Fault Detection and Identification for Wind Turbine Systems: a closed-loop analysis

Stijn Donders

June 2002

Master's Thesis

University of Twente
Faculty of Applied Physics
Systems and Control Engineering

Graduation Committee: Prof. Dr. Ir. M.H.G. Verhaegen
Dr. Ir. V. Verdult
Dr. J.W.J. Verschuur
Ir. N.H. Bergboer

University of Twente
The Netherlands

Printed in The Netherlands
Typeset by the author using LaTeX

"As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality..."

Albert Einstein (1879 - 1955)

Abstract

This thesis assesses the usability of time-domain model-based Fault Detection and Identification (FDI) methods for application to a horizontal axis wind turbine (HAWT) that uses pitch-to-vane control. In particular, two FDI scenarios are considered: the estimation of an unknown actuator gain and an unknown actuator delay.

The wind acts as a disturbance to the system, for which no measurement is available. This, together with the fact that only closed-loop measurements are available, represents the main challenge in this thesis. Two classes of model-based approaches are analyzed: the system identification approach, and the observer-based approach using the Kalman filter. A representative linear HAWT model has been designed and used as a simulation environment to test the various approaches.

It has been shown that no reliable system model can be identified from the input/output data because of the closed-loop measurements, so a system identification approach cannot be used for FDI purposes in closed loop.

Two feasible FDI algorithms that are based on the Kalman filter are presented: the discrete-time Kalman filter and the Interacting Multiple-Model (IMM) estimator. Both algorithms provide a reliable estimate of the wind speed by including it in an augmented system state. They have been applied for online estimation of an unknown actuator gain. The discrete-time Kalman filter requires this gain together with the wind to be included in an augmented system state. The IMM estimator requires a suitable model set that describes a fault-free and a faulty HAWT to be defined instead. The discrete-time Kalman filter has a superior performance for the detection and identification of an unknown pitch actuator gain. The performance of the IMM estimator is slightly lower, but still good enough for FDI purposes. However, the IMM estimator is a more versatile algorithm: it has been applied successfully to estimate an unknown actuator delay, which affects the system behavior in a nonlinear way.

Acknowledgements

This master's thesis is the result of ten months of work at the chair of Systems & Control Engineering, held by Michel Verhaegen. The idea of this project arose from Michel's contacts with Tim van Engelen of the Energy research Centre of the Netherlands (ECN), which resulted in a NOVEM project headed by Vincent Verdult. In the framework of this NOVEM project, I investigated model-based fault detection and identification for wind turbine systems.

First of all, I would like to thank Michel Verhaegen for realizing the importance of this research topic. Michel, you showed me how to do research in the field of systems engineering and made sure that I stayed on the right course. Thank you for your constructive ideas and helpful tips during the course of the research. Vincent, thank you for guiding me through my master's thesis project. With your innovative ideas and critical comments you sometimes managed to keep me off the street, as you would say it, but the result wouldn't have been the same without you. I owe much gratitude to Stoyan Kanev, who on many occasions shared his knowledge of systems engineering with me. He was always there to help me with the implementation of algorithms in Matlab and gave useful tips on a clear formulation of the results. Furthermore, I would like to thank Niek Bergboer for his support on LaTeX and systems identification, and Samir Mešic for his careful review of my first results and his constructive remarks.

I would also like to thank Tim van Engelen and Pieter Schaak from ECN for their help in understanding the field of wind engineering and the physics involved, with a special word of gratitude to Pieter for his large contribution to the realistic wind turbine model. Furthermore, I want to thank Koert Lindenburg from ECN for organizing an introduction course on wind turbine simulation in October, which Vincent and I were able to attend.

In addition, I would like to thank Jeroen Verschuur for being in my graduation committee and for his help with Simulink, and Bas Benschop, who kept the group hardware and software running during the course of the project. I am also grateful to Peter Moll, for our teamwork during several chair courses and for reading the draft of my thesis. Finally, I want to thank my other fellow students Karel Hinnen, Sara van der Hoeven and Benno Aalderink and the former group members Gerard Nijsse, Rufus Fraanje, Hiroshi Oku and Ichiro Jikuya for their company and support.

Stijn Donders
Enschede, June 2002

Contents

List of notations and symbols

The mathematical notations and symbols that have been used in this thesis are listed below.

$\mathbb{N}$	the set of natural numbers (0,1,2,...)
$\mathbb{N}^+$	the set of positive integers (1,2,...)
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^n$	the set of real-valued n-dimensional vectors
$\mathbb{R}^{n \times m}$	the set of real-valued n by m matrices
$\mathbb{I}_n$	the n by n identity matrix
$0_{n \times m}$	the n by m matrix, with all elements equal to zero
$\mathcal{S}$	the discrete model set S
$\in$	belongs to
$=$	equal
$\approx$	approximately equal
$\equiv$	equivalent
min	minimum
max	maximum
arg	argument
log	$^{10}\log$
$E[\cdot]$	statistically expected value
A_{ij}	the (i, j) entry of the matrix A
A^T	the transpose of the matrix A
A^{-1}	the inverse of the matrix A
$A^\dagger$	the Moore-Penrose pseudo inverse of the matrix A
$A^{1/2}$	the symmetric positive definite square root of the matrix A
$\|A\|_2$	the 2-norm of the matrix A
$\hat{A}$	the estimate of the matrix A
$\mathrm{diag}(x)$	an $n \times n$ matrix, whose (i, i)th entry is the ith entry of the n-dimensional vector x.
x_i	the ith entry of the vector x
$\|x\|_2$	the 2-norm of the vector x
$\hat{x}$	the estimate of the vector or variable x
$\dot{x}$	the derivative of the variable x
$\ddot{x}$	the second derivative of the variable x
$\bar{x}$	the mean value of the variable x; used in this thesis to denote the operating point value of a turbine system variable

t	continuous time $t \in \mathbb{R}$
k	discrete time $k \in \mathbb{N}$
z	z-transform parameter
$\Delta(k)$	the unit pulse
A	state transition matrix
B	input to state transition matrix
C	observation matrix
D	direct feed-through matrix
x	the state of a linear system
y	the output of a linear system
u	the input of a linear system
n	system order, dimension of the state
ℓ	dimension of the input
m	dimension of the output
$w \frown (0, Q)$	zero-mean white process noise sequence with covariance matrix Q
$v \frown (0, R)$	zero-mean white measurement noise sequence with covariance matrix R
P	state error covariance matrix
K	Kalman gain

List of abbreviations

ANN	Artificial Neural Network
ECN	Energy research Centre of the Netherlands
EWEA	European Wind Energy Association
FDI	Fault Detection and Identification
FNN	Feedforward Neural Network
FTC	Fault Tolerant Control
HAWT	Horizontal Axis Wind Turbine
IMM	Interacting Multiple-Model
LMS	Least Mean Squares
LTI	Linear Time Invariant
LTV	Linear Time Variant
MIMO	Multiple Input, Multiple Output
MM	Multiple-Model
MOESP	Multi-variable Output Error State sPace
NOVEM	Nederlandse Onderneming Voor Energie en Milieu
ODS	Operational Deflection Shape
PD	Proportional Derivative
PI	Proportional Integral
rpm	rotations per minute
SISO	Single Input, Single Output
SLDV	Scanning Laser Doppler Vibrometer
SLS	Separable Least Squares
VAF	Variance Accounted For

Chapter 1

Introduction

1.1 Background

Fault Detection and Identification (FDI) is a topic that becomes increasingly important in industrial processes, due to growing demands on operational reliability, safety and product quality. The general idea is to use a scheme based on measured process data to detect a fault occurrence in a physical process, e.g. a component deterioration, to isolate the fault location in the process and to identify the time characteristics of the fault.

To provide a framework for this thesis, an overview is given below of current trends in offshore wind engineering and the problems involved in practical realization. More information can be found in Hau (2000) and EWEA (1997). It will be shown that an FDI scheme allows predictive maintenance to be introduced: a component can then be replaced before a system breaks down, which saves money and increases the annual availability and energy yield.

Current trends in offshore wind engineering

The development of offshore wind energy has only just begun. Currently, the world's largest offshore wind farm of 17 MW can be found in Dronten, The Netherlands; however, it merely consists of regular wind turbines with some additional protection against the marine environment, which stand in a shallow inland sea.

As explained in EWEA (1997), placing a wind farm in the open sea yields several benefits. Under offshore conditions, there is a more suitable wind regime for wind turbine energy: the average wind speed is higher, which is energetically more profitable, and there is less turbulence because of the flat surface structure of the sea. Apart from the fish and the fishermen, there are no local residents who complain about the noise and sight of a wind farm. Since sound emission is less of a problem, higher tip speed ratios can be allowed, which increases the energy yield. This extra energy yield is necessary to compensate for the increasing costs: unfortunately, a wind turbine on an offshore location requires higher foundation costs and has higher losses due to electrical transmission from land to shore. Also, the costs for inspection and maintenance are expected to be much higher. While the costs for foundation and transmission are an intrinsic property of the offshore location, a lot can be gained by reducing the inspection and maintenance costs, which therefore is a key objective for offshore development. If the turbine size increases, a lower amount of wind turbines is required to produce the same amount of energy. This decreases the number of maintenance points per megawatt. For successful application of offshore wind turbines, a very challenging goal is therefore to build larger wind turbines, which need less maintenance than current land based turbines.

The reliability of a wind turbine can be increased by making the construction more robust. However, since enlarging the current designs already leads to severe material costs, it is certainly not desirable to make even heavier constructions. To make offshore wind turbines feasible, a lighter construction must be designed that has a higher reliability than the current generation of wind turbines. Lighter and stronger materials have to be developed to achieve this ambitious goal; a large challenge for materials scientists. Another large step forward can be made by control engineers: the design of better maintenance schemes for turbine components can lead to cost reduction and higher availability of the turbine.

The importance of a better maintenance scheme

The main idea is to develop a scheme that monitors the condition of the turbine and its components, which makes it possible to introduce a better maintenance strategy. As described in Caselitz et al. (1994) and Barron (1996), there are three different schemes for component maintenance:

1. Breakdown maintenance, better known as "fix it when it breaks",
2. preventive maintenance, which is based on the average component lifetime, and
3. predictive maintenance, which is based on the operating health of a component.

Nowadays, often preventive maintenance is applied, combined with breakdown maintenance. A component is replaced after the average lifetime; unfortunately, this cannot reduce the number of failures to zero, so in case a failure occurs, it is repaired as quickly as possible. Especially in an offshore turbine, the number of failures must be as low as possible: if a turbine has to be shut down due to a failure, this means that no energy is produced and a maintenance crew has to be sent to repair the damage. The remote offshore location has the result that maintenance costs for offshore sites are much higher than for sites on land. As described in Barron (1996), the lifetime estimate of a component is always on the low side in order to minimize the number of failures. Since individual components may last much longer, this often leads to unnecessary component replacement, so that a lot of money is wasted.

It can be concluded that it is very profitable to monitor the condition of the turbine components and introduce predictive maintenance. To achieve this, component deterioration must be detected in a very early stage, so that measures can be taken to replace the damaged component before a major failure occurs. A successful predictive maintenance scheme means an increased availability of the plant, since the annual number of failures can be reduced, and lower costs for maintenance and materials, since the average component lifetime can be increased.

Fault detection and identification

The above analysis clearly demonstrates the importance of FDI methods in offshore wind turbine systems, since an FDI scheme allows the measurement of component degradation, so that predictive maintenance can be introduced. Though this reduces the annual number of failures, it cannot reduce the number of failures to zero: certain faults have an immediate effect, e.g. a turbine blade failure caused by a stroke of lightning or a collision with a big bird. The FDI scheme must therefore be extended with an Fault-Tolerant Control (FTC) scheme: if a fast method is available to detect a fault occurrence and identify the cause, the turbine can be switched to a safe operation mode to avoid further damage while a maintenance crew is on its way. This increases the annual operation time and the annual energy yield, since for certain failures it might be possible to keep the plant in operation; it then may run at a lower energy yield, while currently the plant is shut down. More information on FTC can be found in Patton (2000) and Blanke et al. (1997), applications can be found in e.g. Noura et al. (2000) and Kanev & Verhaegen (2000).

Fault detection methods can be classified into two categories: model-based methods, which are based on system modeling and model evaluation, and feature-based methods, where faults are detected by analyzing certain features of the measurement signals. Section 2.2 describes this classification in more detail. According to Notash & Moore (2002), a common strategy for FDI is based on sensor redundancy: additional sensors are introduced, so that the correct sensor output can be obtained with a majority-voting scheme, despite a single faulty sensor output. A recent evolution is to introduce artificial sensors, based on both measurements and process modeling. Since the systems behavior is different when a fault occurs, it can be described with a different system model. A sufficiently accurate mathematical model of the supervised process is needed to implement a model-based FDI scheme.

1.2 Problem statement

Considering the above analysis, the following problem statement can be defined:

Develop and test a linear wind turbine model for a Horizontal Axis Wind Turbine (HAWT) that uses pitch-to-vane control, and use it as a simulation environment for model-based Fault Detection and Identification (FDI) methods in the time domain. Analyze to what extent these methods can be applied in an offshore wind turbine under operating conditions.

Wind measurements in practice are not very trustworthy since the wind flow is disturbed by the rotor blades and there are large spatial variations in the rotor-swept area. Therefore, an extra constraint is introduced: no measurement of the wind speed is to be included in the FDI schemes, which means that the driving force on the wind turbine system is unknown.

A short description of a HAWT is given in appendix B. Note that the FDI methods must be applied under operating conditions, so that the closed-loop wind turbine model must be considered. Furthermore, no additional sensors and actuators are allowed: only existing sensors and actuators in a real wind turbine system can be used.

1.3 Organization of the thesis

This thesis is organized as follows:

Chapter 2 gives a short introduction on fault detection and identification. Important terminology is explained, and FDI methods are classified into several categories, which are explained shortly. Furthermore, an overview is given of FDI methods that have successfully been applied to HAWT systems up till now.

Chapter 3 gives a short description of the physics of a HAWT using pitch-to-vane control. Relevant physical equations are linearized in the vicinity of the most important operating point, in which the wind turbine reaches full load operation. A block structure of a HAWT system is defined, and a prototype of a future offshore wind turbine is described. A simulation model of the wind turbine is then designed, and appropriate values of the measurement noise in the system outputs are calculated, so that the wind turbine model can be used as a realistic simulation environment for FDI methods.

Chapter 4 describes two FDI scenarios for a HAWT: the identification of an unknown blade pitch actuator gain and an unknown pitch actuator delay. A fault is defined as a deviation of these process parameters from their normal values. The effect of these faults on the closed-loop system behavior is studied by deriving a state-space representation and viewing the Bode plots of relevant closed-loop transfer functions.

The remainder of this thesis discusses model-based FDI schemes. Two method classes are considered: the system identification approach (Isermann, 1997) and the observer-based approach (Patton & Chen, 1997).

Chapter 5 analyzes the system identification approach, that identifies a system model from a batch of input/output data. A parameter estimation scheme is then used to compare this system model description with the known fault-free system description, so that a process parameter can be estimated.

Chapter 6 describes methods that directly estimate the unknown pitch actuator gain, using the Kalman filter. Knowledge of the nominal system model is assumed, and a reference model is run in parallel to the physical process, which uses the controlled system input to estimate the system's state variables and to reconstruct the system outputs. The unknown variables, the wind speed and the pitch actuator gain, are included in an augmented system state, so that they can be estimated by the discrete-time Kalman filter.

The Kalman filter is also implemented as a least-mean squares (LMS) problem; later on, this problem is rewritten for a moving time window, both as a linear and a nonlinear LMS scheme.

Chapter 7 describes the multiple-model estimation approach, a special class of observer-based estimation. A bank of Kalman filters is designed, each based on a model description of the process, either fault-free or faulty. In a recursive scheme, the probability that each model is in effect is calculated. This yields weighting factors for the state estimates from the filter bank, so that an overall state estimate can be calculated. The fault can then either be found as a weighted combination of the faults associated with the model descriptions in the filter bank, or be defined as the fault associated with the most likely model.

Chapter 8 presents the conclusions of this thesis, and gives recommendations for further research.

Appendix C presents an overview of Matlab and Simulink files that are named throughout this thesis. These files contain the HAWT simulation model, relevant data, functions and algorithm implementations.

Chapter 2

Fault detection and identification

Introduction

Fault detection and identification schemes in industrial processes are becoming increasingly important, because of growing demands for higher product quality, safety and operational reliability. As described in chapter 1, an FDI scheme for an offshore wind turbine system allows a predictive maintenance scheme to be introduced, which saves money and increases the annual energy yield.

As described in section 1.2, the aim of this thesis is to develop model-based FDI methods for application in a wind turbine system under pre-defined operating conditions. As an introduction, some useful terminology for the field of fault detection and identification is explained. A short overview of the existing FDI methods is given, and a literature study is presented on previous applications of FDI methods in wind turbine systems.

2.1 Terminology

In Isermann & Ballé (1997), some useful definitions are suggested for the field of supervision, fault detection and diagnosis. They were discussed and reviewed at the SAFEPROCESS 2000 conference, and published in Isermann & Ballé (2000). A selection of these definitions is presented below, as well as several classifications for faults.

2.1.1 States and signals

Faults and failures are two fundamental concepts in fault-tolerant design.

Fault: an unpermitted deviation of at least one characteristic property or parameter of the system from acceptable/usual/standard condition.

Failure: a permanent interruption of a systems ability to perform a required function under specified operating conditions.

A fault is a physical defect, imperfection or flaw that occurs within the system. This may cause a failure: the non-performance of some action that is due or expected. It is important to remember, that not all system faults lead to a system failure. The aim of fault-tolerant design is, to avoid failures for a selected number of fault scenarios.

According to Basseville & Nikiforov (1993) and Isermann (1997), three classifications of faults can be made. These classifications are listed and explained below.

- **location in the physical system:** sensor, actuator and component faults. Consider a physical process, represented as a discrete-time Linear Time Invariant (LTI) system, such as described in appendix A.2. Neglecting the noise sequences, it is defined as:

  $$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

  with the state matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{\ell \times n}$ and $D \in \mathbb{R}^{\ell \times m}$, the state vector $x \in \mathbb{R}^n$, the input vector $u \in \mathbb{R}^m$ and the output vector $y \in \mathbb{R}^l$, and $k \in \mathbb{N}^+$ the time instant.

  A sensor fault results in a faulty output vector $y(k)$, as the fault appears in the state matrices C and D. An actuator fault results in a faulty control action on the state and output vector: the matrices B and D are affected. A component fault denotes a fault in any system component, other than a sensor or actuator. A component fault may change the dynamics of the physical process: it may result in a different A matrix.

- **mathematical properties:** multiplicative and additive faults. Again consider an LTI system. An additive fault is a fault term which is added to the state vector $x(k)$ or the output vector $y(k)$. In a physical sense, an additive fault appears as a sensor or actuator offset. A multiplicative fault influences one variable in $x(k)$ or $y(k)$ as the product of a fault term and another variable. In a physical sense, multiplicative faults can be used to describe a sensor or actuator degradation. It can also be used to describe component faults: parameter changes within the process.

- **time behavior characteristics:** abrupt, incipient and intermittent faults. As illustrated in figure 2.1, abrupt faults have a step-like behavior: a fault term changes abruptly from the nominal value to a faulty value. Incipient faults have a drift-like behavior: the fault term gradually changes from the nominal value to a faulty value, e.g. in a linear way. Intermittent faults have a temporary effect: the fault term changes from the nominal value to a faulty value, and returns to the nominal value after a short period of time.

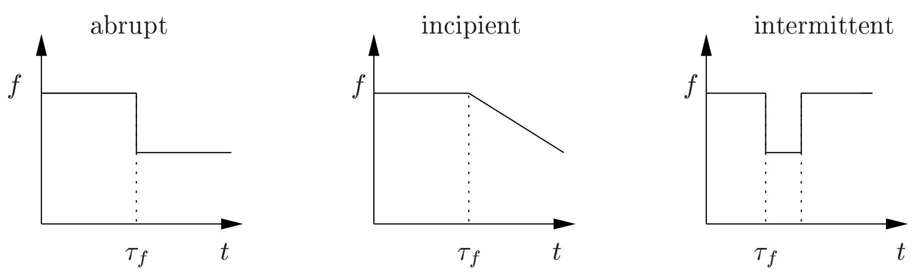

Figure 2.1: *Time-dependency of faults: an example of an abrupt fault, an incipient fault and an intermittent fault. The fault in the feature f occurs at time $t = \tau_f$.*

2.1.2 Functions

A key element in fault-tolerant design is a fault diagnosis scheme. Isermann & Ballé (2000) provide definitions for important functions of such a scheme:

Fault detection: determination of faults present in a system and time of detection.

Fault isolation: determination of kind, location and time of detection of a fault; follows fault detection.

Fault identification: determination of size and time-variant behavior of a fault; follows fault isolation.

Fault diagnosis: determination of kind, size, location and time of a fault; follows fault detection, and includes fault isolation and identification.

In section 2.2, a brief overview of fault detection and identification methods is given.

2.1.3 System properties

Isermann & Ballé (1997) describe several system properties, which are used to compare the performance of fault-tolerant systems:

Reliability: ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time. Measured in mean time between failures.

Safety: ability of a system not to cause danger to persons or equipment nor the environment.

Availability: probability that a system or equipment will operate satisfactorily and effectively at any point in time.

2.2 Fault detection and identification methods

Fault detection and identification methods can be classified by the way process knowledge is incorporated in the signal processing, into model-based methods and signal-based methods. When a process is too complex to be modeled analytically and signal analysis doesn't yield an unambiguous diagnosis, a fault detection approach based on expert systems or artificial intelligence can be used.

2.2.1 Model-based methods

A model-based method uses the system inputs $u(k)$ and outputs $y(k)$. According to Isermann & Ballé (1997), there are three basic categories:

System identification and parameter estimation: using a system identification technique on input/output measurements, process parameters are estimated and compared with a nominal parameter set. The difference, or residual, is used as a fault indicator. Chapter 5 investigates this approach.

State and output observer: an observer, often a Kalman filter, uses knowledge of the nominal system model. A reference model is run in parallel to the observed process, which uses the controlled system input to estimate the system's state variables and reconstruct the system outputs. The residual, defined as the difference between the real and estimated output, can be used as a fault indicator. A clear overview is given in Patton & Chen (1997). Methods based on the Kalman filter are analyzed in chapter 6. A special class of observer-based approaches is the multiple-model estimation approach, described in Rong Li (1996) and analyzed in chapter 7.

Residual generation: primary residuals are formed as the difference between the actual plant outputs and those predicted by the model; these residuals are then subjected to a linear transformation to obtain the desired fault-detection and isolation properties, such as sensitivity to faults, reduced (or no) sensitivity to noise, etc. These methods are not considered in this thesis; a short description is given in Gustafsson (2000), a more thorough description and historical overview can be found in Gertler (1997).

Model-based techniques require a sufficiently accurate mathematical model of the supervised process.

2.2.2 Signal-based methods

A signal-based (or feature-based) method for fault detection is based on analysis of the measured output signals $y(k)$. Suitable features of the measured signals are used to evaluate the operating conditions. These features can be in both time and frequency domain, some examples are the signal mean, variance, skewness, kurtosis, crest factor or the power in a certain frequency band.

Signal-based fault detection methods are particularly interesting for vibration detection. Vibration spectra can be compared to find deviations in rotational frequencies, which may correspond to particular instabilities. However, this thesis only considers model-based fault detection schemes. Information about signal-based methods can be found in e.g. Isermann (1997), Isermann & Ballé (1997) and Gustafsson (2000).

2.2.3 Expert system and artificial intelligence approaches

Sometimes a process is too complex to be modeled analytically and a regular signal analysis approach doesn't yield a reliable FDI scheme, e.g. certain fault combinations have different effects on the system behavior. It is then possible to classify faulty behavior by using qualitative process knowledge to evaluate relations between measured signals and the current operating conditions. These methods are further divided into probabilistic methods, fuzzy logic techniques and artificial neural network approaches; a short description of these methods can be found in Isermann & Ballé (1997), a more detailed overview is given in Leonhardt & Ayoubi (1997). More information on fuzzy techniques can be found in Frank (1990). These methods are also outside the scope of this thesis.

2.3 State of the art of FDI for wind turbine systems

This section gives an overview of FDI methods that have been applied successfully to the main components of a HAWT. These and other components are shortly described in appendix B. The aim of this literature study is to investigate whether model-based FDI schemes have been used in the field of wind turbine engineering.

2.3.1 Rotor blades

Infrared thermography

Smith et al. (1993) state that infrared thermography can be used as an in-field technique for condition monitoring of the wind turbine blades. When the blades experience dynamic mechanical loads, the temperature increases to a certain equilibrium temperature. In an experiment, strip specimens of glass reinforced polyester were subjected to cyclic loads; it was demonstrated, that higher loads correspond to a higher equilibrium temperature and that both stress concentrations and regions of damaged materials give rise to hot spots. It was shown, that the sensitivity of thermal imaging is suitable for non-destructive examination during fatigue testing. Smith et al. (1993) expect that for blades in situ, the wind loads provide enough excitation to create effects that can be detected by thermal imaging.

Vibrational analysis

Ghoshal et al. (2000) test four different algorithms that use vibrational analysis for detecting damage on wind turbine blades. For these experiments, two piezoceramic actuator patches are bonded to the blade. The blade is excited, and the vibrations are measured with four piezoceramic sensors or with a Scanning Laser Doppler Vibrometer (SLDV). Blade damage is modeled by patching an additional weight on the blade, which causes a rotational instability.

Transmittance function monitoring: Schultz et al. (1999) introduced the transmittance function monitoring technique for composite structures, which uses piezoceramic sensors and actuators bonded to these structures. The transmittance function is defined as the ratio of the response cross-spectral density between two sensors r and s and the auto spectral density of a sensor s. It defines how vibration is transmitted between r and s as a function of the frequency. The spectral densities have many peaks and valleys. They misalign when damage occurs, which changes the transmittance function. Ghoshal et al. (2000) made this method better suitable for practical application. The blade is excited with a piezoceramic actuator, and the vibration responses are measured with an SLDV, therefore no additional sensors are required. It was shown in an experiment that damage indication works fine, but that damage location is not always correct. This might be solved by a smaller grid.

Operational deflection shapes: The Operational Deflection Shape (ODS) is the actual shape of the blade structure when subjected to in-service regular dynamic loads. It is established by measuring the three-dimensional motion at critical points on the system. This method doesn't require additional actuator and sensor patches, but a special setup is required to measure the ODS. Ghoshal et al. (2000) use a chirp excitation from the SLDV to obtain the complex vibration response at each measurement point of the blade. The real amplitudes are plotted at selected phase angles, yielding the ODS. In a healthy case, the ODS is symmetric. Once damage occurs, the ODS becomes asymmetric, a clear indication of damage. The validity of this method was verified in an experiment.

Resonance comparison: Ghoshal et al. (2000) compare the response of a healthy and a damaged blade. The piezoceramic actuator is excited by a sine input at a resonance frequency of the blade. The response is measured with the piezoceramic sensors. When a blade is damaged, its resonance frequency changes. Therefore, the resonance response of a damaged blade will be lower than that of a healthy blade; this was verified in an experiment. Ghoshal et al. (2000) consider this method to be the most hopeful method for practical application, since it requires minimal historic data, has a very simple algorithm and can be used in real time. A drawback is, that the method needs both additional sensors and actuators.

Wave Propagation: The wave propagation method uses the same instrumentation as the resonance comparison method. The actuator is excited by an impulse signal instead of a sine wave, and the responses are measured with the piezoceramic sensors. The damage could be indicated in an experiment, but it turned out that the sensitivity is low if the damage is not in the path between actuator and sensor.

2.3.2 Rotor and drive train

Caselitz et al. (1994) study the frequency spectrum of the electric power output of a wind turbine to analyze structural oscillations and drive train vibrations. Order analysis was applied to enhance spectral components that vary with the rotational frequency, and the main peak frequencies were identified, e.g. rotor frequency, blade passing frequency (see section 3.1.1) and blade eigenfrequencies. A lead strip was stuck to the blades to study rotor imbalance, and the resulting spectrum was compared with the spectrum of the fault-free blade. It was found, that the amplitude of the peak at rotor frequency increases linearly with the added mass, while the harmonics are almost unchanged.

2.3.3 Gear box

Elhor et al. (1999) present a diagnosis system for on-line supervision of a wind turbine machine, based on an autoassociator neural network (ANN). A radial and axial accelerometer were installed in the gear box. Data was recorded during normal operation at different wind conditions and mechanical loads. The data is split in batches, of which the frequency spectra are calculated. After feature extraction (choosing the frequencies of interest), this yields spectra containing frequencies that correspond to the contributions of the main rotating components. These spectra are then used to train a neural network: it adapts its internal weight factors in order to memorize the training data samples presented as input, until the learning error becomes smaller than a certain threshold. Thus, the network learns the normal behavior of a wind turbine. Any deterioration of a mechanical component of the gear box changes the spectrum: peaks can shift or resize, or new peaks can appear. Therefore, after the training phase, it is possible to compare the network output and the real output on-line. Any deviation can be interpreted as a machine failure.

2.3.4 Generator

Extensive literature is available on fault diagnosis and condition monitoring of turbine generator systems. Some articles focus on detection of shorts in generator windings that might cause vibrations and lead to mechanical damage of rotor elements. Up till now, these methods havent been applied to a wind turbine generator system; see section 3.1.4 for a brief description of wind turbine generator systems.

Poljakov & Tsvetkov (1999) use thermoresistor measurements for early detection of cooling system deterioration due to leakage or stator overheating in large turbine generators that use water cooling. Since this method deals with generator powers in the order of hundreds of MW, this method is of little relevance to wind turbine generators.

Kulkarni et al. (2000) use the so-called twin signal sensing technique for online detection of incipient faults in field windings of synchronous turbine generator rotors. Two identical pulses are applied to both ends of the winding through capacitors, in order to isolate the high winding voltage from the detection circuits. The reflected pulses are symmetric for an undamaged winding and show asymmetry if the winding is damaged. Therefore, the difference between the reflected signals is further processed as it is a signature for the condition of the winding. The method doesn't require modifications to the generator under test, and on-line experiments gave excellent results.

Megahed & Malik (2000) use a feedforward neural network (FNN) based digital differential relay for fault detection and classification in the stator windings of synchronous generators. Seven generator currents are used as input to two neural networks: one for the fault detection module, which uses a three-neuron output layer to discriminate between normal state, internal and external fault state, and one for the fault classifier module, which is activated in case of an internal fault, and uses a three-neuron output layer to identify three internal phases as faulty or healthy. After the training phase, an output close to one indicates a fault of the associated type. Experiments showed that the designed relay works very well, with outputs greater than 0.9 in case of an external fault while the other outputs are under 0.1. When an internal phase is faulty, the fault classifier maps the corresponding output to a value > 0.95.

Darwish et al. (2001) use ANNs for an on-line fault diagnosis scheme for synchronous generator windings. Three different schemes are tested, the first one similar to that in Megahed & Malik (2000). However, the third scheme proved to be superior in an on-line experiment, both in sensitivity to internal fault detection and stability against external faults. It uses 7 neural networks: three for phase-to-ground faults, three for phase-to-phase faults and one for symmetrical three-phase faults; an output close to one yields a fault of the associated type. The networks have six inputs: three phase currents of both line side and neutral end. This ANN has larger discrimination capabilities, since all the faulty phases as well as the sound phases participate in decision making.

2.4 Conclusions

FDI methods are classified into model-based and signal-based methods. They can be implemented with or without the use of artificial intelligence. The object of this thesis is to design a model-based FDI scheme for a HAWT, which can be used under normal operation conditions.

A study of prior FDI research showed, that model-based FDI has not yet found its way to application in the field of wind turbine engineering. A short description was given of existing FDI schemes for monitoring the condition of various turbine components: rotor blades, drive train, gear box and generator. Only signal-based methods were described in literature, as well as methods based on artificial intelligence. Moreover, most of these methods use additional sensors and actuators, or can only be applied under special measurement conditions. No model-based fault detection scheme was found in literature, not to mention a fault tolerant control scheme.

It can be concluded, that model-based fault detection and identification for wind turbine systems is an open research field. A feasibility study of model-based FDI methods might yield interesting new methods, which can be used to monitor the condition of the components of a HAWT.

Chapter 3

Wind turbine physics

Introduction

Consider an offshore HAWT system with 3 rotor blades that uses pitch-to-vane control; see appendix B for a brief introduction. When the wind passes the rotor swept area, it exerts a torque on the turbine blades. A part of the kinetic energy of the wind flow is therefore converted into rotational energy of the turbine blades and the rotor shaft. The shaft drives a generator, where the rotational energy is converted into electrical energy. The wind also exerts an axial force on the tower head, which together with the torque results in tower vibrations.

In this chapter, a HAWT simulation model is designed. Section 3.1 gives a description of the physics of a HAWT, and presents the physical equations that describe its behavior. Five topics are considered: wind signal properties, aerodynamics, mechanics, electronics and control. In section 3.2, the relevant equations are linearized just above the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$), where the wind turbine reaches full load operation. In section 3.3, the wind turbine system is divided into certain functional parts: it is described as a wind signal fed into a cascade of 3 interacting modules, governed by a controller. Section 3.4 presents a case study of a 6MW HAWT prototype: realistic values are given for the unknown parameters in the physical equations. Using these parameters and the linear equations of section 3.2, the wind turbine modules of section 3.3 can be implemented. This yields a realistic wind turbine simulation model, which is presented in section 3.5. Finally, section 3.6 describes the sensors in a real HAWT. Realistic measurement noise is calculated, to be used in the system outputs of the simulation model.

3.1 Physics

Hau (2000) introduces the basic concepts of wind turbine physics and energy conversion. Van Engelen et al. (2001a,b) give a more thorough description of a HAWT, focusing on the aspects that are relevant for control purposes. This section considers five topics: wind signal physics, aerodynamics, mechanics, electronics and control. The variables that describe the turbine behavior are presented, and the relevant physical equations are presented.

3.1.1 Wind physics

symbol	*[unit]*	*description*
$V_w(t)$	$[m/s]$	rotor-effective wind speed
Ω_r	$[rad/s]$	rotor rotational frequency
H	$[m]$	altitude, measured from earth's surface
$\bar{V}_H$	$[m/s]$	wind speed at altitude H
H^*	$[m]$	altitude, where the wind speed is undisturbed by terrain roughness
V^*	$[m/s]$	the undisturbed wind speed at altitude H^*
α_{ws}	$[-]$	wind shear exponent

In practice, it is never possible to know the exact wind disturbance on the wind turbine. The wind can be described as a wind field, with a different speed at every location in the rotor swept area, and at every time instant. The spatial variations in the wind field are mainly caused by three effects:

Turbulence: According to Hau (2000), wind turbulence consists of short-term fluctuations of the wind speed. The magnitude of the turbulence is approximately proportional to the wind speed; its effect can be seen as a stochastic wind-speed dependent value which is added to the mean wind speed. It has both periodic and non-periodic components, as described in Van Engelen et al. (2001b).

Wind shear: The mean wind speed increases with altitude. Let V^* be the undisturbed mean wind speed at a high altitude H^*, e.g. 1km. The wind flow is affected by the friction against the surface of the earth, so that the wind speed is slowed down from the undisturbed value to zero just above ground. In an engineering approach, an empirical wind shear exponent α_{ws} is specified for each terrain type. A rough surface has a higher α_{ws} than a smooth terrain, see Hau (2000) for some typical values. The mean wind speed at a certain altitude H can then be calculated using

$$\bar{V}_H = V^* \cdot \left(\frac{H}{H^*}\right)^{\alpha_{ws}} \tag{3.1}$$

Tower interference: Hau (2000) describes that when the wind flows around the tower, the internal friction of the flowing medium and the surface friction of the tower body cause an area of disturbed wind flow behind the body, the so-called flow wake area or tower shadow. There is a similar but smaller effect in front of the tower: the tower dam. Since the wind flow has to bend around the tower, the wind speed in front of the tower is a little lower than elsewhere in the rotor swept area. Once every rotation, a blade passes this area with lower wind speed.

