Northumbria Research Link

Citation: Rahimilarki, Reihane (2021) Fault Diagnosis for Wind Turbine Systems by Using Neural Network and Deep Learning Techniques. Doctoral thesis, Northumbria University.

Fault Diagnosis for Wind Turbine Systems by Using Neural Network and Deep Learning Techniques

Reihane Rahimilarki

Submitted for the Degree of Doctor of Philosophy
from Northumbria University

Supervisors:
Dr. Zhiwei Gao
Dr. Nanlin Jin

Department of Mathematics, Physics, and Electrical Engineering
Northumbria University
Newcastle, Tyne and Wear NE1 8ST, U.K.

July 2021

Abstract

Concerning the fact that the number of wind turbines is increasing worldwide, it seems necessary to implement monitoring systems. To respond to this demand, this PhD thesis studies different fault diagnosis techniques in order to improve the reliability and reduce maintenance costs. Based on the fact that a considerable amount of data is stored via SCADA in every industry nowadays, the methods developed on historical data (called data-driven methods) can be very beneficial. By analysing the historical data, the changing trends of a nonlinear dynamics, such as a wind turbine, can be predicted. Moreover, by applying suitable approaches, one can distinguish different faults based on the output of the system.

The first part in this research reviews a neural network identification method by decoupling linear and nonlinear parts of a wind turbine model. As for the linear part, a Luenberger observer is designed, while for the nonlinear part, a neural network observer is proposed. By having an identification model for a wind turbine system, residual-based fault detection is studied.

The second part in this research proposes a novel neuro-robust fault estimation method to deal with the occurred faults on actuators or sensors. The challenge in this method is environmental disturbances and sensor noises. To overcome these problems and simultaneously estimate the faults and the states, an augmented system is proposed in different scenarios of actuator faults or sensor faults. Then, a neural network updating rule is calculated along with the robust performance index to fully achieve this goal. The stability of the augmented system is guaranteed by having a Lyapunov function and input-to-state stability criteria.

The third and final part in this research studies different structures of Convolutional Neural Networks for the problem of fault classification in a wind turbine. As working with time-series signals is challenging in deep learning classification, a pre-processing analysis is applied to prepare the data of system outputs for the input of the model.

Each proposed method is applied to a 4.8 MW wind turbine benchmark and obtained results are illustrated and discussed to validate the accuracy and performance of the approach.

Keywords *Wind Turbine, fault estimation, fault detection, deep learning, neural networks, convolutional neural networks.*

Acknowledgements

It would not have been possible to carry out this research without the kind support of many people. I would like to thank all of them that helped me during this journey. First of all, I would like to thank my supervisor, Dr Zhiwei Gao, for his kind support and his great personality. His constructive discussions and scientific feedback have helped me do my PhD as best as I can. Without his kindness, constant motivation, and moral support, this work could not have been accomplished. I would also like to thank my second supervisor, Dr Nanlin Jin, who helped me through the concept of machine learning and deep learning. Moreover, I would like to express my thanks to all the staffs and colleagues in the Electrical Engineering Department, who have supported me with their kindness and for being there whenever I needed them. Finally, I would like to gratefully acknowledge Northumbria University, for providing a good research atmosphere, facilities and my scholarship, without which this PhD would be impossible.

Dedicated to 176 passengers of #PS752, whom losses
brought us never-ending grief.

And to all Iranian women, who struggle constantly
for their rights and freedom.

Declaration of Authorship

I declare that the work contained in this thesis is my own work, and to the best of my knowledge, it reproduces no material previously published or written, nor material has not been submitted for any other degree. Any ethical clearance for the research presented in this thesis has been approved.

Approval has been sought and granted by the University Ethics Committee on 5th of November, 2017. I declare that the word count of this thesis is 31,533 words.

