\theta]\} \\ G(\theta, R) &= \lim_{t\to\infty} E\{\varphi(t)\varphi^T(t) - R] \end{aligned} \tag{B.28}$$

Under an adaptive feedback law the $y(t)$, $u(t)$ and $\varphi(t)$ will be functions of θ.

If we assume that the true system is given by

$$y(t) = \varphi^T(t)\theta_0 + e(t) \tag{B.29}$$

where $e(t)$ is assumed to be zero mean white noise with variance σ_e^2, and $E[\varphi(t)e(t)] = 0$. Substituting (B.29) into (B.28) yields

$$\begin{aligned} f(\theta) &= G(\theta)(\theta_0 - \theta) \\ G(\theta, R) &= G(\theta) - R \end{aligned} \tag{B.30}$$

where $G(\theta) = \lim_{t\to\infty} E[\varphi(t)\varphi^T(t)]$ is a symmetric nonnegative definite matrix.

If $\hat{\theta}(t) \to \theta^*$ and $R \to R^*$, the local convergence point for the recursion (B.27) will satisfy

$$\begin{aligned} f(\theta^*) &= G(\theta^*)(\theta_0 - \theta^*) = 0 \\ R^* &= G(\theta^*) \end{aligned} \tag{B.31}$$

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

$$\lim_{t\to\infty} E[\varphi(t)\varphi^T(t)] > 0 \tag{B.32}$$

which is a generalized persistent excitation condition and this usually requires that inputs to the system vary sufficiently to excite the system, it implies that $\theta^* = \theta_0$, and

$$H(\theta^*) = (R^*)^{-1}\frac{d}{d\theta}f(\theta)|_{\theta=\theta_0} = -I \tag{B.33}$$

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

System identification is usually the main goal of an adaptive control algorithm, lack of sufficient excitation to guarantee a nonsingular R^* indicates that possible ill-conditioning problems may occur in the recursion (B.27) as time evolves. By reconstructing the $R(t)$ matrix in recursion (B.27), a positive definite matrix $R(t)$ for all t can be guaranteed.

$$R(t) = R(t-1) + \frac{1}{t}[\varphi(t)\varphi^T(t) + \delta I - R(t-1)] \tag{B.34}$$

where δ is a small positive scalar. This is known as the Levenberg-Marguardt regularization. Finally, note that (B.34) leads to the replacement of ODE by

$$\frac{dR}{d\tau} = G(\theta) + \delta I - R \tag{B.35}$$

This means that if the estimates converge

$$R^* = G(\theta^*) + \delta I \tag{B.36}$$

which is invertible.

Appendix C

The EKF as a Joint state and Parameter Estimator

The extended Kalman filter 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 in [Ljung 79b] in a systematic and comprehensive way. In this appendix we just give a brief description for understanding Chapter 4

C.1 Extended Kalman filter

For further reference, we shall give a brief account of the extended Kalman filter (EKF) algorithm. The method is based on linearization of the state equations at each time step and use of linear estimation theory.

Let the nonlinear discrete-time system be given by

$$\begin{aligned} x(t+1) &= f(t,x(t)) + w(t) \\ y(t) &= h(t,x(t)) + e(t) \end{aligned} \tag{C.1}$$

where $y(t)$ and $x(t)$ are output and state vectors. The sequences $\{v(t)\}$ and $\{w(t)\}$ are independent random vectors with zero means and variances $Q^w(t) = E\{w(t)w^T(t)\}$ and $Q^e(t) = E\{e(t)e^T(t)\}$.

The EKF estimate of the state $x(t)$ is given by

$$\hat{x}(t+1) = f(t,\hat{x}(t)) + K(t)[y(t) - h(t,\hat{x}(t))] \tag{C.2}$$

where $K(t)$ is

$$\begin{aligned} K(t) &= F(t,\hat{x}(t))P(t)H^T(t,\hat{x}(t)) \\ &\quad \cdot [H(t,\hat{x}(t))P(t)H^T(t,\hat{x}(t)) + Q^e(t)]^{-1} \end{aligned} \tag{C.3}$$

$$\begin{aligned} P(t+1) &= F(t,\hat{x}(t))P(t)F^T(t,\hat{x}(t)) + Q^w(t) \\ &\quad - K(t)[H(t,\hat{x}(t))P(t)H^T(t,\hat{x}(t)) + Q^e(t)]K^T(t) \end{aligned} \tag{C.4}$$

and

$$\begin{aligned} F(t,\hat{x}) &= \tfrac{\partial}{\partial x} f(t,x)\big|_{x=\hat{x}} \\ H(t,\hat{x}) &= \tfrac{\partial}{\partial x} h(t,x)\big|_{x=\hat{x}} \end{aligned} \tag{C.5}$$

C.2 The system

Assume the system is

$$\begin{aligned} x(t+1) &= A_0 x(t) + B_0 u(t) + w(t) \\ y(t) &= C_0 x(t) + e(t) \end{aligned} \tag{C.6}$$

where $\{w(t)\}$ and $\{e(t)\}$ are independent random sequences with zero means and variances

$$\begin{aligned} E\{w(t)w^T(t)\} &= Q_0^w \\ E\{e(t)e^T(t)\} &= Q_0^e \\ E\{w(t)e^T(t)\} &= Q_0^{we} \end{aligned} \tag{C.7}$$

Furthermore, it is assumed that the initial state $x(0)$ is a random vector with zero-mean and covariance matrix Π_0. It is independent of future values of $\{w(t)\}$ and $\{e(t)\}$. All the matrices A_0, B_0, C_0, Q_0^w, Q_0^e and Q_0^{we} are assumed to be time invariant.