The rotating blades sample the wind field as they rotate through the rotor swept area, so that only a small part of the wind field influences the wind turbine system. Measuring the wind speed at one location in the rotor swept area doesn't contain all information that is needed to account for the wind disturbance; moreover, the measured wind flow is disturbed by the rotating blades.

To model the influence of the wind field on the rotor, a one-point wind input signal is used that induces an aerodynamic torque T_a and axial force F_a on the wind turbine system, comparable with the torque and force that a real wind turbine experiences in a wind field. This one-point wind signal is called the rotor-effective wind speed, described in Van Engelen et al. (2001b) as the sum of a mean value and a certain variation. The variation is obtained by a spatial integration over the rotor swept area of the effects of turbulence, the tower interference and the wind shear.

The disturbance caused by tower interference and wind shear depends on the position of the blade. Because of the turning of the rotor, the HAWT experiences disturbance loads with a frequency equal to the 'blade passage frequency' or its multiples. In a first order approximation, these loads can be seen as cyclic loads. For a HAWT with three rotor blades, the blade passage frequency is equal to three times the rotor rotational frequency Ω_r. It is denoted as '$3p$', where p represents the rotor frequency in rotations per minute (rpm).

The sampling of the turbulence field by the rotating blades has the effect that low-frequent energy in the wind spectrum is shifted to multiples of the blade passage frequency in the disturbance spectrum; hence, the rotor-effective wind speed has a higher spectral energy around frequencies '$3p$', '$6p$', etc. The time characteristics of this so-called rotor effective turbulence are obtained by an inverse Fourier transform of the spectrum, where an uniformly distributed random phase angle between 0 and 2π is applied for every frequency point. This realization is then added to the periodic disturbance that is caused by wind shear and tower interference, which yields the rotor-effective wind speed. In Van Engelen et al. (2001b), the design procedure of this wind speed is described in more detail.

A rotor-effective wind signal $V_w(t)$ was developed by ECN, to be used as the wind input signal for the simulation model designed in this thesis; see *wind1p2.mat*. A uniform wind direction is assumed, therefore the yawing system (see appendix B) for turning the rotor into the wind direction is omitted in this thesis.

3.1.2 Aerodynamic equations

symbol	*[unit]*	*description*
C_p	$[-]$	power coefficient
C_t	$[-]$	thrust coefficient
T_a	$[Nm]$	aerodynamic torque on rotor
F_a	$[N]$	axial force on rotor
P_{aero}	$[W]$	aerodynamic power
R_b	$[m]$	rotor blade radius
$\dot{x}_{nd}$	$[m/s]$	nodding speed of the tower head
λ	$[-]$	tip speed ratio
ρ	$[kg/m^3]$	air density
Ω_r	$[rad/s]$	rotor rotational frequency
θ_p	$[rad]$	physical pitch angle
θ_p^d	$[rad]$	pitch angle with a delay
p_1, p_2	$-$	parameters of the pitch delay
$\tau_{i_{DI}}$	$[s]$	integral time constant of lead-lag filter
$\tau_{d_{DI}}$	$[s]$	differential time constant of lead-lag filter

The wind disturbance results in tower vibrations that are described in the nodding-naying reference frame, see figure B.2. The nodding direction is opposite to the wind direction, the naying direction is perpendicular to the nodding direction and parallel to the earth's surface.

Neglecting several details in geometry and aerodynamics, Van Engelen et al. (2001a,b) describe the aerodynamic torque T_a and axial force F_a that are exerted by the wind on the rotor as:

$$T_a = \frac{C_p(\theta_p, \lambda)}{\lambda} \cdot \frac{1}{2}\rho\pi R_b^3 \cdot (V_w - \dot{x}_{nd})^2 \tag{3.2}$$

$$F_a = C_t(\theta_p, \lambda) \cdot \frac{1}{2}\rho\pi R_b^2 \cdot (V_w - \dot{x}_{nd})^2 \tag{3.3}$$

with the tip-speed ratio λ defined as the ratio of the blade tip speed and the effective wind speed, which is the wind speed minus the naying vibration speed $\dot{x}_{nd}$:

$$\lambda = \frac{\Omega_r \cdot R_b}{(V_w - \dot{x}_{nd})} \tag{3.4}$$

The aerodynamic power P_{aero} can be defined as

$$P_{aero} = T_a \cdot \Omega_r \tag{3.5}$$

A common way to control the electrical power output of a wind turbine is by adjusting the pitch angle of the rotor blades. This thesis considers a pitch-to-vane control strategy; see section 3.1.5 for an explanation. To model the aerodynamic effect of the blade pitch actuator on the turbine blades, two effects play an important role: dynamic inflow and pitch delay.

Dynamic inflow is the 3-dimensional effect of the rotor wake on the unsteady aerodynamics of the rotor; a description can be found in Krothapalli et al. (1999). To model the dynamic inflow effect of pitching, the controlled system input θ_p is led through first-order lead-lag filters in order to account for the dynamic influence on the aerodynamic torque T_a and axial force F_a.

For torque variations ΔT_a, the following transfer function applies:

$$\Delta T_a = \frac{\partial T_a}{\partial \theta_p} \Delta\theta_{p,eff}^{T_a}$$

where $\Delta\theta_{p,eff}^{T_a}$ is defined as

$$\Delta\theta_{p,eff}^{T_a} = \frac{\tau_{d_{DI}}^{T_a} s + 1}{\tau_{i_{DI}} s + 1}$$

Similarly, for variations in the axial force F_a, the following holds:

$$\Delta F_a = \frac{\partial F_a}{\partial \theta_p} \Delta\theta_{p,eff}^{F_a}$$

where $\Delta\theta_{p,eff}^{F_a}$ is defined as

$$\Delta\theta_{p,eff}^{F_a} = \frac{\tau_{d_{DI}}^{F_a} s + 1}{\tau_{i_{DI}} s + 1}$$

Up till now, this dynamic inflow effect is not included in the simulation model.

The second dynamic effect in pitching concerns the pitch servo mechanism behavior. This can be modeled accurately enough by a straightforward delay. This delay can be modeled with a second order Padé approximation (3.6), a lead-lag filter with two zeros and two poles.

$$\frac{\theta_p^d}{\theta_p} = \frac{(s-(p_1+j\cdot p_2))\cdot(s-(p_1-j\cdot p_2))}{(s-(-p_1+j\cdot p_2))\cdot(s-(-p_1-j\cdot p_2))} \tag{3.6}$$

A Padé approximation is useful to model time delay effects such as transport and computation delays within the context of continuous-time systems. The Laplace transform of a time delay of T seconds is e^{-sT}. This exponential transfer function is approximated by a rational transfer function using the Padé approximation formulas, as described in Golub & Loan (1989).

3.1.3 Mechanical equations

symbol	*[unit]*	*description*
x_{nd}	$[m]$	nodding displacement of tower head
x_{ny}	$[m]$	naying displacement of tower head
Ω_r	$[rad/s]$	rotor rotational frequency
Ω_r^a	$[rad/s]$	rotor rotational frequency used in aerodynamic conversion
Ω_g	$[rad/s]$	generator rotational frequency, slow shaft equivalent
i_{gb}	$[-]$	gear ratio
J_r	$[kgm^2]$	rotor moment of inertia
J_g	$[kgm^2]$	generator moment of inertia
γ	$[rad]$	shaft distortion
θ_{tilt}	$[rad]$	tilt angle between F_a and $\ddot{x}_{nd}$
F_{hydr}	$[N]$	hydrodynamic load (tower top equivalent)
L_t	$[m]$	tower height
T_l	$[Nm]$	loss torque between rotor and gear box
T_{nac}	$[Nm]$	sideward tower bending torque
T_e	$[Nm]$	generator torque (slow shaft equivalent)
c_d	$[Nm/rad]$	torsional stiffness constant, 1^{st} distortion mode
k_d	$[Nm/(rad/s)]$	torsional damping constant, 1^{st} distortion mode
c_t	$[N/m]$	tower stiffness constant, 1^{st} bending mode
k_t	$[N/(m/s)]$	damping constant, 1^{st} tower bending mode
m_t	$[kg]$	tower top equivalent mass of the 1^{st} order bending mode

Van Engelen et al. (2001a,b) describe the relevant mechanical equations in a wind turbine system. In order to keep them orderly, slow shaft equivalent variables are used to describe the generator behavior. These are the equivalent values on the slow shaft, in front of the gear box, as described in appendix B. The differential equations that describe the tower bending and drive train dynamics are shown in (3.7)-(3.10).

$$\ddot{x}_{nd} = -\frac{k_t}{m_t}\dot{x}_{nd} - \frac{c_t}{m_t}x_{nd} + \frac{F_a}{m_t}\cos\theta_{tilt} \tag{3.7}$$

$$\ddot{x}_{ny} = -\frac{k_t}{m_t}\dot{x}_{ny} - \frac{c_t}{m_t}x_{ny} + \frac{3}{2m_t}\frac{T_{nac}}{L_t} + \frac{F_{hydr}}{m_t} \tag{3.8}$$

$$\ddot{\gamma} = -\frac{k_d(J_r+J_g)}{J_r\cdot J_g}\dot{\gamma} - \frac{c_d(J_r+J_g)}{J_r\cdot J_g}\gamma + \frac{T_a-T_l}{J_r} + \frac{T_e}{J_g} \tag{3.9}$$

$$\dot{\Omega}_r = -\frac{k_d}{J_r}\dot{\gamma} - \frac{c_d}{J_r}\gamma + \frac{T_a-T_l}{J_r} \tag{3.10}$$

In this thesis, the 1^{st} distortion mode of the drive train (with c_d, k_d) and the 1^{st} order bending mode of the tower (with c_t, k_t) are considered. The shaft distortion speed $\dot{\gamma}$ is defined as

$$\dot{\gamma} \quad = \quad \Omega_r - \Omega_g \tag{3.11}$$

T_e is the slow shaft equivalent generator torque. T_{nac} is the sideward tower bending torque, exerted by the drive train on the nacelle at hub height L_t:

$$T_{nac} = \frac{i_{gb} \pm 1}{i_{gb}} J_g \dot{\Omega}_r + T_e. \tag{3.12}$$

Here, the '$\pm$' is a '$+$' for positive naying speed, and a '$-$' for negative naying speed. At a constant rotor speed, $\dot{\Omega}_r = 0$, so (3.12) yields $T_{nac} = T_e$.

The naying speed introduces an extra rotational frequency term Ω_{ny}, to be added to Ω_r everywhere in the system, except in the aerodynamic conversion (3.4), since its influence can be neglected there. From here on, the rotational frequency used in aerodynamics is denoted Ω_r^a. The rotor rotational frequency Ω_r will be described as

$$\Omega_r = \Omega_r^a + \frac{\dot{x}_{ny}}{L_t} \tag{3.13}$$

This is a linear approximation for small vibration amplitudes, using $\phi \approx \tan\phi = x_{ny}/L_t$.

F_{hydr} is the tower top equivalent hydrodynamic load. A detailed expression is given in Van Engelen et al. (2001b). This load is included in the simulation model, but set to zero for now. Effectively, an onshore wind turbine is simulated.

3.1.4 Electrical equations

symbol	*[unit]*	*description*
P_e	$[W]$	electrical power
P_{nom}	$[W]$	nominal electrical power
T_e	$[Nm]$	generator torque (slow shaft equivalent)
T_e^{ref}	$[Nm]$	reference value for generator torque
τ_g	[s]	time constant of (T_e^{ref}, T_e) servo system
K_{opt}	[-]	constant for calculating T_e^{ref} in partial load

The electrical system of a HAWT consists of the generator, which converts the mechanical power into electrical power, and the connection to the grid.

There are two basic generator types, as described in Hau (2000). In an asynchronous (induction) generator, an alternating current is produced in the rotor coils, which rotate through the magnetic field of electromagnetic stators. In a synchronous radiator, a rotating electromagnet is used to induce an alternating current in the stator coils.

Two methods are available to couple the generator output to the grid. Direct grid connection means, that the generator is connected directly to the alternating current grid. Often this is not possible; for instance, if Ω_g doesn't have a constant value, an alternating current is generated with varying frequency, while the grid requires a constant frequency. Indirect grid connection means, that the generator output passes through a series of electrical devices that adjust the current to match the grid frequency.

From here on, these electrical properties are neglected: the generator type and the net coupling are outside the scope of this thesis. No energy losses in the electric system are considered, so the electrical power P_e is defined as:

$$P_e = T_e \cdot \Omega_g \tag{3.14}$$

A simple model will be used to describe the wind turbine generator. A reference value T_e^{ref} that depends on Ω_r is used as input to a linear filter to yield the generator torque T_e:

$$T_e(s) = \frac{1}{\tau_g s + 1} T_e^{ref}(s) \tag{3.15}$$

In figure 3.1, the reference value T_e^{ref} of the generator torque is shown as a function of the generator rotational frequency Ω_g. Below and above the operating point, T_e^{ref} is defined in a different way:

$$T_e^{ref} = \begin{cases} K_{opt} \cdot \Omega_g^2 & \text{for } \Omega_g < \bar{\Omega}_g\text{; Partial load} \\ \frac{P_{nom}}{\Omega_g} & \text{for } \Omega_g \geq \bar{\Omega}_g\text{; Full load} \end{cases} \tag{3.16}$$

Since the partial load and full load characteristics in figure 3.1 intersect in the operating point, the value of K_{opt} is equal to $\frac{P_{nom}}{\bar{\Omega}_g^3}$.

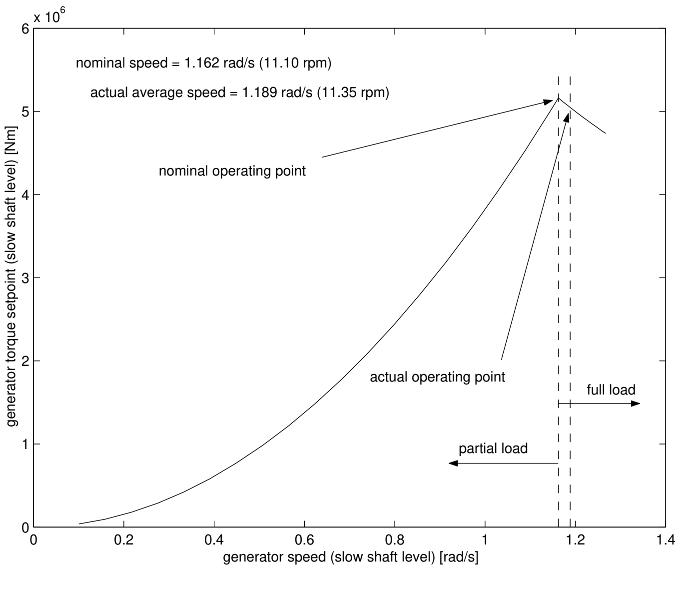

Figure 3.1: *Generator characteristics for partial and full load operation.*

When the generator rotational frequency Ω_g increases, full load is reached at the point of operation ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$); from here on, $\bar{a}$ denotes the operating point value of the parameter or variable a. For higher values of Ω_g, the power P_e is kept at the constant level P_{nom}. According to (3.14), a constant power is obtained if the reference value T_e^{ref} is taken proportional to Ω_g^{-1}.

The actual operating point is chosen just above the nominal operating point, so the generator behavior is modeled in the full load regime; see figure 3.1. This results in a small shift in the turbine parameters (see section 3.4), with no relevant effect on the turbine dynamics. Therefore, this shift will be neglected in this thesis.

3.1.5 Controller

A common way to control the electrical power output of a HAWT is by adjusting the pitch angle of the rotor blades. The pitch angle is defined as the angle between the shape axis and the direction of the wind speed. Hau (2000) describes two general approaches for pitch control. Consider an undesired increase in output power. In pitch-to-vane control, the pitch angle is decreased in order to reduce aerodynamic power capture of the blades. When pitch-to-stall control is applied, the blade pitch angle is increased up to the point where the flow separates at the rotor blades' surface, which creates a stall effect that limits the power capture.

In this thesis, only pitch-to-vane control is considered for control of the power output. A Proportional Derivative (PD) controller is used to control $\dot{\theta}_p$ (which has a Laplace transform of $s\theta_p$), in order to keep the rotor rotational frequency Ω_r close to a desired level Ω_r^{ref}. The controller has the following transfer function:

$$\frac{s\theta_p}{\Omega_r} = K_{PD}(1 + \tau_{PD}s) \tag{3.17}$$

This is equivalent with a Proportional Integrating (PI) controller on θ_p:

$$\frac{\theta_p}{\Omega_r} = K_{PD}(\frac{1}{s} + \tau_{PD}) \tag{3.18}$$

An extra control option is control of T_e to reduce the naying vibrations $\dot{x}_{ny}$; these vibrations are largely caused by T_{nac}, the reaction torque of T_e on the tower head. This control loop was not implemented in this thesis. The primary control loop is visualized in Figure 3.2.

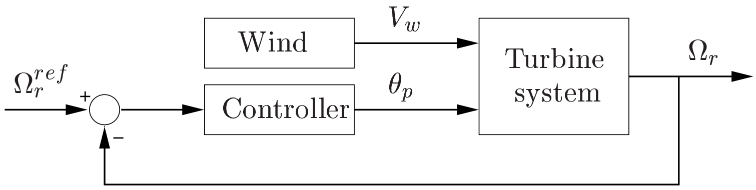

Figure 3.2: *The power control loop for the wind turbine system: in the full load regime, the rotor rotational frequency Ω_r is to be kept at a constant level Ω_r^{ref} by a PI controller on θ_p.*

3.2 Linearization and simplification

In order to fit the described physics in a model representation, it is necessary to simplify a few of the physical equations that are described in section 3.1. The physics describing the aerodynamics, mechanics and electronics will be linearized just above the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$). The variables and parameters are the same as used in section 3.1; new variables and parameters are specified in this section. Note that the value of a variable or parameter a in the operating point is denoted as $\bar{a}$.

3.2.1 Aerodynamic equations

symbol	*[unit]*	*description*
ρ	$[kg/m^3]$	air density
ΔT_a	$[Nm]$	T_a-$\bar{T}_a$
ΔF_a	$[N]$	F_a-$\bar{F}_a$
$\Delta\Omega_r^a$	$[rad/s]$	Ω_r^a-$\bar{\Omega}_r^a$
ΔV_w	$[m/s]$	V_w-$\bar{V}_w$
$\Delta\theta_p$	$[rad]$	$\theta_p - \bar{\theta}_p$

According to Van Engelen et al. (2001b), the aerodynamic conversion equations for aerodynamic torque T_a (3.2) and axial force F_a (3.3) are determined by 4 variables: the pitch angle θ_p, wind speed V_w, tower nodding speed $\dot{x}_{nd}$ and the rotor rotational frequency Ω_r (via tip speed ratio λ). These equations are linearized as a superposition of 4 products, each consisting of a partial derivative in the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$) for the variation of one of the variables described above. This results in the following linearization of the aerodynamic torque T_a:

$$\partial T_a = (T_a)'_{\theta_p}\partial\theta_p + (T_a)'_{\Omega_r}\partial\Omega_r + (T_a)'_{V_w}\partial V_w + (T_a)'_{\dot{x}_{nd}}\partial\dot{x}_{nd}$$

with:

$$\begin{aligned}
(T_a)'_{\theta_p} &= \left.\frac{\partial T_a}{\partial\theta_p}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = \left.\frac{\partial C_q}{\partial\theta_p}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} \cdot \tfrac{1}{2}\rho\pi R_b^3 \cdot \bar{V}_w^2 \\
(T_a)'_{\Omega_r} &= \left.\frac{\partial T_a}{\partial\Omega_r}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = \left.\frac{\partial C_q}{\partial\lambda}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} \cdot \tfrac{1}{2}\rho\pi R_b^4 \cdot \bar{V}_w \\
(T_a)'_{V_w} &= \left.\frac{\partial T_a}{\partial V_w}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = -\left.\frac{\partial C_q}{\partial\lambda}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} \cdot \bar{\lambda} \cdot \tfrac{1}{2}\rho\pi R_b^3 \cdot \bar{V}_w + C_q(\bar{\theta}_p,\bar{\lambda}) \cdot \tfrac{1}{2}\rho\pi R_b^3 \cdot 2\bar{V}_w \\
(T_a)'_{\dot{x}_{nd}} &= \left.\frac{\partial T_a}{\partial\dot{x}_{nd}}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = -(T_a)'_{V_w}
\end{aligned}$$

For the axial force F_a, this yields:

$$\partial F_a = (F_a)'_{\theta_p}\partial\theta_p + (F_a)'_{\Omega_r}\partial\Omega_r + (F_a)'_{V_w}\partial V_w + (F_a)'_{\dot{x}_{nd}}\partial\dot{x}_{nd}$$

with:

$$\begin{aligned}
(F_a)'_{\theta_p} &= \left.\frac{\partial F_a}{\partial \theta_p}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = \left.\frac{\partial C_t}{\partial \theta_p}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} \cdot \tfrac{1}{2}\rho\pi R_b^2 \cdot \bar{V}_w^2 \\
(F_a)'_{\Omega_r} &= \left.\frac{\partial F_a}{\partial \Omega_r}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = \left.\frac{\partial C_t}{\partial \lambda}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} \cdot \tfrac{1}{2}\rho\pi R_b^3 \cdot \bar{V}_w \\
(F_a)'_{V_w} &= \left.\frac{\partial F_a}{\partial V_w}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = -\left.\frac{\partial C_t}{\partial \lambda}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} \cdot \bar{\lambda} \cdot \tfrac{1}{2}\rho\pi R_b^2 \cdot \bar{V}_w + C_t(\bar{\theta}_p,\bar{\lambda}) \cdot \tfrac{1}{2}\rho\pi R_b^2 \cdot 2\bar{V}_w \\
(F_a)'_{\dot{x}_{nd}} &= \left.\frac{\partial F_a}{\partial \dot{x}_{nd}}\right|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w} = -(F_a)'_{V_w}
\end{aligned}$$

This yields the following linearizations (3.19)-(3.19) around the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$).

$$\begin{aligned}
\partial T_a &= (T_a)'_{\theta_p}\partial\theta_p + (T_a)'_{\Omega_r}\partial\Omega_r + (T_a)'_{V_w}(\partial V_w - \partial\dot{x}_{nd}) && (3.19) \\
\partial F_a &= (F_a)'_{\theta_p}\partial\theta_p + (F_a)'_{\Omega_r}\partial\Omega_r + (F_a)'_{V_w}(\partial V_w - \partial\dot{x}_{nd}) && (3.20)
\end{aligned}$$

Based on (3.13), (3.19) and (3.20), the wind turbine model will be implemented with the variables defined as variations around the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$):

$$\begin{aligned}
\Delta T_a &= (T_a)'_{\theta_p}\Delta\theta_p + (T_a)'_{\Omega_r^a}\Delta\Omega_r^a + (T_a)'_{V_w}(\Delta V_w - \dot{x}_{nd}) && (3.21) \\
\Delta F_a &= (F_a)'_{\theta_p}\Delta\theta_p + (F_a)'_{\Omega_r^a}\Delta\Omega_r^a + (F_a)'_{V_w}(\Delta V_w - \dot{x}_{nd}) && (3.22)
\end{aligned}$$

3.2.2 Mechanical equations

symbol	*[unit]*	*description*
ΔT_e	$[Nm]$	T_e-T_e
ΔT_{nac}	$[Nm]$	T_e-$\bar{T}_{nac}$
ΔF_{hydr}	$[N]$	F_{hydr}-$\bar{F}_{hydr}$
$\Delta\Omega_r$	$[rad/s]$	Ω_r-$\bar{\Omega}_r$
$\Delta\Omega_g$	$[rad/s]$	Ω_g-$\bar{\Omega}_g$

In analogy with the aerodynamic equations, the turbine mechanical equations (3.7)- (3.12) will be linearized just above the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$), and the variables will be defined as variations around this operating point. Several approximations are made:

- $\cos(\theta_{tilt}) = 1$, since θ_{tilt} is small;
- $T_l = 0$;
- $\frac{i_{gb}\pm 1}{i_{gb}} = 1$, since typically, the gear ratio $i_{gb} \approx 20$.

This yields the following mechanical equations:

$$
\begin{aligned}
\ddot{x}_{nd} &= -\frac{k_t}{m_t}\dot{x}_{nd} - \frac{c_t}{m_t}x_{nd} + \frac{\Delta F_a}{m_t} && (3.23)\\
\ddot{x}_{ny} &= -\frac{k_t}{m_t}\dot{x}_{ny} - \frac{c_t}{m_t}x_{ny} + \frac{3}{2m_t}\frac{\Delta T_{nac}}{L_t} + \frac{\Delta F_{hydr}}{m_t} && (3.24)\\
\ddot{\gamma} &= -\frac{k_d(J_r + J_g)}{J_r \cdot J_g}\dot{\gamma} - \frac{c_d(J_r + J_g)}{J_r \cdot J_g}\gamma + \frac{\Delta T_a}{J_r} + \frac{\Delta T_e}{J_g} && (3.25)\\
\Delta\dot{\Omega}_r &= -\frac{k_d}{J_r}\dot{\gamma} - \frac{c_d}{J_r}\gamma + \frac{\Delta T_a}{J_r} && (3.26)\\
\dot{\gamma} &= \Delta\Omega_r - \Delta\Omega_g && (3.27)\\
\Delta T_{nac} &= J_g \cdot \Delta\dot{\Omega}_r + \Delta T_e. && (3.28)\\
\Delta\Omega_r &= \Delta\Omega_r^a + \frac{\dot{x}_{ny}}{L_t} && (3.29)
\end{aligned}
$$

The hydrodynamic load ΔF_{hydr} is included in the model setup, but in the remainder of this thesis, its value is set to zero.

3.2.3 Electrical equations

symbol	*[unit]*	*description*
ΔT_e^{ref}	$[Nm]$	T_e^{ref}-$\bar{T}_e^{ref}$

The generator behavior just above the operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$) can be found by linearizing the reference value T_e^{ref} of the generator torque T_e (3.16) in full load operation:

$$\partial T_e^{ref} = \frac{\partial T_e^{ref}}{\partial \Omega_g}\bigg|_{\bar{\Omega}_g} \cdot \partial\Omega_g = \frac{\partial P_{nom}}{\partial \Omega_g}\bigg|_{\bar{\Omega}_g} \cdot \partial\Omega_g \tag{3.30}$$

Since P_{nom} is a constant, this yields:

$$\Delta T_e^{ref} = -\frac{P_{nom}}{\bar{\Omega}_g^2} \cdot \Delta\Omega_g \tag{3.31}$$

In a real wind HAWT, the electrical power P_e is a measured output. However, if losses in the electrical system are neglected, P_e can be calculated directly from the measurements of the rotational frequency Ω_g by linearizing (3.14), which yields:

$$\Delta P_e = \Delta T_e \cdot \bar{\Omega}_g + \bar{T}_e \cdot \Delta\Omega_g \tag{3.32}$$

3.3 Block structure

In the previous sections, the physics of a wind turbine system was described, simplified and linearized. In this section, a general system model structure for a HAWT is presented, in analogy with Molenaar (2000). In this model representation, the wind is modeled as an input signal that is fed into a cascade of 3 modules: the aerodynamic, mechanical and electrical module. These module blocks are governed by a 4th module: the controller. The linearized physics can be implemented into this block structure. Figure 3.3 shows the modules described by Molenaar (2000), with the input/output relations between these modules.

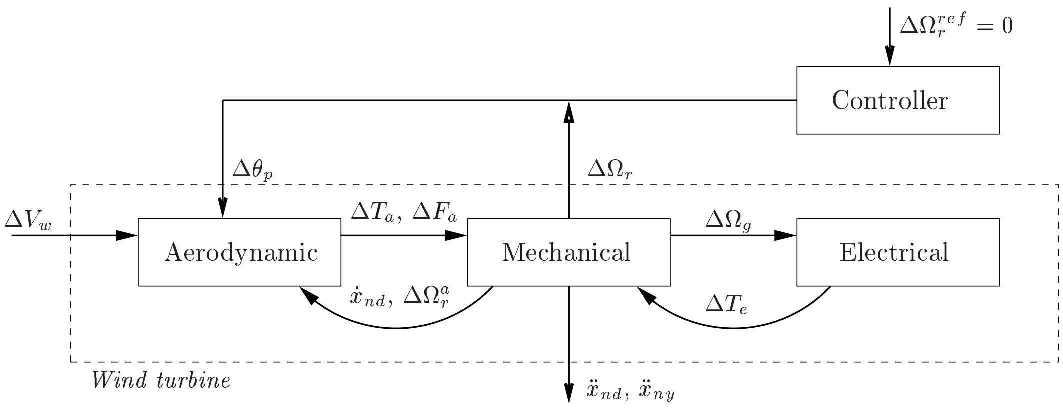

Figure 3.3: *HAWT block structure in analogy with Molenaar (2000): the wind turbine is modeled as a cascade of three modules, with the wind speed ΔV_w as an unknown disturbance input. A PI controller is used to control the output $\Delta\Omega_r$: the pitch angle $\Delta\theta_p$ is adjusted in order to compensate for variations in $\Delta\Omega_r$.*

The dotted block represents the system boundary. Arrows between the modules are within this block; these are internal system variables. The arrows that cross the system boundary represent the input/output variables of the wind turbine system:

Unknown input:	ΔV_w
Input by controller:	$\Delta\theta_p$
Controlled output:	$\Delta\Omega_r$
Other measured outputs:	$\ddot{x}_{nd}$, $\ddot{x}_{ny}$

The PI controller (3.18) is implemented to compensate for variations in $\Delta\Omega_r$, by adjusting the pitch angle $\Delta\theta_p$.

3.4 Case presentation: model parameters

The wind turbine simulation model is designed to represent a prototype of a 6 MW wind turbine. Wind turbines of this size are a topic of current research, as they might be realized offshore in the near future. This section presents realistic values for the unknown parameters in the physical equations and linearizations in the previous sections, as designed by ECN.

Tower and drive train properties:

symbol	*[unit]*	*description*
m_t	$[kg]$	$4.9 \cdot 10^8$
L_t	$[m]$	95
R_b	$[m]$	64.5
c_t	$[N/m]$	$2.0 \cdot 10^6$
k_t	$[N/(m/s)]$	$1.0 \cdot 10^3$
c_d	$[Nm/rad]$	$3.3 \cdot 10^8$
k_d	$[Nm/(rad/s)]$	$3.1 \cdot 10^6$
J_r	$[kgm^2]$	$2.6 \cdot 10^7$
J_g	$[kgm^2]$	$2.9 \cdot 10^6$

Operating point values:

symbol	*[unit]*	*description*
ρ	$[kg/m^3]$	1.125
P_{nom}	$[W]$	$6 \cdot 10^6$
F_{hydr}	$[N]$	0, for now
$\bar{F}_a$	$[N]$	$5.54 \cdot 10^5$
$\bar{T}_a$	$[Nm]$	$5.346 \cdot 10^6$
$\bar{T}_e$	$[Nm]$	$5.346 \cdot 10^6$
$\bar{\Omega}_r$	$[rad/s]$	1.162
$\bar{\Omega}_g$	$[rad/s]$	1.162
$\bar{V}_w$	$[m/s]$	14.53
$\bar{\dot{x}}_{nd}$	$[m/s]$	0
$\bar{\theta}_p$	$[rad]$	0.136
$\left.\frac{\partial C_q}{\partial \theta_p}\right\|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w}$	$[1/rad]$	-0.290
$\left.\frac{\partial C_q}{\partial \lambda}\right\|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w}$	$[-]$	$-3.74 \cdot 10^{-3}$
$C_q(\bar{\theta}_p, \bar{\lambda})$	$[-]$	$4.9 \cdot 10^{-2}$
$\left.\frac{\partial C_t}{\partial \theta_p}\right\|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w}$	$[1/rad]$	-2.1142
$\left.\frac{\partial C_t}{\partial \lambda}\right\|_{\bar{\theta}_p,\bar{\lambda},\bar{V}_w}$	$[-]$	$-1.73 \cdot 10^{-3}$
$C_t(\bar{\theta}_p, \bar{\lambda})$	$[-]$	0.325

Delay implementations and controller parameters:

symbol	*[unit]*	*description*
τ_g	$[s]$	0.1
p_1	$[-]$	15
p_2	$[-]$	8.660254
K_{PD}	$[-]$	$-4.02508 \cdot 10^{-2}$
τ_{PD}	$[s]$	14.254

Table 3.1: *Input and output signals of the wind turbine simulation model.*

Inputs and Outputs	*symbol*	*[unit]*	*description*	*Simulink variable*
disturbance input	ΔV_w	$[rad/s]$	V_w-$\overline{V}_w$	`deltaVW`
controlled input	$\Delta\theta_p$	$[rad]$	θ_p-$\overline{\theta}_p$	`deltaTH`
measured output	$\ddot{x}_{nd}$	$[m/s^2]$	$\ddot{x}_{nd}$	`ddxnd`
measured output	$\ddot{x}_{ny}$	$[m/s^2]$	$\ddot{x}_{ny}$	`ddxny`
controlled output	$\Delta\Omega_r$	$[rad/s]$	Ω_r-$\overline{\Omega}_r$	`deltaWR`

3.5 Simulink implementation

Using the linearized physics of section 3.2 and the model parameters of section 3.4, a linear HAWT system model was implemented in Simulink, see the model file *versie2.mdl*. Table 3.1 shows the input and output variables of the turbine model. The constants and parameters presented in section 3.4 are stored in the file *constants.m*.

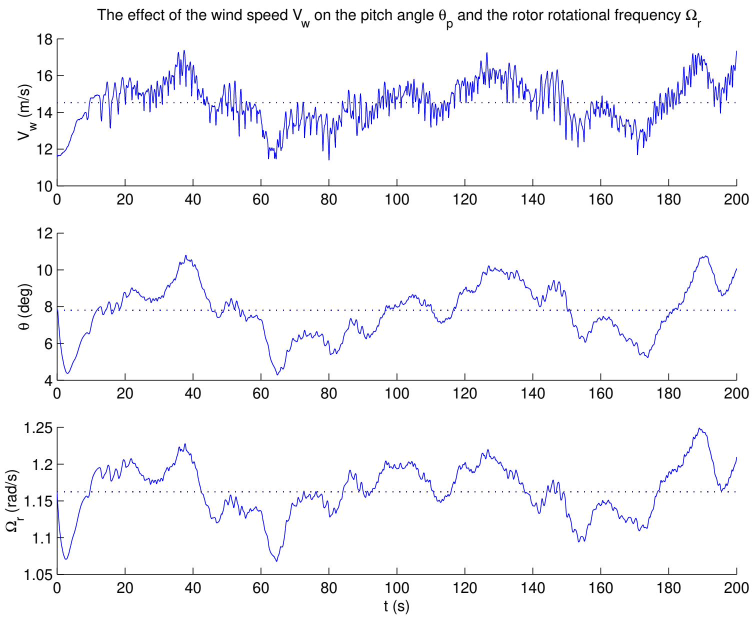

Figure 3.4: *Simulation results - a realistic wind speed V_w is used as an input to the turbine system. The resulting pitch angle θ_p and the rotor rotational frequency Ω_r are shown. The dotted lines are the operating point values $\bar{V}_w$, $\bar{\theta}_p$ and $\bar{\Omega}_r$, respectively. No measurement noise in the system outputs.*

Figure 3.4 shows the closed-loop simulation results of the wind turbine model: a plot of the wind speed V_w (disturbance input), the pitch angle θ_p (controlled input) and the rotor rotational frequency Ω_r (controlled output). The operating point values are represented by the dotted lines.

The behavior can roughly be explained as follows. If the wind speed increases, the aerodynamic torque T_a on the rotor blades increases, so that Ω_r increases. The control objective is to to keep Ω_r constant: an increase of Ω_r therefore leads to an increase of the pitch angle θ_p.