Name:

Date:

Publications

- **R. Rahimilarki** and Z. Gao, "Grey-box model identification and fault detection of wind turbines using artificial neural networks," in 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), pp. 647–652, IEEE, 2018.
- **R. Rahimilarki**, Z. Gao, A. Zhang, and R. Binns, "Robust neural network fault estimation approach for nonlinear dynamic systems with applications to wind turbine systems," IEEE Transactions on Industrial Informatics, vol. 15, no. 12, pp. 6302–6312, 2019.
- **R. Rahimilarki**, Z. Gao, N. Jin, and A. Zhang, "Time-series deep learning fault detection with the application of wind turbine benchmark," in 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), pp. 1337–1342, IEEE, 2019.
- **R. Rahimilarki**, Z. Gao, N. Jin, R.Binns, and A. Zhang, "Data-driven sensor fault estimation for the wind turbine systems," in 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), pp. 1211-1216, IEEE, 2020.

Contents

Abstract ii

Acknowledgements iv

Declaration of Authorship vii

Publications viii

Contents ix

List of Figures xiii

List of Tables xv

Abbreviations xvii

Notations xix

1 Introduction 1
- 1.1 Research Motivations 1
- 1.2 Contributions and Overview 3

2 Literature Review 7
- 2.1 Model Dynamics of the Wind Turbine 7
  - 2.1.1 Wind Model 10
  - 2.1.2 Blade and Pitch System 11
    - 2.1.2.1 Aerodynamic Model 11
    - 2.1.2.2 Pitch System Model 12
  - 2.1.3 Drive Train Model 12
  - 2.1.4 Generator and Converter Model 13
  - 2.1.5 Controller 13
  - 2.1.6 Overall Model 14

2.1.7 Breakdown and Faults in a Wind Turbine 17
2.2 Fault Diagnosis Techniques 18
2.2.1 Model-Based Fault Diagnosis Approach 18
2.2.2 Signal-Based Fault Diagnosis Approach 19
2.2.3 Knowledge-Based Fault Diagnosis Approach 20
2.2.4 Hybrid Fault Diagnosis Approach 22
2.3 Machine Learning 22
2.3.1 Artificial Neural Networks 23
2.3.2 Applications of ANN in Fault Diagnosis of Wind Turbines . 24
2.3.3 Deep Learning 25
2.3.3.1 Convolutional neural network 25
2.3.3.2 Applications of CNN 28
2.4 Summary 28

3 Grey-box Model Identification and Fault Detection Using Artificial Neural Networks 31
3.1 Model Identification and Fault Detection method 32
3.1.1 Model Identification 32
3.1.2 Fault Detection 35
3.2 Validating on Wind Turbine 36
3.3 Fault Detection for Wind Turbine 37
3.4 Summary and Conclusions 39

4 Robust Neural Network Fault Estimation Approach for Nonlinear Dynamic Systems 41
4.1 Neuro-Robust Actuator Fault Estimation 42
4.1.1 Actuator Fault Estimation for 4.8 MW Wind Turbine Benchmark 54
4.2 Neuro-Robust Sensor Fault Estimation 64
4.2.1 Simulation and Results 70
4.3 Summary and Conclusions 73

5 Time-series Deep Learning Fault Detection 83
5.1 Introduction to Deep Learning 83
5.2 Preprocess the Input Data 84
5.3 Scenario 1: One Actuator Fault 85
5.3.1 CNN with One convolutional Layer 86
5.3.2 CNN with Two convolutional Layer 86
5.3.3 Simulation 87
5.4 Scenario 2: Two Actuator Faults 90
5.4.1 Converting Two Signals into One Image 90
5.4.2 CNN Structure 92
5.4.3 Simulation 92
5.5 Scenario 3: Four Sensor Faults 94
5.5.1 Converting Four Signals into One Image 94

5.5.2 CNN Structure 96
5.5.3 Simulation 96
5.6 Summary and Conclusions 99

6 Conclusions and Future Works 101
6.1 Summary and Conclusions 101
6.2 Future Works 103

A Simulation of Wind Turbine Benchmark in Matlab 105

References 109

List of Figures

2.1 (a) Vertical-axis Structure. (b) Horizontal-axis Structure. 8
2.2 A typical schematic of a wind turbine. 9
2.3 Wind turbine subsystems. 10
2.4 The variation of wind in the model. 11
2.5 The ratio of occurred faults in wind turbines 17
2.6 A Three-layer Fully Connected Neural Network. 23
2.7 A simplified CNN consists of three main layers, used for classification. 26
2.8 The kernel and its stride in convolutional layer. 27
2.9 The max pooling layer. 27