If these matrices are all known, then the linear least-squares state estimate for the system is

$$\hat{x}_0(t+1) = A_0\hat{x}_0(t) + B_0 u(t) + K_0(t)[y(t) - C_0\hat{x}_0(t)] \tag{C.8}$$

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

where

$$K_0(t) = [A_0 P_0(t) C_0^T + Q_0^{we}][C_0 P_0(t) C_0^T + Q_0^e]^{-1} \tag{C.9}$$

$$P_0(t+1) = A_0 P_0(t) A_0^T + Q_0^w - K_0(t)[C_0 P(t) C_0^T + Q_0^e]K_0^T(t) \tag{C.10}$$

$$P_0(0) = \Pi_0$$

Let

$$\bar{K}_0 = \lim_{t\to\infty} K_0(t) \tag{C.11}$$

C.3 The model

If the system (C.6) - (C.7) is assumed to be (partly) unknown to the user, the problem faced is to determine the matrices A_0, B_0, C_0 and possibly also Q_0^w, Q_0^e and Q_0^{we} together with the state estimates, based on the measurements of input-output data.

The following model is assumed for the system (C.6) - (C.7)

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + w_\theta(t) \\ y(t) &= C(\theta)x(t) + e_\theta(t) \end{aligned} \tag{C.12}$$

where

$$\begin{aligned} E\{w_\theta(t)w_\theta^T(t)\} &= Q^w(\theta) \\ E\{e_\theta(t)e_\theta^T(t)\} &= Q^e(\theta) \\ E\{w_\theta(t)e_\theta^T(t)\} &= Q^{we}(\theta) \\ E\{x(0)\} &= x_0 \\ E\{x(0)x^T(0)\} &= \Pi(\theta) \end{aligned} \tag{C.13}$$

The matrices $A(\theta)$, $B(\theta)$, $C(\theta)$, $Q^w(\theta)$, $Q^e(\theta)$ and $Q^{we}(\theta)$ depend on a parameter vector θ in an arbitrary way. It is assumed that the matrix elements are differentiable with respect to θ.

C.4 Joint parameter and state estimation

Usually in model (C.12) - (C.13), the matrices $Q^w(\theta)$, $Q^e(\theta)$ and $Q^{we}(\theta)$ do not depend on θ, but are chosen fixed in some ad hoc way, most often with $Q^{we} = 0$. This corresponds to the fact that noise characteristics are independent of the state. We shall assume that $\{w_\theta\}$ and $\{e_\theta\}$ are independent

of θ in the remainder of this section, i.e., $w_\theta(t) = w(t)$, $e_\theta(t) = e(t)$ and $Q^w(\theta) = Q^w$, $Q^e(\theta) = Q^e$, $Q^{we}(\theta) = Q^{we}$.

The joint state and parameter estimation can of course be understood as a state estimation problem for a nonlinear system. It is a fairly natural thing to include the unknown parameters in the state vector, and once this is done, standard Kalman filter programs can be applied for the estimation.

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

$$\underline{x}(t) = \begin{pmatrix} x(t) \\ \theta(t) \end{pmatrix} \tag{C.14}$$

We then have the state-space form

$$\begin{aligned} \underline{x}(t+1) &= f(\underline{x}(t), u(t)) + \begin{pmatrix} w(t) \\ 0 \end{pmatrix} \\ y(t) &= h(\underline{x}(t)) + e(t) \end{aligned} \tag{C.15}$$

where

$$\begin{aligned} f(\underline{x}(t), u(t)) &= \begin{bmatrix} A(\theta)x(t) + B(\theta)u(t) \\ \theta(t) \end{bmatrix} \\ h(\underline{x}(t)) &= C(\theta)x(t) \end{aligned} \tag{C.16}$$

Now the problem belongs to a nonlinear filtering problem. If the EKF (C.2) - (C.5) is applied

$$\hat{\underline{x}}(t+1) = f(\hat{\underline{x}}(t), u(t)) + K_{\underline{x}}(t)[y(t) - h(\hat{\underline{x}}(t))] \tag{C.17}$$

$$\begin{aligned} \hat{\underline{x}}(0) &= \hat{\underline{x}}_0 \\ K_{\underline{x}}(t) &= [F(\hat{\underline{x}}(t), u(t))\bar{P}(t)H^T(\hat{\underline{x}}(t)) + \bar{Q}^{we}] \\ &\quad \cdot [H(\hat{\underline{x}}(t))\bar{P}(t)H^T(\hat{\underline{x}}(t)) + Q^e]^{-1} \end{aligned} \tag{C.18}$$

$$\begin{aligned}\bar{P}(t+1) &= F(\underline{\hat{x}}(t),u(t))\bar{P}(t)F^T(\underline{\hat{x}}(t),u(t)) + \bar{Q}^w \\ &\quad - K_{\underline{x}}(t)[H(\underline{\hat{x}}(t))\bar{P}(t)H^T(\underline{\hat{x}}(t)) + Q^e]K_{\underline{x}}^T(t) \\ \bar{P}(0) &= \bar{P}_0\end{aligned} \tag{C.19}$$