As a result, the turbine blades experience a smaller aerodynamic torque T_a, so that the rotor rotational frequency Ω_r decreases to the desired level. This results in a realistic variation of Ω_r of $\pm 10\%$ around the desired value $\bar{\Omega}_r$.

Figure 3.5 shows a plot of the nodding acceleration $\ddot{x}_{nd}$ and the naying acceleration $\ddot{x}_{ny}$ during the first 40 seconds of the simulation shown in figure 3.4. The low amplitude of the naying vibrations indicates that they are only weakly excited. A reason is, that the hydrodynamic load F_{hydr} is taken zero in the simulations.

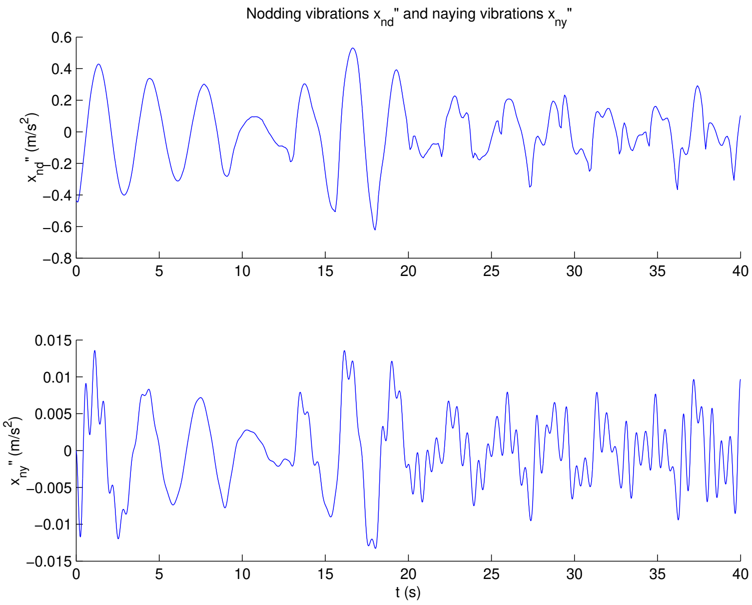

Figure 3.5: *Simulation results - For the first* 40 *seconds of the simulation shown in figure 3.4, the nodding acceleration* $\ddot{x}_{nd}$ *and naying acceleration* $\ddot{x}_{ny}$ *are shown. Note the difference in amplitude: the naying vibrations are much smaller, since they are only weakly excited. Note that the nodding and naying vibrations have the same eigenfrequencies, since the tower stiffness and elasticity is equal in both nodding and naying direction. No measurement noise in the system outputs.*

The continuous-time state matrices of the wind turbine system model have been found with Matlab's *linmod* command. These state matrices describe the wind turbine system as a continuous LTI system of order 10. The system was found to be stable, which is logical from a physical point of view: an unstable system could produce an infinite amount of energy. The balanced state-space realization can be calculated, which balances the values of the state matrix elements and thus decreases the numerical errors in calculations. This balanced realization can then be discretized with a sample time of 0.1 seconds, yielding a realistic discrete-time LTI system representation of the HAWT. In the remainder of this thesis, the discrete-time state matrices will be used to simulate the wind turbine in Matlab, together with a discretized version of the PI controller.

3.6 Sensors of the wind turbine system

In this section, typical sensors are described which are used to measure the outputs in a real HAWT. Based on these sensors, realistic values are calculated of the amount of measurement noise in the outputs of the simulation model presented in section 3.5. Zero-mean white measurement noise is assumed.

3.6.1 Measurement of rotor rotational frequency

In a real HAWT, the generator rotational frequency Ω_g^{fse} is measured on the fast shaft, hence the superscript fse for fast shaft equivalent. If Ω_g^{fse} is divided by the gear ratio i_{gb}, the slow shaft equivalent generator rotational frequency Ω_g^{sse} is obtained. In this thesis, the superscript sse is removed: Ω_g^{sse} is denoted as Ω_g.

The measured signal Ω_g is processed to remove $3p$ disturbance contributions (caused by turbulence, tower passage and wind shear effects, see section 3.1) and contributions of the drive train vibrations. A reconstruction $\hat{\Omega}_r$ of the low frequency behavior of the rotor rotational frequency Ω_r is thus obtained, which is used for turbine control. The only difference between the generator rotational frequency Ω_g and the rotor rotational frequency Ω_r is caused by drive train vibrations, which manifest itself in Ω_g: the generator oscillates around the rotor shaft, because of the finite stiffness of the shaft. Since in a real wind turbine a good reconstruction of Ω_r is used for control, it is a logical choice to use the signal Ω_r directly as the controlled output of the turbine model, as has been implemented in the simulation model in section 3.5.

The generator rotational frequency Ω_g is measured with a position encoder. A thorough calculation of the measurement noise is presented in Van Engelen et al. (2001b); relevant equations are shown below. Since the encoder creates N pulses per rotation, the position is measured with a resolution of

$$\mathcal{R}_{\phi_g} = \frac{2\pi}{N} \text{ rad} \tag{3.33}$$

The standard deviation $\sigma_{\epsilon^q_{\phi_g^m}}$ of the position measurement is equal to

$$\sigma_{\epsilon^q_{\phi_g^m}} = \sqrt{1/12} \cdot \mathcal{R}_{\phi_g} = \sqrt{1/3}\frac{\pi}{N} \text{ rad} \tag{3.34}$$

Ω_g is measured by numerical differentiation of the position measurement over a sample period ΔT:

$$\Omega_g(k\Delta T) = \frac{\phi_g^m(k\Delta T) - \phi_g^m((k-1)\Delta T)}{\Delta T} \tag{3.35}$$

The measurement noise in Ω_g is obtained from numerical differentiation of the measurement error $\epsilon^q_{\phi_g^m}$:

$$\epsilon_{\Omega_g^m}(k\Delta T) = \frac{\epsilon^q_{\phi_g^m}(k\Delta T) - \epsilon^q_{\phi_g^m}((k-1)\Delta T)}{\Delta T} \tag{3.36}$$

This yields, that the measurement Ω_g^m of the generator rotational frequency Ω_g has a standard deviation $\sigma_{\epsilon^q_{\Omega_g^m}}$ equal to

$$\sigma_{\epsilon^q_{\Omega_g^m}} = \sqrt{2} \cdot \frac{\sigma_{\epsilon^q_{\phi_g^m}}}{\Delta T} = \sqrt{2/3} \cdot \frac{\pi}{N\Delta T} \text{ rad/s} \tag{3.37}$$

A typical value of the number of pulses N per rotation is $2^{12} = 4096$. The operating point value $\bar{\Omega}_g$ of the generator rotational frequency Ω_g is 1.16 rad/s, as can be seen in section 3.4. The sample time ΔT is equal to 0.1 s. For the measurement noise on Ω_g, the standard deviation $\sigma_{\epsilon^q_{\Omega^m_g}}$ can now be found:

$$\sigma_{\epsilon^q_{\Omega^m_g}} = \sqrt{2/3} \cdot \frac{\pi}{4096 \cdot 0.1} = 0.0063 \text{ rad/s} \tag{3.38}$$

If the sampling time is not constant enough, this can be taken into account with an additional measurement noise contribution. This is neglected in this thesis.

Measurement noise has high-frequent components, and therefore the processing that is performed to reconstruct the low-frequent control signal will reduce the noise contribution. Therefore, if necessary, the model can also be simulated with less measurement noise to take the noise reduction by filtering into account.

According to Van Engelen et al. (2001b), a sensible measurement of Ω_g can only be obtained if the sample time doesn't exceed the time it takes for the fast generator axis to complete one revolution. This yields a maximum to the sample time allowed in a real wind turbine setup. In the turbine model described in section 3.5, the slow shaft equivalent Ω_g has a mean value of 1.16 rad/s, which can rise up to 1.4 rad/s when a wind gust occurs. Therefore, with a typical gear ratio of 20, a maximal fast shaft equivalent Ω_g^{fse} of 28 rad/s can occur. This yields:

$$\Delta T \leq \frac{2\pi}{\Omega^{fse}_{g_{max}}} \leq 0.22 \tag{3.39}$$

In the simulation model, a sample time of 0.1 s is used, which satisfies (3.39).

3.6.2 Measurement of tower vibration accelerations

The vibrations in the tower head are measured with a digital wind turbine structural vibration monitor, such as the PCH 1026. The product data (PCH Engineering, 1993) of this type of monitor are used to calculate a typical measurement noise value that can be used in the simulations.

In the product data (PCH Engineering, 1993), it can be found that all accelerometers measure a statistical spectral noise density of 500 μg/$\sqrt{\text{Hz}}$ over the signal bandwidth. The sample time of the wind turbine system is 0.1 s (corresponding to 10 Hz). The use of an anti-aliasing filter is assumed, therefore the noise bandwidth is equal to 5 Hz, half the bandwidth associated with the sample time. Parseval's theorem can then be used, stating that the noise energy of the time domain signal $x(t)$ and the frequency domain signal $X(\omega)$ must be equal:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega \tag{3.40}$$

Since $X(\omega) = 500$ μg/$\sqrt{Hz}$, the noise energy contained in the frequency domain is equal to $(X(\omega))^2 \cdot 5$. This yields the standard deviation of the measurement noise on the vibration acceleration measurements: 0.011 m/s^2.

When the measurements of the tower vibrations are used for turbine control, a Chebychev band filter around the first tower eigenfrequencies is applied to reduce the noise contribution, as described in Van Engelen et al. (2001b). Since this control loop was not implemented in this thesis, such a filter was omitted here.

3.6.3 Simulations with standard measurement noise

From here on, the noise values calculated in the previous sections will be referred to as standard noise conditions: zero-mean white noise with a standard deviation of 0.011 m/s^2 for the noise in the tower vibration accelerations, and a standard deviation of 0.006 rad/s for the noise in the rotor rotational frequency Ω_r. The simulation experiment shown in figures 3.4 and 3.5 was repeated, with standard measurement noise on the system outputs. The simulation results are shown in figures 3.6 and 3.7.

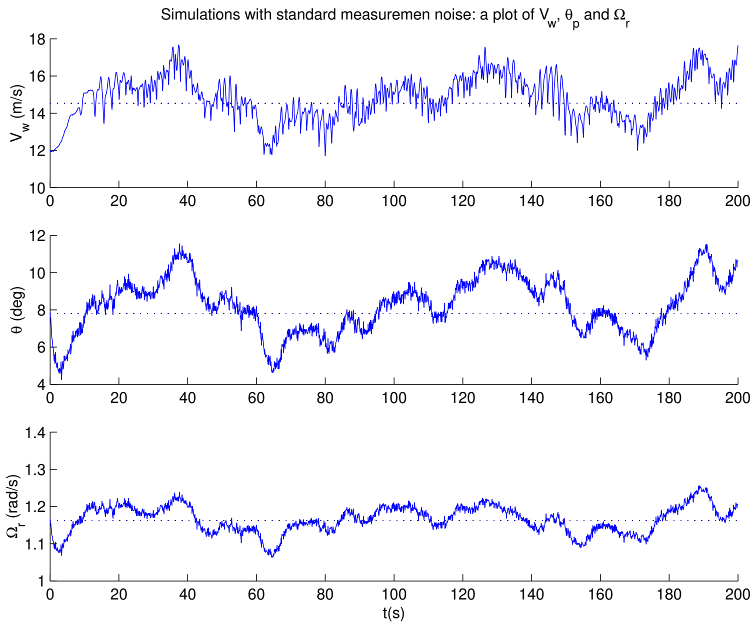

Figure 3.6: *Same simulation as in figure 3.4, but with standard measurement noise present in the system outputs: the wind speed V_w, the pitch angle θ_p and the rotor rotational speed Ω_r as a function of time, with the dotted lines representing the operating point.*

Figure 3.6 shows that the measurement noise on Ω_r results in a noisy controlled system input θ_p. In reality, a filtered version of the measured output might be used to reduce this noise on the control signal; this was not implemented in this thesis.

Figure 3.5 shows that the naying vibration accelerations $\ddot{x}_{ny}$ are excited very weakly, due to the fact that F_{hydr} was set to zero. In figure 3.7, it is clear that this results in a very bad signal-to-noise ratio: the amplitude of $\ddot{x}_{ny}$ and the variance of the measurement noise are almost equal. The naying vibrations are not used for analysis in the remainder of this thesis, as the measurements will be too noisy in the defined simulation setup.

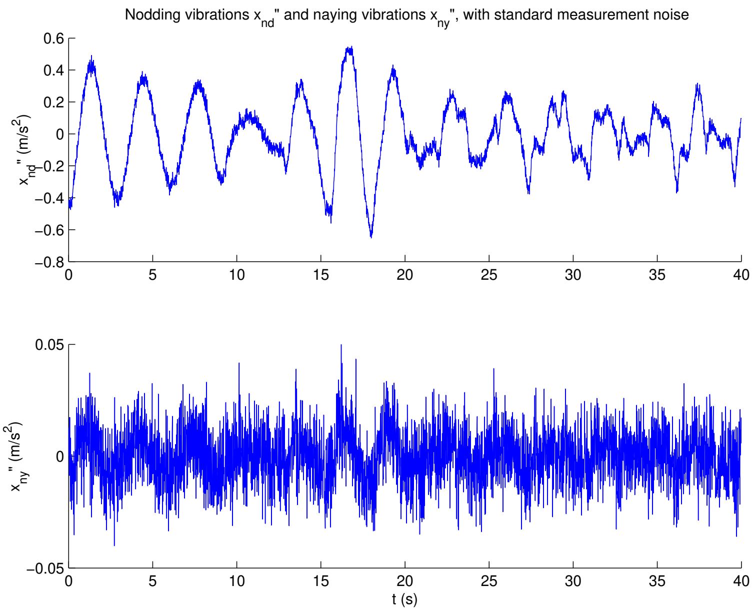

Figure 3.7: *For the first* 40 *seconds of the simulation shown in figure 3.6, with standard measurement noise present in the system outputs, the nodding acceleration* $\ddot{x}_{nd}$ *and naying acceleration* $\ddot{x}_{ny}$ *are shown. Since the naying vibrations are weakly excited in the simulation setup, the signal-to-noise ratio is very bad.*

3.7 Conclusions

In this chapter, a simulation model of a HAWT with pitch-to-vane control was derived. The physics were presented that describe such a wind turbine, and the relevant equations were linearized around the most important operating point ($\bar{V}_w$, $\bar{\lambda}$, $\bar{\theta}_p$), where the HAWT reaches full load operation. A block representation of a wind turbine system was presented, with modules for aerodynamics, mechanics and electronics that are governed by a controller. For a prototype of a 6 MW wind turbine, realistic values of the physical parameters are given.

Using all this information, a wind turbine simulation model was designed in Simulink, with a controlled output Ω_r and measured vibration outputs $\ddot{x}_{nd}$ and $\ddot{x}_{ny}$. The sensors of a realistic wind turbine were described, and realistic values of the measurement noise in the outputs were calculated. It was shown that the measurement of $\ddot{x}_{ny}$ is weakly excited and has a bad signal-to-noise ratio in the defined simulation setup; it will not be used for analysis.

The continuous-time state-space representation of the HAWT model with the two outputs Ω_r and $\ddot{x}_{nd}$ was saved in *turbmat_vib.mat*. In the remainder of this thesis, the balanced discrete-time state-space representation of this continuous-time HAWT is used to simulate the wind turbine system directly in Matlab. The HAWT has stable eigenvalues, and can be simulated with the function *simulator.m*.

Chapter 4

Two fault scenarios

A common way to control the electrical power output of a wind turbine is by adjusting the pitch angle of the rotor blades. As described in section 3.1.5, this thesis only considers pitch-to-vane control. In this chapter, two fault detection and identification (FDI) problems are introduced: the identification of an unknown blade pitch actuator gain, and the identification of an unknown pitch delay. These FDI problems will be studied by simulating the wind turbine model derived in chapter 3. For each problem, a block diagram is presented to show how the fault appears in the wind turbine system. The closed-loop behavior is analyzed by deriving the state-space representation of the closed-loop system and presenting relevant Bode plots of the closed-loop system.

4.1 Detection of an unknown pitch actuator gain

This case study deals with the identification of an unknown blade pitch actuator gain in wind turbines. The unknown pitch actuator gain can be included in the model description by introducing a scaling factor σ on the output of the controller. The nominal system model is then represented with $\sigma = 1$, a faulty model with $0 < \sigma < 1$. For instance, in a wind turbine configuration with three blades, failure of one of the blade pitch actuators results in a reduction of the total actuator gain by a factor 2/3. The objective is to develop an FDI scheme which is able to estimate this unknown scaling factor.

4.1.1 The wind turbine system with an unknown pitch actuator gain

In close cooperation with ECN, a simulation model of a HAWT using pitch-to-vane control was derived as a linearization of the relevant physical equations in the vicinity of the most common operating point. A detailed description can be found in chapter 3, a short overview is given below. All simulation experiments take place in closed-loop; the inputs and outputs of the wind turbine model are described in chapter 3, see table 3.1. In pitch-to-vane control, the variations in rotor rotational frequency $\Delta\Omega_r$ are controlled by adjusting the pitch angle $\Delta\theta_p$. The goal is to keep Ω_r at a constant value, despite the changes in the wind speed ΔV_w. The controller that is used is the PI controller (3.18) on θ_p, designed by ECN.

The closed-loop system that results from connecting the controller to the wind turbine is shown in figure 4.1. The wind turbine system is denoted by G_0, the controller by C_0 and the unknown pitch actuator gain by σ. The model variables are defined as variations around the operating point values, so that they vary around zero. Therefore, the reference input signals r_1 and r_2 are equal to zero.

For reasons of convenience, the operational inputs $u_1 = \Delta V_w$, $\bar{u}_2 = \Delta\theta_p^{con}$ and $u_2 = \Delta\theta_p$ are defined, as well as the outputs $y_1 = [\ddot{x}_{nd}\ \ddot{x}_{ny}]^T$ and $y_2 = \Delta\Omega_r$. This yields the system model shown in figure 4.2.

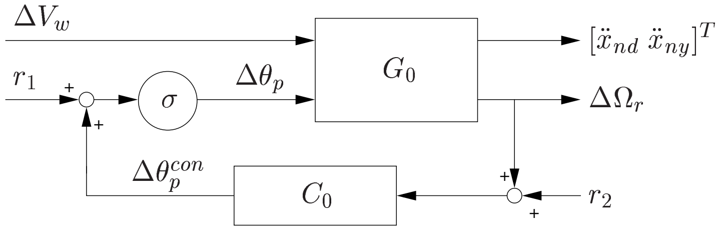

Figure 4.1: *Closed-loop wind turbine system; operational inputs and outputs.*

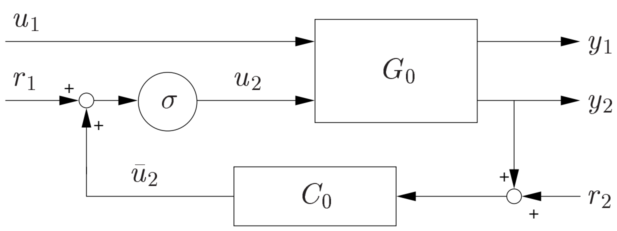

Figure 4.2: *Closed-loop wind turbine system; symbolic inputs and outputs.*

Both the wind turbine system G_0 and the controller C_0 can be denoted as LTI systems, as defined in (A.3)-(A.4). The state equations are as follows:

$$G_0 \quad = \quad \begin{cases} x(k+1) & = Ax(k) + B_1 u_1(k) + B_2 u_2(k) \\ y_1(k) & = C_1 x(k) + D_{11} u_1(k) + D_{12} u_2(k) \\ y_2(k) & = C_2 x(k) + D_{21} u_1(k) + D_{22} u_2(k) \end{cases} \tag{4.1}$$

$$C_0 \quad = \quad \begin{cases} \bar{x}(k+1) & = \bar{A}\bar{x}(k) + \bar{B} y_2(k) \\ \bar{u}_2(k) & = -\big(\bar{C}\bar{x}(k) + \bar{D} y_2(k)\big) \end{cases} \tag{4.2}$$

Since $r_1 = 0$, the faulty pitch actuator gain σ is a scaling factor on the controller output $\bar{u}_2(k)$:

$$u_2(k) = \sigma \bar{u}_2(k) \tag{4.3}$$

4.1.2 Analysis of the closed-loop system

Transfer function from $\bar{u}_2$ to y

For the system shown in figure 4.2, it is now possible to derive the state matrices for the closed-loop relation between $\bar{u}_2$ and y. Substitution of (4.3) into the system equations (4.1) yields

$$G_0 \quad = \quad \begin{cases} x(k+1) & = Ax(k) + B_1 u_1(k) + B_2 \sigma \bar{u}_2(k) \\ y_1(k) & = C_1 x(k) + D_{11} u_1(k) + D_{12} \sigma \bar{u}_2(k) \\ y_2(k) & = C_2 x(k) + D_{21} u_1(k) + D_{22} \sigma \bar{u}_2(k) \end{cases} \tag{4.4}$$

Subsequently, y_2 from (4.4) is substituted into (4.2). For $\bar{u}_2$, this yields

$$\begin{aligned} \bar{u}_2(k) &= -\big(\bar{C}\bar{x}(k) + \bar{D}(C_2 x(k) + D_{21} u_1(k) + D_{22}\sigma\bar{u}_2(k))\big) \\ &= -\big(\bar{C}\bar{x}(k) + \bar{D}C_2 x(k) + \bar{D}D_{21} u_1(k) + \bar{D}D_{22}\sigma\bar{u}_2(k)\big) \\ (1 + \bar{D}D_{22}\sigma)\bar{u}_2(k) &= -\big(\bar{C}\bar{x}(k) + \bar{D}C_2 x(k) + \bar{D}D_{21} u_1(k)\big) \\ \bar{u}_2(k) &= -(1 + \bar{D}D_{22}\sigma)^{-1}\big(\bar{C}\bar{x}(k) + \bar{D}C_2 x(k) + \bar{D}D_{21} u_1(k)\big) \end{aligned}$$

Therefore, (4.2) becomes

$$C_0 \quad = \quad \begin{cases} \bar{x}(k+1) & = \bar{B}C_2 x(k) + \bar{A}\bar{x}(k) + \bar{B}D_{21} u_1(k) + \bar{B}D_{22}\sigma\bar{u}_2(k) \\ \bar{u}_2(k) & = -(1 + \bar{D}D_{22}\sigma)^{-1}\big(\bar{C}\bar{x}(k) + \bar{D}C_2 x(k) + \bar{D}D_{21} u_1(k)\big) \end{cases} \tag{4.5}$$

For the wind turbine system shown in figure 4.2, y_1 is equal to $[\ddot{x}_{nd}\ \ddot{x}_{ny}]^T$, therefore D_{11}=$[D_{11}^{nd}\ D_{11}^{ny}]^T$ and D_{12}=$[D_{12}^{nd}\ D_{12}^{ny}]^T$. Since the wind turbine system model uses $(\Delta V_w - \dot{x}_{nd})$ as the effective wind disturbance input, the values of D_{11}^{nd} and D_{12}^{nd} are non-zero. On physical grounds, all other terms D_{ij} are expected to be zero, which is indeed the case for the turbine matrices derived in chapter 3. Therefore, in (4.4)- (4.5), the substitutions $D_{21} = 0$, $D_{22} = 0$ and $(1 - \bar{D}D_{22}\sigma)^{-1} = 1$ can be made, yielding

$$G_0 \quad = \quad \begin{cases} x(k+1) & = Ax(k) + B_1 u_1(k) + B_2 \sigma \bar{u}_2(k) \\ y_1(k) & = C_1 x(k) + D_{11} u_1(k) + D_{12} \sigma \bar{u}_2(k) \\ y_2(k) & = C_2 x(k) \end{cases} \tag{4.6}$$

$$C_0 \quad = \quad \begin{cases} \bar{x}(k+1) & = \bar{B}C_2 x(k) + \bar{A}\bar{x}(k) \\ \bar{u}_2(k) & = -\big(\bar{C}\bar{x}(k) + \bar{D}C_2 x(k)\big) \end{cases} \tag{4.7}$$

Easily, the state equations can now be obtained:

$$\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 \\ \bar{B}C_2 & \bar{A} \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \begin{bmatrix} B_1 & B_2\sigma \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ \bar{u}_2(k) \end{bmatrix} \tag{4.8}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12}\sigma \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ \bar{u}_2(k) \end{bmatrix} \tag{4.9}$$

From (4.8)-(4.9) it follows, that when the transfer function from $\bar{u}_2$ to y is analyzed, the fault shows itself as a scaling factor on the controller output $\bar{u}_2(k)$.

Transfer function from u_1 to y

In order to investigate the behavior of the closed-loop system as a function of the disturbance input $u_1(k)$, the equation for $\bar{u}_2(k)$ from (4.7) must be substituted into the system equations (4.8)-(4.9). This yields:

$$\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \end{bmatrix} = \begin{bmatrix} A - B_2\sigma\bar{D}C_2 & -B_2\sigma\bar{C} \\ \bar{B}C_2 & \bar{A} \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} u_1(k) \tag{4.10}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 - D_{12}\sigma\bar{D}C_2 & -D_{12}\sigma\bar{C} \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \begin{bmatrix} D_{11} \\ 0 \end{bmatrix} u_1(k) \tag{4.11}$$

The state equations of the closed-loop system (4.10)-(4.11) can now be written as follows:

$$\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \end{bmatrix} = \tilde{A}(\sigma) \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \tilde{B}u_1(k) \tag{4.12}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \tilde{C}(\sigma) \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \tilde{D}u_1(k) \tag{4.13}$$

Combining (4.10)-(4.11) with (4.12)-(4.13), it is clear that a faulty actuator gain σ shows itself as a component fault: it changes the system dynamics, since it changes the state matrix $\tilde{A}(\sigma)$.

Bode plots of the closed-loop system

It is interesting to analyze how the closed-loop system behaves for different values of the pitch actuator gain σ, and to find out how the wind speed $u_1(k) = \Delta V_w(k)$ affects the system behavior. A customary approach is to compute the transfer functions $G(e^{j\omega})$ from $\Delta V_w(k)$ to certain variables of interest. A transfer function describes the system behavior as a function of the signal frequency ω, and can be displayed graphically in a Bode plot, where the magnitude $\log|G(e^{j\omega})|$ and the phase $\arg(G(e^{j\omega}))$ are plotted against $\log(\omega)$.

Figure 4.3 shows the Bode plot of the transfer function from ΔV_w to $\Delta\Omega_r$. From this plot it is obvious that the largest change in behavior occurs in the low frequency area. At low frequencies, the magnitude of $\Delta\Omega_r$ increases if σ decreases.

Figure 4.4 shows the Bode plot of the transfer function from ΔV_w to the faulty controlled input $\Delta\theta_p$. Again, the largest change in behavior occurs in the low frequency area: the magnitude of the magnitude of $\Delta\theta_p$ increases if σ decreases. The similarity between figure 4.3 and figure 4.4 is obvious; this is logical, considering that the transfer function from $\Delta\Omega_r$ to $\Delta\theta_p$ is that of the low-order PI controller (3.18).

The Bode plot of the transfer function from ΔV_w to the nodding vibration $\ddot{x}_{nd}$ is shown in figure 4.5. A decrease of σ slightly increases the magnitude of the low-frequent nodding vibrations. It must be noted here, that a faulty actuator gain caused by a broken blade in a real wind turbine gives rise to rotational instabilities, which result in a larger change of the vibration spectrum; this behavior is not included in the wind turbine model described in chapter 3.

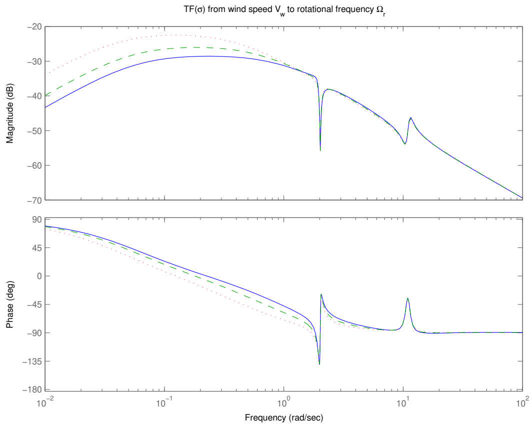

Figure 4.3: *Bode plot of the transfer function from* ΔV_w *to* $\Delta\Omega_r$*, for* $\sigma = 1$ *(solid),* $\sigma = 2/3$ *(dashed) and* $\sigma = 1/3$ *(dotted).*

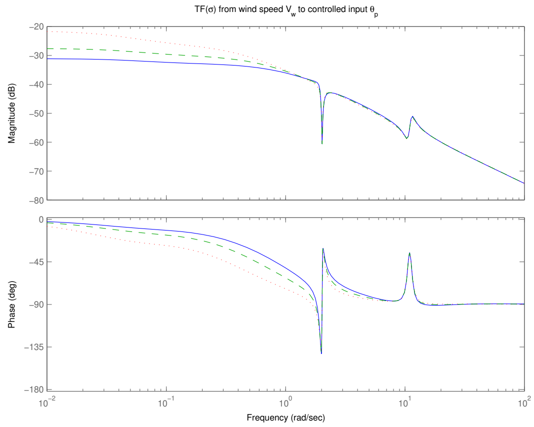

Figure 4.4: *Bode plot of the transfer function from* ΔV_w *to the controlled input* $\Delta\theta_p$*, for* $\sigma = 1$ *(solid),* $\sigma = 2/3$ *(dashed) and* $\sigma = 1/3$ *(dotted).*

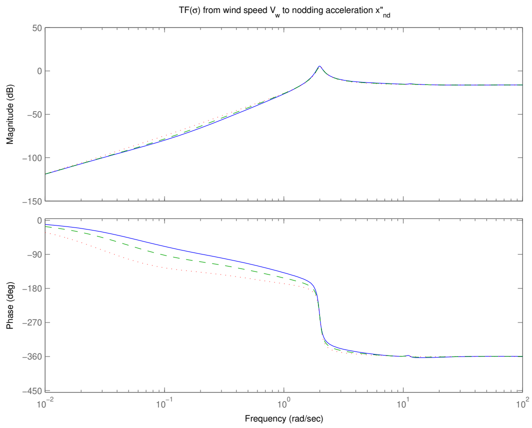

Figure 4.5: *Bode plot of the transfer function from* ΔV_w *to the nodding vibration* $\ddot{x}_{nd}$*, for* $\sigma = 1$ *(solid),* $\sigma = 2/3$ *(dashed) and* $\sigma = 1/3$ *(dotted).*

4.2 Detection of an unknown pitch actuator delay

This case study deals with the identification of an unknown delay of the controlled pitch angle $\Delta\theta_p$. In the wind turbine model described in chapter 3, this delay is modeled with a Padé approximation: a lead-lag filter (3.6) with two zeros and two poles, which is repeated below:

$$\frac{\theta_p^d}{\theta_p} = \frac{(s - (p_1 + j \cdot p_2)) \cdot (s - (p_1 - j \cdot p_2))}{(s - (-p_1 + j \cdot p_2)) \cdot (s - (-p_1 - j \cdot p_2))}$$

A change in the delay time modifies the two poles in equation (3.6), and therefore changes two eigenvalues of the wind turbine system. In figure 4.6, the delay transfer (3.6) is modeled in a separate block d_0, which has the controller output $\bar{u}_2 = \Delta\theta_p$, the desired pitch angle, as the input signal, and the delayed pitch angle $u_2 = \Delta\theta_p^d$ as the output signal. The continuous-time state-space representation of the delay system block d_0 is as follows:

$$A_c = \begin{bmatrix} -2p_1 & -2p_2 \\ 2p_2 & 0 \end{bmatrix} \qquad B_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C_c = \begin{bmatrix} -4p_1 & 0 \end{bmatrix} \qquad D_c = \begin{bmatrix} 1 \end{bmatrix}$$

For further analysis, the continuous-time system must be discretized with a sample time of 0.1s; this discrete-time system representation is denoted as $[\tilde{A},\ \tilde{B},\ \tilde{C},\ \tilde{D}]$ in (4.16). It was found that $\tilde{D} = 1$ for all values of the delay parameters p_1 and p_2, and that the discrete state matrices $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ change when the delay parameters are changed.

The wind turbine system is designed with a delay of 0.2s, the design values of the parameters p_1 and p_2 are shown in section 3.4. When a delay differs from the design delay, the delay time is reversely proportional to these parameters, as can be seen in table 4.1. The objective is to develop an FDI scheme that is able to estimate the unknown pitch delay.

Table 4.1: *Values of the parameters p_1 and p_2 in equation (3.6) for various implementations of the delay time in the blade pitch actuator. A healthy wind turbine system has a delay of* 0.2s.

delay (s)	p_1	p_2
0.1	30.000000	17.320508
0.2	15.000000	8.660254
0.3	10.000000	5.773503
0.4	7.500000	4.330127

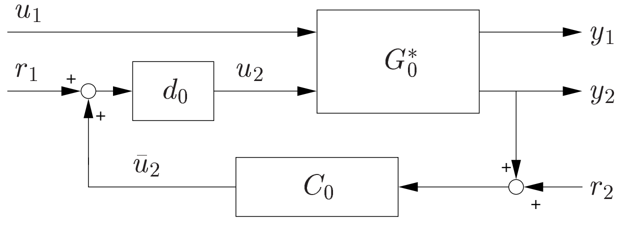

Figure 4.6: *Closed-loop wind turbine system, where d_0 represents the pitch actuator delay; symbolic inputs and outputs.*

4.2.1 The wind turbine system with an unknown pitch delay

The wind turbine system G_0 in figure 4.2 includes the pitch actuator delay. To analyze the effect of the pitch delay on the closed-loop behavior, the wind turbine system G_0 must be divided into a block d_0, which contains the pitch delay, and a block G_0^*, which represents the rest of the wind turbine system. The result is shown in figure 4.6, where the reference signals r_1 and r_2 are equal to zero. The same inputs and outputs are considered as in section 4.1.1. As mentioned before, the controlled input u_2 now represents the delayed pitch angle θ_p^d.

The state equations for the turbine system G_0^*, the controller C_0 and the delay d_0 can again be defined as LTI systems, and written as follows:

$$G_0^* \;=\; \begin{cases} x(k+1) & = Ax(k) + B_1u_1(k) + B_2u_2(k) \\ y_1(k) & = C_1x(k) + D_{11}u_1(k) + D_{12}u_2(k) \\ y_2(k) & = C_2x(k) + D_{21}u_1(k) + D_{22}u_2(k) \end{cases} \tag{4.14}$$

$$C_0 \;=\; \begin{cases} \bar{x}(k+1) & = \bar{A}\bar{x}(k) + \bar{B}y_2(k) \\ \bar{u}_2(k) & = -\big(\bar{C}\bar{x}(k) + \bar{D}y_2(k)\big) \end{cases} \tag{4.15}$$

$$d_0 \;=\; \begin{cases} \tilde{x}(k+1) & = \tilde{A}\tilde{x}(k) + \tilde{B}\bar{u}_2(k) \\ u_2(k) & = \tilde{C}\tilde{x}(k) + \tilde{D}\bar{u}_2(k) \end{cases} \tag{4.16}$$

The matrix $\tilde{D}$ is included in the analysis, though $\tilde{D} = 1$ for all values of the delay parameters.