3.1 Block diagram of neural network observer. 34
3.2 Rotor speed signal and its estimation. 37
3.3 Generator speed signal and its estimation. 38
3.4 Pitch angle signal and its estimation. 38
3.5 Generator torque signal and its estimation. 39
3.6 The healthy and faulty signal of τ_g. 40
3.7 The norm of residual. 40

4.1 The scheme of ANN based observer. 45
4.2 Flowchart of the Combined Algorithm. 54
4.3 Pitch angle signal and its estimation using Luenbeger observer. 55
4.4 Generator torque signal and its estimation using Luenbeger observer. 55
4.5 Rotor speed signal and its estimation. 59
4.6 Generator speed signal and its estimation. 60
4.7 Pitch angle signal and its estimation. 60
4.8 Generator torque signal and its estimation. 61
4.9 Torsion angle signal and its estimation. 61
4.10 Pitch angular velocity signal and its estimation. 62
4.11 20% Faults on reference of generator torque actuator and its estimation. . . . 62
4.12 20% Faults on reference of pitch angle actuator and its estimation. 63
4.13 The model of the observer based on ANN. 65
4.14 Comparison of occurred fault on ω_r sensor in the first scenario. 74
4.15 Comparison of occurred fault on ω_g sensor in the first scenario. 75
4.16 Comparison of occurred fault on β sensor in the first scenario. 76
4.17 Comparison of occurred fault on τ_g sensor in the first scenario. 77
4.18 Comparison of occurred fault on ω_r sensor in the second scenario. 78

4.19 Comparison of occurred fault on ω_g sensor in the second scenario. 79
4.20 Comparison of occurred fault on β sensor in the second scenario. 80
4.21 Comparison of occurred fault on τ_g sensor in the second scenario. 81

5.1 Fulfilling an image of $n \times n$ matrix with a time-series signal. 85
5.2 The schematic of proposed CNN with one convolutional layer. 86
5.3 The schematic of proposed CNN with two convolutional layers. 87
5.4 The conversion of raw data to 2-D gray scale image. 88
5.5 Classification of testing dataset of Scenario 1. 89
5.6 Healthy form of the two actuators' signals. 91
5.7 The process of converting a pair of signals into one image. 91
5.8 The converted image of a pair of healthy actuators' signals. 92
5.9 The schematic of proposed CNN with three convolutional layers. 92
5.10 Classification of testing dataset of Scenario 2. 94
5.11 Healthy form of the four sensors' signals. 95
5.12 The process of converting four signals into one image. 95
5.13 The converted image of four healthy sensors' signals. 96
5.14 The schematic of proposed CNN with four convolutional layers. 96
5.15 Classification of testing dataset of Scenario 3. 98

A.1 Blade & Pitch System with Actuator Fault Blocks. 106
A.2 Drive Train. 106
A.3 Generator System with Actuator Fault Blocks. 106
A.4 Controller. 107
A.5 Sensors with Fault Blocks. 107

List of Tables

2.1 Numerical quantity and their physical meanings of wind turbine parameters. 16

4.1 RMSE value of each states and faults. 63
4.2 Sensor faults for the first scenario. 72
4.3 Sensor Faults for the Second Scenario 72

5.1 Comparison results between different CNN structures in Scenario 1. . . . 90
5.2 Comparison results between different CNN structures in Scenario 2. . . . 93
5.3 Comparison results between different CNN structures in Scenario 3. . . . 97