where

$$\begin{aligned}F(\underline{\hat{x}}(t),u(t)) &= \frac{\partial}{\partial \underline{x}} f(\underline{x},u)\Big|_{\underline{x}=\underline{\hat{x}}(t)} \\ &= \begin{bmatrix} A(\hat{\theta}(t)) & M(\hat{\theta}(t),\hat{x}(t),u(t)) \\ 0 & I \end{bmatrix}\end{aligned} \tag{C.20}$$

$$\begin{aligned}H(\underline{\hat{x}}(t)) &= \frac{\partial}{\partial \underline{x}} h(\underline{x})\Big|_{\underline{x}=\underline{\hat{x}}(t)} \\ &= [C(\hat{\theta}(t)) \quad D(\hat{\theta}(t),\hat{x}(t))]\end{aligned} \tag{C.21}$$

$$\begin{aligned}\bar{Q}^w &= \begin{bmatrix} Q^w & 0 \\ 0 & 0 \end{bmatrix} \\ \bar{Q}^{we} &= \begin{bmatrix} Q^{we} \\ 0 \end{bmatrix}\end{aligned} \tag{C.22}$$

$$\begin{aligned}\underline{\hat{x}}_0 &= \begin{bmatrix} \hat{x}_0 \\ \hat{\theta}_0 \end{bmatrix} \\ \bar{P}_0 &= \begin{pmatrix} \Pi(\hat{\theta}_0) & 0 \\ 0 & \Sigma_0 \end{pmatrix}\end{aligned} \tag{C.23}$$

Here

$$M(\hat{\theta}(t),\hat{x}(t),u(t)) = \frac{\partial}{\partial\theta}(A(\theta)\hat{x}(t) + B(\theta)u(t))\Big|_{\theta=\hat{\theta}(t)} \tag{C.24}$$

$$D(\hat{\theta}(t),\hat{x}(t)) = \frac{\partial}{\partial\theta}(C(\theta)\hat{x}(t))\Big|_{\theta=\hat{\theta}(t)} \tag{C.25}$$

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

Introduce for short

$$A_t = A(\hat{\theta}(t))$$

$$B_t = B(\hat{\theta}(t))$$

$$C_t = C(\hat{\theta}(t))$$

$$D_t = D(\hat{\theta}(t),\hat{x}(t))$$

$$M_t = M(\hat{\theta}(t),\hat{x}(t),u(t))$$

If we write matrices $K_{\underline{x}}(t)$ and $\bar{P}(t)$ in the form

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

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

equations (C.17) can then be rewritten explicitly as

$$\hat{x}(t+1) = A_t\hat{x}(t) + B_t u(t) + K_x(t)[y(t) - C_t\hat{x}(t)] \tag{C.26}$$

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

and

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K_\theta(t)[y(t) - C_t\hat{x}(t)] \tag{C.27}$$

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

Rewriting the equation (C.18) as

$$\begin{pmatrix} K_x(t) \\ K_\theta(t) \end{pmatrix} = \left(\begin{bmatrix} A_t & M_t \\ 0 & I \end{bmatrix}\begin{bmatrix} P_1(t) & P_2(t) \\ P_2^T(t) & P_3(t) \end{bmatrix}\begin{bmatrix} C_t^T \\ D_t^T \end{bmatrix} + \begin{bmatrix} Q^{we} \\ 0 \end{bmatrix}\right)\cdot S^{-1}$$

where

$$S_t = [C_t \quad D_t] \begin{bmatrix} P_1(t) & P_2(t) \\ P_2^T(t) & P_3(t) \end{bmatrix} \begin{bmatrix} C_t^T \\ D_t^T \end{bmatrix} + Q^e$$

$K_x(t)$ and $K_\theta(t)$ can then be obtained

$$K_x(t) = [A_t P_1(t) C_t^T + M_t P_2^T(t) C_t^T + A_t P_2(t) D_t^T + M_t P_3(t) D_t^T + Q^{we}] S_t^{-1} \tag{C.28}$$

$$K_\theta(t) = [P_2^T(t) C_t^T + P_3(t) D_t^T] S_t^{-1} \tag{C.29}$$

and

$$S_t = C_t P_1(t) C_t^T + C_t P_2(t) D_t^T + D_t P_2^T(t) C_t^T + D_t P_3(t) D_t^T + Q^e \tag{C.30}$$

In the same way the equation (C.19) is rewritten as

$$\begin{pmatrix} P_1(t+1) & P_2(t+1) \\ P_2^T(t+1) & P_3(t+1) \end{pmatrix} = \begin{bmatrix} A_t & M_t \\ 0 & I \end{bmatrix} \begin{bmatrix} P_1(t) & P_2(t) \\ P_2^T(t) & P_3(t) \end{bmatrix} \begin{bmatrix} A_t^T & 0 \\ M_t^T & I \end{bmatrix} + \begin{bmatrix} Q^w & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} K_x(t) \\ K_\theta(t) \end{bmatrix} S_t [K_x^T(t) \quad K_\theta^T(t)]$$