4.2.2 Analysis of the closed-loop system

Transfer function from $\bar{u}_2$ to y

For the system model shown in figure 4.6, the state matrices of the transfer function from $\bar{u}_2$ to y are derived. The delayed output $u_2(k)$ of (4.16) is substituted into (4.14). Using the fact that $D_{21} = 0$ and $D_{22} = 0$, this yields:

$$G_0^* \;=\; \begin{cases} x(k+1) & = Ax(k) + B_1u_1(k) + B_2\tilde{C}\tilde{x}(k) + B_2\tilde{D}\bar{u}_2(k) \\ y_1(k) & = C_1x(k) + D_{11}u_1(k) + D_{12}\tilde{C}\tilde{x}(k) + D_{12}\tilde{D}\bar{u}_2(k) \\ y_2(k) & = C_2x(k) \end{cases} \tag{4.17}$$

The subsequent substitution of $y_2(k)$ from (4.17) into (4.15) yields:

$$C_0 \;=\; \begin{cases} \bar{x}(k+1) & = \bar{A}\bar{x}(k) + \bar{B}C_2x(k) \\ \bar{u}_2(k) & = -\big(\bar{C}\bar{x}(k) + \bar{D}C_2x(k)\big) \end{cases} \tag{4.18}$$

The closed-loop relation between $\bar{u}_2$ and y now follows from (4.17), (4.18) and (4.16). It is shown in (4.19)-(4.20):

$$\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 & B_2\tilde{C} \\ \bar{B}C_2 & \bar{A} & 0 \\ 0 & 0 & \tilde{A} \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} B_1 & B_2\tilde{D} \\ 0 & 0 \\ 0 & \tilde{B} \end{bmatrix} \begin{bmatrix} u_1(k) \\ \bar{u}_2(k) \end{bmatrix} \tag{4.19}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 & 0 & D_{12}\tilde{C} \\ C_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12}\tilde{D} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ \bar{u}_2(k) \end{bmatrix} \tag{4.20}$$

It is intuitively logical that a variation in the pitch delay changes the dynamics of the system, and therefore results in a different state matrix. From (4.19)-(4.20), it is immediately clear that the transfer function $\bar{u}_2$ to y shows a different dynamic behavior for a different delay.

Transfer function from u_1 to y

In order to investigate the behavior of the closed-loop system as a function of the disturbance input $u_1(k)$, the equation for $\bar{u}_2(k)$ from (4.18) must be substituted into (4.17) and (4.16). This yields:

$$
\begin{aligned}
G_0^* &= \begin{cases} x(k+1) & = Ax(k) + B_1u_1(k) + B_2\tilde{C}\tilde{x}(k) - B_2\tilde{D}\bar{C}\bar{x}(k) - B_2\tilde{D}\bar{D}C_2x(k) \\ y_1(k) & = C_1x(k) + D_{11}u_1(k) + D_{12}\tilde{C}\tilde{x}(k) - D_{12}\tilde{D}\bar{C}\bar{x}(k) - D_{12}\tilde{D}\bar{D}C_2x(k) \\ y_2(k) & = C_2x(k) \end{cases} \\
C_0 &= \begin{cases} \bar{x}(k+1) & = \bar{A}\bar{x}(k) + \bar{B}C_2x(k) \end{cases} \\
d_0 &= \begin{cases} \tilde{x}(k+1) & = \tilde{A}\tilde{x}(k) - \tilde{B}\bar{C}\bar{x}(k) - \tilde{B}\bar{D}C_2x(k) \end{cases}
\end{aligned}
$$

These equations can be rewritten as the state-space representation (4.21)-(4.21). It describes the influence of $u_1(k)$ on the closed-loop behavior and the system outputs. Logically, a variable delay again shows itself as a component fault.

$$
\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A - B_2\tilde{D}\bar{D}C_2 & -B_2\tilde{D}\bar{C} & B_2\tilde{C} \\ \bar{B}C_2 & \bar{A} & 0 \\ -\tilde{B}\bar{D}C_2 & -\tilde{B}\bar{C} & \tilde{A} \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \\ 0 \end{bmatrix} u_1(k) \tag{4.21}
$$

$$
\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 - D_{12}\tilde{D}\bar{D}C_2 & -D_{12}\tilde{D}\bar{C} & D_{12}\tilde{C} \\ C_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} D_{11} \\ 0 \end{bmatrix} u_1(k) \tag{4.22}
$$

Bode plots of the closed-loop system

Analogous to the previous FDI problem, the closed-loop behavior is visualized by showing three relevant bode-plots for three different values of the pitch delay.

Figure 4.7 shows the Bode plot of the transfer function from ΔV_w to $\Delta\Omega_r$. Note that the frequency range is smaller than in figure 4.3. Below $\omega = 5$ rad/s, an increasing delay results in a higher magnitude of $\Delta\Omega_r$, except close to the minimum around $\omega = 2$ rad/s. Above $\omega = 5$ rad/s, an increasing delay results in a slightly lower magnitude of $\Delta\Omega_r$.

Figure 4.8 shows the Bode plot of the transfer function from ΔV_w to the controlled input $\Delta\theta_p$. Below $\omega = 5$ rad/s, an increasing delay results in a higher magnitude of $\Delta\Omega_r$, except close to the minimum around $\omega = 2$ rad/s. Above $\omega = 5$ rad/s, an increasing delay results in a slightly lower magnitude of $\Delta\Omega_r$. The similarity between figure 4.7 and figure 4.8 results from the low-order controller, which determines the transfer function from $\Delta\Omega_r$ to $\Delta\theta_p$.

The Bode plot of the transfer function from ΔV_w to the nodding vibration $\ddot{x}_{nd}$ is shown in figure 4.9. A decrease of σ slightly increases the magnitude of the nodding vibration acceleration, for $\omega < 5$, and not in the peak at $\omega = 2$. For $\omega > 5$, the magnitude of the nodding vibration acceleration slightly decreases.

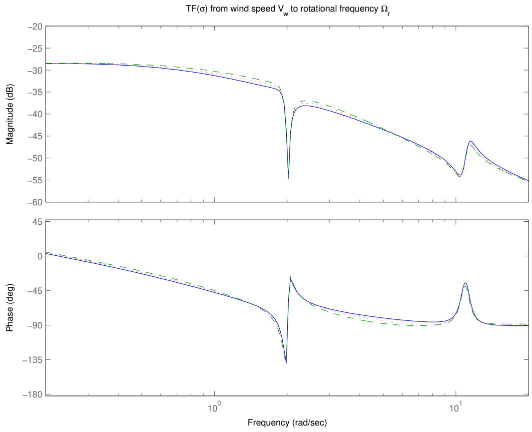

Figure 4.7: *Bode plot of the transfer function from* ΔV_w *to* $\Delta\Omega_r$*, for a pitch delay of* 0.2*s (solid),* 0.3*s (dashed) and* 0.4*s (dotted).*

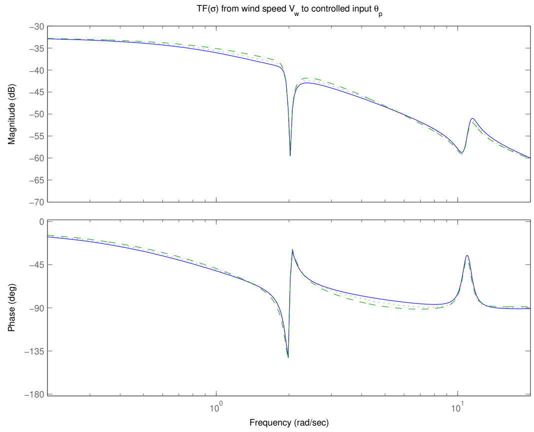

Figure 4.8: *Bode plot of the transfer function from* ΔV_w *to the controlled input* $\Delta\theta_p$*, for a pitch delay of* 0.2*s (solid),* 0.3*s (dashed) and* 0.4*s (dotted).*

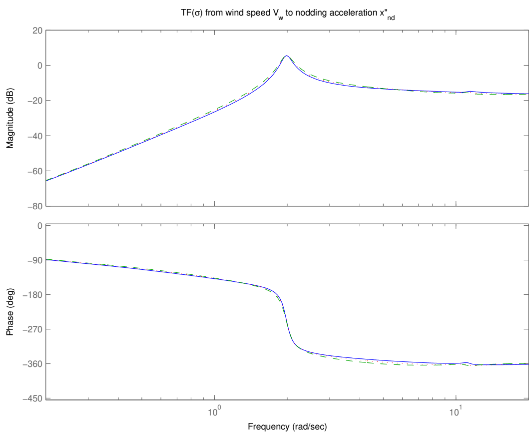

Figure 4.9: *Bode plot of the transfer function from* ΔV_w *to the nodding vibration* $\ddot{x}_{nd}$*, for a pitch delay of* 0.2*s (solid),* 0.3*s (dashed) and* 0.4*s (dotted).*

4.3 Conclusions

In chapter 3, a realistic simulation model was designed of a HAWT that uses pitch-to-vane control. This model will be used as a simulation environment for FDI problems in the remainder of this thesis.

This chapter introduced two realistic FDI problems for a HAWT: the identification of an unknown blade pitch actuator gain and the identification of an unknown pitch delay. A fault in the wind turbine system is defined as a deviation of these system parameters from their normal operating values. In this thesis, methods have to be found that are able to estimate these unknown parameters from closed-loop input/output data of the turbine system.

A block scheme of the closed-loop wind turbine system was presented for each fault scenario, in order to visualize where the fault appears in the turbine system. The state-space representations of the transfer function from the controlled input $\Delta\theta_p$ to the system outputs, and of the wind disturbance input ΔV_w to the system outputs were derived, to be used in the remainder of this thesis. Relevant Bode plots were shown for several fault values, to visualize the effect of a fault on the dynamic behavior of the closed-loop system.

Chapter 5

System identification methods

5.1 Introduction

In general, there are two methods to arrive at a model of an engineering system. First principles modeling uses physical knowledge in order to find a theoretical description of the process. Experimental modeling, or system identification, uses measurements of several process inputs and outputs to identify a model that approximates the physical process as well as possible.

This chapter investigates whether the system identification approach can be used to identify a model of the wind turbine system, using closed-loop input and output measurements. If a reliable model of the wind turbine system can be identified from a data batch, a fault can be represented as a slowly varying parameter that has a constant value in the data batch. A different value of this unknown parameter has the result that a different model description is identified from the data batch. A nominal system model can be identified from a fault-free wind turbine system, so that an identified model can be compared with the nominal model in a parameter estimation scheme (Isermann, 1997), in order to to estimate the unknown parameter during the batch of process data.

5.2 Closed-loop spectral analysis

Van den Hof (1997) presents a general closed-loop system, shown in figure 5.1. Here, $e(t)$ is an unknown disturbance signal, r_1 is a reference value, a setpoint or noise disturbance, and r_2 a setpoint or measurement noise. G_0 represents the plant, and H_0 is a filter which converts the disturbance input $e(t)$ to the disturbance $v(t)$ on the outputs. The controller is denoted C, and $y(t)$ represents the controlled system output Ω_r. The data equations of the general closed-loop system in figure 5.1 are as follows:

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \tag{5.1}$$

$$u(t) = C(q)[r_2(t) - y(t)] + r_1(t) \tag{5.2}$$

If it is not important to consider the separate effects of r_1 and r_2, the total reference signal $r(t)$ can be defined as

$$r(t) = r_1(t) + C(q)r_2(t) \tag{5.3}$$

The feedback law (5.2) can be rewritten as

$$u(t) = r(t) - C(q)y(t) \tag{5.4}$$

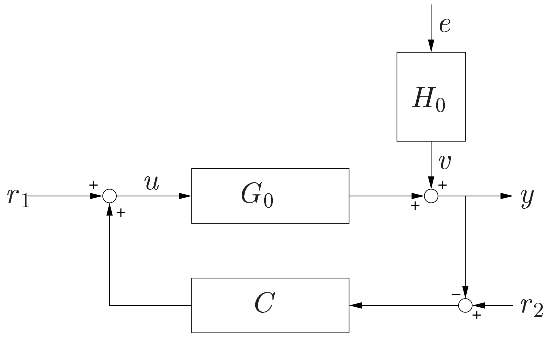

Figure 5.1: *A general closed-loop system, presented by Van den Hof (1997)*

Van den Hof (1997) rewrites the closed-loop equations for $u(t)$ and $y(t)$ as

$$y(t) = (I + G_0(q)C(q))^{-1}[G_0(q)r(t) + H_0(q)e(t)] \tag{5.5}$$
$$u(t) = (I + C(q)G_0(q))^{-1}[r(t) - C(q)H_0(q)e(t)] \tag{5.6}$$

The input sensitivity $S_0(z)$ and the output sensitivity $W_0(z)$ are defined as:

$$S_0(z) = (I + C(z)G_0(z))^{-1} \tag{5.7}$$
$$W_0(z) = (I + G_0(z)C(z))^{-1} \tag{5.8}$$

Using the fact that $W_0G_0 = G_0S_0$ and $CW_0 = S_0C$, equations (5.5) - (5.6) simplify into

$$y(t) = G_0(q)S_0(q)r(t) + W_0(q)H_0(q)e(t) \tag{5.9}$$
$$u(t) = S_0(q)r(t) - C(q)W_0(q)H_0(q)e(t) \tag{5.10}$$

Van den Hof (1997) describes the closed-loop identification problem of Söderström & Stoica (1989): the spectral analysis estimate $\hat{G}(e^{i\omega})$ of the system G_0 is calculated by using direct identification: system identification that uses the controlled input data $u(t)$ and the output data $y(t)$.

Let $r_2(t) \equiv 0$, r_1 and e uncorrelated, and y and u available from measurements. The signal $z(t) = C(q)H_0(q)e(t)$ is introduced, yielding the following descriptions for the input and output signals:

$$y(t) = S_0[G_0(q)r_1(t) + \frac{1}{C(q)}z(t)] \tag{5.11}$$
$$u(t) = S_0(q)[r(t) - z(t)] \tag{5.12}$$

In the frequency domain, this yields

$$\Psi_u(\omega) = |S_0(e^{i\omega})|^2[\Psi_{r_1}(\omega) + \Psi_z(\omega)] \tag{5.13}$$
$$\Psi_{yu}(\omega) = |S_0(e^{i\omega})|^2[G_0(e^{i\omega})\Psi_{r_1}(\omega) - \frac{1}{C(e^{i\omega})}\Psi_z(\omega)] \tag{5.14}$$

Assuming that the spectral densities $\Psi_u(\omega)$ and $\Psi_{yu}(\omega)$ can be estimated exactly (which is asymptotically true for a large number of data points), the spectral analysis estimate $\hat{G}(e^{i\omega})$ is found to be

$$\hat{G}(e^{i\omega}) = \frac{\Psi_{yu}(\omega)}{\Psi_u(\omega)} = \frac{G_0(e^{i\omega})\Psi_{r_1}(\omega) - \frac{\Psi_z(\omega)}{C(e^{i\omega})}}{\Psi_{r_1}(\omega) + \Psi_z(\omega)} \tag{5.15}$$

The spectral analysis estimate $\hat{G}(e^{i\omega})$ (5.15), obtained with direct identification from $u(k)$ and $y(k)$, has two extreme cases:

No external disturbance: $e(t) = 0$, so $z(t) \equiv 0$ and $\Psi_z(\omega) = 0$, simplifying (5.15) into

$$\hat{G}(e^{i\omega}) = G_0(e^{i\omega}) \tag{5.16}$$

No external signal: $r_1(t) = 0$, therefore $\Psi_{r_1}(\omega) = 0$, which simplifies (5.15) into

$$\hat{G}(e^{i\omega}) = \frac{-1}{C(e^{i\omega})} \tag{5.17}$$

If both the disturbance $e(t)$ and the reference input $r_1(t)$ are nonzero, $\hat{G}(e^{i\omega})$ is found as a weighted combination of the extreme cases (5.16)-(5.17).

A similar proof is presented in Ljung (1987, Theorem 14.2). In a system setup such as figure 5.1, the closed-loop identification can provide information about the system G_0 if and only if $r_1(t)$ is a persistently exciting input signal. Horch (2000, Theorem 6.1) presents a more general formulation of this theorem, and a generalization of the closed-loop identification problem described by Van den Hof (1997). It is shown, that in case a persistently exciting disturbance $v(t)$ is applied to the system and the reference signal $r_1(t)$ has a constant value, the relation between the control signal $u(t)$ and the measured output $y(t)$ is always determined by the controller, i.e. $u(t) = -C(q)y(t)$. No matter how much $u(t)$ and $y(t)$ vary, as long as no persistently exciting signal $r_1(t)$ enters the loop, only the controller can be identified. This broadens the application range of (5.17): the zero input reference $r_1(t)$ of Van den Hof (1997) can be seen as a special case of a constant input reference. This is mentioned here, to show that the closed-loop identification problem cannot be solved in a trivial way by designing the controller around a non-zero constant reference input.

5.3 Experiment - system identification in closed-loop

It is interesting to investigate whether FDI schemes can be based on parameter estimation methods, which compare the fault-free and the faulty system model. A necessary prerequisite is that a reliable system model can be identified from measured data.

The wind turbine model of chapter 3, which is shown in figure 4.2, is compared with the general closed-loop system of Van den Hof (1997), as shown in figure 5.1. The wind turbine system has a non-white disturbance input $e(t) = \Delta V_w$, and reference signals r_1 and r_2 equal to zero. It is therefore expected that the wind turbine system cannot be identified directly from $u(t)$ and $y(t)$: this identification will probably yield the negative inverse $-1/C$ of the controller instead of the turbine system, a result similar to (5.17). This will be verified with an experiment; for the implementation, see the file *clident.m*.

The continuous-time wind turbine system matrices $[A, B, C, D]$ and the disturbance matrices $[A, B_h, C, D_h]$ have been derived from the Simulink model presented in section 3.5. The continuous-time wind turbine has been reconstructed in Matlab's control toolbox, with the disturbance input $e(t) = \Delta V_w$, the controlled input $u(t)$ and controlled output $y(t) = \Delta\Omega_r$.

Errors-in-variables (EIV-)MOESP is a subspace identification method that is particularly suitable for closed-loop identification, see Chou & Verhaegen (1997). For more information about subspace identification, see e.g. Haverkamp (2001) and Verhaegen & Verdult (2001).

Using EIV-MOESP on $u(t)$ and $y(t)$ data from the wind turbine model, a discrete-time system model $[Ae, Be, Ce, De]$ of order 5 has been estimated with a sample time of 0.1s. A Variance-Accounted-For (VAF), which is a common measure of accuracy, of 100% was found on validation data.

The PI controller (3.18) and the turbine system $[A, B, C, D]$ have been discretized with a sample time of 0.1s. In figure 5.2, the identified system model is compared with $-1/C_d$, where C_d is the discrete-time system model of the controller. The result is as expected: the estimated model doesn't describe the wind turbine system, but the negative inverse controller. The slight mismatch of the two plots originates from the fact that a 5th order system model is identified of the 1st order controller, so that small numerical errors are included in the model description. If a model of order 1 is estimated, an even better fit can be obtained.

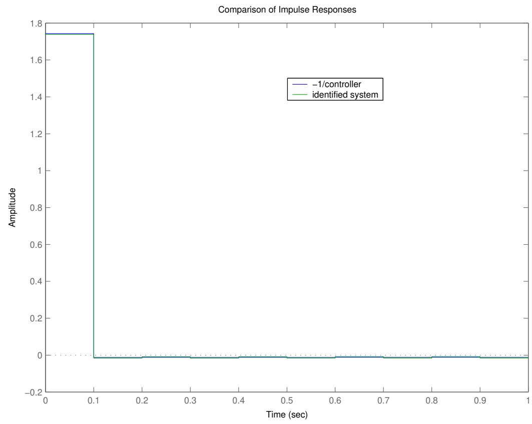

Figure 5.2: *Comparison of impulse responses of the identified system and of the negative inverse controller* $-1/C_d$.

5.4 Conclusions

In this chapter, it was analyzed whether the system identification approach can be used to identify a reliable model of the wind turbine system from inputs and outputs measured in closed-loop. It can be concluded, that this is not possible.

An operational wind turbine system has reference inputs equal to zero, and the wind speed is an unknown input that is the driving disturbance of the system. A well-known closed-loop identification problem was encountered: in the closed-loop setup defined for the the wind turbine, a system identification approach yields a model of the inverse controller instead of a model of the wind turbine system. This was verified in a simulation experiment. It is therefore clear, that a fault in the turbine system cannot be detected with an approach based on closed-loop system identification: the identified model only contains information about the controller, and contains no information about the wind turbine system.

Chapter 6

Kalman filter methods

Introduction

In this chapter, it is investigated whether a fault detection and identification scheme for a wind turbine system can be based on a linear Kalman filter, also known as a Kalman observer. The discrete-time Kalman filter was introduced by Kalman (1960), and has been the subject of extensive research and application; see Ljung (1987) and Godwin et al. (2001) for a thorough description. If a Kalman filter is used in practice, it is necessary that a sufficiently accurate mathematical model of the wind turbine system is available; this is assumed to be the case. The Kalman filter uses the known inputs of the real system to reconstruct the system states in such a way, that the system outputs are predicted as accurately as possible. An unknown variable can be estimated, if it is included in an augmented system state.

This chapter only deals with the FDI problem defined in section 4.1, the detection of the unknown pitch actuator gain σ. In section 6.1, the discrete-time Kalman filter is introduced, and it is investigated to what extent it can be used for FDI purposes. Section 6.2 rewrites the Kalman filter formulation as a least squares problem. This least squares problem is redefined in section 6.3 as a Kalman filter problem over a moving time window, which allows σ to be estimated as an average value during this time window. Section 6.4 presents a nonlinear Kalman filter problem over a moving time window.

6.1 The discrete-time Kalman filter

Consider the LTI system (A.3)-(A.4):

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned}$$

with $w(k)$ the process noise, a zero-mean white noise sequence with covariance matrix Q, and $v(k)$ the measurement noise: zero-mean white noise with covariance matrix R. From here on, the noise sequences will be denoted compactly as $w(k) \frown (0, Q)$ and $v(k) \frown (0, R)$, in analogy with Verhaegen & Verdult (2001).

The recursive Kalman filter algorithm starts at time instant k; using an initial (*a priori*) state estimate $\hat{x}(k|k-1)$ and an initial process noise estimate $\hat{P}(k|k-1)$, it consists of two update steps: the measurement update and the time update.

The measurement update adjusts the estimate $\hat{x}(k|k-1)$ by an actual measurement:

$$K(k) = \hat{P}(k|k-1)C^T(C\hat{P}(k|k-1)C^T + R(k))^{-1} \quad (6.1)$$
$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(y(k) - Du(k) - C\hat{x}(k|k-1)) \quad (6.2)$$
$$\hat{P}(k|k) = (I - KC)\hat{P}(k|k-1) \quad (6.3)$$

First, the current Kalman gain $K(k)$ is computed. Then, the process measurement $y(k)$ is used to generate the *a posteriori* state estimate $\hat{x}(k|k)$: the measurement update. Finally, the a posteriori error covariance estimate $\hat{P}(k|k)$ is computed.

The time update projects the current state estimate $\hat{x}(k|k)$ ahead in time:

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \quad (6.4)$$
$$\hat{P}(k+1|k) = A\hat{P}(k|k)A^T + Q(k) \quad (6.5)$$

Using the measurement update $\hat{x}(k|k)$, the time update calculates the one-step ahead predictor $\hat{x}(k+1|k)$, which is the a priori state estimate for the next time step. Also, the a priori error covariance estimate $\hat{P}(k+1|k)$ for the next time step is computed.

The function *it_kalm.m* was created in Matlab to compute the current Kalman gain $K(k)$ and the predicted noise covariance $\hat{P}(k+1|k)$ during each time step. Here, $K(k)$ is calculated as in (6.1), and $\hat{P}(k+1|k)$ results from the Riccati equation (Godwin et al., 2001), obtained by substituting (6.1) and (6.3) into (6.5):

$$\begin{aligned}\hat{P}(k+1|k) = {} & A\big(\hat{P}(k|k-1) \\ & -\hat{P}(k|k-1)C^T(R + C\hat{P}(k|k-1)C^T)^{-1}C\hat{P}(k|k-1)\big)A^T + Q \quad (6.6)\end{aligned}$$

6.1.1 The wind turbine system

The detection problem of the unknown pitch actuator gain σ was described in section 4.1. The state-space representation of the transfer function from the pitch angle θ_p to the system outputs is shown in (4.8)-(4.9). The naying vibrations $\ddot{x}_{ny}$ are not used for analysis, so $y_1 = \ddot{x}_{nd}$ and $y_2 = \Delta\Omega_r$ from here on. The wind speed $u_1(k)$ is denoted as $v_w(k)$, and the pitch angle $\bar{u}_2(k)$ as $u(k)$. The state matrices are renamed: the substitutions $B_1 = B_h$ and $B_2 = B$ are made, as well as $D_{11} = D_h$ and $D_{12} = D$. A process noise term $w(k) \frown N(0, Q)$ is added to the state vector, which can be written as a unit covariance zero-mean noise term $\tilde{w}(k) \frown (0, I)$ multiplied by the process noise covariance $Q^{1/2}$. Similarly, a measurement noise term $v(k) \frown (0, R)$ is added to the output vector. This yields the following state-space representation of the wind turbine system, with a faulty actuator gain $\sigma(k)$:

$$x(k+1) = Ax(k) + B_h v_w(k) + B\sigma(k)u(k) + Q^{1/2}\tilde{w}(k) \quad (6.7)$$
$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x(k) + \begin{bmatrix} D_h & D\sigma(k) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_w(k) \\ u(k) \end{bmatrix} + R^{1/2}\tilde{v}(k) \quad (6.8)$$

with the state vector $x \in \mathbb{R}^n$, the input vector $u \in \mathbb{R}^m$ and the output vector $y \in \mathbb{R}^\ell$; $k \in \mathbb{N}^+$ is the time instant.

6.1.2 The wind speed in an augmented state

The wind speed $v_w(k)$ is considered unknown. It is interesting to verify whether the Kalman filter can be used to estimate the wind speed. This requires $v_w(k)$ to be included in an augmented system state. The wind speed is defined as a random walking variable:

$$v_w(k+1) = v_w(k) + Q_{v_w}^{1/2}\tilde{w}_{v_w}(k) \tag{6.9}$$

with $\tilde{w}_{v_w}(k) \frown (0, I)$ zero-mean unit variance white noise, multiplied with the square root of the covariance matrix Q_{v_w}. Here, Q_{v_w} represents the random walk size: a larger value of Q_{v_w} results in bigger random steps, yielding faster but less accurate estimation. The wind turbine system (6.7)-(6.8) can now be rewritten as an augmented state-space system:

$$\begin{bmatrix} x(k+1) \\ v_w(k+1) \end{bmatrix} = \begin{bmatrix} A & B_h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \sigma(k)u(k) + \begin{bmatrix} Q^{1/2} & 0 \\ 0 & Q_{v_w}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{w}(k) \\ \tilde{w}_{v_w}(k) \end{bmatrix} \tag{6.10}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 & D_h \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} \sigma(k)u(k) + R^{1/2}\tilde{v}(k) \tag{6.11}$$

With the appropriate definitions of the augmented system matrices, and the augmented state vector $\breve{x} \in \mathbb{R}^{n+1}$, the augmented system (6.10)-(6.11) can be denoted as

$$\begin{aligned} \breve{x}(k+1) &= \breve{A}\breve{x}(k) + \breve{B}\sigma(k)u(k) + \breve{Q}^{1/2}\breve{w}(k) \\ y(k) &= \breve{C}\breve{x}(k) + \breve{D}\sigma(k)u(k) + R^{1/2}\tilde{v}(k) \end{aligned}$$

In section 3.6, the standard measurement noise conditions were described as zero-mean white noise in the system outputs, with a standard deviation of 0.011 m/s^2 for y_1 and 0.006 rad/s^2 for y_2; with the squares of these values on the diagonal of a 2 by 2 matrix, the standard measurement noise covariance matrix R_{smn} is obtained:

$$R_{smn} = \begin{bmatrix} 1.2 \cdot 10^{-4} & 0 \\ 0 & 3.6 \cdot 10^{-5} \end{bmatrix} \tag{6.12}$$

Experiments

The pitch actuator gain $\sigma(k)$ is set to 1 for all time instants k, so that (6.10)-(6.11) becomes an LTI system of the fault-free wind turbine. Solving the Kalman filter for this augmented system yields estimates of the wind turbine states and the wind speed. Two cases are considered: the SISO experiment only uses the output $y_2(k) = \Delta\Omega_r(k)$, the MIMO experiment uses both outputs $y_1(k) = \ddot{x}_{nd}$ and $y_2(k)$. The implementation can be found in the m-file *mrk_w.m*.

It was found, that good results are obtained if the process noise covariance Q is set to zero. The measurement noise covariance R has to be chosen non-zero, otherwise the inversion in (6.6) is not possible.

In the first experiment, no measurement noise was simulated on the system outputs; the value of the measurement noise covariance was set to $R = 0.01 \cdot R_{smn}$ (6.12), corresponding to 10% of the noise amplitude of standard measurement noise. The random walk size Q_{v_w} was set to 5. The wind estimates obtained with the SISO and MIMO experiment are shown in figure 6.1 and figure 6.2, respectively. The VAF between real and estimated wind speed was found to be 94.8% in the SISO experiment, and 100.0% in the MIMO experiment.

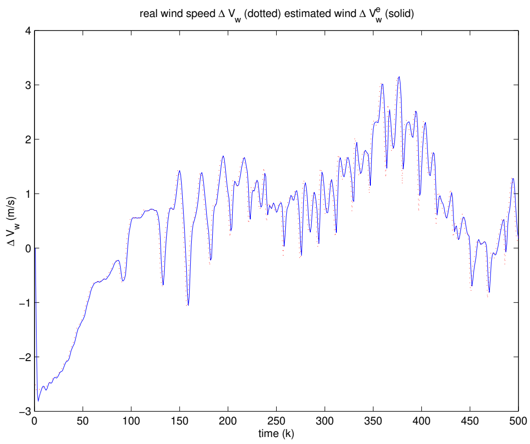

Figure 6.1: *The wind speed $v_w(k)$, estimated with the regular discrete-time Kalman filter as an augmented system state, in the SISO setup, using the measurement of y_2. No measurement noise in the output, and $Q_{v_w} = 5$.*

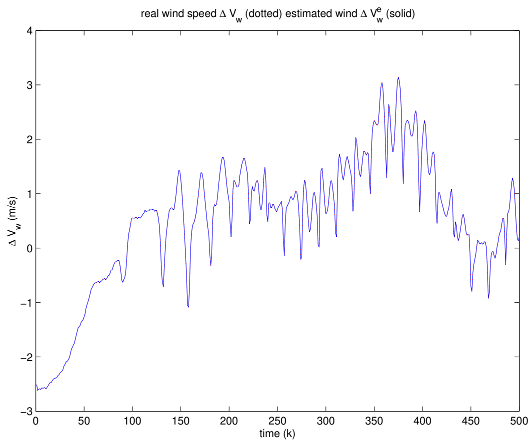

Figure 6.2: *The wind speed $v_w(k)$, estimated with the regular discrete-time Kalman filter as an augmented state, in the MIMO setup, using the measurements of y_1 and y_2. No measurement noise in the outputs, and $Q_{v_w} = 5$.*

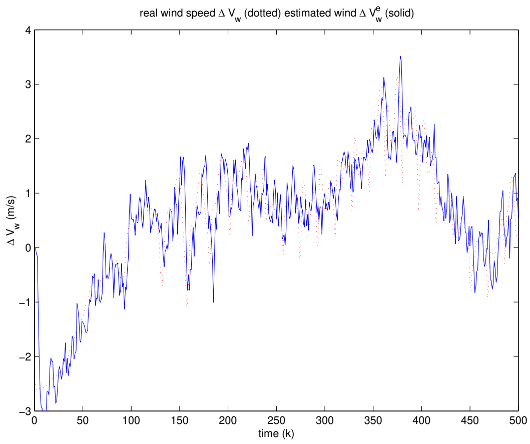

Figure 6.3: *The wind estimate $v_w(k)$ in the SISO setup: same experiment as shown in figure 6.1, with standard measurement noise present in the output and $Q_{v_w} = 0.1$*

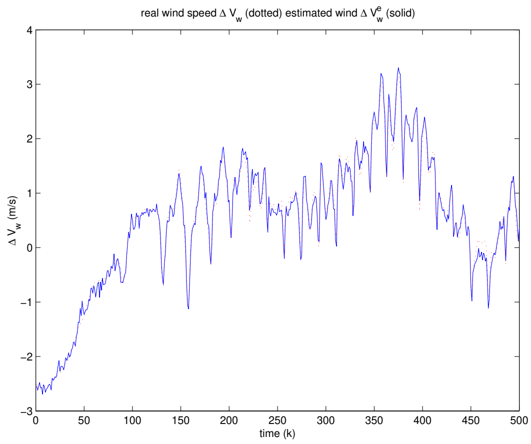

Figure 6.4: *The wind estimate $v_w(k)$ in the MIMO setup: same experiment as shown in figure 6.2, with standard measurement noise present in the outputs and $Q_{v_w} = 5$.*

In a second experiment, realistic measurement noise was present in the system outputs, and the corresponding $R = R_{smn}$ (6.12) was selected. In the SISO setup, the random walk size Q_{v_w} had to be lowered to 0.1, otherwise the estimate became too noisy. In the MIMO setup, Q_{v_w} was again set to 5. The resulting wind estimates for the SISO and MIMO setup are shown in figure 6.3 and figure 6.4, respectively. The VAF between real and estimated wind speed was found to be 81.7% in the SISO experiment, and 99.0% in the MIMO experiment.

Analysis of the results

The wind speed $v_w(k)$ was included in an augmented system state, so that it could be estimated with the discrete-time Kalman filter algorithm, as described in section 6.1. The measurement noise covariance R had to be chosen non-zero, even when no measurement noise was simulated. The process noise on the turbine states Q could be set to zero; it was found that a non-zero noise covariance Q_{v_w} on the wind estimate provides enough excitation for all states. This means that Q_{v_w} is the only tuning parameter of the algorithm.

Two simulation setups were considered: the SISO setup only uses the measurement of Ω_r, and the MIMO setup uses the measurements of $\ddot{x}_{nd}$ and $\Delta\Omega_r$. The experiments were performed with and without standard measurement noise in the system outputs. In the MIMO setup without measurement noise, a perfect estimate of the wind speed was obtained. When standard measurement noise was present in the outputs, the accuracy of the estimate was a little lower, but still very good. The estimates in the SISO setup were less accurate; this was especially clear when standard measurement noise was present in the controlled output.

6.1.3 The pitch actuator gain in an augmented state

In section 6.1.2, it was shown that the Kalman filter can be used to estimate the wind speed, by including it in an augmented system state. The best results were obtained in the MIMO setup, using both the measurements $\ddot{x}_{nd}$ and $\Delta\Omega_r$. In this section, only the MIMO setup is considered, and the unknown pitch actuator gain $\sigma(k)$ is included in an augmented system state as well. It is therefore defined as a random walking variable, in analogy with the wind speed in (6.9):

$$\sigma(k+1) = \sigma(k) + \Psi^{1/2}\tilde{p}(k) \tag{6.13}$$

with $\tilde{p}(k) \frown (0, I)$ zero-mean unit variance white noise, multiplied with the square root of the covariance matrix Ψ. The value of Ψ is a measure for the step size of the random walk: a larger Ψ results in faster but less accurate estimation. The wind turbine system (6.10)-(6.11) can now be written as:

$$\begin{bmatrix} x(k+1) \\ v_w(k+1) \\ \sigma(k+1) \end{bmatrix} = \begin{bmatrix} A & B_h & Bu(k) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \\ \sigma(k) \end{bmatrix} + \begin{bmatrix} Q^{1/2} & 0 & 0 \\ 0 & Q_{v_w}^{1/2} & 0 \\ 0 & 0 & \Psi^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{w}(k) \\ \tilde{w}_{v_w}(k) \\ \tilde{p}(k) \end{bmatrix} \tag{6.14}$$

$$y(k) = \begin{bmatrix} C_1 & D_h & Du(k) \\ C_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \\ \sigma(k) \end{bmatrix} + R^{1/2}\tilde{v}(k) \tag{6.15}$$

The augmented system (6.14)-(6.15) with both wind speed v_w and faulty pitch actuator gain $\sigma(k)$ included in an augmented system state can be denoted as

$$\begin{aligned} \breve{x}(k+1) &= \breve{A}(k)\breve{x}(k) + \breve{Q}^{1/2}\breve{w}(k) \\ y(k) &= \breve{C}(k)\breve{x}(k) + R^{1/2}\tilde{v}(k) \end{aligned}$$

with the appropriate definitions of the augmented system matrices, and the augmented state vector $\breve{x} \in \mathbb{R}^{n+2}$. The augmented system (6.14)-(6.15) is time variant: $\breve{A}(k)$ and $\breve{C}(k)$ depend on the time instant k, as they contain the controlled input $u(k)$. Such a system is called a Linear Time Variant (LTV) system, as presented in (A.1)-(A.2). The state matrices have to be calculated at every time instant k, using the known value of $u(k)$.