Abbreviations

ANN	**A**rtificial **N**eural **N**networks
AI	**A**rtificial **I**ntelligence
BPNN	**B**ack **P**ropagation **N**eural **N**etworks
CI	**C**omputational **I**ntelligence
CNN	**C**onvolutional **N**eural **N**etwork
DL	**D**eep **L**earning
FCNN	**F**ully **C**onnected **N**eural **N**etwork
ISS	**I**nput to **S**tate **S**tability
LMI	**L**inear **M**atrix **I**nequalities
LSTM	**L**ong **S**hort **T**erm **M**emory
MIMO	**M**ulti **I**nput **M**ulti **O**utput
ML	**M**achine **L**earning
MLP	**M**ulti **L**ayer **P**erceptron
NNO	**N**eural **N**etwork **O**bserver
NRMSE	**N**ormalized **R**oot **M**ean **S**quare **E**rror
PID	**P**roportional **I**ntegral **D**erivative
ReLU	**Re**ctified **L**inear **U**nit
RGB	**R**ed **G**reen **B**lue
RMS	**R**oot **M**ean **S**quare
RMSE	**R**oot **M**ean **S**quare **E**rror
SCADA	**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition
SVM	**S**upport **V**ector **M**achine

Notations

0	Zero Matrix with Suitable Dimension
I	Identity matrix with Suitable Dimension
I_n	Identity matrix with Dimension of $n \times n$
R	Set of Real Numbers
R^n	n-dimensional Euclidean Space
$R^{n \times m}$	Set of $n \times m$ real matrices
M^T	Transpose of the Matrix M
$\|\|x\|\|$	Norm of the Vector x
$exp(.)$	Exponential Function, $e^{(\cdot)}$
$\sigma(.)$	Sigmoid Transfer Function
$sgn(.)$	Sign Function
L_∞	The Space of all Bounded Measurable Signals

Chapter 1

Introduction

1.1 Research Motivations

As environmental pollution and concerns about global warming are increasing year by year, the importance of using renewable and green energy becomes more vital. As a result, as one of the cleanest energy resources, wind turbine industries, have been receiving considerable attention and budgets. Gradually, wind energy has become an integrated component in the grid worldwide. In the UK, wind turbines are responsible for over 20% of the consumed electricity in 2020 [1]. Precisely speaking, by August 2020, there are 10911 installed wind turbines with the capacity of 24000 MW, of which 13600 MW onshore and the rest offshore. This production makes the United Kingdom the sixth-largest producer of wind power in the world [1]. The plan for expanding this potential is to increase the capacity of wind power up to 50000 MW by 2030[1]. Therefore, regarding this plan, careful considerations should be made to have a reliable energy resource in the grid.

However, wind turbines, just like any other electro-mechanical system, may encounter several unexpected and severe faults [2, 3]. These faults can reduce the reliability of the wind turbines in the grid, increase the number of unanticipated shutdowns, and increase maintenance requirements. Detecting any abnormalities in wind turbines in a timely manner, may also reduce the potential hazards and risks, especially off-shore. In addition, due to the high loads and extreme conditions in which a wind turbine normally works, its maintenance may be quite challenging [4]. Therefore, it seems only logical to invest in developing a fault monitoring system, helping prevent the system's undesirable behaviours and extra costing.

Fault monitoring systems can be extended to different strategies. In many industrial systems, there are conventional alarm boards, showing different faults and their severity happening in a specific section. They are merely working based on defining several thresholds for each signal and data directly from SCADA [5]. There are also fault detection methods, which are developed based on the model or the historical data of the plants. The aim of these methods is detecting any occurrence of faults even if they are so low in magnitude as soon as possible [6, 7]. In addition to the fault detection methods, industrial companies can benefit from some more complicated approaches, called fault diagnosis, which can help identify the magnitude, the pattern and/or the probabilities of repetitions of the faults [8, 9]. These techniques can also be beneficial in designing more robust controller [10] and can lead to the stability of the system.

As it is evident in recent years, more and more artificial intelligence (AI) methods have been introduced in everyday life. The applications of them are varied from software in recognizing traffic lights and autonomous driving [11, 12] to the fraud and anomaly detection in financial transactions [13, 14]. The most crucial ability of AI is its strength in predicting and coping with unknown dynamics and processing a massive amount of data to provide the most accurate and comprehensive outputs [15]. The reliable performance of these methods attracts many research interests to them [16]. This point is also valid in the field of fault diagnosis, in which there are always some unknown dynamics, perturbed parameters and a huge amount of data to analyse, that simply conventional methods are unable to deal with. Therefore, it is very reasonable to use AI in order to improve data diagnosis techniques.