The matrices $P_1(t)$, $P_2(t)$ and $P_3(t)$ can be obtained

$$P_1(t+1) = A_t P_1(t) A_t^T + A_t P_2(t) M_t^T + M_t P_2^T(t) A_t^T + M_t P_3(t) M_t^T - K_x(t) S_t K_x^T(t) + Q^w \tag{C.31}$$

$$P_1(0) = \Pi_0(\hat{\theta}_0)$$

$$P_2(t+1) = A_t P_2(t) + M_t P_3(t) - K_x(t) S_t K_\theta^T(t) \tag{C.32}$$

$$P_2(0) = 0$$

$$P_3(t+1) = P_3(t) - K_\theta(t) S_t K_\theta^T(t) \tag{C.33}$$

$$P_3(0) = \Sigma_0$$

By inserting matrix $K_\theta(t)$ into equation (C.32), $P_2(t+1)$ can be rewritten as

$$P_2(t+1) = (A_t - K_x(t) C_t) P_2(t) + (M_t - K_x(t) D_t) P_3(t) \tag{C.34}$$

It could be mentioned that certain numerical problems will arise in the algorithm if $M_t - K_x(t) D_t$ is not a full rank stochastic process (i.e., its covariance matrix is singular). This a question associated with the parameterization of the model (C.12). In such a case, we shall in the sequel assume that some measures are taken to come around these numerical problems. A simple way is to replace (C.33) by

$$P_3(t+1) = [\{P_3(t) - K_\theta(t) S_t K_\theta^T(t)\}^{-1} + \delta I]^{-1} \tag{C.35}$$

for some small positive δ. Notice that for small δ, (C.35) can be approximated by

$$P_3(t+1) = P_3(t) - K_\theta(t) S_t K_\theta^T(t) - \delta P_3(t) P_3(t) \tag{C.36}$$

C.5 Convergence analysis

Convergence of the algorithm (C.26) - (C.33) will be analysed using the ordinary differential equation (ODE) method which is developed by Ljung. We shall in this section determine the differential equation that is associated with the algorithm (C.26) - (C.33).

The differential equation is defined in terms of the process that (C.26) - (C.33) would produce if the model parameters were kept constant= θ. This means that (C.27) would be replaced by $\hat{\theta}(t) = \theta$. Consequently in the θ-dependent matrices $[A_t, B_t, C_t, M_t$ and $D_t]$ the estimate $\hat{\theta}(t)$ should be replaced by θ. It is easy to see that P_2 and P_3 would tend to zero and

$$\bar{P}_1(\theta) = A(\theta)\bar{P}_1(\theta)A^T(\theta) + Q^w - \bar{K}_x(\theta)\bar{S}(\theta)\bar{K}_x^T(\theta) \tag{C.37}$$

$$\bar{S}(\theta) = C(\theta)\bar{P}_1(\theta)C^T(\theta) + Q^e \tag{C.38}$$

$$\bar{K}_x(\theta) = [A(\theta)\bar{P}_1(\theta)C^T(\theta) + Q^{we}]\bar{S}^{-1}(\theta) \tag{C.39}$$

Then define the process $\bar{\hat{x}}(t,\theta)$ as the estimates that would be obtained with this constant model, corresponding to the parameters value θ

$$\bar{\hat{x}}(t+1,\theta) = A(\theta)\bar{\hat{x}}(t,\theta) + B(\theta)u(t) + \bar{K}_x(\theta)\bar{\epsilon}(t,\theta) \tag{C.40}$$

where

$$\bar{\epsilon}(t,\theta) = y(t) - C(\theta)\bar{\hat{x}}(t,\theta) \tag{C.41}$$

It is natural to interpret the EKF as an attempt to minimize the expected value of the squared residuals associated with model θ. A suitable criterion to seek to minimize would be

$$V(\theta) = E|\bar{\epsilon}(t,\theta)|^2 \tag{C.42}$$

A reasonable adjustment scheme to achieve minimization of $V(\theta)$ should be related to the gradient of $V(\theta)$

$$\frac{d}{d\theta}V(\theta) = 2E\left[\left(\frac{d}{d\theta}\bar{\epsilon}^T(t,\theta)\right)\bar{\epsilon}(t,\theta)\right] \tag{C.43}$$

Denote the matrix

$$-\frac{d}{d\theta}\bar{\epsilon}^T(t,\theta) = \bar{\psi}(t,\theta) \tag{C.44}$$

Then the negative gradient of $V(\theta)$ can be written

$$-\frac{d}{d\theta}V(\theta) = 2E\bar{\psi}(t,\theta)\bar{\epsilon}(t,\theta) \tag{C.45}$$

the parameter values of θ would be corrected in this direction.

Differentiating (C.41) gives

$$\frac{d}{d\theta}\bar{\epsilon}(t,\theta) = -\left[\frac{d}{d\theta}C(\theta)\right]\bar{\hat{x}}(t,\theta) - C(\theta)\left[\frac{d}{d\theta}\bar{\hat{x}}(t,\theta)\right] \tag{C.46}$$

where $d/d\theta\bar{\hat{x}}(t,\theta)$ can be found by differentiating (C.40)

$$\begin{aligned}\frac{d}{d\theta}\bar{\hat{x}}(t+1,\theta) = {} & [A(\theta) - \bar{K}_x(\theta)C(\theta)]\frac{d}{d\theta}\bar{\hat{x}}(t,\theta) + \left[\frac{d}{d\theta}A(\theta)\right]\bar{\hat{x}}(t,\theta) \\ & + \left[\frac{d}{d\theta}B(\theta)\right]u(t) - \bar{K}_x(\theta)\left[\frac{d}{d\theta}C(\theta)\right]\bar{\hat{x}}(t,\theta) \\ & + \left[\frac{d}{d\theta}\bar{K}_x(\theta)\right]\bar{\epsilon}(t,\theta)\end{aligned} \tag{C.47}$$