First experiment: no measurement noise

The Kalman filter algorithm for the augmented system (6.14)-(6.15) was implemented in the m-file *mrk_ws.m*. Only the MIMO setup is considered: the Kalman filter uses the measurements of $\ddot{x}_{nd}$ and Ω_r. An abrupt fault is simulated. Using the HAWT simulation function *simulator_abrupt.m*, the pitch actuator gain σ was changed from 1 to 2/3 at time instant $k = 500$.

In the first experiment, no measurement noise was present in the system outputs. The covariance matrices $Q_{v_w}^{1/2}$ and $\Psi^{1/2}$ are the tuning parameters of this Kalman filter. It was found, that the process noise covariance Q on the turbine states can be set to zero. As in section 6.1.2, the measurement noise covariance R must be chosen non-zero: it was set to $R = 0.01 \cdot R_{smn}$ (6.12). With $Q_{v_w}^{1/2} = 5$ and $\Psi^{1/2} = 0.01$, the wind estimate shown in figure 6.5 was obtained. The VAF between real and estimated wind speed was found to be 99.98%, an almost perfect result. The estimate of the $\sigma(k)$ is shown in figure 6.6. A VAF of 99.23% between the real and estimated pitch actuator gain was found. The mismatch in the wind estimate occurs mainly just after the fault: if $\sigma = 1$ for all k, the VAF between real and estimated wind is equal to 100.00%.

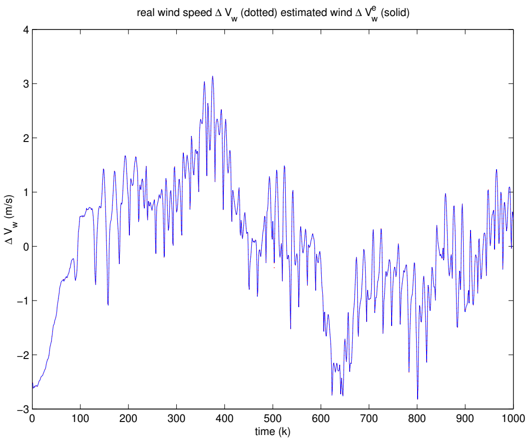

Figure 6.5: *The estimate of the wind speed $v_w(k)$, when both wind speed and pitch actuator gain are included in an augmented system state. No measurement noise was present in the system outputs.*

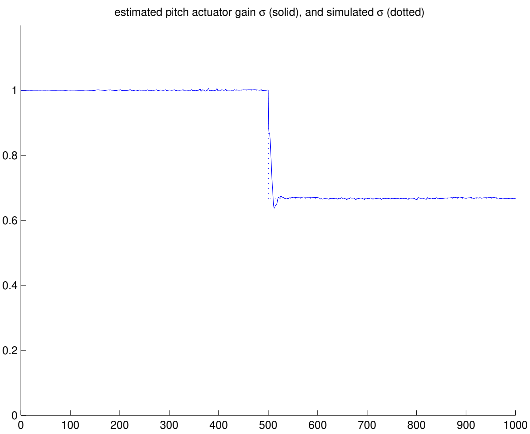

Figure 6.6: *The estimate of the pitch actuator gain $\sigma(k)$, when $v_w(k)$ and $\sigma(k)$ are included in an augmented system state. No measurement noise in the system outputs, and $\sigma(k)$ changes abruptly at $k = 500$.*

Figure 6.6 shows that fault identification is obtained after 20 time instants, when the estimated pitch actuator gain equals the simulated $\sigma = 2/3$. A quick fault detection can be obtained with the numerical derivative $\dot{\sigma}_{est}$ (6.16) of the estimated pitch actuator gain σ_{est}:

$$\dot{\sigma}_{est}(k) = \frac{\sigma_{est}(k) - \sigma_{est}(k-1)}{T_s} \tag{6.16}$$

Figure 6.7 shows a plot of $\dot{\sigma}_{est}(k)$. The steepest descent is very clear. It occurs at $k = 501$, one time instant after the fault occurrence: an immediate fault detection. The estimate of the turbine states is very good, as shown in figure 6.9. The values of the VAF between real

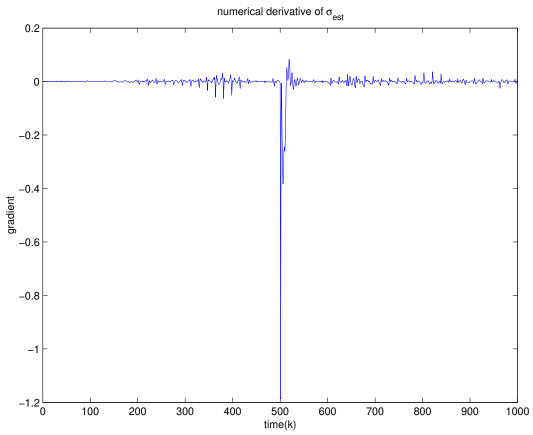

Figure 6.7: *The numerical derivative $\dot{\sigma}_{est}(k)$ of the estimated $\sigma(k)$ shown in figure 6.6.*

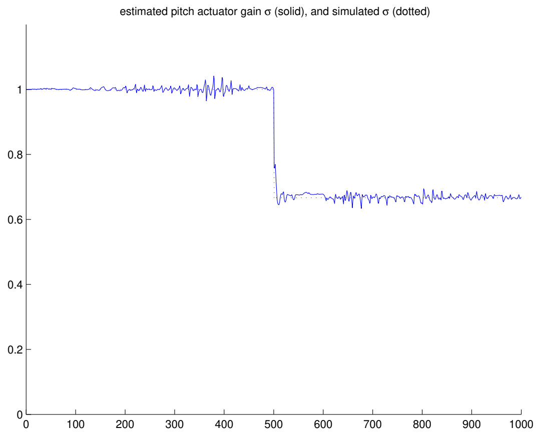

Figure 6.8: *Simulation as in figure 6.6, but with a larger random walk size $\Psi = 0.1$ for $\sigma(k)$.*

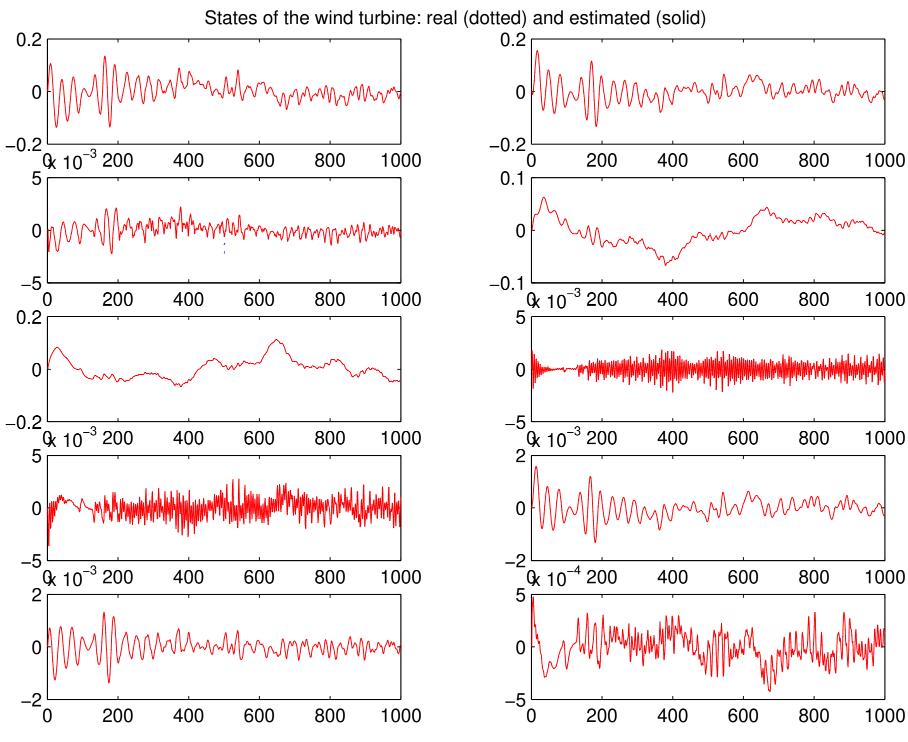

Figure 6.9: *The real turbine states x_r and the estimated turbine states x_e of the simulation shown in figure 6.6. On this scale, only the third state shows a slight mismatch just after the change time* $k = 500$.

states x_r and estimated states x_e were found to be very high (in %):

$$VAF(x_e, x_r) = \left[\begin{array}{cccccccccc} 100.0 & 100.0 & 99.1 & 100.0 & 100.0 & 100.0 & 100.0 & 100.0 & 100.0 & 100.0 \end{array}\right]^T$$

To illustrate the importance of the choice of the random walk size Ψ of the pitch actuator gain $\sigma(k)$, the experiment in figure 6.6 is repeated with a larger value of $\Psi = 0.1$. The resulting estimate of $\sigma(k)$ is shown in figure 6.8. The change detection is even faster, but the estimate is less accurate, since the allowed random walk size is larger.

Second experiment: standard measurement noise

The previous experiment was repeated with standard measurement in the system outputs, as described in section 3.6; the corresponding measurement noise covariance matrix $R = R_{smn}$ (6.12) was chosen.

The measurement noise deteriorates the estimates. The random step sizes Q_{v_w} and Ψ of the augmented state variables $v_w(k)$ and $\sigma(k)$ turned out to be very important tuning parameters to obtain a good result. A suitable pair was $Q_{v_w} = 0.08$ and $\Psi = 0.0002$. The wind estimate is shown in figure 6.10, the corresponding VAF is 99.0%. The estimate of the fault is shown in figure 6.11, with a VAF of 90.0%. The state estimates are still very good; the plot (which is not shown) looks similar to figure 6.9; the VAF values between real states x_r and estimated states x_e are a little lower, but still quite good (in %):

$$VAF(x_e, x_r) = \begin{bmatrix} 99.3 & 99.5 & 99.4 & 99.7 & 99.7 & 96.7 & 97.7 & 98.5 & 99.2 & 98.6 \end{bmatrix}^T$$

The numerical derivative $\dot{\sigma}(k)$ of the pitch actuator gain was derived as in (6.16); it is shown in figure 6.12. It is clear that the derivative is less reliable for FDI purposes: the negative descent just after the fault occurrence is only slightly larger than other peaks. However, a threshold in the estimate of $\sigma(k)$ in figure 6.13 can be used to detect a fault, e.g. $\sigma(k) < 0.8$ for $k > 520$, so the fault is detected after 20 time instants. The fault is identified as well: the estimate of $\sigma(k)$ tracks the simulated $\sigma(k)$ quite well.

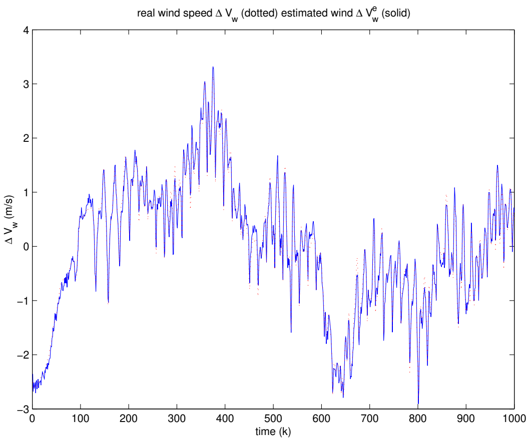

Figure 6.10: *The estimate of the wind speed $v_w(k)$: same simulation conditions as in figure 6.5, but with standard measurement noise present in the outputs.*

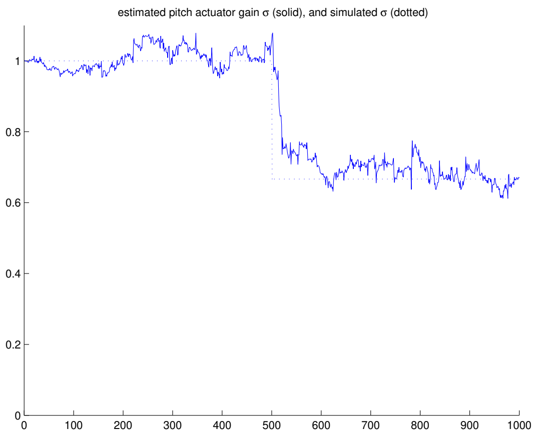

Figure 6.11: *The estimate of the pitch actuator gain $\sigma(k)$: same simulation conditions as in figure 6.6, but with standard measurement noise present in the outputs.*

To illustrate the importance of the choice of the random walk size Ψ of the pitch actuator gain $\sigma(k)$, the experiment in figure 6.11 is repeated with a smaller value of $\Psi = 0.00005$. The estimate of $\sigma(k)$ is shown in figure 6.13. The smaller step size results in slower fault tracking after the change at $k = 500$, therefore fault detection is slower. However, there is better fault isolation: before the fault occurrence and for $k > 650$, the estimated $\sigma(k)$ is a better approximation of the simulated $\sigma(k)$, when compared to the result in figure 6.11. The overall VAF between simulated and estimated fault is 83.7%, due to the larger mismatch for $500 < k < 650$.

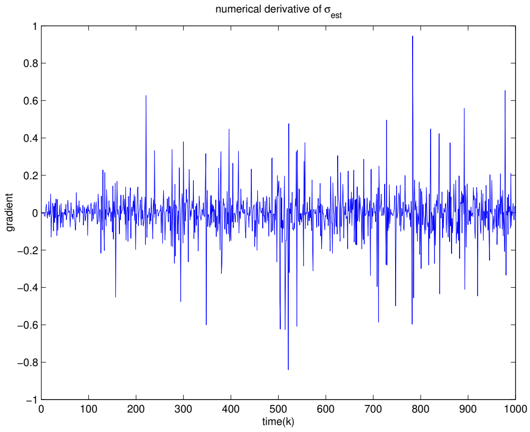

Figure 6.12: *The numerical derivative $\dot{\sigma}_{est}(k)$ of the estimated $\sigma_{est}(k)$ shown in figure 6.11.*

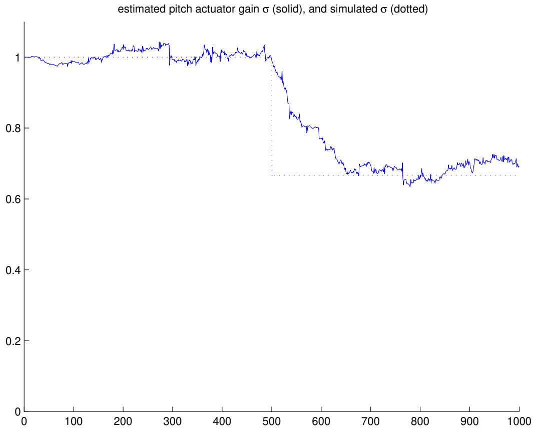

Figure 6.13: *Simulation as in figure 6.11, but $\sigma_{est}(k)$ has a smaller random walk size $\Psi =$* 0.0005.

Analysis of the results

The discrete-time Kalman filter algorithm, described in section 6.1, was used to estimate both the wind speed $v_w(k)$ and the pitch actuator gain $\sigma(k)$. The wind speed and actuator gain were defined as random walking variables, with random walk sizes Q_{v_w} and Ψ, respectively, and included in an augmented system state.

It was found, that the algorithm has two tuning parameters: the random walk sizes of the wind speed and the pitch actuator gain. It was shown that it is possible to tune these parameters in such a way, that an abrupt fault in the pitch actuator gain can be detected, isolated and identified. Only the MIMO case was considered: the measurements of $\ddot{x}_{nd}$ and $\Delta\Omega_r$ are used in the algorithm. An abrupt change in the pitch actuator gain σ was simulated.

Without measurement noise present in the system outputs, very good estimates were obtained of the wind speed $v_w(k)$, the pitch actuator gain $\sigma(k)$ and the turbine states. It was demonstrated that the numerical derivative $\dot{\sigma}(k)$ can be used to for quick fault detection, and that the estimate of the pitch actuator $\sigma(k)$ can be used for fault isolation and identification.

With standard measurement noise present in the system outputs, the quality of the estimates deteriorated. It was shown, that tuning the parameters Q_{v_w} and Ψ improves the quality of the estimates. These parameters must be tuned to a lower value, so that less freedom is allowed to the estimates. A reliable estimate of the pitch actuator gain $\sigma(k)$ could be obtained, but the measurement noise results in a more noisy estimate. Therefore, the numerical derivative $\dot{\sigma}(k)$ is more noisy as well, so it is less reliable for FDI purposes. However, the estimated pitch actuator gain is a reliable indicator of a change in the real actuator gain $\sigma(k)$, even in the presence of measurement noise.

6.2 The Kalman filter as a least squares problem

In this section, the Kalman filter formulation is written as a least mean squares (LMS) problem, in analogy with Verdult & Verhaegen (2001). This LMS framework is used in sections 6.3 and 6.4, where new methods are introduced to detect an unknown actuator gain $\sigma(k)$.

6.2.1 A general covariance representation

Consider the LTI system (A.3)-(A.4), as defined in appendix A.2.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned}$$

where the covariance matrices of the process noise $v(k)$ and measurement noise $w(k)$ are defined as:

$$E \begin{bmatrix} v(k) \\ w(k) \end{bmatrix} \begin{bmatrix} v(j)^T & w(j)^T \end{bmatrix} = \begin{bmatrix} R(k) & S(k)^T \\ S(k) & Q(k) \end{bmatrix} \Delta(k-j) \tag{6.17}$$

where $E[\cdot]$ denotes the statistically expected value and $\Delta(k)$ the unit pulse. Suppose that an estimate $\hat{x}(0|-1)$ of the initial state $x(0)$ is available, along with its covariance matrix $P(0|-1)$. Furthermore, suppose that the initial state $x(0)$ is a random variable which is independent of $w(k)$ and $v(k)$ and has a mean value $\hat{x}(0|-1)$ and covariance matrix $P(0|-1)$. The upper or lower triangular matrix $S(0|-1)$ is the Cholesky factor of $P(0|-1)$. This yields:

$$\begin{aligned} E[x(0)] &= \hat{x}(0|-1) \\ E\Big[\big(x(0) - \hat{x}(0|-1)\big)\big(x(0) - \hat{x}(0|-1)\big)^T\Big] &= P(0|-1) \\ &= S(0|-1)S(0|-1)^T \end{aligned}$$

In Verhaegen & Verdult (2001), a general covariance representation is introduced for the statistical information of the random variables. To describe $x(0)$, an auxiliary stochastic variable $\tilde{u}(k) \frown (0, I)$ is introduced. Using the Cholesky factor $S(0|-1)$, this yields:

$$\hat{x}(0|-1) = x(0) + S(0|-1)\tilde{u}(0) \tag{6.18}$$

If all covariance matrices in (6.17) are positive definite, a similar representation can be introduced for the measurement noise $v(k)$ and the process noise $w(k)$:

$$\begin{bmatrix} v(k) \\ w(k) \end{bmatrix} = \begin{pmatrix} R(k)^{1/2} & 0 \\ X(k) & Q_x(k) \end{pmatrix} \begin{bmatrix} \tilde{v}(k) \\ \tilde{w}(k) \end{bmatrix} \qquad \text{with} \begin{bmatrix} \tilde{v}(k) \\ \tilde{w}(k) \end{bmatrix} \frown (0, I) \tag{6.19}$$

and the matrices $X(k)$ and $Q_x(k)$ satisfying

$$X(k) = S(k)R(k)^{-T/2} \tag{6.20}$$

$$Q_x(k) = Q(k) - S(k)R(k)^{-1}S(k)^T \tag{6.21}$$

It is assumed that there is no cross correlation between $v(k)$ and $w(k)$, so $S(k) = 0$, simplifying (6.20)-(6.21) into

$$X(k) = 0 \tag{6.22}$$

$$Q_x(k) = Q(k) \tag{6.23}$$

6.2.2 The least squares problem

The general covariance representation (6.18) containing the statistical information of the initial state can now be combined with the state equations (A.3)-(A.4), as shown in Verhaegen & Verdult (2001). With $S(k) = 0$, all this information can be represented together for $k = 0, 1, .., N$:

$$\begin{aligned} \hat{x}(k|k-1) &= x(k) + S(0|k-1)\tilde{u}(0) && (6.24) \\ y(k) - Du(k) &= Cx(k) + R(k)^{1/2}\tilde{v}(k) && (6.25) \\ -Bu(k) &= Ax(k) - x(k+1) + Q^{1/2}\tilde{w}(k) && (6.26) \end{aligned}$$

The left hand side of the equations (6.24)-(6.26) contains all known information, and the right hand side contains the unknown quantities and the noise contributions. In Verhaegen & Verdult (2001), the D matrix was assumed zero, which is not the case for the wind turbine system model. It can be seen in (6.25) that the term $Du(k)$ appears as an offset of the output vector $y(k)$. The equations (6.24)-(6.26) can be written as a least squares problem:

$$\begin{bmatrix} \hat{x}(k|k-1) \\ y(k) - Du(k) \\ -Bu(k) \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ C & 0 \\ A & -I_n \end{bmatrix} \begin{bmatrix} x(k) \\ x(k+1) \end{bmatrix} + \begin{bmatrix} S(k|k-1) & 0 & 0 \\ 0 & R(k)^{1/2} & 0 \\ 0 & 0 & Q(k)^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{u}(k) \\ \tilde{v}(k) \\ \tilde{w}(k) \end{bmatrix} \quad (6.27)$$

Let this be denoted as

$$m(k) = F(k)\xi(k) + L(k)\mu(k) \quad (6.28)$$

Duncan & Horn (1972) showed that the state estimates can be obtained by minimizing

$$\min_{\xi} \mu^T \mu \quad (6.29)$$

Verdult & Verhaegen (2001) solve this minimization problem by exploiting the sparseness of the matrices $F(k)$ and $L(k)$. This leads to a recursive procedure. Assuming invertibility of $L(k)$, which is always the case if there is no cross correlation between $v(k)$ and $w(k)$, solving (6.29) is equivalent to minimizing

$$\|L^{-1}(k)F(k)\xi(k) - L^{-1}(k)m(k)\|_2^2 \quad (6.30)$$

Verdult & Verhaegen (2001) use the QR decomposition

$$L^{-1}(k)F(k) = T(k)R(k) = T(k) \begin{bmatrix} \bar{R}(k) & \bar{G}(k) \\ 0 & S(k|k+1)^{-1} \\ 0 & 0 \end{bmatrix} \quad (6.31)$$

where $T(k)$ is an orthogonal matrix, to rewrite the minimization problem (6.30) as

$$\left| \begin{bmatrix} \bar{R}(k) & \bar{G}(k) \\ 0 & S(k+1|k)^{-1} \end{bmatrix} \begin{bmatrix} x(k) \\ x(k+1) \end{bmatrix} - \begin{bmatrix} \bar{c}(k) \\ c(k+1) \end{bmatrix} \right|_2^2 \quad (6.32)$$

with

$$\begin{aligned} \bar{c}(k) &= [I_n \; 0_{n\times(n+\ell)}]T(k)^T L(k)^{-1} m(k) \\ c(k+1) &= [0_{n\times n} \; I_n \; 0_{n\times\ell}]T(k)^T L(k)^{-1} m(k) \end{aligned}$$

The solution to this set of equations only depends on measurements up to time instant k, therefore it equals the one-step ahead predictor $\hat{x}(k+1|k)$ of the state and the measurement update $\hat{x}(k|k)$ of the state:

$$\begin{aligned}\hat{x}(k+1|k) &= S(k+1|k)c(k+1)\\ \hat{x}(k|k) &= \bar{R}(k)^{-1}\big(\bar{c}(k) - \bar{G}(k)\hat{x}(k+1|k)\big)\end{aligned}$$

The one-step ahead predictor $\hat{x}(k+1|k)$ and the Cholesky factor $S(k+1|k)$ can be extracted from $R(k)$ in (6.31), and they can be used to initialize the LMS problem in the next time step.

A more direct way to solve (6.30) is obtained, when (6.31) remains written as

$$L^{-1}(k)F(k) = T(k)R(k)$$

The minimization problem (6.30) can then be reduced to

$$\min_{\xi} \|R(k)\xi(k) - T(k)^T L^{-1}(k)m(k)\|_2^2 \tag{6.33}$$

which allows $\xi(k)$ to be found:

$$\xi(k) = R^{\dagger}(k)T(k)^T L^{-1}(k)m(k) \tag{6.34}$$

where the superscript $\dagger$ stands for the left Penrose-Moore pseudo-inverse of the matrix $R(k)$, denoted as $R^{\dagger}(k)$, given by $(R^T(k)R(k))^{-1}R^T(k)$.

6.3 The Kalman filter over a time window

This section introduces a new method to identify the unknown pitch actuator gain $\sigma(k)$, which is based on the least squares formulation of the Kalman filter problem described in section 6.2. A moving time window w is chosen, and the unknown pitch actuator gain $\sigma(k)$ is considered as a constant during this time window. The Kalman filter formulation as a least squares problem is then rewritten in a way, that allows to make an estimate of $\sigma(k)$ as an average value during this time window. Only the MIMO setup is considered: the algorithm uses the measurements $y_1 = \ddot{x}_{nd}$ and $y_2 = \Delta\Omega_r$.

6.3.1 The algorithm

Recall the wind turbine system (6.10)-(6.11) with the wind speed $v_w(k)$ included in an augmented state, and the unknown pitch actuator gain denoted as $\sigma(k)$:

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ v_w(k+1) \end{bmatrix} &= \begin{bmatrix} A & B_h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \sigma(k)u(k) + \begin{bmatrix} Q^{1/2} & 0 \\ 0 & Q_{v_w}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{w}(k) \\ \tilde{w}_{v_w}(k) \end{bmatrix} \\
\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} &= \begin{bmatrix} C_1 & D_h \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} \sigma(k)u(k) + R^{1/2}\tilde{v}(k)
\end{aligned}$$

This augmented system can be denoted in a more compact way:

$$\begin{aligned}
\breve{x}(k+1) &= \breve{A}\breve{x}(k) + \breve{B}\sigma(k)u(k) + \breve{Q}^{1/2}\breve{w}(k) && (6.35) \\
y(k) &= \breve{C}\breve{x}(k) + \breve{D}\sigma(k)u(k) + R^{1/2}\tilde{v}(k) && (6.36)
\end{aligned}$$

with the augmented state vector $\breve{x}(k) \in \mathbb{R}^{n+1}$. The least squares problem for the augmented system (6.35)-(6.36) can be written in analogy with (6.27):

$$\begin{aligned}
\begin{bmatrix} \hat{x}(k|k-1) \\ y(k) - \breve{D}\sigma(k)u(k) \\ -\breve{B}\sigma(k)u(k) \end{bmatrix} &= \begin{bmatrix} I_{n+1} & 0 \\ \breve{C} & 0 \\ \breve{A} & -I_{n+1} \end{bmatrix} \begin{bmatrix} x(k) \\ x(k+1) \end{bmatrix} \\
&+ \begin{bmatrix} S(k|k-1) & 0 & 0 \\ 0 & R(k)^{1/2} & 0 \\ 0 & 0 & \breve{Q}(k)^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{u}(k) \\ \tilde{v}(k) \\ \breve{w}(k) \end{bmatrix} \qquad (6.37)
\end{aligned}$$

A time window of size $w \in \mathbb{N}^+$ can now be defined. The LMS problem (6.37) is then rewritten for a moving time window as the large LMS problem (6.38) that contains past and present input data $u(k-w+1), \ldots, u(k)$ and output data $y(k-w+1), \ldots, y(k)$. This LMS problem can be solved to estimate the turbine states and the wind speed, and it also yields an estimate of the pitch actuator gain σ as an average value during the time window w.

In comparison with (6.37), this requires that the terms $-\breve{D}\sigma(k)u(k)$ and $-\breve{B}\sigma(k)u(k)$ for all $k \in [k-w+1, k]$ are moved to the right hand side of the LMS problem (6.38), of which a compact notation is shown in (6.39). The elements $\breve{D}u(k)$ and $\breve{B}u(k)$ for all $k \in [k-w+1, k]$ appear in the matrix that corresponds to $F(k)$ in (6.39). The unknown pitch actuator gain $\sigma(k)$ is assumed to be constant during the time window w, and appears as the last term of the vector that corresponds to $\xi(k)$ in (6.39).

$$\begin{bmatrix} \hat{x}(k-w+1|k-w) \\ y(k-w+1) \\ 0 \\ y(k-w+2) \\ 0 \\ \vdots \\ y(k) \\ 0 \end{bmatrix} = \begin{bmatrix} I_{n+1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \breve{C} & 0 & 0 & \cdots & 0 & 0 & \breve{D}u(k-w+1) \\ \breve{A} & -I_{n+1} & 0 & \cdots & 0 & 0 & \breve{B}u(k-w+1) \\ 0 & \breve{C} & 0 & \cdots & 0 & 0 & \breve{D}u(k-w+2) \\ 0 & \breve{A} & -I_{n+1} & \cdots & 0 & 0 & \breve{B}u(k-w+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \breve{C} & 0 & \breve{D}u(k) \\ 0 & 0 & 0 & \cdots & \breve{A} & -I_{n+1} & \breve{B}u(k) \end{bmatrix} \begin{bmatrix} x(k-w+1) \\ x(k-w+2) \\ x(k-w+3) \\ \vdots \\ x(k) \\ x(k+1) \\ \sigma(k) \end{bmatrix} +$$
$$\begin{bmatrix} \hat{S}(k-w+1|k-w) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & R^{1/2} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \breve{Q}^{1/2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & R^{1/2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \breve{Q}^{1/2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & R^{1/2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \breve{Q}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{u}(k-w+1) \\ \tilde{v}(k-w+1) \\ \breve{w}(k-w+1) \\ \tilde{v}(k-w+2) \\ \breve{w}(k-w+2) \\ \vdots \\ \tilde{v}(k) \\ \breve{w}(k) \end{bmatrix} \tag{6.38}$$

As mentioned, this large LMS problem can be denoted in a compact way as (6.39), in analogy with (6.28):

$$m(k) = F(k)\xi(k) + L(k)\mu(k) \tag{6.39}$$

This LMS problem can be solved in the exact same way as (6.29) in section 6.2. The solution yields the measurement update $\hat{x}(k-w+1|k-w)$ and the one-step ahead predictor $\hat{x}(k-w+2|k-w+1)$, as well as $\sigma(k)$, based on the past and present inputs and outputs during the selected time window. Effectively, the state estimates are calculated with a delay of w sample times, in order to obtain an average fault estimate over the selected time window. At the end of a recursive procedure from $k = 1, \ldots, N$, estimates of the turbine states and the wind speed are available over the interval $k = 1, \ldots, N - w$, as well as an estimate of the pitch actuator gain σ over the interval $k = w + 1, \ldots, N$.

The solution $\xi(k)$ of the minimization problem (6.39) and the matrix $R(k)$ from the QR factorization have the structure shown in (6.40)-(6.41). Analogous to Verdult & Verhaegen (2001), the one-step ahead predictor $\hat{x}(k-w+2|k-w+1)$ can be taken from $x(k-w+2)$ in $\xi(k)$ in (6.41) and the measurement update $\hat{x}(k-w+1|k-w+1)$ from $x(k-w+1)$. The last element of the vector $\xi(k)$ is the estimate of $\sigma(k)$, as an average over the time window $k - w + 1, \ldots, k$.

$$R(k) = \begin{bmatrix} \bar{R}_{k-w+1}(k) & \bar{G}_{k-w+1}(k) & 0 & 0 & 0 & \Sigma_{k-w+1}(k) \\ 0 & \bar{R}_{k-w+2}(k) & \bar{G}_{k-w+2}(k) & 0 & 0 & \vdots \\ 0 & 0 & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & \bar{R}_k(k) & \bar{G}_k(k) & \vdots \\ 0 & 0 & 0 & 0 & \bar{R}_{k+1}(k) & \Sigma_{k+1}(k) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.40)$$

$$\xi(k) = \begin{bmatrix} x(k-w+1) \\ x(k-w+2) \\ \vdots \\ x(k) \\ x(k+1) \\ \sigma(k) \end{bmatrix} \quad (6.41)$$

The matrix $R(k)$ shown in (6.40) consists of a block diagonal matrix, containing matrices $\bar{R}(k)$ and $\bar{G}(k)$ analogous to (6.31), and an extra column which contains the vector $\Sigma \in \mathbb{R}^{(w+1)\cdot(n+1)+1}$ that contains the R factorization terms corresponding to $\sigma(k)$, which are not used for analysis. In analogy with Verdult & Verhaegen (2001), $\bar{R}_{k-w+2}(k)$ is equal to $S(k-w+2|k-w+1)^{-1}$, the inverse of the Cholesky factor of the covariance matrix of the predicted state $\hat{x}(k-w+2|k-w+1)$ that is needed to initialize the next time step.

6.3.2 Experimental issues

The Kalman filter over a time window w, as described in section 6.3.1, was implemented in the m-file *linkalwin.m*. It was demonstrated in experiments that this algorithm can be used to obtain an estimate of the faulty actuator gain $\sigma(k)$. However, it turned out to be difficult to obtain reliable estimates. To understand this, the algorithm must be compared to the regular discrete-time Kalman filter algorithm that is described in section 6.1.

If the augmented wind turbine system (6.10)-(6.11) is solved with a regular discrete-time Kalman filter, the noise covariance Q on the state estimates can be set to zero. As can be seen in (6.6), it is not required that the matrix Q is invertible. The augmented state variables $v_w(k)$ and $\sigma(k)$ have random walk parameters Q_{v_w} and Ψ, respectively. These random walk parameters can be tuned in such a way that both $v_w(k)$ and $\sigma(k)$ can be estimated despite the presence of measurement noise.

In the Kalman filter algorithm over a moving time window, several problems arise. First of all, unlike in the regular discrete-time Kalman filter formulation, no tunable step size is available in the Kalman filter over a moving time window. Such a step size effectively introduces a memory functionality in the estimate of $\sigma(k)$, since every next estimate is only allowed to deviate from the previous estimate to a certain extent. In the Kalman filter (6.38) over a time window, the pitch actuator gain $\sigma(k)$ is estimated as an average value during the time window w, without a constraint on the allowed step size.

Secondly, the covariance matrix Q of the noise on the estimated states cannot be set to zero in the Kalman filter algorithm over a time window. The matrix $L(k)$ is not invertible for $Q = 0$, so that the minimization problem (6.39) cannot be solved. The tuning parameters of the LMS problem (6.38) are therefore the covariance matrix Q of the turbine system and the covariance matrix Q_{v_w} of the augmented wind state. This means that the Kalman filter over a time window has $n+1$ tuning parameters, with the system order $n = 10$, while the regular discrete-time Kalman filter has only two. It is therefore difficult to tune the parameters in such a way that reliable estimates are obtained for the turbine states, the wind speed and the pitch actuator gain. It was found that a large value of Q yields a biased estimate of σ. A possible explanation is that the actual disturbance by the wind signal is non-white and has a small non-zero mean value, which results in an offset in the estimate of σ. When Q is large, this offset is multiplied with a larger value so that it shows itself clearly on the estimate of σ. When a smaller value of Q is taken, this bias is not visible. However, it was found that the estimate of σ is very noisy in this case, with a lot of variation. Two experiments will be used to illustrate this behavior.