Based on the facts mentioned earlier, this PhD thesis concentrates on proposing novel data-driven methods, using artificial intelligence to develop fault diagnosis methods to cope with the problem of unknown dynamics, environmental disturbances, sensor noises and data loss. The investigating faults in this thesis, may come from electrical system or mechanical parts, such as gear boxes. The origins of faults will be further discussed in 2.1.7. However, the proposed algorithms can be extended to different categories of faults. The accuracy and performance of the proposed methods can increase the reliability of wind turbines and decrease maintenance costs.

1.2 Contributions and Overview

The contributions of this thesis can be categorised as follows:

- Grey-box model identification and fault detection using artificial neural networks.

  As there are some uncertainties and unknown parameters in a wind turbine benchmark, an artificial neural network (ANN) based identification and fault detection is proposed. This method contains two parts: in the first one, an ANN observer is developed to be considered as a substitution for the real dynamics. The problem is that we cannot apply neural networks state estimation directly for a wind turbine system, because we need all the states measurable, which does not happen in a wind turbine benchmark. In a typical wind turbine dynamics, only four out of six states are measurable. Hence, to confront this problem, a Luenberger observer is designed conjointly with ANN to identify the dynamics. The second part of this method is applying a residual-based method to detect faults.

The results of this research contribution is highlighted in the following publication:

- **R. Rahimilarki** and Z. Gao, "Grey-box model identification and fault detection of wind turbines using artificial neural networks," in 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), pp. 647–652, IEEE, 2018.

- Robust neural network fault estimation approach for nonlinear dynamic systems.

  In wind turbine systems, there are always some unexpected faults, environmental disturbances, and sensor noises. Designing a fault estimation method to tackle all these problems can be challenging. The novel robust neural network fault estimation method is proposed to not only confront these challenging points but also to estimate the occurred faults on both actuator signals or the four of the sensors. In this method, the idea of back-propagation neural networks is combined with robust optimization theorem. It also guarantees the stability of the system via Lyapunov function and input-to-state stability criteria.

  The results of this research contribution are highlighted in the following publications:

  - **R. Rahimilarki**, Z. Gao, A. Zhang, and R. Binns, "Robust neural network fault estimation approach for nonlinear dynamic systems with applications to wind turbine systems," IEEE Transactions on Industrial Informatics, vol. 15, no. 12, pp. 6302–6312, 2019.

  - **R. Rahimilarki**, Z. Gao, N. Jin, R.Binns, and A. Zhang, "Data-driven sensor fault estimation for the wind turbine systems," in 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), pp. 1211-1216, IEEE, 2020.

- Time-series deep learning fault detection.

  In order to achieve a fault detection method to classify even minimal anomalies, four Convolutional Neural Network (CNN) structures are proposed. The data from the benchmark contains sensor noises, rather than actuator and sensor faults. By considering deep learning methods and a pre-processing technique of converting time-series signals into 2-D images, a novel approach is proposed that can classify different faults in a wind turbine.

  The results of this research contribution are presented in the following publications:

  - **R. Rahimilarki**, Z. Gao, N. Jin, and A. Zhang, "Time-series deep learning fault detection with the application of wind turbine benchmark," in 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), pp. 1337–1342, IEEE, 2019.

This thesis is divided into six chapters. As we overviewed the introduction in this chapter, Chapter 2 is going through the dynamics of a wind turbine in details and the main occurred faults for this system. The literature review and complementary explanation about different methods of fault detection and diagnosis are also studied in this chapter. Moreover, at the end of this chapter, supplementary information of neural networks and convolutional neural networks are explained.