Furthermore we define $\bar{w}(t,\theta) = d/d\theta\bar{\hat{x}}(t,\theta)$

$$\begin{aligned}\bar{w}(t+1,\theta) = {} & [A(\theta) - \bar{K}_x(\theta)C(\theta)]\bar{w}(t,\theta) \\ & + [M(\theta,\bar{\hat{x}}(t,\theta),u(t)) - \bar{K}_x(\theta)D(\theta,\bar{\hat{x}}(t,\theta))] \\ & + \left[\frac{d}{d\theta}\bar{K}_x(\theta)\right]\bar{\epsilon}(t,\theta)\end{aligned} \tag{C.48}$$

and then $\bar{\psi}(t,\theta)$ can be found

$$\bar{\psi}(t,\theta) = [C(\theta)\bar{w}(t,\theta) + D(\theta,\bar{\hat{x}}(t,\theta))]^T \tag{C.49}$$

where matrices M and D were defined by (C.24) and (C.25).

If we remove the column $[d/d\theta\bar{K}_x(\theta)]\bar{\epsilon}(t,\theta)$ from $\bar{w}(t+1,\theta)$ (the reason will be shown later) and compare it with (C.34), it can be easily found that $P_2 \sim \bar{w}(t,\theta)\tilde{P}_3$ for given constant θ. Again, by comparing $\bar{\psi}(t,\theta)$ with $K_\theta(t)$, having in mind that $P_2 \sim \bar{w}(t,\theta)\tilde{P}_3$, we see that $K_\theta(t) \sim \tilde{P}_3\bar{\psi}(t,\theta)\bar{S}^{-1}(\theta)$ for given constant θ and $\tilde{P}_3$.

Since, according to (C.27) and the definition of ODE in Appendix B, it seems reasonable that

$$f(\theta) = E\bar{\psi}(t,\theta)\bar{S}^{-1}(\theta)\bar{\epsilon}(t,\theta) \tag{C.50}$$

where E denotes expectation with respect to the stochastic process $y(t)$ and $u(t)$. Define also

$$G(\theta) = E\bar{\psi}(t,\theta)\bar{S}^{-1}(\theta)\bar{\psi}^T(t,\theta) \tag{C.51}$$

we may thus interpret $f(\theta)$ as the direction (modified by $R^{-1} = \tilde{P}_3$) in which the estimates asymptotically are adjusted.

What we have done until now is to give a formal definition of the functions $f(\theta)$ and $G(\theta)$. Thus the ODE can be obtained by

$$\begin{aligned}\frac{d}{d\tau}\theta(\tau) &= R^{-1}(\tau)f(\theta(\tau)) \\ \frac{d}{d\tau}R(\tau) &= G(\theta(\tau)) + \delta I - R(\tau)\end{aligned} \tag{C.52}$$

if $\theta(t)$ converges to θ^* (θ^* is a stationary point of the differential equation), then

$$f(\theta^*) = 0$$

$$R^* = G(\theta^*) + \delta I \tag{C.53}$$

Now, as seen from the definition of $f(\theta)$ in (C.50), this function is the correlation between the residuals (innovations) $\bar{\epsilon}(t,\theta)$ obtained from model θ and variable $\bar{\psi}(t,\theta)$. This random variable is according to (C.48) and (C.49) obtained by linear filtering of the state estimates corresponding to the same model θ. Therefore $f(\theta)$ is a measure of the correlation between the current innovation and previous state estimates, which in turn can be obtained from previous residuals. Therefore, it is clear that $f(\theta)$ measures the correlation of the sequence $\{\bar{\epsilon}(t,\theta)\}$.

If the sequence $\{\bar{\epsilon}(t,\theta)\}$ is uncorrelated for some $\theta = \theta^*$, then $f(\theta^*) = 0$ and $\theta = \theta^*$, $R = G(\theta^*) + \delta I$ is a stationary point of (C.52). However, the converse is not necessarily true, i.e., $f(\theta^*) = 0$ does not in general imply that $\{\bar{\epsilon}(t,\theta^*)\}$ is a sequence of uncorrelated random vectors. The problem is caused by that we have removed the column $[d/d\theta\bar{K}_x(\theta)]\bar{\epsilon}(t,\theta)$ from $\bar{w}(t+1,\theta)$, when we discuss the convergence properties of the EKF. It means that only in the special case where $\bar{K}_x(\theta)$ happens to be independent of θ, i.e., $[d/d\theta\bar{K}_x(\theta)] = 0$, the convergence properties of the EKF can be satisfied. This can be explained below.

Suppose that for some θ_0,

$$\begin{aligned}A_0 &= A(\theta_0) \\ B_0 &= B(\theta_0) \\ C_0 &= C(\theta_0)\end{aligned} \tag{C.54}$$

where matrices A_0, B_0 and C_0 are the true system matrices, which are obtained for a certain parameter vector θ_0. Then $\{\bar{\epsilon}(t,\theta_0)\}$ will be in general

an orthogonal sequence only if

$$\bar{K}_0 = \bar{K}_x(\theta_0) \tag{C.55}$$

where $\bar{K}_0$ is defined by (C.11) and $\bar{K}_x(\theta_0)$ by (C.39). This condition holds only if the assumptions about noise structure of the model (C.13) are in accordance with those of the true system (C.7). Consequently, a value θ_0 corresponding to the true system will in general be a stationary point of (C.52) and hence a possible convergence point only if the assumed noise structure coincides with the true one. Otherwise the estimates will be biased.