Experiments

The same abrupt fault as in section 6.1 was simulated: the pitch actuator gain $\sigma(k)$ changes from 1 to 2/3 at time instant $k = 500$. Standard measurement noise is present in the turbine system outputs, as described in section 3.6. The first experiment uses a relatively small value of Q: the diagonal matrix denoted in (6.42), with $q = 1.0 \cdot 10^{-3}$.

$$Q = q \cdot \text{diag}\left(\begin{bmatrix} 1.5 \cdot 10^{-3} \\ 1.5 \cdot 10^{-3} \\ 7.0 \cdot 10^{-5} \\ 5.5 \cdot 10^{-4} \\ 2.1 \cdot 10^{-3} \\ 4.5 \cdot 10^{-7} \\ 8.2 \cdot 10^{-7} \\ 1.5 \cdot 10^{-7} \\ 1.2 \cdot 10^{-7} \\ 2.2 \cdot 10^{-8} \end{bmatrix}\right) \tag{6.42}$$

The window size w was set to 25, and $Q_{v_w} = 0.1$. This yields the wind estimate shown in figure 6.14, which has a VAF between real and estimated wind speed of 99.5%. The estimate of $\sigma(k)$ is shown in figure 6.15. Though the VAF between real and simulated σ was found to be 91.9%, it can be seen that the estimated $\sigma(k)$ is quite noisy. These rapid variations have the result that the numerical derivative of $\sigma(k)$ is not a good indicator for a fault occurrence: the derivative is too noisy.

A discrete-time low-pass filter (6.43) was implemented in analogy with Kanev & Verhaegen (2000), yielding a filtered value $\sigma^{filt}(k)$ of the estimate $\sigma(k)$. This filtered estimate $\sigma^{filt}(k+1)$ is mainly based on the previous filtered estimate $\sigma^{filt}(k)$, and not on the actual estimate $\sigma(k)$, so that an artificial restriction on the step size of the estimate is introduced.

$$\sigma^{filt}(k+1) = 0.98 \cdot \sigma^{filt}(k) + 0.02 \cdot \sigma(k) \tag{6.43}$$

The filtered estimate $\sigma^{filt}(k)$ is shown as the dashed line in figure 6.15. It follows the simulated $\sigma(k)$ reasonably well, but reacts slowly to a change. The following VAF between real and estimated turbine states (which are not plotted) was found (in %):

$$VAF(x_e, x_r) = \begin{bmatrix} 99.8 & 99.7 & 98.1 & 99.6 & 99.8 & 97.3 & 98.2 & 85.4 & 83.7 & 99.2 \end{bmatrix}^T$$

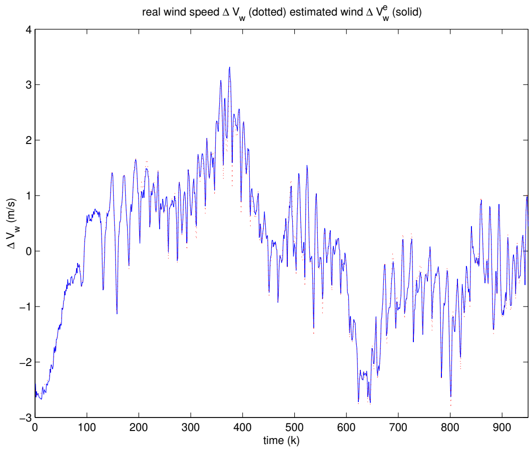

Figure 6.14: *First experiment with the linear Kalman filter over a time window: the estimate of the wind speed $v_w(k)$. Here, $w = 25$ and Q is defined as (6.42) with $q = 1.0 \cdot 10^{-3}$.*

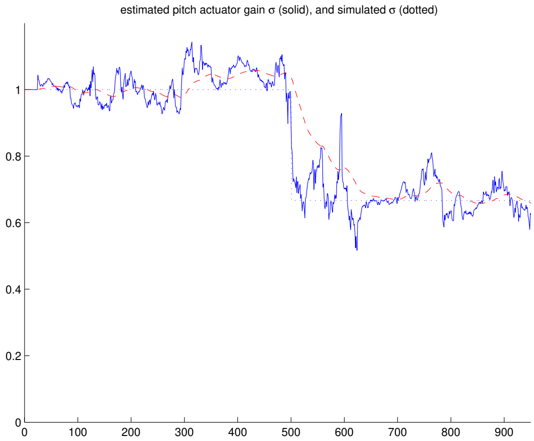

Figure 6.15: *First experiment with the linear Kalman filter over a time window: simulated pitch actuator gain (dotted), the estimated pitch actuator gain $\sigma(k)$ (solid) and the filtered estimate $\sigma^{filt}(k)$ (6.43). Here, $w = 25$ and Q is defined as (6.42) with $q = 1.0 \cdot 10^{-3}$.*

A second experiment, with larger values of Q and Q_{v_w}, yields an example of a biased fault estimator. The value of q in (6.43) was set to 1; with $Q_{v_w} = 1.4$ and $w = 30$, this yields the estimate of $\sigma(k)$ shown in figure 6.16. This is a biased estimate, since the estimated $\sigma(k)$ has an offset: it is approximately 0.1 larger than the simulated value, for all k.

Apart from the difficult tuning and the fact that there is no restriction on the step size of $\sigma(k)$, there is a third problem that complicates practical application of the Kalman filter algorithm over a time window. The computation time is very large: it was found to be proportional to w^3, the cube of the window size. This is a logical result, since a QR factorization was used to obtain the solution $\xi(k)$ for the minimization problem (6.39). Golub & Loan (1989) show that the required computation time for QR factorization is proportional to the cube of the rank of the matrix $F(k)$ in (6.39), and the rank of the (full-rank) matrix $F(k)$ is indeed proportional to the window size. A more efficient QR factorization algorithm can be implemented that is proportional to the square of the window size, as described in Golub & Loan (1989), but this still yields a heavy computational burden. The computation time T_{cpu} that required for one cycle of the Kalman filter problem over a time window is shown in figure 6.17, with the empirical approximation (6.44). As a comparison, the regular discrete-time Kalman filter has a T_{cpu} of $5.4 \cdot 10^{-4}$s for one cycle of the algorithm.

$$T_{cpu} = 5.4 \cdot 10^{-5} \cdot w^3 \tag{6.44}$$

It follows from (6.44) that the computation time is already larger than the sample time $T_s = 0.1$s for a window size of 15; this size turns out to be too small to yield a reliable estimate of $\sigma(k)$. A larger window size w reduces the variation in the estimate, but the results were never as reliable as with the regular discrete-time Kalman filter described in section 6.1.

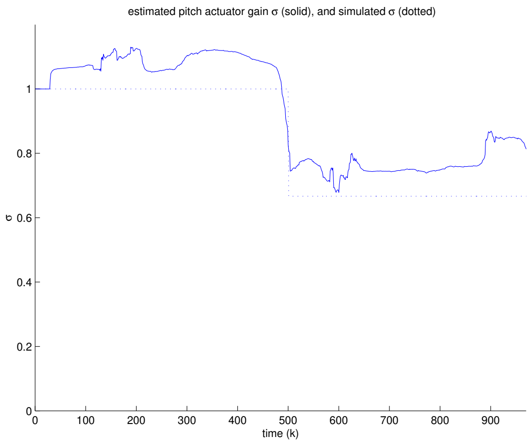

Figure 6.16: *Second experiment with the linear Kalman filter over a time window: simulated pitch actuator gain (dotted) and the estimated pitch actuator gain $\sigma(k)$ (solid). Here, $w = 30$ and Q relatively large: defined as in (6.42), with $q = 1.0$.*

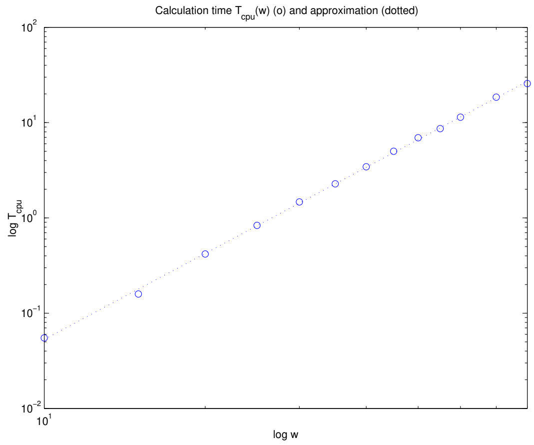

Figure 6.17: *The computation time T_{cpu} required for one cycle of the linear Kalman filter problem over a time window, for several values of the window size w. The empirical relation (6.44) was found on this data, which is shown as a dotted line.*

Analysis of the results

In section 6.2, the Kalman filter algorithm was formulated as an LMS problem. This LMS framework was rewritten in this section into a linear LMS problem, defined during a moving time window of size w. Past and present input and output data were used to estimate the pitch actuator gain as an average value during this time window. Only the MIMO case was considered: the algorithm uses measurements of $\ddot{x}_{nd}$ and $\Delta\Omega_r$. Standard measurement noise was present in the system outputs, and an abrupt change in the pitch actuator gain σ was simulated.

It was demonstrated that this algorithm can be used to estimate the states of the wind turbine system, as well as the wind speed and the pitch actuator gain. However, it turned out that the algorithm is not very promising for practical application: no real-time application is to be expected.

In comparison with the discrete-time Kalman filter algorithm that is described in section 6.1, the Kalman filter over a time window has a much higher computation time for one cycle of the algorithm. This computation time T_{cpu} is proportional to the cube of the window size w, and a relatively large value of w is needed. Furthermore, since all values of the state noise covariance matrix Q have to be non-zero, the algorithm has 11 tuning parameters instead of 2. Finally, there is no limited step size for the estimate of σ, that serves as a memory to prevent large steps in the subsequent estimates. This results in an algorithm that is very difficult to tune: either a noisy estimate of the pitch actuator gain $\sigma(k)$ is obtained, or the estimate is biased. A filtered version of this noisy estimate can be used to detect and isolate the fault; considering the fact that a regular discrete-time Kalman filter doesn't yield a noisy estimate, it can be stated that the Kalman filter over a time window first creates a problem, which is then partly solved. It is clear that this method is not worth the computational effort.

6.4 A nonlinear Kalman problem

In this section, another new method to estimate the unknown pitch actuator gain $\sigma(k)$ is presented. Analogous to section 6.3.1, the wind speed $v_w(k)$ is included in an augmented system state. A nonlinear LMS problem with a particular structure can be defined over a moving time window w, which can be solved as a so-called separable least squares (SLS) problem. A nonlinear optimization scheme is used to estimate an optimal value $\hat{\sigma}^{opt}$ as an average over the time window w. This $\hat{\sigma}^{opt}$ reduces the nonlinear LMS problem to a linear LMS problem, from which the turbine states and the wind speed can be estimated. Only the MIMO setup is considered: the algorithm uses the measurements $y_1 = \ddot{x}_{nd}$ and $y_2 = \Delta\Omega_r$.

6.4.1 The algorithm

Recall the state-space representation (4.10)-(4.11) of the transfer function from $u_1(k) = v_w(k)$ to the system outputs $y(k)$. In analogy with section 6.1.1, the substitutions $u_1(k) = v_w(k)$, $B_1 = B_h$ and $B_2 = B$ are made, as well as $D_{11} = D_h$ and $D_{12} = D$. The unknown pitch actuator gain is represented as σ, which yields the state-space representation (6.45)-(6.46):

$$\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \end{bmatrix} = \begin{bmatrix} A - B\sigma\bar{D}C_2 & -B\sigma\bar{C} \\ \bar{B}C_2 & \bar{A} \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \begin{bmatrix} B_h \\ 0 \end{bmatrix} v_w(k) \tag{6.45}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 - D\sigma\bar{D}C_2 & -D\sigma\bar{C} \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \end{bmatrix} + \begin{bmatrix} D_h \\ 0 \end{bmatrix} v_w(k) \tag{6.46}$$

Measurement noise $v(k)$ and process noise $w(k)$ is added, in analogy with section 6.1. The wind speed $v_w(k)$ is defined as a random walking variable as in (6.9), and included in an augmented state. The the noise term on the controller state is defined as a zero-mean unit variance term $\tilde{w}_c(k) \frown (0, I)$, multiplied with the square root of the covariance matrix Q_c. This allows (6.45)-(6.46) to be rewritten as (6.47)-(6.48):

$$\begin{bmatrix} x(k+1) \\ \bar{x}(k+1) \\ v_w(k+1) \end{bmatrix} = \begin{bmatrix} A - B\sigma\bar{D}C_2 & -B\sigma\bar{C} & B_h \\ \bar{B}C_2 & \bar{A} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} Q^{1/2} & 0 & 0 \\ 0 & Q_c^{1/2} & 0 \\ 0 & 0 & Q_{v_w}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{w}(k) \\ \tilde{w}_c(k) \\ \tilde{w}_{v_w}(k) \end{bmatrix} \tag{6.47}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 - D\sigma\bar{D}C_2 & -D\sigma\bar{C} & D_h \\ C_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{x}(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} R_1^{1/2} & 0 \\ 0 & R_2^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{v}_1(k) \\ \tilde{v}_2(k) \end{bmatrix} \tag{6.48}$$

This can be denoted in a more compact way:

$$\breve{x}(k+1) = \breve{A}(\sigma)\breve{x}(k) + \breve{Q}\breve{w}(k) \tag{6.49}$$

$$y(k) = \breve{C}(\sigma)\breve{x}(k) + R(k)^{1/2}\tilde{v}(k) \tag{6.50}$$

with the appropriate definitions of the augmented system matrices in (6.47)-(6.48), and the augmented state vector $\breve{x} \in \mathbb{R}^{n+2}$. In analogy with (6.27), this can be written as the LMS problem (6.51), where $\hat{x}(k|k-1)$ is the estimate of $\breve{x}(k)$ from the prior time step, and $S(k|k-1)$ the Cholesky factor of the covariance matrix that is associated with $\hat{x}(k|k-1)$.

$$\begin{bmatrix} \hat{x}(k|k-1) \\ y(k) \\ 0 \end{bmatrix} = \begin{bmatrix} I_{n+2} & 0 \\ \breve{C}(\sigma) & 0 \\ \breve{A}(\sigma) & -I_{n+2} \end{bmatrix} \begin{bmatrix} \breve{x}(k) \\ \breve{x}(k+1) \end{bmatrix} + \begin{bmatrix} \breve{S}(k|k-1) & 0 & 0 \\ 0 & R(k)^{1/2} & 0 \\ 0 & 0 & \breve{Q}(k)^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{u}(k) \\ \tilde{v}(k) \\ \breve{w}(k) \end{bmatrix} \tag{6.51}$$

Note that the controller output $u(k)$ is not included in (6.51): the behavior of the entire closed-loop system is considered, so that $u(k)$ has become an internal variable of the closed-loop system. The controller output $u(k)$ is denoted as $\bar{u}_2(k)$ in section 4.1, and (4.7) shows that it can be reconstructed as a linear combination of the controller state $\bar{x}(k)$ and the turbine system states $x(k)$.

A similar approach as in section 6.3.1 can now be applied. The LMS problem (6.51) can be defined during a moving time window of size $w \in \mathbb{N}^+$, where the pitch actuator gain $\sigma(k)$ is assumed to have the constant value σ during this time window.

$$\begin{bmatrix} \hat{x}(k-w+1|k-w) \\ y(k-w+1) \\ 0 \\ y(k-w+2) \\ 0 \\ \vdots \\ y(k) \\ 0 \end{bmatrix} = \begin{bmatrix} I_{n+2} & 0 & 0 & \cdots & 0 & 0 \\ \breve{C}(\sigma) & 0 & 0 & \cdots & 0 & 0 \\ \breve{A}(\sigma) & -I_{n+2} & 0 & \cdots & 0 & 0 \\ 0 & \breve{C}(\sigma) & 0 & \cdots & 0 & 0 \\ 0 & \breve{A}(\sigma) & -I_{n+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \breve{C}(\sigma) & 0 \\ 0 & 0 & 0 & \cdots & \breve{A}(\sigma) & -I_{n+2} \end{bmatrix} \begin{bmatrix} \breve{x}(k-w+1) \\ \breve{x}(k-w+2) \\ \breve{x}(k-w+3) \\ \vdots \\ \breve{x}(k) \\ \breve{x}(k+1) \end{bmatrix} + \begin{bmatrix} \breve{S}(k-w+1|k-w) & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & R^{1/2} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \breve{Q}^{1/2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & R^{1/2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \breve{Q}^{1/2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & R^{1/2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \breve{Q}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{u}(k-w+1) \\ \tilde{v}(k-w+1) \\ \breve{w}(k-w+1) \\ \tilde{v}(k-w+2) \\ \breve{w}(k-w+2) \\ \vdots \\ \tilde{v}(k) \\ \breve{w}(k) \end{bmatrix} \tag{6.52}$$

This LMS problem can be denoted as:

$$m(k) = F(\sigma)\xi(k) + L(k)\mu(k) \tag{6.53}$$

According to Duncan & Horn (1972) and Verdult & Verhaegen (2001), the following minimization leads to the solution:

$$\min_{\xi,\sigma} \|L^{-1}(k)F(\sigma)\xi(k) - L^{-1}(k)m(k)\|_2^2 \tag{6.54}$$

Unfortunately, finding the solution for (6.53) is not straightforward: (6.54) is a nonlinear minimization problem, which has the following general structure:

$$\min_{\xi,\sigma} \|Y(k) - \Phi(\sigma)X(k)\|_2^2 \tag{6.55}$$

This minimization problem is linear in $X(k)$ and nonlinear in σ. This means that for a given value of σ, it reduces to an ordinary linear LMS problem.

This class of problems was described by Golub & Pereyra (1973), and the so-called separable least-squares (SLS) method was presented as a solution. A thorough description of the method is given by Verdult (2002). The problem can be solved in two steps. In the first step, a nonlinear minimization problem in σ is solved, yielding an optimal $\hat{\sigma}^{opt}$ for the minimization problem described in (6.55). A theorem by Golub & Pereyra (1973) shows, that solving (6.55) is equal to solving (6.56) in order to find the optimal value $\hat{\sigma}^{opt}$ of the faulty pitch actuator gain σ.

$$\hat{\sigma}^{opt} = \min_{\sigma} \|Y(k) - \Phi(\sigma)\big(\Phi(\sigma)^T\Phi(\sigma)\big)^{-1}\Phi(\sigma)^T Y(k)\|_2^2 \tag{6.56}$$

The minimization problem (6.56) is solved with a nonlinear optimization scheme to find $\hat{\sigma}^{opt}$. Subsequently, $\hat{\sigma}^{opt}$ can be substituted into (6.53) in order to reduce it to (6.28):

$$m(k) = F(k)\xi(k) + L(k)\mu(k)$$

This is an ordinary linear LMS problem that can be solved in the same way as (6.29) in section 6.2.

6.4.2 Experimental issues

In order to find the optimal $\hat{\sigma}_{opt}$, the minimization problem (6.56) was solved with Matlab's nonlinear optimization function *lsqnonlin.m*. This required a function *jnsig.m* to be made, that creates the output vector $J_N(\sigma)$ that represents the cost function of (6.56) as a function of σ:

$$J_N(\sigma) = Y(k) - \Phi(\sigma)\big(\Phi(\sigma)^T\Phi(\sigma)\big)^{-1}\Phi(\sigma)^T Y(k) \tag{6.57}$$

The function *lsqnonlin.m* is used to minimize (6.56), with the initial estimate $\sigma(k|k-1)$ set to 1 for the first time step. The upper and lower boundary for σ were set in advance, so that $0 < \sigma < 1.2$. The result of this minimization is an optimal value $\hat{\sigma}^{opt}(k)$ that can be used as the initial estimate for σ in the next time step; compared with initializing $\sigma = 1$ at every time step, this saves calculation time. Furthermore, the optimal value $\hat{\sigma}^{opt}$ is substituted into (6.52), so that it can be solved as an ordinary linear LMS problem. This yields estimates for the turbine and controller states, as well as an estimate for the wind speed. The complete implementation of this procedure can be found in the m-file *nonlinkalwin.m*.

It was found that the nonlinear Kalman filter problem during a moving time window requires even more computation time than the linear Kalman filter problem during a moving time window that was described in section 6.3.2. For a certain window size w, both algorithms solve a QR factorization of matrices of comparable size, but before this, the nonlinear Kalman filter problem solves a nonlinear optimization problem to estimate the pitch actuator gain σ. Furthermore, the nonlinear Kalman filter algorithm has even more tuning parameters: instead of two tuning parameters in the regular discrete-time Kalman filter of section 6.1 and $n+1$ in the linear Kalman filter problem during a time window, the nonlinear problem has $n+2$ tuning parameters. The reason is that the controller state is included, so that its covariance Q_c must be tuned, as it is not allowed to be zero. Similar to the linear Kalman filter problem over a time window, the nonlinear Kalman filter algorithm yields a biased fault estimate when relatively high values of Q, Q_c and Q_{v_w} are used. Tuning the parameters was found to be very difficult, as lower parameter values yield an estimate that is more noisy, so that the estimated pitch actuator gain $\hat{\sigma}^{opt}$ sometimes lies on the upper (or lower) boundary.

Experiment

An example of a biased estimate of σ is obtained, when Q is chosen as in (6.43) with $q=1$, $Q_{v_w}=1.4$, $Q_c=0.001$ and $w=30$. The estimated wind speed is shown in figure 6.18, with corresponding VAF of 94.1%. The pitch actuator gain is shown in figure 6.19, a biased estimate with corresponding VAF of 78.9%, that sometimes lies on the upper boundary.

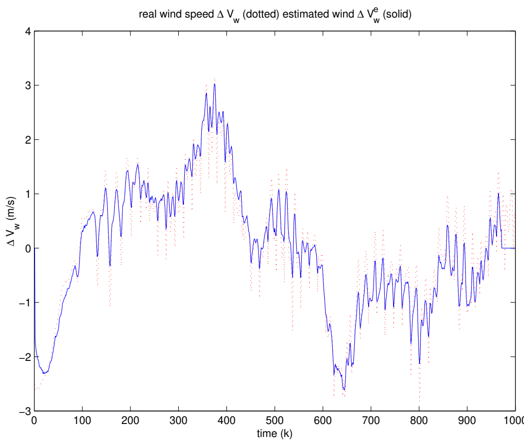

Figure 6.18: *The estimate of the wind speed $v_w(k)$ with the nonlinear Kalman filter algorithm, when a relatively large value of Q is chosen, and $w=30$.*

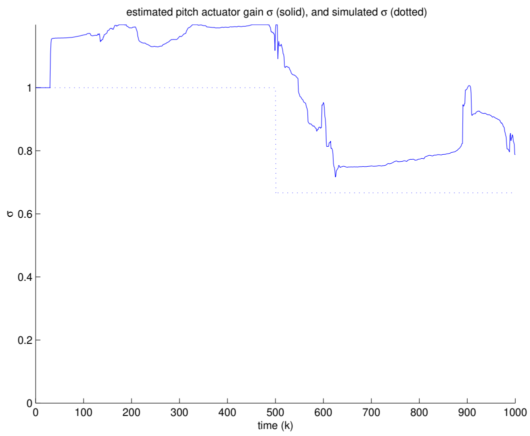

Figure 6.19: *The dotted line represents the simulated pitch actuator gain σ, the solid line shows the biased estimate of the pitch actuator gain σ that was obtained with the nonlinear Kalman filter algorithm.*

Analysis of the results

The nonlinear Kalman filter problem defined over a moving time window is not a very reliable method to obtain estimates of the wind turbine states, the wind speed and the pitch actuator gain. It is even slower than the linear Kalman filter algorithm over a moving time window presented in section 6.3.1: for a given window size w, it solves a linear LMS problem problem of comparable size, but during each time cycle it performs an extra nonlinear minimization procedure to estimate the pitch actuator gain. Similar to the linear Kalman filter over a time window, there is no limited change rate for the estimated pitch actuator gain in the nonlinear Kalman filter algorithm. Finally, tuning the nonlinear algorithm is even more difficult: the controller state is included, which introduces the associated noise covariance Q_c as an extra tuning parameter.

It was demonstrated that the nonlinear LMS problem can be used to estimate the states of the wind turbine system, as well as the wind speed and the pitch actuator gain. Because of the difficult tuning, either a biased estimate is found, or a noisy estimate that sometimes lies on the upper (or lower) boundary. Clearly, this method is even more troublesome than the linear Kalman filter over a time window and wastes even more computation time.

6.5 Conclusions

In this chapter, methods based on the Kalman filter were used to detect and identify an unknown pitch actuator gain in a HAWT simulation model.

The discrete-time Kalman filter algorithm was shown to be a very powerful method to estimate unknown variables in a wind turbine system, when the system representation of the fault-free system is known. The unknown variables can be defined as random walking variables and included in an augmented system state, so that they can be estimated by the Kalman filter. Both system outputs $\Delta\Omega_r$ and $\ddot{x}_{nd}$ had to be used to obtain good results.

For the problem of identifying an unknown pitch actuator gain, it was shown that the wind speed can be estimated to a high degree of accuracy, and that a reliable value of the unknown pitch actuator gain can be estimated from closed-loop measurements, even when measurement noise is present in the system outputs. The random walk sizes of the unknown variables turned out to be important tuning parameters: lower values were required when measurement noise was present in the system outputs, in order to reduce the freedom of the estimates.

An alternative path that was explored turned out to be less promising. It was demonstrated that the Kalman filter can be rewritten over a moving time window both as a linear and a nonlinear LMS algorithm. The linear problem is based on the open-loop transfer function from the controller input to the system outputs. The nonlinear problem is based on the closed-loop transfer function of the wind turbine system, where the unknown pitch actuator gain shows itself as a component fault that can be estimated by solving a nonlinear minimization problem.

It was demonstrated that both methods can be used to estimate the pitch actuator gain $\sigma(k)$ as an average during the time window, as well as the turbine states and wind speed. However, these algorithms are not very reliable and practical. The main shortcoming is the lack of a tuning parameter that restricts the freedom of the fault estimate, and the fact that the algorithms have too much other tuning parameters. Therefore, there is too much freedom on the estimate of $\sigma(k)$, even for a large window size. It was demonstrated that these problems result in either noisy or biased estimates. Finally, the computation time for these algorithms is very high, especially for the nonlinear algorithm. Considering that the estimation results were worse than with the regular discrete-time Kalman filter algorithm, it can be concluded that these methods are of little practical relevance.

Chapter 7

Multiple-model estimation

7.1 Introduction

In this chapter, the hybrid estimation approach to fault detection and identification is introduced. The general idea is to describe the state of a system subject to failures or parameter changes as a hybrid dynamic state; this concept is explained in section 7.2.

The mainstream and most natural approach for hybrid estimation is one based on the use of multiple models. The operating range of the system is divided into several system modes, yielding a set of equations to describe all relevant system behavior. A bank of filters is run in parallel, each matching the current system mode to a particular system mode. The bank of filters may consist of any single-model based filter; the Kalman filter is an obvious choice. The probability is calculated that each mode is in effect; these probabilities are then used as weighting factors for the filter's state estimates to calculate an overall state estimate. The system behavior may change if a fault occurs, which may result in updated weighting factors for the system modes. These weighting factors can therefore be used for FDI purposes: if the probability of a fault mode exceeds the probability of the nominal mode to a certain extent, it can be concluded that this fault mode is in effect.

Several multiple-model (MM) algorithms have been presented in literature, which have different procedures for re-initialization of the filter banks, different switching schemes between modes and different ways of estimate fusion: how to obtain an overall state estimate as a weighted combination of all estimates from the filter bank. Efe & Atherton (1997) successfully applied the Interacting Multiple-Model (IMM) algorithm of Bar-Shalom & Fortmann (1988) to the industrial actuator benchmark for fault detection and isolation, presented by Blanke et al. (1995). Zhang & Rong Li (1998) showed that the IMM approach has a superior performance for fault detection purposes, when compared to other multiple-model (MM) algorithms such as the Generalized Pseudo-Bayesian approaches described in Maybeck & Stevens (1999) and Griffin, Jr. & Maybeck (1997). The IMM approach has become a standard method for abrupt change detection: it was applied successfully for FDI in combination with controller reconfiguration (Kanev & Verhaegen, 2000) and applied to target tracking problems (Gustafsson, 2000).

In this chapter, it is investigated whether the IMM algorithm can be used to solve the two FDI problems described in chapter 4: identification of the unknown blade pitch actuator gain and the unknown blade pitch delay in a wind turbine system.

7.2 Hybrid systems

As described by Rong Li (1996), there are two basic types of estimation, classified in terms of the quantity to be estimated: parameter estimation and state/signal estimation. A parameter usually refers to a constant (or slowly-varying) quantity while a state or signal refers to a rapidly-varying quantity.

When estimation has to be done in the presence of structural uncertainty, when there is an unknown structure or a random structural change, it is often possible to define a discrete set of model descriptions to describe different behavior modes of the system. In this case, there is a combined decision and estimation problem: a decision, which is a selection among discrete alternatives, has to be made to determine the model currently in effect; after that, the states of the system can be estimated in a continuous range. Hybrid estimation can be seen as a unification of conventional estimation and decision (Rong Li, 1996). Before a hybrid estimation approach can be applied, the overall system must be represented as a hybrid system.

A stochastic hybrid system is a system which has both a continuous-valued base state and discrete-valued structural/parametric uncertainty (Zhang & Rong Li, 1998). The components of the base state $x(k)$ are the usual state variables; the system mode is a mathematical characterization of a certain behavior pattern or structure of the system. When hybrid estimation is applied to a system subject to structural failures, the operating range of the system is divided into several system modes: one system description for the nominal system behavior, and several linear descriptions for each relevant fault scenario. Hybrid estimation consists of estimating both the base state and the modal state; a jump from the nominal system mode to a particular faulty system mode can be seen as detection of that particular fault.

Zhang & Rong Li (1998) give a general linear description of a stochastic hybrid system, for a hybrid system with a D matrix equal to zero. With a non-zero D-matrix, the following system is obtained:

$$\begin{aligned} x(k+1) &= A(k,m(k+1))x(k) + B(k,m(k+1))u(k) + \\ &\quad T(k,m(k+1))\xi(k,m(k+1)) \end{aligned} \tag{7.1}$$

$$y(k) = C(k,m(k+1))x(k) + D(k,m(k+1))u(k) + \eta(k,m(k)) \tag{7.2}$$

where the system mode sequence is a first-order Markov chain with transition probabilities

$$P\{m_j(k+1)|m_i(k)\} = \pi_{ij}(k) \qquad \forall \quad m_i, m_j \in \mathcal{S} \tag{7.3}$$

and

$$\sum_j \pi_{ij}(k) = 1, \quad i = 1,..,s. \tag{7.4}$$

Here, $x(k) \in \mathbb{R}^{n_x}$ is the base state vector, $y(k) \in \mathbb{R}^{n_y}$ the mode-dependent measurement vector, $u(k) \in \mathbb{R}^{n_u}$ the control input vector; $\xi(k) \in \mathbb{R}^{n_\xi}$ and $\eta(k) \in \mathbb{R}^{n_\eta}$ are independent discrete-time process and measurement noises with means $\bar{\xi}(k)$ and $\bar{\eta}(k)$ and covariance matrices $Q(k)$ and $R(k)$; $m(k)$ is the discrete-valued modal state at time instant k, that denotes the mode in effect during the sample period ending at k. π_{ij} is the transition probability from mode m_i to mode m_j. The set $\mathcal{S} = \{m_1, m_2, ..m_s\}$ is the set of all possible system modes.

The overall state of the hybrid system (7.1)-(7.2) is called the hybrid state: it is defined in a hybrid space $\mathbb{R}^{n_x} \times \mathcal{S}$, the Cartesian product of the continuous-valued base-state space $\mathbb{R}^{n_x}$ and the discrete model set $\mathcal{S}$, the collection of the system's modes.

The hybrid system (7.1)-(7.2) is also known as a jump-linear system (Gustafsson, 2000): it has a linear behavior given the system mode, but it may jump from one system mode to another at a random time. It can be seen from (7.2) that the base state measurements are in general noisy and mode dependent. The mode information is embedded (i.e., not directly measured) in the measurement sequence $y(k)$; it is an indirectly observed Markov chain (Zhang & Rong Li, 1998).

A necessary prerequisite for application of hybrid estimation with a multiple-model method such as the IMM estimator is, that it is possible to design the set of models $\mathcal{S} = \{m_1, ..., m_N\}$, which contains a model description M_j for every jth possible system mode, such that the set of models describes the operating range of the hybrid system to a certain degree of accuracy. Designing this model set is a complex task; there exists no systematic procedure for the choice of S. From now on it is assumed that the model set is given for each particular fault detection and identification problem.

Using this model set S, it is possible to approximate the hybrid system description (7.1)-(7.2) by the following sets of equations, for $j = 1, 2, \ldots, N$:

$$x(k+1) = A_j(k)x(k) + B_j(k)u(k) + T_j(k)\xi_j(k) \quad (7.5)$$

$$y(k) = C_j(k)x(k) + D_j(k)u(k) + \eta_j(k) \quad (7.6)$$

The matrices A_j, B_j, C_j, D_j and T_j may be of different structure for different j; however, it is assumed in this thesis that the state matrices have the same dimensions for all $j = 1, 2, \ldots, N$.

7.3 Interacting multiple-model estimator

The IMM estimator is considered to be one of the most cost-effective approaches to hybrid estimation. It was first introduced by Bar-Shalom & Fortmann (1988), and first applied to fault detection by Efe & Atherton (1997). It is a recursive algorithm; a clear overview of one cycle is presented in Zhang & Rong Li (1998), see table 7.3 for an overview. For implementation purposes, Kanev & Verhaegen (2000) present some practical aspects on model set design and algorithm implementation.

The IMM algorithm consists of four steps, as can be seen in table 7.3:

1. **Mixing of the estimates:** the input to the filter matched to every jth particular mode is obtained by mixing the estimates of all N filters at the previous time instant, assuming that this particular mode is in effect at this time;

2. **Model-conditional filtering:** a Kalman algorithm cycle as described in section 6.1 is run in parallel for all N Kalman filters in the filter bank. A prediction of the next state and covariance is made, and the current state is updated;

3. **Mode probability update:** Based on the model-conditional likelihood functions, the mode probabilities for every mode are updated;

4. **Combination of estimates:** The overall state is calculated as a probabilistically weighted sum of the updated state estimates of all filters.

According to Zhang & Rong Li (1998), the introduction of the IMM estimator proved to be a great advance in MM estimation. When a non-interacting MM approach is used, the single-mode-based filters run in parallel without mutual interaction. Such an approach is quite effective in handling problems with an unknown structure, without structural change. The IMM estimator also performs well in case of abrupt structural changes: it switches from one model to another in a probabilistic manner. The single-model-based filters interact with each other, which accounts for the superior performance.

The algorithm in table 7.3 is slightly modified. As mentioned before, Zhang & Rong Li (1998) consider an LTI system with a zero D-matrix. For the wind turbine system, this is not the case. This results in the term $-D_j(k+1)u(k+1)$ as a correction for the output vector $y(k+1)$ in the calculation of the measurement residual. Note that the residual covariance S_j is a 2 by 2 matrix, so the matrix norm must be used instead of the vector norm, and a matrix exponent must be used in the likelihood function.

Apart from the model set S, the only design parameter of the IMM algorithm is the transition probability matrix π_{ij}. This must be defined such, that the probability that the system stays in the same mode is relatively high. More details on the choice of π_{ij} will be given on the basis of the experimental results in the following sections.