It is of course somewhat unrealistic to assume that the noise structure of the system is known, while the dynamics of the system are unknown. However we might expect the improved asymptotic convergence properties of the EKF if $[d/d\theta\bar{K}_x(\theta)]\epsilon(t)$, where $\epsilon(t)$ is the current residual, is added to the matrix M_t in the algorithm (C.26) - (C.33). Ljung in his paper [Ljung 79b] gives a modified algorithm and a Gauss-Newton algorithm by using innovations model. These two algorithms have the same asymptotic convergence properties. The Gauss-Newton algorithm could be called a recursive prediction error algorithm which is given is Appendix D.

Appendix D

The RPE Method Applied to the Innovations Model

The recursive prediction error (RPE) method applied to an innovations model is given by Ljung in his book *Theory and Practice of Recursive Identification* [Ljung and Söderström 83].

D.1 The model

The effect of the assumptions associated with the noise covariance matrices $Q^w(\theta)$, $Q^e(\theta)$ and $Q^{we}(\theta)$ in state-space model (C.12) - (C.13) in Appendix C is in fact only to provide the Kalman filter gain. It is this gain that has the algorithmic importance. The noise assumptions are only the vehicles to

arrive at it. Therefore in most cases it should be a good idea to parameterize the steady-state Kalman gain rather than the covariance matrices. This will normally involve fewer parameters. The only cases, when this may be undesirable is when important *a priori* information of the noise structure in the model (C.12) is available or when it is important to have a time-varying Kalman gain for the initial part of the recorded data [Ljung 79b].

An innovations model is

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + K_x(\theta)\epsilon(t) \\ y(t) &= C(\theta)x(t) + \epsilon(t) \end{aligned} \tag{D.1}$$

with

$$Ex(0) = x_0(\theta)$$

$$E[x(0) - x_0(\theta)][x(0) - x_0(\theta)]^T = \Pi(\theta)$$

where $\epsilon(t)$ is the prediction error or innovation $y(t) - \hat{y}(t|\theta)$ with variance

$$E\{\epsilon(t)\epsilon^T(s)\} = \Lambda\delta_{ts} \tag{D.2}$$

Instead of model (C.12) - (C.13) in Appendix C, the variance matrices are

$$\begin{aligned} Q^w(\theta) &= K_x(\theta)\Lambda K_x^T(\theta) \\ Q^e(\theta) &= \Lambda \\ Q^{we}(\theta) &= K_x(\theta)\Lambda \end{aligned} \tag{D.3}$$

D.2 The algorithm

The prediction of the output for the model (D.1) - (D.2) is obtained by

$$\begin{aligned} \hat{x}(t+1) &= A(\theta)\hat{x}(t) + B(\theta)u(t) + K_x(\theta)\epsilon(t,\theta) \\ \hat{y}(t|\theta) &= C(\theta)\hat{x}(t) \end{aligned} \tag{D.4}$$

with

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

and the prediction error is

$$\epsilon(t,\theta) = y(t) - \hat{y}(t|\theta) \tag{D.5}$$

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

$$V(\theta) = E\frac{1}{2}\epsilon^T(t,\theta)\Lambda^{-1}\epsilon(t,\theta) \tag{D.6}$$

where Λ is a positive definite covariance matrix.

The first derivative of the criterion is

$$\left[\frac{d}{d\theta}V(\theta)\right]^T = \left[\frac{d}{d\theta}\epsilon(t,\theta)\right]^T \Lambda^{-1}\epsilon(t,\theta) \tag{D.7}$$

where

$$\frac{d}{d\theta}\epsilon(t,\theta) = \frac{d}{d\theta}[y(t) - \hat{y}(t|\theta)] = -\frac{d}{d\theta}\hat{y}(t|\theta) = -\psi^T(t,\theta) \tag{D.8}$$

The second derivative of $V(\theta)$ (the Hessian) can be approximated by

$$\frac{d^2}{d\theta^2}V(\theta) \approx E\psi(t,\theta)\Lambda^{-1}\psi(t,\theta) \tag{D.9}$$

An approximation of the Hessian is often referred to as the *Gauss-Newton direction*, which guarantees positive semidefinite.

If we denote the second derivative approximation by R(t), the Gauss-Newton iteration becomes

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)R(t)^{-1}\psi(t)\Lambda^{-1}\epsilon(t) \tag{D.10}$$

the search direction of the estimation is *Gauss-Newton direction*. $R(t)$ can be computed recursively by

$$R(t) = R(t-1) + \gamma(t)[\psi(t)\Lambda^{-1}\psi^T(t) - R(t-1)] \tag{D.11}$$

$$R(0) = R_0$$

The covariance matrix Λ can be estimated by

$$\hat{\Lambda}(t) = \hat{\Lambda}(t-1) + \gamma(t)[\epsilon(t)\epsilon^T(t) - \hat{\Lambda}(t-1)] \tag{D.12}$$