1. **Mixing of the estimates** (for $j = 1, ..., N$):	
predicted mode probability:	$\mu_j(k+1\|k) = \sum_{i=1}^{N} \pi_{ij}\mu_i(k)$
mixing probability:	$\mu_{i\|j} = \pi_{ij}\mu_i(k)/\mu_j(k+1\|k)$
mixing estimate:	$\hat{x}_j^0(k\|k) = \sum_{i=1}^{N} \hat{x}_i(k\|k)\mu_{i\|j}(k)$
mixing covariance:	$P_j^0(k\|k) = \sum_{i=1}^{N} \{P_i(k\|k) + [\hat{x}_j^0(k\|k) - \hat{x}_i(k\|k)][\hat{x}_j^0(k\|k) - \hat{x}_i(k\|k)]^T\}\mu_{i\|j}(k)$
2. **Model-conditional filtering** (for $j = 1, ..., N$):	
predicted state:	$\hat{x}_j(k+1\|k) = A_j(k)\hat{x}_j^0(k\|k) + B_j(k)u(k) + T_j(k)\bar{\xi}_j(k)$
predicted covariance:	$P_j(k+1\|k) = A_j(k)P_j^0(k\|k)A_j(k)^T + T_j(k)Q_j(k)T_j(k)^T$
measurement residual:	$\nu_j(k+1) = y(k+1) - C_j(k+1)\hat{x}_j(k+1\|k) - D_j(k+1)u(k+1) - \bar{\eta}_j(k+1)$
residual covariance:	$S_j(k+1) = C_j(k+1)P_j(k+1\|k)C_j(k+1)^T + R_j(k+1)$
filter gain:	$K_j(k+1) = P_j(k+1\|k)C_j(k+1)^T S_j(k+1)^{-1}$
updated state:	$\hat{x}_j(k+1\|k+1) = \hat{x}_j(k+1\|k) + K_j(k+1)\nu_j(k+1)$
updated covariance:	$P_j(k+1\|k+1) = P_j(k+1\|k) - K_j(k+1)S_j(k+1)K_j(k+1)^T$
3. **Mode probability update** (for $j = 1, ..., N$):	
likelihood function:	$L_j(k+1) = \frac{1}{\sqrt{\|2\pi S_j(k+1)\|}} exp[-\frac{1}{2}\nu_j(k+1)^T S_j(k+1)^{-1}\nu_j(k+1)]$
mode probability:	$\mu_j(k+1) = \frac{\mu_j(k+1\|k)L_j(k+1)}{\sum_i \mu_i(k+1\|k)L_i(k+1)}$
4. **Combination of estimates**:	
overall state estimate:	$\hat{x}(k+1\|k+1) = \sum_{i=1}^{N} \hat{x}_i(k+1\|k+1)\mu_i(k+1)$
overall covariance:	$P(k+1\|k+1) = \sum_{i=1}^{N} \{P_i(k+1\|k+1) + [\hat{x}(k+1\|k+1) - \hat{x}_i(k+1\|k+1)] \times [\hat{x}(k+1\|k+1) - \hat{x}_i(k+1\|k+1)]^T\}\mu_i(k+1)$

Table 7.1: *One cycle of the IMM estimator.*

7.4 IMM estimation of an unknown pitch actuator gain

In this section it is verified whether the IMM algorithm described in section 7.3 can be used to solve the FDI problem defined in section 4.1: estimation of the unknown pitch actuator gain σ. The same fault simulation will be used as in the previous chapter: the pitch actuator gain $\sigma(k)$ changes abruptly from 1 to 2/3 at time instant $k = 500$. Only the MIMO setup is considered: both measurements $y_1 = \ddot{x}_{nd}$ and $y_2 = \Delta\Omega_r$ are used. The IMM algorithm uses a bank of Kalman filters, and it was shown in section 6.1.3 that a discrete-time Kalman filter gives the best results when multiple system outputs are used.

7.4.1 Design of the model set

A set of two models must be defined, that can be used to estimate the unknown pitch actuator gain σ. Consider the state-space representation (6.10)-(6.11) of the turbine system, where the wind speed $v_w(k)$ is included in an augmented state. The parameter of interest for the model set is the unknown pitch actuator gain $\sigma_j(k)$, which is set to 1 for the nominal (fault-free) mode, and to 2/3 for the faulty system mode. Only the pitch actuator gain depends on the mode j; all state matrices are the same for both modes. This yields the model set (7.7)-(7.8).

$$\begin{bmatrix} x(k+1) \\ v_w(k+1) \end{bmatrix} = \begin{bmatrix} A & B_h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} B\sigma_j(k) \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} Q^{1/2} & 0 \\ 0 & Q_{v_w}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{w}(k) \\ \tilde{w}_{v_w}(k) \end{bmatrix} \quad (7.7)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 & D_h \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} D\sigma_j(k) \\ 0 \end{bmatrix} u(k) + R^{1/2}\tilde{v}(k) \quad (7.8)$$

$$\begin{cases} \sigma = 1 & \text{for the nominal mode} \\ \sigma = 2/3 & \text{for the faulty mode} \end{cases}$$

The model set (7.7)-(7.8) can be rewritten as

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}\tilde{x}(k) + \tilde{B}_j u(k) + \tilde{Q}^{1/2}\tilde{w}_{v_w}(k) \\ y(k) &= \tilde{C}\tilde{x}(k) + \tilde{D}_j u(k) + R^{1/2}\tilde{v}(k) \end{aligned}$$

7.4.2 Experimental issues

One cycle of the IMM algorithm was implemented in the m-file *imm_mimo.m*. The m-file *mmod_sigma.m* contains the rest of the implementation of the IMM algorithm.

In analogy with Kanev & Verhaegen (2000), the overall estimate of the pitch actuator gain $\sigma(k)$ can be determined as a linear combination of the modal probabilities $\mu_j(k)$ and the associated actuator gains σ_j:

$$\sigma_{est}(k) = \sum_{j=1}^{N} \sigma_j \mu_j(k) \quad (7.9)$$

The step size of the fault estimate is determined by the transition probability matrix π_{ij}. It turned out to be important that the diagonal elements of π_{ij} are equal and the anti-diagonal elements are equal as well, otherwise one mode may be favored over the other based on this transition probability. Therefore π_{ij} is always chosen symmetrical, as a function of $\delta = 1 - \pi_{11}$, a small non-zero value:

$$\pi_{ij} = \begin{bmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{bmatrix}$$

It was found, that the noise covariance Q on the turbine states can be set to zero. In analogy with section 6.1.3, the covariance R of the measurement noise must be chosen non-zero, otherwise a problem might occur in the IMM algorithm shown in table 7.3: the residual covariance $S_j(k)$ may become singular, so that the likelihood function cannot be calculated.

First IMM experiment: no measurement noise

In the first experiment, no measurement noise is present in the system outputs. Similar to the discrete-time Kalman filter in section 6.1.3, the algorithm has two tuning parameters: the random walk size Q_{v_w} of the wind estimate, and the value of π_{11}, the diagonal element of the transition probability matrix π_{ij}. As in section 6.1.3, the measurement noise covariance matrix is set to $R = 0.01 \cdot R_{smn}$ (6.12), and the covariance of the wind estimate to $Q_{v_w} = 5$. Figure 7.1 compares the modal probabilities for several values of π_{11}, the diagonal element of the transition probability matrix.

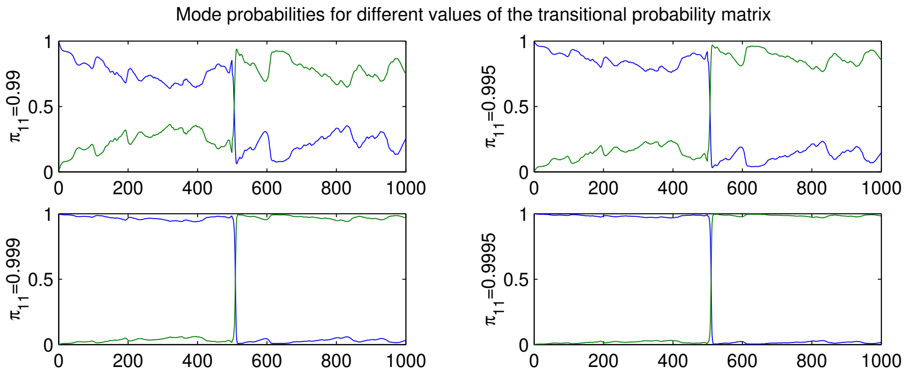

Figure 7.1: *First IMM experiment: modal probabilities for different values of the transition probability matrix π_{ij}; abrupt change in σ at $t = 500$.*

Clearly, the value of π_{11} must be chosen very close to 1. For example, when $\pi_{11} = 0.99$ the IMM algorithm cannot distinguish very well between the models, which deteriorates the estimate: it results in a relatively high modal probability for the model that is not in effect. However, as $\pi_{11} \uparrow 1$, the probability increases that the same system mode is estimated in the next time step, which results in a slower response of the estimated $\sigma_{est}(k)$ to a change in the simulated $\sigma(k)$. In extreme cases, a low value of π_{11} shows a 50/50 choice between the system modes, and a value of π_{11} very close to one yields a 100% choice for one system modes. But, if π_{11} is chosen closer to one, the switch time between the model sets increases.

The choice of the value of π_{11} is therefore a trade-off between estimation accuracy and estimation speed: π_{11} plays the same role as the random walk size Ψ of the augmented state $\sigma(k)$ in the discrete-time Kalman filter implementation in section 6.1.3.

Based on the results in figure 7.1, a value of $\pi_{11} = 0.999$ was selected. Figure 7.2 shows the modal probabilities and figure 7.3 shows the estimate of the wind speed $v_w(k)$; the VAF between real and estimated wind speed is equal to 99.5%. The estimated pitch actuator $\sigma_{est}(k)$ was calculated as (7.9), and the result is shown in figure 7.4; the VAF between real and estimated pitch actuator gain was found to be 95.6%. The numerical derivative $\dot{\sigma}(k)$ was calculated according to (6.16); the result is shown in figure 7.5. The steepest descent is found at $k = 510$, 10 time steps after the simulated change in σ.

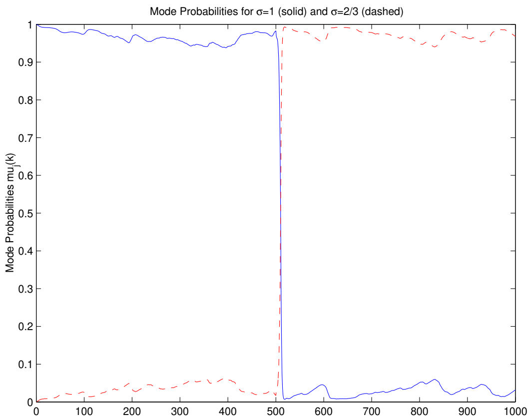

Figure 7.2: *First IMM experiment: modal probabilities $\mu_j(k)$, when $\pi_{11} = 0.999$ and $Q_{v_w} = 5$. No measurement noise is present in the outputs, and σ changes abruptly at $k = 500$.*

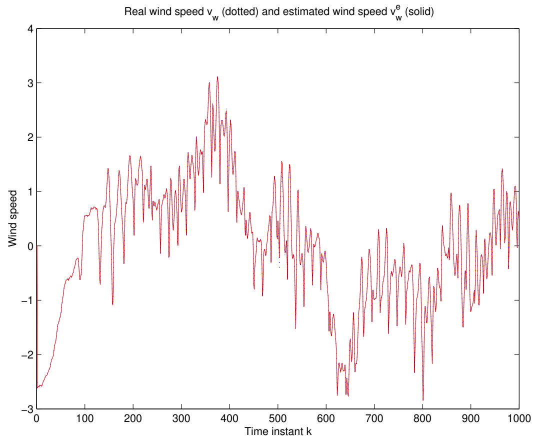

Figure 7.3: *First IMM experiment: wind speed estimate $v_w(k)$, when $\pi_{11} = 0.999$ and $Q_{v_w} = 5$. No measurement noise is present in the outputs, and σ changes abruptly at $k = 500$.*

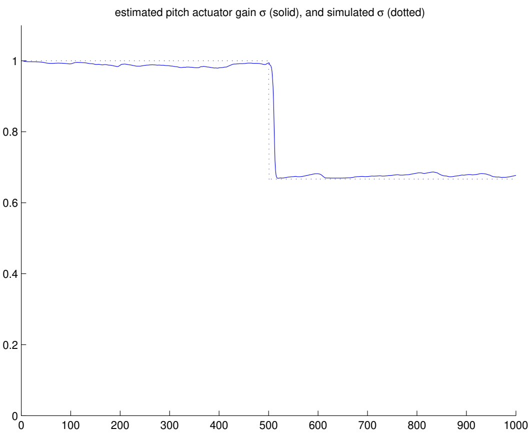

Figure 7.4: *First IMM experiment: estimated pitch actuator gain $\sigma_{est}(k)$, calculated according to (7.9) as a linear combination of the modal probabilities shown in figure 7.2.*

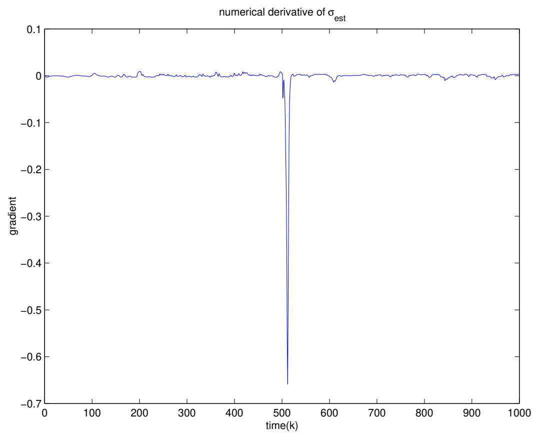

Figure 7.5: *First IMM experiment: numerical derivative $\dot{\sigma}_{est}(k)$ of the estimated pitch actuator gain $\sigma_{est}(k)$ that is shown in figure 7.4, calculated as in (6.16).*

The estimated turbine states are shown in figure 7.6; they are a little less accurate than with the regular discrete-time Kalman filter of section 6.1.3, since the overall state estimate is produced as a weighted sum of the estimates for each model in the filter bank. Therefore, at every time instant there is a contribution from the model that is not in effect at the time. However, the VAF values between real and estimated states are still almost perfect (in %):

$$VAF(x_e, x_r) = \left[\begin{array}{cccccccccc} 100.0 & 100.0 & 98.8 & 100.0 & 100.0 & 99.9 & 100.0 & 99.3 & 99.2 & 100.0 \end{array}\right]^T$$

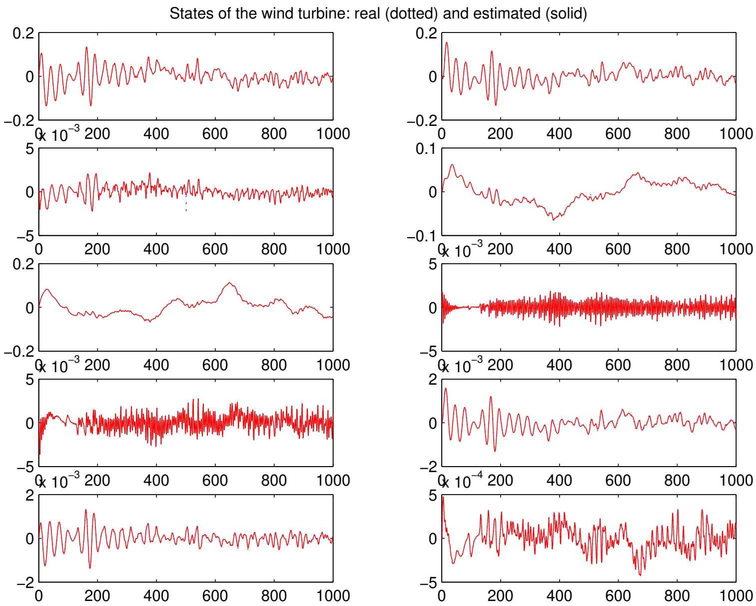

Figure 7.6: *First IMM experiment: the real turbine states* x_r *and the estimated turbine states* x_e*.*

Second IMM experiment: standard measurement noise

In this second experiment, standard measurement noise is present in the system outputs, as described in section 3.6; the corresponding measurement noise covariance matrix $R = R_{smn}$ (6.12) was selected.

The measurement noise in the system outputs has a similar effect as in section 6.1.3: the quality of the estimates deteriorates slightly, and to obtain reliable results it is required that a smaller value of Q_{v_w} is selected; with $Q_{v_w} = 0.1$ and π_{11} again set to 0.999, the modal probabilities shown in figure 7.7 were obtained. The wind estimate $v_w(k)$ is shown in figure 7.8, with a VAF of 98.2% between real and estimated wind speed. The estimate of the pitch actuator gain σ is shown in figure 7.9, with a VAF of 96.2% between the estimated and simulated $\sigma(k)$. The numerical derivative $\dot{\sigma}(k)$ is shown in figure 7.10. It can still be used for change detection: the steepest descent is found at $k = 520$, 20 time steps after the simulated change in σ. However, it is clear that this method is less reliable: the derivative is more noisy, because the estimate of σ is more noisy.

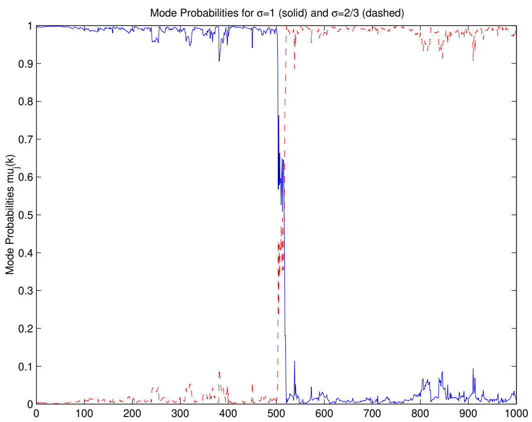

Figure 7.7: *Second IMM experiment: modal probabilities* $\mu_j(k)$*, when* $\pi_{11} = 0.999$ *and* $Q_{v_w}) = 0.1$*. Standard measurement noise was present in the system outputs, and* σ *changes abruptly at* $k = 500$*.*

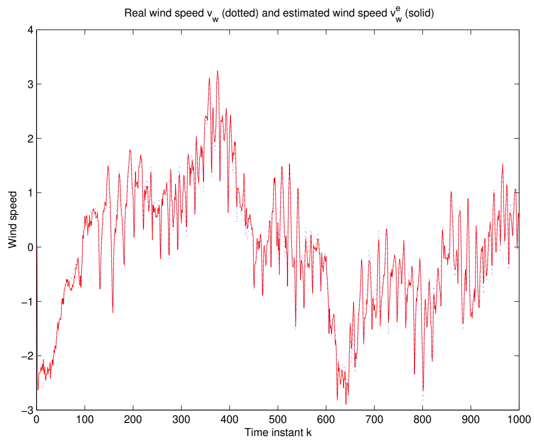

Figure 7.8: *Second IMM experiment: wind speed estimate* $v_w(k)$*, when* $\pi_{11} = 0.999$ *and* $Q_{v_w}) = 0.1$*. Standard measurement noise was present in the system outputs, and* σ *changes abruptly at* $k = 500$*.*

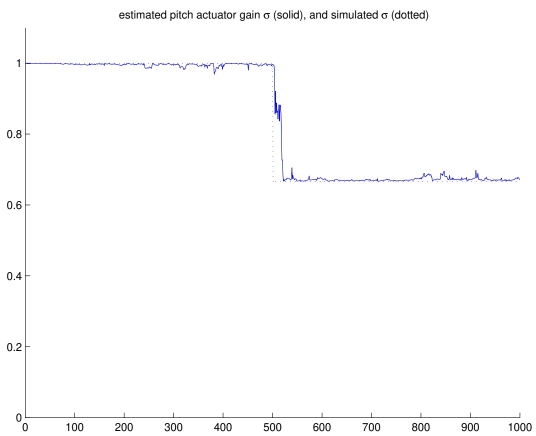

Figure 7.9: *Second IMM experiment: estimated pitch actuator gain* $\sigma_{est}(k)$*, calculated according to (7.9) as a linear combination of the modal probabilities shown in figure 7.7.*

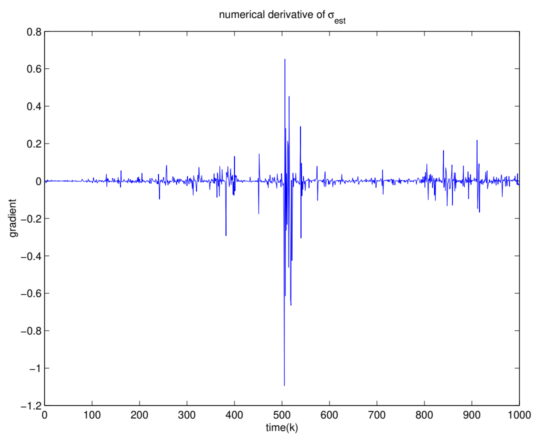

Figure 7.10: *Second IMM experiment: numerical derivative* $\dot{\sigma}_{est}(k)$ *of the estimated pitch actuator gain* $\sigma_{est}(k)$ *that is shown in figure 7.9, calculated as in (6.16).*

The state estimates (not shown, as they are similar to figure 7.6) are not as good as in the simulation without measurement noise. However, the VAF values between real and estimated states are still very high (in %):

$$VAF(x_e, x_r) = \begin{bmatrix} 99.2 & 99.4 & 99.6 & 99.9 & 99.6 & 95.9 & 97.3 & 98.6 & 99.5 & 98.3 \end{bmatrix}^T$$

From simulations, an estimate was made of the required computation time T_{cpu} for one time step of the IMM algorithm. An average of 100 cycles yields $T_{cpu} = 1.7 \cdot 10^{-3}$s. As a comparison, one cycle of the regular discrete-time Kalman filter algorithm described in section 6.1.3 has a computation time of $T_{cpu} = 5.4 \cdot 10^{-4}$s. The IMM algorithm is about three times slower, which is logical: two Kalman filters are run in parallel instead of just one, which means the algorithm should be at least twice as slow. As shown in table 7.3, each cycle step also includes mixing of the estimates, as well as a mode probability update and a combination of the estimates, which accounts for the rest of the extra computation time.

Analysis of the results

It was demonstrated in this section that the IMM estimator can be used to estimate the states of the turbine system, the wind speed $v_w(k)$ and the pitch actuator gain $\sigma(k)$. The IMM algorithm requires a suitable model set to be defined for each FDI problem. The wind speed $v_w(k)$ was included in an augmented system state, as described in section 6.1.2. A set of two models was defined: the nominal model has a value of $\sigma = 1$, and the faulty model has a value of $\sigma = 2/3$. An abrupt change in the pitch actuator gain from $\sigma = 1$ to $\sigma = 2/3$ was simulated.

To evaluate the performance of the IMM algorithm, it is compared with the regular discrete-time Kalman filter in section 6.1.3, that produces estimates for the turbine states, wind speed and pitch actuator gain as well. Just as the Kalman filter implementation for this FDI problem, the IMM algorithm has two tuning parameters: the random step Q_{v_w} of the wind estimate $v_w(k)$, and the diagonal value π_{11} of the transition probability matrix.

A rapid switch in the mode probabilities occurs at the time of change when suitable values of these tuning parameters are selected, so that the estimated pitch actuator gain follows the change in the simulated pitch actuator gain in a reliable way. Even in the presence of measurement noise, the wind speed and the turbine states were estimated very well. However, these estimates were a little less accurate than those obtained with the regular discrete-time Kalman filter algorithm.

The explanation of the lower performance lies in the choice of the transition probability matrix π_{ij}. If the value of π_{11} is chosen very close to 1, the response of the estimate to a change becomes very slow. A lower value of π_{11} results in faster change detection, but less accurate estimates. This trade-off between estimation accuracy and estimation speed has the result that π_{11} must be chosen in such a way that there is enough sensitivity to change detection. Unfortunately, this means that at every time instant there is a non-zero contribution of the model that is not in effect at that time, which accounts for the inaccuracy.

The IMM algorithm also requires more computation time: the algorithm is about three times slower than the regular discrete-time Kalman filter algorithm. This is not a practical limitation: both algorithms are certainly fast enough for real-time application.

7.5 IMM estimation of an unknown pitch actuator delay

In this section, it is verified whether the IMM estimator algorithm described in section 7.3 can be used to solve the FDI problem defined in section 4.2: estimation of the unknown pitch actuator delay. Only the MIMO setup is considered: the system outputs $y_1 = \ddot{x}_{nd}$ and $y_2 = \Delta\Omega_r$ are used in the IMM algorithm.

7.5.1 Design of the model set

Recall the transfer function (3.6) from the desired pitch angle θ_p (the controller output) to the delayed pitch angle θ_p^d:

$$\frac{\theta_p^d}{\theta_p} = \frac{(s-(p_1+j\cdot p_2))\cdot(s-(p_1-j\cdot p_2))}{(s-(-p_1+j\cdot p_2))\cdot(s-(-p_1-j\cdot p_2))}$$

The parameter pair $(p_1,\, p_2)$ in this transfer function determines the pitch actuator delay. Table 4.1 shows different values of $(p_1,\, p_2)$ and the corresponding delay times. A fault-free system has a delay of 0.2s; a deviation from this value can be seen as a fault in the system.

In section 4.2, the wind turbine system G_0 was divided into a block d_0 that contains the delay (3.6), and a block G_0^* that contains the rest of the wind turbine system. For a system with a variable delay, the transfer function from the turbine system inputs $u_1(k) = v_w(k)$ and $\bar{u}_2 = \Delta\theta_p(k)$ to the turbine system outputs $y_1 = \ddot{x}_{nd}$ and $y_2 = \Delta\Omega_r$ was derived; the discrete-time state-space representation of this transfer function is shown in (4.19)-(4.20). When the controller state $\bar{x}(k)$ is omitted, it reduces to (7.10)-(7.11):

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_2\tilde{C} \\ 0 & \tilde{A} \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} B_1 & B_2\tilde{D} \\ 0 & \tilde{B} \end{bmatrix} \begin{bmatrix} u_1(k) \\ \bar{u}_2(k) \end{bmatrix} \tag{7.10}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_1 & D_{12}\tilde{C} \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12}\tilde{D} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ \bar{u}_2(k) \end{bmatrix} \tag{7.11}$$

where $[\tilde{A},\ \tilde{B},\ \tilde{C},\ \tilde{D}]$ denote the state matrices of the delay block d_0. It was shown in section 4.2, that the matrices $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ have a different value for a different pair $(p_1,\, p_2)$, and that $\tilde{D} = 1$ for all pairs $(p_1,\, p_2)$.

To design the model set, a different delay is associated with a different system mode j. It follows from (7.10)-(7.11) that the turbine matrices A_j, B_j and C_j are mode dependent and that and D_j is the same for every mode j. Again, the wind speed $v_w(k)$ is included in an augmented state system state; this yields a model set similar to (7.7)-(7.8):

$$\begin{bmatrix} x(k+1) \\ v_w(k+1) \end{bmatrix} = \begin{bmatrix} A_j & B_{h,j} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} B_j \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} Q^{1/2} & 0 \\ 0 & Q_{v_w}^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{w}(k) \\ \tilde{w}_{v_w}(k) \end{bmatrix} \tag{7.12}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} C_{1,j} & D_h \\ C_{2,j} & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v_w(k) \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} u(k) + R^{1/2}\tilde{v}(k) \tag{7.13}$$

The model set (7.12)-(7.13) is denoted shortly as (7.14)-(7.15):

$$\tilde{x}(k+1) = \tilde{A}_j\tilde{x}(k) + \tilde{B}_j u(k) + \tilde{Q}^{1/2}\tilde{w}_{v_w}(k) \tag{7.14}$$

$$y(k) = \tilde{C}_j\tilde{x}(k) + \tilde{D}u(k) + R^{1/2}\tilde{v}(k) \tag{7.15}$$

As in the previous FDI problem in section 7.4, a model set with two modes is defined:

$$\begin{cases} \text{delay}=0.2\text{s} & \text{for the nominal mode} \\ \text{delay}=0.4\text{s} & \text{for the faulty mode} \end{cases}$$

7.5.2 Experimental issues

The wind turbine simulation model *versie2.mdl* was used to obtain the continuous-time state matrices of the wind turbine for different values of the delay parameters (p_1, p_2). The balanced realizations of these state matrices were calculated, and discretized with a sample time of 0.1s. For delay times of 0.2s, 0.3s and 0.4s, these system representations were saved as *turbmat_d02.mat*, *turbmat_d03.mat* and *turbmat_d04.mat*, respectively. The file *mmod_delay.m* contains the implementation of the IMM algorithm for detection of the unknown pitch actuator delay; again, the function *imm_mimo.m* is used to compute each cycle of the IMM algorithm.

It must be noted that the delay affects the turbine behavior in a nonlinear way. Therefore, the analogy with (7.9) cannot be made: it is not straightforward to define the estimated delay as a linear combination of the delay times associated with the different modes in the model set, with the modal probabilities as weighting factors. The model set consists of turbine models with a pitch delay of 0.2s and 0.4s. An initial model probability of 1 is set for the nominal system with a pitch delay of 0.2s. Batch-wise simulation experiments are used to test the IMM algorithm, since they proved to give much insight in the problem. Three simulation setups are considered: turbine systems with a pitch delay of 0.2s, 0.3s and 0.4s are simulated.

Experiments without measurement noise

For now, no measurement noise is present in the system outputs. As described in section 7.4, the IMM algorithm requires a non-zero covariance matrix R of the measurement noise; it was set to $R = 0.01 \cdot R_{smn}$ (6.12), a value that corresponds to 10% of standard measurement conditions. Again, the random walk size Q_{v_w} of the wind estimate and the value of the diagonal element π_{11} in the transition probability matrix π_{ij} are the tuning parameters of the IMM algorithm.

In the first experiment, a wind turbine with a pitch delay of 0.2s is simulated, the delay value associated with the nominal system mode. The tuning parameters were set to $Q_{v_w} = 5$ and $\pi_{11} = 0.997$. This yields the modal probabilities shown in figure 7.11. Clearly, the nominal mode is identified by the IMM algorithm: the modal probability that corresponds to the nominal turbine model is larger than 0.93 for all k. The estimate of the wind speed is shown in figure 7.12, with a VAF between real and estimated wind speed of 99.5%. The turbine states were estimated very accurately: the result is very similar to figure 7.6, therefore no plot is shown here. The following VAF values between real and estimated states were found (in %):

$$VAF(x_e, x_r) = \begin{bmatrix} 100.0 & 100.0 & 99.5 & 100.0 & 100.0 & 99.9 & 99.9 & 100.0 & 100.0 & 100.0 \end{bmatrix}^T$$

In the second experiment, a wind turbine with a faulty pitch delay of 0.4s is simulated, the delay value associated with the faulty system mode. Again using $Q_{v_w} = 5$ and $\pi_{11} = 0.997$, the modal probabilities shown in figure 7.13 were found. Clearly, the IMM algorithm detects the faulty pitch delay: the modal probabilities corresponding to the faulty mode with a delay

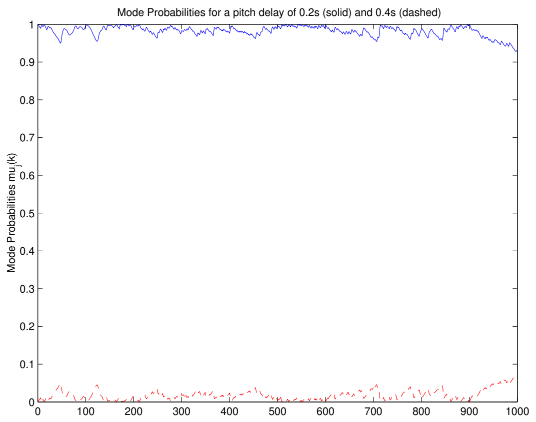

Figure 7.11: *First IMM experiment to detect the pitch delay: modal probabilities* $\mu_j(k)$*, when* $\pi_{11} = 0.997$ *and* $Q_{v_w} = 5$*. No measurement noise is present in the outputs; a pitch delay of* $0.2s$ *was simulated.*

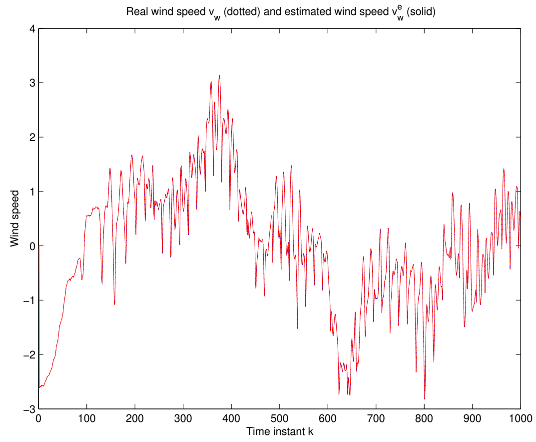

Figure 7.12: *First IMM experiment to detect the pitch delay: wind speed estimate* $v_w(k)$*, when* $\pi_{11} = 0.997$ *and* $Q_{v_w} = 5$*. No measurement noise is present in the outputs; a pitch delay of* $0.2s$ *was simulated.*

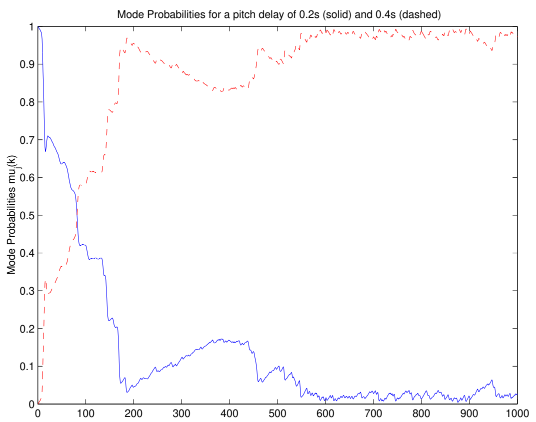

Figure 7.13: *Second IMM experiment to detect the pitch delay: modal probabilities* $\mu_j(k)$*, when* $\pi_{11} = 0.997$ *and* $Q_{v_w} = 5$*. No measurement noise is present in the outputs; a pitch delay of* $0.4s$ *was simulated.*

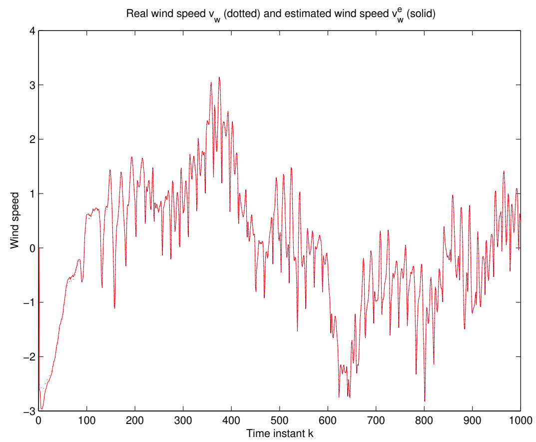

Figure 7.14: *Second IMM experiment to detect the pitch delay: wind speed estimate* $v_w(k)$*, when* $\pi_{11} = 0.997$ *and* $Q_{v_w} = 5$*. No measurement noise is present in the outputs; a pitch delay of* $0.4s$ *was simulated.*

of of 0.4s start to increase at $k = 1$. A clear identification of the delay is obtained for $k > 600$, since the modal probability corresponding to the faulty turbine model is larger than 0.93 for $k > 600$. The estimate of the wind speed $v_w(k)$ is shown in figure 7.14; the VAF between real and estimated wind speed was found to be 99.4%. The state estimates are shown in figure 7.15. For low values of k, there is a mismatch between the simulated and estimated turbine states, which results in lower VAF values (in %):

$$VAF(x_e, x_r) = \left[\begin{array}{cccccccccc} 99.0 & 98.7 & 93.9 & 99.6 & 100.0 & 80.8 & 68.3 & 18.0 & 36.8 & 100.0 \end{array}\right]^T$$

The IMM algorithm calculates the state estimates as a weighted sum of the states of the two turbine system modes in the model set. For low values of k, there is a large contribution from the nominal model with a delay of 0.2s. Therefore, the overall state estimates do not match the states of the simulated model, which results in the lower VAF values. For $k > 600$, the modal probability for the faulty turbine model with a delay of 0.4s has a value larger 0.93, so the overall state estimates are a much better approximation of the real state estimates. In fact, for $k > 600$ the quality of the state estimates is comparable to the state estimates in the first experiment.

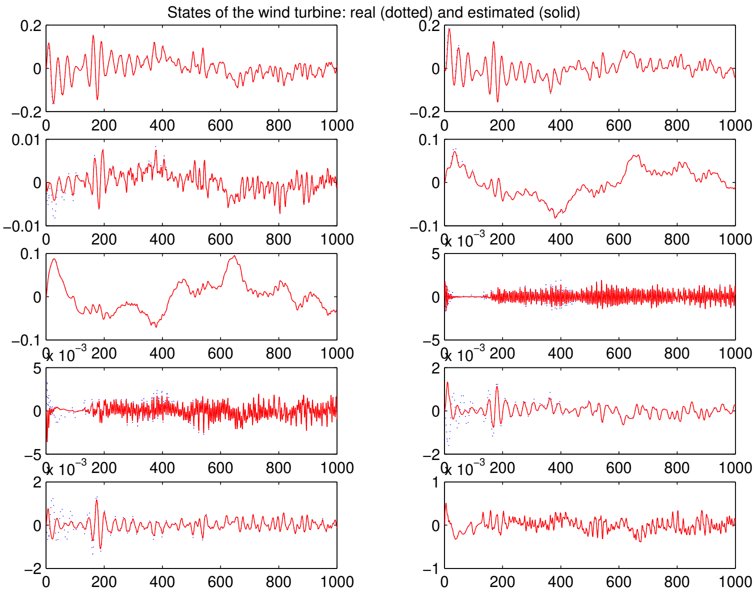

Figure 7.15: *Second IMM experiment to detect a pitch delay: the real turbine states x_r and the estimated turbine states x_e.*

In the third experiment, a wind turbine is simulated with a faulty pitch delay of 0.3s. The model set still consists of a nominal model with a delay of 0.2s and a faulty model with a delay of 0.4s. Again using $Q_{v_w} = 5$ and $\pi_{11} = 0.997$, the modal probabilities were found that are shown in figure 7.16. For $k < 400$, not much happens; this was the case for various values of the tuning parameters. After that, the modal probabilities change: almost equal probabilities are found for delay values of 0.2s and 0.4s, though 0.4s is favored a little. Apparently, the dynamics of a wind turbine model with a delay of 0.3s is slightly better approximated by a model with a delay of 0.4s than by a model with a delay of 0.2s. The estimate of $v_w(k)$ (not shown) is still good; the VAF between real and estimated wind speed is 99.5%. The state estimates (not shown) are not so good, especially for the states with a low amplitude, as can be seen from the VAF values (in %) between real and estimated states:

$$VAF(x_e, x_r) = \begin{bmatrix} 98.5 & 97.9 & 83.1 & 98.9 & 100.0 & -97.7 & -103.2 & -233.3 & -237.3 & 100.0 \end{bmatrix}^T$$

This is an understandable result. The IMM algorithm reconstructs the state estimates of the simulated wind turbine as a weighted combination of the state estimates from Kalman filter bank. Since the delay affects in the turbine behavior in a nonlinear way, it is simply not possible to reconstruct all states of the simulated model as a linear combination of the states of the filter bank models, since the model set only contains models with a delay of 0.2s and 0.4s. The estimates of the states with a low amplitude have the lowest quality, but these states don't contribute much to the system output. Therefore, the IMM algorithm still yields a good wind estimate and indicates that the simulated delay lies between 0.2s and 0.4s. For a discrete system with a sample time of 0.1s, this is an easy choice: 0.3s is the only possible value for the delay. However, it is clear from figure 7.16 that this result is not very reliable: no detection is obtained during the first 400 time instants.

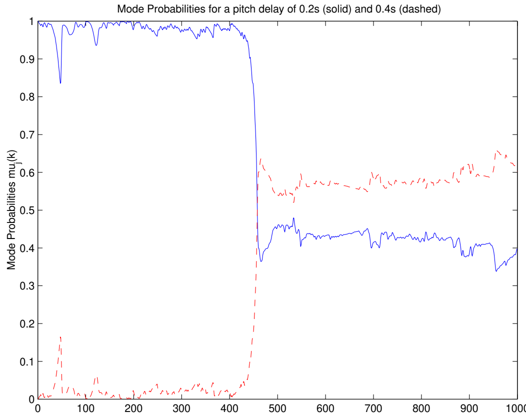

Figure 7.16: *Third IMM experiment to detect the pitch delay: modal probabilities* $\mu_j(k)$, *when* $\pi_{11} = 0.997$ *and* $Q_{v_w} = 5$. *No measurement noise is present in the outputs; a pitch delay of* 0.3s *was simulated.*

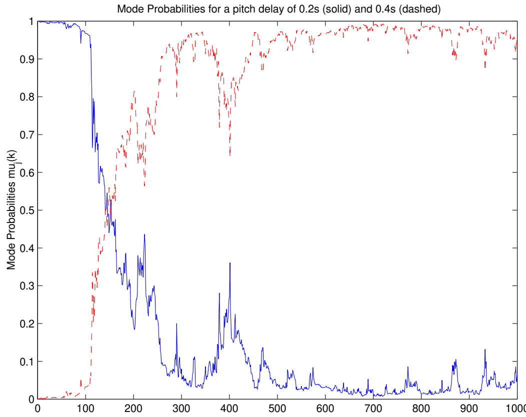

Figure 7.17: *Fourth IMM experiment to detect the pitch delay: a pitch delay of* 0.4s *was simulated (as in figure 7.13, with standard measurement present in the outputs. This figure shows the modal probabilities* $\mu_j(k)$, *when* $\pi_{11} = 0.9995$ *and* $Q_{v_w} = 0.5$.

An experiment with measurement noise

To illustrate the effect of measurement noise in the outputs, the second experiment is repeated: a delay of 0.4s is simulated with standard measurement noise in the outputs, as defined in section 3.6; the measurement noise covariance matrix was set to $R = R_{smn}$ (6.12).

The tuning of the parameters Q_{v_w} and π_{11} was found to be analogous to section 6.1.3: the performance of the algorithm deteriorates in the presence of measurement noise, and the freedom of the estimates must be reduced to obtain reliable estimates. Therefore, Q_{v_w} was lowered to 0.5 and π_{11} was increased to 0.9995. The modal probabilities are shown in figure 7.17. Compared with figure 7.13, the modal probabilities for the same simulation without measurement noise, it is clear that the IMM estimator can identify the faulty delay of 0.4s, even though there is more noise in the modal probabilities. The VAF between the real wind speed $v_w(k)$ and the wind estimate (not shown) was found to be 98.2%.

Analysis of the results

It was found, that the IMM estimator can be used to identify the unknown pitch delay, even though the delay has a nonlinear effect on the behavior of the wind turbine. Four batch-wise experiments were used to demonstrate this. The wind speed was included in an augmented system state, and the model set consisted of the nominal model (the initial model, with a delay of 0.2s), and a faulty model (with a delay of 0.4s).

In the first experiment, a pitch delay of 0.2s was simulated without measurement noise present in the system outputs. The pitch delay was clearly identified, and the state and wind estimates were found to be very reliable.

In the second experiment, a pitch delay of 0.4s was simulated without measurement noise present in the system outputs. The delay of 0.4s was detected and accurately identified after a transition time of the modal probabilities. Some of the estimated turbine states were not so accurate during the transition period. However, after the transition period, the state and wind estimates were just as accurate as in the first experiment.

In the third experiment, a pitch delay of 0.3s was simulated, again without measurement noise present in the system outputs. The state estimates were not reliable: due to the nonlinear effect of the delay on the turbine behavior, it is simply not possible to reconstruct all states of the wind turbine with a pitch delay of 0.3s as a linear combination of the state estimates of the model set, that contains models of wind turbines with delay times of 0.2s and 0.4s. The FDI performance of this third experiment was not very good: for the first 400 time instants, the modal probabilities stayed the almost the same. For $k > 400$, the modal probabilities indicated a delay time between 0.2s and 0.4s. Since the sample time of the system is 0.1s, this is equivalent to identification of a delay time of 0.3s.

Combining this result with the quick fault detection and clear fault identification in the second experiment, it can be concluded that it is a good idea to include the model for 0.3s as a third model in the bank of Kalman filters, if it is desirable to estimate this particular value of the pitch delay time.

The fourth experiment demonstrated that the IMM estimator can be used to detect a faulty pitch delay when standard measurement noise is present in the system outputs. As in the previous FDI problem, the measurement noise deteriorates the quality of the estimates. The tuning parameters of the algorithm had to be tuned so that less freedom is allowed on the estimates; with a lower value of Q_{v_w} and a higher value of π_{11}, the faulty pitch delay could be detected and identified in a reliable way.

7.6 Conclusions

In this chapter, it was investigated whether multiple-model (MM) estimation, a special class of observer-based estimation, can be applied for FDI purposes in a closed-loop HAWT system. Only the Interacting Multiple-Model (IMM) algorithm was considered, as it has been demonstrated in literature that this algorithm has a superior performance for FDI problems.

For each FDI problem, a model set must be designed that contains models to describe different behavior modes of the actual system, e.g. a healthy and a faulty description. A bank of Kalman filters is then designed that compares the current system behavior to the behavior predicted by each system mode, yielding a probability that each mode is in effect. The overall system state is then reconstructed as a linear combination of the different behavior modes, using the modal probabilities as weighting factors.

It was demonstrated in this chapter that the IMM estimator can be used to solve both FDI problems defined in chapter 4: identification of the unknown blade pitch actuator gain and the unknown blade pitch delay in a wind turbine system. The wind speed $v_w(k)$ was included in an augmented system state, so it could be estimated as well.

The IMM algorithm was successfully implemented to detect a change in the pitch actuator gain and to estimate a reliable value. The algorithm has two tuning parameters: the random walk size of the wind estimate and the diagonal value π_{11} of the transition probability matrix determine the change rate of the estimates. When measurement noise is present in the system outputs, the estimate quality deteriorates. In that case, the change rate must be lowered to reduce the influence of the noise on the estimates. This behavior is similar to that of the discrete Kalman filter in section 6.1.3, but the performance of the IMM algorithm is lower: the estimate quality is a little lower, because of the non-zero contribution of the model that is not in effect at every time instant. Furthermore, the computation time is about three times larger.

The IMM algorithm has one large advantage over the discrete-time Kalman filter algorithm: it is applicable to a much wider range of FDI scenarios. While a regular discrete-time Kalman filter can only be used to estimate a scalar parameter or variable, the IMM estimator can distinguish between systems with completely different state matrices. For each FDI scenario, it is only required that a suitable model set is defined.

It was shown that the IMM algorithm can clearly detect a deviation of the pitch actuator gain from normal operating conditions and yield a reliable FDI scheme, provided that the simulated value of the delay matches a delay value that is included in the model set. Again, the presence of measurement noise deteriorates the quality of the estimates, and the change rate of the estimates must be lowered to reduce the influence of the noise on the estimates.

It was demonstrated that it is possible to detect a delay that is not included in the model set. However, this result was found to be unreliable, as the delay affects the turbine behavior in a nonlinear way. The IMM algorithm reconstructs the overall state as a linear combination of the states estimated by the bank of Kalman filters, and it is simply not possible to reconstruct the state of a turbine with a certain delay value as a weighted combination of the states of turbines with different delay values. It is therefore better to design a model set, that contain a turbine model for each value of interest of the pitch delay.

Chapter 8

Conclusions and recommendations

8.1 Conclusions

In this thesis, time-domain model-based fault detection and identification (FDI) methods have been applied to the field of wind engineering; a literature survey showed no prior applications. The aim of this research was to design an algorithm that detects and identifies faulty process parameters in a wind turbine under operating conditions. Such a scheme is particularly important for an offshore wind turbine: if the condition of the turbine components can be monitored, it is possible to reduce the maintenance costs and increase the annual availability and energy yield of the wind turbine.

A realistic simulation model of a Horizontal Axis Wind Turbine (HAWT) under operating conditions was designed, which was used to study two classes of model-based FDI methods: the system identification approach, and observer-based approaches using the Kalman filter. A brief overview of what has been discussed in the thesis follows.

8.1.1 The simulation setup

The physical behavior of a prototype of an offshore HAWT using pitch-to-vane control was linearized in the vicinity of the most important operating point, in which the wind turbine reaches full load operation. It was used as a simulation environment under pre-defined operating conditions: the closed-loop system was considered, with the wind speed as an unknown disturbance input, the blade pitch angle as the controlled input, and the rotor rotational frequency and the tower vibration accelerations as outputs, see figure 4.1.

Two FDI scenarios were defined: the identification of an unknown pitch actuator gain and an unknown pitch actuator delay.

8.1.2 FDI using system identification

The system identification approach to FDI problems is not applicable to a wind turbine. Because of the closed-loop operating conditions and the fact that the wind speed is an unknown disturbance input, no reliable system model can be identified by using direct identification. Instead, the system model that is identified from input/output data describes the negative inverse controller.

8.1.3 FDI based on the Kalman filter

To investigate observer-based approaches, knowledge of the nominal HAWT system model was assumed. Three approaches were investigated, all based on the Kalman filter:

- The regular discrete-time Kalman filter,
- the Kalman filter, rewritten as an LMS problem over a moving time window, and
- the IMM estimator, that uses a bank of Kalman filters.

All algorithms yield better results when the algorithm uses multiple system outputs, rather than only the controlled output. The wind speed was included in an augmented state for all methods, so it could be estimated. Since the wind speed is the driving disturbance input of the wind turbine, it proved to be important to estimate its value.

It was found that the Kalman filter can be rewritten during a moving time window, both as a linear and a nonlinear LMS algorithm. However, these methods have little practical relevance. The main shortcoming is the lack of a tuning parameter that restricts the freedom of the fault estimate, and the fact that the algorithms have too much other tuning parameters. This resulted in either noisy or biased estimates. Finally, these algorithms require way too much computation time.

It was shown that the discrete-time Kalman filter and the IMM estimator can be used in reliable FDI schemes, even when measurement noise is present in the system outputs. Both algorithms have two tuning parameters that determine the rate of change of the fault and wind estimates. The choice of the change rates is a trade-off between estimation speed and estimation accuracy. When measurement noise is present in the system outputs, the change rates must be lowered to reduce the influence of the noise on the estimates. Both algorithms have a lower performance in the presence of measurement noise.

The regular discrete-time Kalman filter proved to be the best algorithm to identify the unknown pitch actuator gain, when the actuator gain was included in the augmented system state as well. The IMM algorithm with a filter bank consisting of two models is about three times slower, but both algorithms can be applied in real-time. The IMM algorithm yields estimates of a lower quality, since at each time instant there is a non-zero contribution of the model that is not in effect at the time. However, this is only a slight difference.

The IMM algorithm proved to be the best algorithm to estimate the unknown pitch actuator delay. The reason is that the regular discrete-time Kalman filter cannot be implemented to solve this FDI problem: the delay affects the system behavior in a nonlinear way, which cannot be represented by a single parameter that can be included in an augmented system state. The IMM algorithm can clearly detect a deviation of the pitch actuator gain from normal operating conditions and it can be used in a reliable FDI scheme, provided that the simulated value of the delay matches a delay value in the bank of Kalman filters.

It can be concluded that the discrete-time Kalman filter algorithm has a superior performance when the parameter that has to be estimated affects the system linearly. For such an FDI problem, the IMM algorithm has a slightly lower performance, but this algorithm is more versatile: when a suitable model set can be defined to cover the dynamic range of an FDI problem, it can also be used to identify nonlinear effects.

8.2 Recommendations

This thesis presented two FDI methods based on the Kalman filter, that can be applied to a HAWT simulation model. It should be verified whether these methods can be applied to a real wind turbine system. Some recommendations towards this verification are given below.

8.2.1 Improving the simulation model: a nonlinear model?

There are several options to modify the linear HAWT model so that it better describes a realistic offshore HAWT. It is expected that the introduction of a non-zero hydrodynamic load disturbance will result in a higher excitation of the naying vibrations, so that it becomes important to use this output in the Kalman algorithms. It might then be necessary to implement a secondary control loop that reduces the naying vibrations by controlling the generator torque, in order to make the simulation more realistic.

Including the dynamic inflow effect of blade pitching results in a more realistic description of the control action.

It should be investigated whether it is possible to obtain an accurate linear model that describes the behavior of a real wind turbine under operating conditions. Since a real wind turbine is a complex nonlinear system under various operating conditions, this is not expected to be possible. Therefore, it is a good idea to design a nonlinear HAWT simulation model, that can be used to design more realistic FDI schemes.

8.2.2 Local linear models

It was demonstrated in this thesis that FDI methods can be implemented for a local linear model of the wind turbine system. Based on the divide-and-conquer strategy, Verdult (2002) showed that it is possible to design a set of local linear models to approximate a nonlinear system in the entire operating range. If a model set can be designed for a wind turbine system under nominal as well as faulty operating conditions, this yields a batch of models that together describe all possible behavior modes of the wind turbine.

The IMM algorithm might then be used to switch between these modes to identify the most probable mode at each time. Since it was shown in this thesis that the IMM algorithm is also sensitive to nonlinear changes in the turbine behavior, a suitable model set might yield an FDI scheme that covers several fault scenarios in the entire dynamic range of a wind turbine.

8.2.3 Nonlinear Kalman filter

It was demonstrated in this thesis that FDI problems in a linear HAWT model can be solved by methods based on the linear Kalman filter. It is an interesting parallel to verify whether nonlinear Kalman filter approaches (Boutayeb et al., 1997) can be used for FDI in a nonlinear HAWT.

8.2.4 Controller reconfiguration

In this thesis, two FDI methods were presented that can detect a change in the blade pitch actuator gain and identify a reliable value of this actuator gain. It should be verified whether it is always necessary to shut down the wind turbine when a faulty actuator gain is detected and identified. In some situations, a fault might not be critical to the system. It is then profitable to reconfigure the controller, so that the wind turbine can still yield energy while a maintenance crew is on its way to repair the damage that causes the fault.

As described in Kanev & Verhaegen (2000), controller reconfiguration for a faulty actuator gain is straightforward: a faulty non-zero actuator gain g can be compensated by multiplying the controller gain by g^{-1}, so that the desired control action is reconstructed. It might be possible to implement a controller reconfiguration strategy for other fault scenarios as well.

Appendix A

Linear state-space systems

Several methods are described in literature for the modeling of linear systems, based on transfer function models, parametric models and state-space models. Extensive overviews are given in Ljung (1987) and Verhaegen & Verdult (2001). This appendix gives an overview of discrete-time linear state-space system descriptions, which can be classified according to their time characteristics. A Linear Time Variant (LTV) system is used to describe a system that changes in time, and a Linear Time Invariant (LTI) system is used to describe a system that is constant in time.

A.1 The discrete-time LTV system

According to Ljung (1987) and Verhaegen & Verdult (2001), a discrete Linear Time Variant (LTV) system can be represented as

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) \tag{A.1}$$

$$y(k) = C(k)x(k) + D(k)u(k) + v(k) \tag{A.2}$$

with the state matrices $A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times m}$, $C(k) \in \mathbb{R}^{\ell \times n}$ and $D(k) \in \mathbb{R}^{\ell \times m}$, the state vector $x \in \mathbb{R}^n$, the input vector $u \in \mathbb{R}^m$ and the output vector $y \in \mathbb{R}^l$. $k \in \mathbb{N}^+$ is the time instant. The state noise vector $w \in \mathbb{R}^n$ represents the process noise, and $v \in \mathbb{R}^\ell$ represents the measurement noise. For many applications, these noise sequences can be considered as zero-mean white noise sequences. Often, the matrix $D(k)$ is equal to zero, which means that the system input $u(k)$ does not have an immediate effect on the system output $y(k)$.

A.2 The discrete-time LTI system

When the system matrices $A(k)$, $B(k)$, $C(k)$ and $D(k)$ in (A.1)-(A.2) are not depending on the time instant k, the system dynamics are constant in time. The system can then be described as a Linear Time Invariant (LTI) system:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{A.3}$$

$$y(k) = Cx(k) + Du(k) + v(k) \tag{A.4}$$

Appendix B

Horizontal axis wind turbine

In Hau (2000), various types of wind energy converters are described. In this thesis, only the horizontal axis wind turbine (HAWT) is considered. Though a vertical axis design has the advantage of rotational symmetry, almost all wind turbines for generating electricity are of the HAWT type (EWEA, 1997). A HAWT has a higher aerodynamic efficiency; see Hau (2000) for a calculation. Since a blade pitch mechanism can be used, it is easier to control the power output of a HAWT. Finally, there is a technological lead in the development of the propeller design.

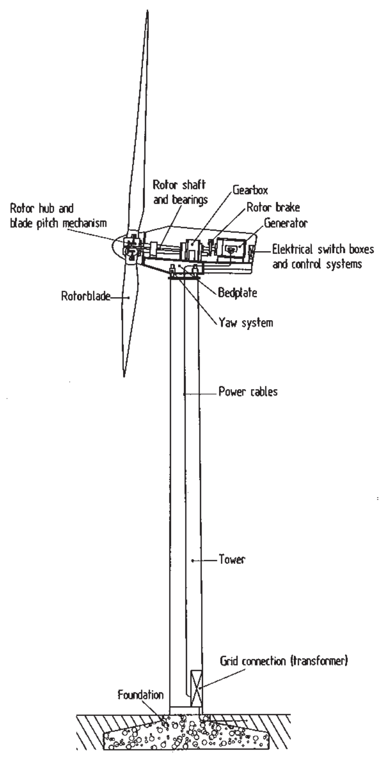

Figure B.1: *A representative HAWT, as described in Hau (2000).*

A representative HAWT with its most important components is shown in figure B.1. These and other components are briefly introduced in this appendix. As far as deemed relevant for this thesis, more information on several components is presented in chapter 3. For further reading, see e.g. Hau (2000) and Van Engelen et al. (2001b).

Simply said, a HAWT consists of a *rotor* on top of a *tower*. The *foundation* ensures that the tower stays upright despite the forces of the wind. The purpose of the rotor, of course, is to convert the linear motion of the wind into rotational energy: the *rotor shaft* is set into motion, which is used to drive a *generator*, where electrical energy is generated.

The *rotor blades* are attached to the rotor shaft at the *rotor hub* with the *blade pitch mechanism*, that is used for control of the rotor speed and the power output: the blades are pitched around their longitudinal axis to decrease or increase the aerodynamic torque on the rotor.

The *rotor bearings* are used to maintain the position and alignment of the rotor shaft. Often, the rotor shaft consists of two parts: the rotor blades set the so-called *slow shaft* into motion. The slow shaft is the input to a *gear box* that transfers the rotational energy to the *fast shaft*. The fast shaft has a higher rotational frequency, which improves the energy yield in the generator. Often there is a *rotor brake* on the fast shaft, to stop the rotation if needed.

All rotating parts of the wind turbine are commonly referred to as the *drive train*. The box on top of the wind turbine is called the *nacelle*; its *bed plate* is connected to the tower. The nacelle and rotor can be turned into the wind direction by the *yaw system*.

The wind turbine *control system* consists of a number of computers which continuously monitor the condition of the wind turbine and collect statistics on its operation. As the name implies, the controller also controls a large number of *electrical switches*, hydraulic pumps, valves and motors within the wind turbine. The electrical current generated by the wind turbine generator is fed through the tower by the *power cables*, to a *grid connection* at the tower basis.

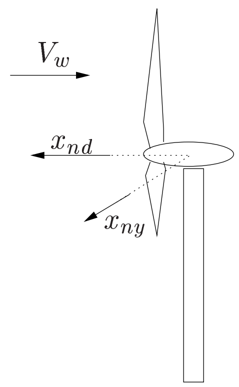

Figure B.2: *The nodding-naying reference frame.*

Tower vibrations are often described in the nodding-naying reference frame, as defined in figure B.2. The nodding direction the direction opposite to the wind direction; the naying direction is the direction perpendicular to the nodding direction, and parallel to the earth's surface.

Appendix C

List of files

The Matlab and Simulink files that have been used in this thesis are listed below. These files are only available for members of the graduation committee and participants in the NOVEM project. For download, see `http://www.stijndonders.nl` (June 2002). Please contact the author for a username and a password.

C.1 The HAWT simulation model and the associated data

- *versie2.mdl* (section 3.5):
  The Simulink model for a HAWT model with measured outputs $\Delta\Omega_r$, $\ddot{x}_{nd}$ and $\ddot{x}_{ny}$, linearized in the operating point where the HAWT reaches full load operation.

- *constants.m* (section 3.4):
  The constants and parameters for the HAWT simulation model of a 6 MW prototype.

- *wind1p2.mat* (section 3.1.1):
  A rotor-effective wind speed input sequence, designed by ECN.

- *turbmat_vib.mat* (section 3.5):
  Continuous-time state matrices of a HAWT with measured outputs $\Delta\Omega_r$ and $\ddot{x}_{nd}$: the fault-free HAWT model, with a pitch actuator gain of 1 and a pitch delay of 0.2s.

- *turbmat_d02.mat*, *turbmat_d03.mat*, *turbmat_d04.mat* (section 7.5):
  The balanced realizations of the discrete-time HAWT system with measured outputs $\Delta\Omega_r$ and $\ddot{x}_{nd}$, with a pitch actuator delay of 0.2s, 0.3s and 0.4s, respectively. Note that *turbmat_d02.mat* contains the discrete-time balanced realization of *turbmat_vib.mat*.

C.2 Functions

- *simulator.m* (section 3.5):
  Function for simulating a discrete-time HAWT system; see help contents for syntax.

- *simulator_abrupt.m* (section 6.1.3):
  Modified version of *simulator.m*, for simulating a change in the pitch actuator gain σ. A change time can be set, as well as the value of σ before and after the change; see help contents for syntax.

- *it_kalm.m* (section 6.1):
Computes the current Kalman gain $K(k)$ and the predicted noise covariance $\hat{P}(k+1|k)$ for one cycle of the discrete-time Kalman algorithm.

- *jnsig.m* (section 6.4):
Creates the vector $J_N(\sigma)$ that represents the cost function of the nonlinear minimization problem (6.56) for a given σ.

- *imm_mimo.m* (chapter 7):
Computes one cycle of the IMM algorithm, as described in table 7.3.

C.3 Algorithms

- *clident.m* (chapter 5):
EIV-MOESP is used for system identification based on a batch of closed-loop measurements of the wind turbine system.

- *mrk_w.m* (section 6.1.2):
The wind speed is estimated with the discrete-time Kalman filter. It is included in an augmented system state.

- *mrk_ws.m* (section 6.1.3):
The wind speed and the unknown pitch actuator gain are estimated with the discrete-time Kalman filter. Both the unknown variables are included in an augmented system state.

- *linkalwin.m* (section 6.3):
The wind speed and the unknown pitch actuator gain are estimated with the Kalman filter, defined as a linear LMS problem over a moving time window. The wind speed is included in an augmented system state, and the pitch actuator gain is estimated as an average value during the moving time window.

- *nonlinkalwin.m* (section 6.4):
The wind speed and the unknown pitch actuator gain are estimated with the Kalman filter defined as a nonlinear LMS problem over a moving time window. The wind speed is included in an augmented system state. The pitch actuator gain is estimated as a solution to a nonlinear optimization procedure, yieling an average value during the moving time window.

- *mmod_sigma* (section 7.4):
The wind speed and the unknown pitch actuator gain are estimated with the IMM estimator. The wind speed is included in an augmented system state, and a model set is defined that describes a HAWT with a nominal and faulty pitch actuator gain.

- *mmod_delay* (section 7.5):
The wind speed and the unknown pitch actuator delay are estimated with the IMM estimator. The wind speed is included in an augmented system state, and a model set is defined that describes a HAWT with a nominal and a faulty pitch actuator delay.

Bibliography

Bar-Shalom, Y. & Fortmann, T. (1988). Tracking and Data Association. *Mathematics in Science and Engineering*, **179**. Academic Press.

Barron, R. (1996). *Engineering Condition Monitoring - Practice, Methods and Applications.* Longman, Essex.

Basseville, M. & Nikiforov, I. V. (1993). *Detection of abrupt changes.* Prentice Hall, Englewood Cliffs, New Jersey 07632.

Blanke, M., Bøgh, S., Jørgensen, R., & Patton, R. (1995). Fault detection for a Diesel Engine Actuator: a benchmark for FDI. *Control Engineering Practice*, **3**(12), 1731–1740.

Blanke, M., Izadi-Zamanabadi, R., Bøgh, S., & Lunau, C. (1997). Fault-tolerant Control Systems - a holistic view. *Control Engineering Practice*, **5**(5), 693–702.

Boutayeb, M., Rafaralahy, H., & Darouach, M. (1997). Convergence Analysis for the Extended Kalman Filter used ass an Observer for Nonlinear Deterministic Discrete-Time Systems. *IEEE Transactions on Automatic Control*, **42**(4), 581–586.

Caselitz, P., Giebhardt, J., & Mevenkamp, M. (1994). On-line fault detection and prediction in wind energy converters. *Proceedings of EWEC'94, Thessaloniki, Greece*, pages 623–627.

Chou, C. T. & Verhaegen, M. (1997). Subspace Algorithms for the Identification of Multivariable Dynamic Errors-in-Variables Models. *Automatica*, **33**(10), 1857–1869.

Darwish, H. A., Taalab, A.-M. I., & Kawady, T. A. (2001). Development and Implementation of an ANN-Based Fault Diagnosis Scheme for Generator Winding Protection. *IEEE Transactions on Power Delivery*, **16**(2), 208–214.

Duncan, D. & Horn, S. (1972). Linear dynamic recursive estimation from the viewpoint of regression analysis. *Journal of the American Statistical Association*, **67**, 815–821.

Efe, M. & Atherton, D. P. (1997). The IMM approach to the fault detection problem. *Conference Proceedings SYSID '97*, **2**, 625–630.

Elhor, N., Bertrand, R., Postaire, J.-G., & Hamad, D. (1999). Neural networks for wind turbine supervision. *Elektrotechnik und Informationstechnik*, **116**(6), 366–369.

EWEA (1997). Wind energy - the facts. Technical report, European Commission, Directorate-General for Energy.

Frank, P. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. *Automatica*, **26**, 459–474.

Gertler, J. (1997). Fault detection and isolation using parity relations. *Control Engineering Practice*, **5**(5), 653–661.

Ghoshal, A., Sundaresan, M. J., Schultz, M. J., & Pai, P. F. (2000). Structural Health Monitoring techniques for wind turbine blades. *Journal of Wind Engineering and Industrial Aerodynamics*, **85**(3), 309–324.

Godwin, G. C., Graebe, S. F., & Salgado, M. E. (2001). *Control System Design*. Prentice Hall, Upper Saddle River, New Jersey 07458.

Golub, G. H. & Loan, C. F. V. (1989). *Matrix Computations*. Johns Hopkins University Press. pp. 557-558.

Golub, G. H. & Pereyra, V. (1973). The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal of Numerical Analysis*, **10**(2), 413–432.

Griffin, Jr., G. C. & Maybeck, P. S. (1997). MMAE/MMAC Control for Bending with Multiple Uncertain Parameters. *IEEE Transactions on Aerospace and Electronic Systems*, **33**(3), 903–911.

Gustafsson, F. (2000). *Adaptive Filtering and Change Detection*. John Wiley, New York, NY, USA.

Hau, E. (2000). *Wind Turbines: Fundamentals, Technology, Application, Economics*. Springer Verlag, Berlin.

Haverkamp, L. R. J. (2001). *State Space Identification: Theory and Practice*. Delft University of Technology, Delft.

Horch, A. (2000). *Condition Monitoring of Control Loops*. Department of Signals, Sensors and Systems.

Isermann, R. (1997). Supervision, Fault Detection and Fault Diagnoiss methods - An Introduction. *Control Engineering Practice*, **5**(5), 639–652.

Isermann, R. & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, **5**(5), 707–719.

Isermann, R. & Ballé, P. (2000). Applied Terminology of Fault Detection, Supervision and Safety for Technical Processes. IFAC Technical Committee SAFEPROCESS 2000, Budapest, Hungary.

Kalman, R. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME - Journal of Basic Engineering*, **82D**, 35–45.

Kanev, S. & Verhaegen, M. (2000). Controller reconfiguration for nonlinear systems. *Control Engineering Practice*, **8**, 1223–1235.

Krothapalli, K., Prasad, J., & Peters, D. A. (1999). Helicopter Rotor Dynamic Inflow Modeling for Maneuvering Flight. Presented at the 55^{th} Annual Forum of the American Helicopter Society, Montreal, Canada.

Kulkarni, A. S., El-Sharkawi, M. A., II, R. J. M., Andexler, G., King, J., & Kerszenbaum, I. (2000). Development of a Technique for On-Line Detection of Shorts in Field Windings of Turbine-Generator Rotors: Circuit Design and Testing. *IEEE Transactions on Energy Conversion*, **15**(1), 8–13.

Leonhardt, S. & Ayoubi, M. (1997). Methods of Fault diagnosis. *Control Engineering Practice*, **5**(5), 683–692.

Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall, first edition.

Maybeck, P. S. & Stevens, R. D. (1999). Reconfigurable Flight Control via Multiple-Model Adaptive Control Methods. *Elektrotechnik und Informationstechnik*, **116**(6), 366–369.

Megahed, A. & Malik, O. (2000). Experimental Testing of a Neural Network Based Digital Differential Relay for Synchronous Generators. *IEEE Transactions on Power Delivery*, **15**(1), 80–85.

Molenaar, D.-P. (2000). Control Relevant Structural Modeling of Flexible Wind Turbines. In Proceedings of Windpower 2000, Wyndham Convention Center, Palm Springs, California, USA.

Notash, L. & Moore, T. N. (2002). Fault analysis in mechatronic systems. In R. Bishop, editor, *The Mechatronics Handbook*, pages 1–15. CRC Press, New York. pre-print.

Noura, H., Sauter, D., Hamelin, F., & Theilliol, D. (2000). Fault Tolerant Control in Dynamic Systems: Application to a Winding Machine. *IEEE Control Systems Magazine*, pages 33–49.

Patton, R. & Chen, J. (1997). Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice*, **5**(5), 671–682.

Patton, R. J. (2000). Fault-Tolerant Control Systems: the 1997 situation. *Conference Safe-Process 2000*, pages 1033–1054.

PCH Engineering (1993). PCH 1026 - Digital windturbine structural vibration monitor. Product Data.

Poljakov, V. & Tsvetkov, V. (1999). Methods of stator winding on-line diagnosis for large turbine generator preventive maintenance. *IEEE Transactions on Energy Conversion*, **14**(4), 1646–1650.

Rong Li, X. (1996). Hybrid estimation techniques. In C. T. Leondes, editor, *Control and Dynamic Systems*, volume 76, pages 213–287. Academic Press, New York.

Schultz, M. J., Pai, P. F., & Inman, D. (1999). Health monitoring and active control of composite structures using piezoceramic patches. *Composites part B*, **30**, 713–725.

Smith, G., Clayton, B., Dutton, A., & Irving, A. (1993). Infra-red condition thermography for condition monitoring of composite wind turbine blades: feasibility studies using cyclic loading tests. *Proceedings of the 15th British Wind Energy Association Conference*, pages 365–371.

Söderström, T. & Stoica, P. (1989). *System Identification*. Prentice-Hall, Hemel Hempstead, UK.

Van den Hof, P. (1997). *System Identification*. Delft University of Technology, Delft, manuscript version edition. Lecture Notes tn3111 2001; version 4, March 1997.

Van Engelen, T., Van der Hooft, E., & Schaak, P. (2001a). Development of Wind Turbine Control Algorithms for Industrial Use. Technical Report ECN-RX–01-060, ECN.

Van Engelen, T., Van der Hooft, E., & Schaak, P. (2001b). Ontwerpgereedschappen voor de Regeling van Windturbines. Technical Report ECN-I-CTRLTOOL, ECN.

Verdult, V. (2002). *Nonlinear System Identification: a State-Space Approach*. Twente University Press, Enschede.

Verdult, V. & Verhaegen, M. (2001). Deriving the Recursive Kalman Filter from a Weighted Least Squares Problem.

Verhaegen, M. & Verdult, V. (2001). *Filtering and System Identification: An Introduction*. University of Twente, Enschede.

Zhang, Y. & Rong Li, X. (1998). Detector and Diagnosis of Sensor and Actuator Failures using IMM Estimator. *IEEE Transactions on Aerospace and Electronic Systems*, **34**(4), 1293–1313.