Since the matrix $\psi(t)$ defined by (D.8) can be represented as

$$\psi(t,\theta) = \left[\frac{d}{d\theta}\hat{y}(t|\theta)\right]^T = \left[\frac{d}{d\theta}[C(\theta)\hat{x}(t,\theta)]\right]^T$$

and we introduce

$$W(t,\theta) = \frac{d}{d\theta}\hat{x}(t,\theta) \tag{D.13}$$

then

$$\psi(t,\theta) = [C(\theta)W(t,\theta) + D(\theta,\hat{x}(t,\theta))]^T \tag{D.14}$$

where

$$D(\hat{\theta},\hat{x}) = \frac{\partial}{\partial\theta}(C(\theta)\hat{x})\bigg|_{\theta=\hat{\theta}(t)} \tag{D.15}$$

We now must find an expression for $W(t,\theta)$. To do this we differentiate (D.4)

$$\begin{aligned} W(t+1,\theta) &= [A(\theta) - K_x(\theta)C(\theta)]W(t,\theta) \\ &\quad + \bar{M}(\theta,\hat{x}(t,\theta),u(t),\epsilon(t,\theta)) - K_x(\theta)D(\theta,\hat{x}(t,\theta)) \end{aligned} \tag{D.16}$$

and $\bar{M}(\hat{\theta}(t),\hat{x}(t),u(t),\epsilon(t))$ will be

$$\bar{M}(\hat{\theta}(t),\hat{x}(t),u(t),\epsilon(t)) = \frac{\partial}{\partial\theta}[A(\theta)\hat{x}(t) + B(\theta)u(t) + K_x(\theta)\epsilon(t)]\bigg|_{\theta=\hat{\theta}(t)} \tag{D.17}$$

The recursive prediction error algorithm applied to the innovations model (D.1) - (D.2) gives the following method:

$$\begin{aligned} \epsilon(t) &= y(t) - \hat{y}(t) \\ \hat{\Lambda}(t) &= \hat{\Lambda}(t-1) + \gamma(t)[\epsilon(t)\epsilon^T(t) - \hat{\Lambda}(t-1)] \\ R(t) &= R(t-1) + \gamma(t)[\psi(t)\hat{\Lambda}^{-1}(t)\psi^T(t) - R(t-1)] \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + \gamma(t)R^{-1}(t)\psi(t)\hat{\Lambda}^{-1}(t)\epsilon(t) \\ \hat{x}(t+1) &= A_t\hat{x}(t) + B_tu(t) + K_t\epsilon(t) \\ \hat{y}(t+1) &= C_t\hat{x}(t+1) \\ W(t+1) &= [A_t - K_tC_t]W(t) + \bar{M}_t - K_tD_t \\ \psi(t+1) &= [C_tW(t+1) + D^T(\hat{\theta}(t),\hat{x}(t+1))]^T \end{aligned} \tag{D.18}$$

where

$$A_t = A(\hat{\theta}(t))$$

$$B_t = B(\hat{\theta}(t))$$

$$C_t = C(\hat{\theta}(t))$$

$$D_t = D(\hat{\theta}(t),\hat{x}(t))$$

$$\bar{M}_t = \bar{M}(\hat{\theta}(t),\hat{x}(t),u(t),\epsilon(t))$$

$$K_t = K_x(\hat{\theta}(t))$$

In this case it is obvious that stability region for the predictor is given by

$$\begin{aligned} D_s &= \{\theta | A(\theta) - K_x(\theta)C(\theta) \textit{ has all eigenvalues} \\ &\quad \textit{strictly inside the unit circle}\} \end{aligned} \tag{D.19}$$

According to Ljung, for the RPE algorithm, the estimate $\hat{\theta}(t)$ steps occasionally out of the stability region will make the algorithm "explode". Hence the RPE algorithm should contain stability monitoring and projection into the stability region. The projection can be implemented in the following way.

1. Choose a factor $0 \leq \mu < 1$
2. Compute $\tilde{\theta}(t) = \gamma(t)R^{-1}(t)\psi(t)\hat{\Lambda}^{-1}(t)\epsilon(t)$
3. Compute $\hat{\theta}(t) = \hat{\theta}(t-1) + \tilde{\theta}(t)$
4. Test if $\hat{\theta}(t) \in D_s$. If yes, go to 6; if no, go to 5
5. Set $\tilde{\theta}(t) = \mu\tilde{\theta}(t)$ and go to 3
6. Stop

In the projection algorithm, the step 4 is to test whether the eigenvalues of the matrix $A(\theta) - K_x(\theta)C(\theta)$ are inside the unit circle. Strictly speaking, the projection algorithm violates the rules of a recursive algorithm, since there is no absolute bound on the number of iterations required. This could be resolved by taking $\mu = 0$. Then a measurement that would take $\hat{\theta}(t)$ out of the stability region is imply ignored.

Appendix E

Tekniske Data for WD34 Wind Turbine

Data for WD34 wind turbine from *Vestas - Danish Wind Technology A/S* are given by *Risø National Laboratory*.

Rotor

Number of blades	3
Rotational axis	Horisontal
Diameter	34 m
Height of rotor axis	32 m
Nominal speed of rotation	35 RPM
Direction of rotor plane	Against wind

Tip speed at 35 RPM	62.65 m/s
Swept area	907.9 m^2
Weight of nacelle and rotor	21000 kg
Rotor inertia	214000 kgm^2
Power regulation	Active pitch regulation

Pitch

Material	Glass fiber
Blade rotation system	Hydraulic
Minimal pitch angle	-2^o
Maximal pitch angle	87^o
Maximal pitch rated	10^o

Gearbox

Exchange ratio	1:28.7	
Efficiency	100%	97.5%
	75%	97.0%
	50%	95.8%
	25%	92.5%

Generator

Type	Asynchronous generator
Rated power	400 kW

Synchronous angular velocity	1010 RPM	
Grid frequency	50 Hz	
Nominal slip	1%	
Number of pole pairs	3	
Generator + brake inertia	$12.6 + 9.5\ kgm^2$	
Efficiency	125%	95.9%
	100%	96.1%
	75%	96.1%
	50%	95.6%
	25%	93.2%

Drive train

Main shaft stiffness	7.87e6 Nm/rad

Tower

Height	32 m
Tower stiffness	1.19e6 kg/s^2
Tower damping	4492.2 kg/s

Operating data

Cut-in wind speed	4 m/s
Cut-out wind speed	25 m/s
Rated power of 400 KW at wind speed	13 m/s

Ph. D. theses from IMM

1. **Larsen, Rasmus.** (1994). *Estimation of visual motion in image sequences. xiv* + 143 pp.

2. **Rygard, Jens Moberg.** (1994). *Design and optimization of flexible manufacturing systems. xiii* + 232 pp.

3. **Lassen, Niels Christian Krieger.** (1994). *Automated determination of crystal orientations from electron backscattering patterns. xv* + 136 pp.

4. **Melgaard, Henrik.** (1994). *Identification of physical models. xvii* + 246 pp.

5. **Wang, Chunyan.** (1994). *Stochastic differential equations and a biological system. xxii* + 153 pp.

6. **Nielsen, Allan Aasbjerg.** (1994). *Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data. xxiv* + 213 pp.

7. **Ersbøll, Annette Kjær.** (1994). *On the spatial and temporal correlations in experimentation with agricultural applications. xviii* + 345 pp.

8. **Møller, Dorte.** (1994). *Methods for analysis and design of heterogeneous telecommunication networks.* Volume 1-2, *xxviii* + 282 pp., 283-569 pp.

9. **Jensen, Jens Christian.** (1995). *Teoretiske og eksperimentelle dynamiske undersøgelser af jernbanekøretøjer.* ATV Erhvervsforskerprojekt EF 435. *viii* + 174 pp.

10. **Kuhlmann, Lionel.** (1995). *On automatic visual inspection of reflective surfaces.* ATV Erhvervsforskerprojekt EF 385. Volume 1, *xviii* + 220 pp., (Volume 2, *vi* + 54 pp., fortrolig).

11. **Lazarides, Nikolaos.** (1995). *Nonlinearity in superconductivity and Josephson Junctions. iv* + 154 pp.

12. **Rostgaard, Morten.** (1995). *Modelling, estimation and control of fast sampled dynamical systems. xiv* + 348 pp.

13. **Schultz, Nette.** (1995). *Segmentation and classification of biological objects. xiv* + 194 pp.

14. **Jørgensen, Michael Finn.** (1995). *Nonlinear Hamiltonian systems. xiv* + 120 pp.

15. **Balle, Susanne M.** (1995). *Distributed-memory matrix computations. iii* + 101 pp.

16. **Kohl, Niklas.** (1995). *Exact methods for time constrained routing and related scheduling problems. xviii* + 234 pp.

17. **Rogon, Thomas.** (1995). *Porous media: Analysis, reconstruction and percolation. xiv* + 165 pp.

18. **Andersen, Allan Theodor.** (1995). *Modelling of packet traffic with matrix analytic methods. xvi* + 242 pp.

19. **Hesthaven, Jan.** (1995). *Numerical studies of unsteady coherent structures and transport in two-dimensional flows.* Risø-R-835(EN) 203 pp.

20. **Slivsgaard, Eva Charlotte.** (l995). *On the interaction between wheels and rails in railway dynamics. viii* + 196 pp.

21. **Hartelius, Karsten.** (1996). *Analysis of irregularly distributed points. xvi* + 260 pp.

22. **Hansen, Anca Daniela.** (1996). *Predictive control and identification - Applications to steering dynamics. xviii* + 307 pp.

23. **Sadegh, Payman.** (1996). *Experiment design and optimization in complex systems. xiv* + 162 pp.

24. **Skands, Ulrik.** (1996). *Quantitative methods for the analysis of electron microscope images. xvi* + 198 pp.

25. **Bro-Nielsen, Morten.** (1996). *Medical image registration and surger simulation. xxvii* + 274 pp.

26. **Bendtsen, Claus.** (1996). *Parallel numerical algorithms for the solution of systems of ordinary differential equations. viii* + 79 pp.

27. **Lauritsen, Morten Bach.** (1997). *Delta-domain predictive control and identification for control. xxii* + 292 pp.

28. **Bischoff, Svend.** (1997). *Modelling Colliding-pulse mode-locked semiconductor lasers. xxii* + 217 pp.

29. **Arnbjerg-Nielsen, Karsten.** (1997). *Statistical analysis of urban hydrology with special emphasis on rainfall modelling.* Institut for Miljøteknik, DTU. *xiv* + 161 pp.

30. **Jacobsen, Judith L.** (1997). *Dynamic modelling of processes in rivers affected by precipitation runoff. xix* + 215 pp.

31. **Sommer, Helle Mølgaard.** (1997). *Variability in microbiological degradation experiments. - Analysis and case stydy. xiv* + 211 pp.

32. **Ma, Xin.** (1997). *Adaptive extremum control and wind turbine control.* *xix* + 293 pp.