In Chapter 3, grey-box model identification and fault detection methods are presented along with validating simulation. Chapter 4 focuses on robust neural network fault estimation approach for nonlinear dynamic systems. In this chapter, the essential mathematics and considerations are explained, and the stability of the system is proved in two different scenarios. Chapter 5 introduces time-series deep learning fault detection using convolutional neural networks. The pre-processing analysis, needed for better accuracy, is also discussed in this chapter. In the end, conclusion and future works are brought in Chapter 6.

Chapter 2

Literature Review

In this chapter, the benchmark of a wind turbine is explained along with the different categories of fault diagnosis and their applications in general and specifically in wind turbine systems. In section 2.1, the model of the wind turbine benchmark is introduced based on its nonlinear dynamics and state-space matrices. The potential faults that can happen in this system and their rate of occurrence are also discussed. Following to this part, section 2.2 presents the various types of fault diagnosis methods and the obstacles in this area. Moreover, the different techniques of fault diagnosis, specifically with the application of a wind turbine system are reviewed in this section. Based on the fact that the main contributions in this thesis are about mathematical analysis of ANN and deep learning in fault diagnosis, a brief introductions to those topics are brought in 2.3. At the end of this chapter, a comprehensive summary is also brought in 2.4.

2.1 Model Dynamics of the Wind Turbine

Over the past few years, wind energy has received significant attention owing to the concerns about global warming, environmental issues, and fossil fuels reduction.

During the past decade, numerous investments have been aimed to wind energy industries and the wind turbine installed capacity had a constant increase. Designing the structure and caging of the wind turbines have been improved due to the more accurate engineering and more robust composites. However, the generators, drive and control systems roughly remain the same.

Wind turbines have been built horizontally or vertically. As it can be seen in Fig. 2.1.a, in vertical wind turbines, the blades are installed vertically [17]. One of the advantages of this kind of wind turbines is that the conversion systems and the gearboxes are located on the ground, while, the disadvantage of this type is that the maintenance is somehow complicated as it normally requires rotor removal [18]. Besides, the efficiency of converting wind energy to electrical one is lower compared to horizontal turbines. For these reasons, nowadays, the modern wind turbines usually have been built horizontally, as shown in Fig. 2.1.b [19].

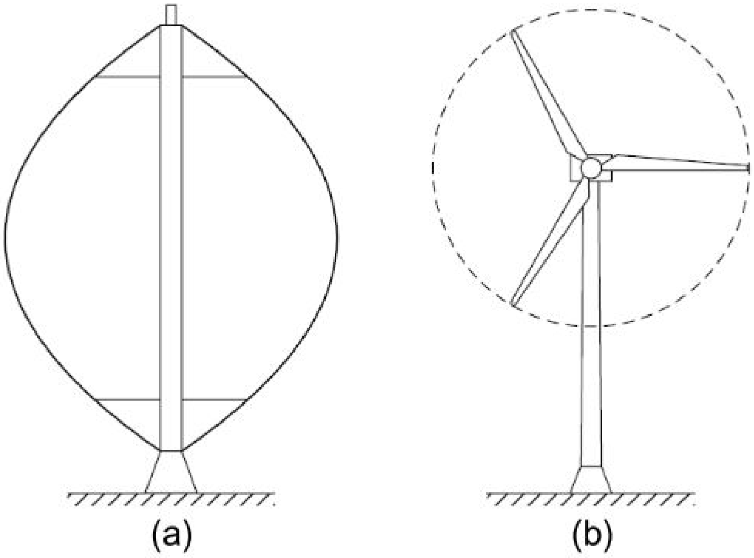


FIGURE 2.1: (a) Vertical-axis Structure. (b) Horizontal-axis Structure.

In this thesis, only the three-blade horizontal wind turbine is discussed. However, the approaches may be extended to any other structure with little adjustments.

The schematic structure and different components of a typical wind turbine are presented in Fig. 2.2 [20]. Wind energy rotates the blades and produces mechanical power, transmitted to the system via a shaft, which is connected to these blades. A generator converts mechanical energy to electrical one based on the rotation of the blades. The blades angles can vary to handle the wind speed variations. Meanwhile, the yaw structure is designed to align the whole wind turbine based on the direction of the anemometer.

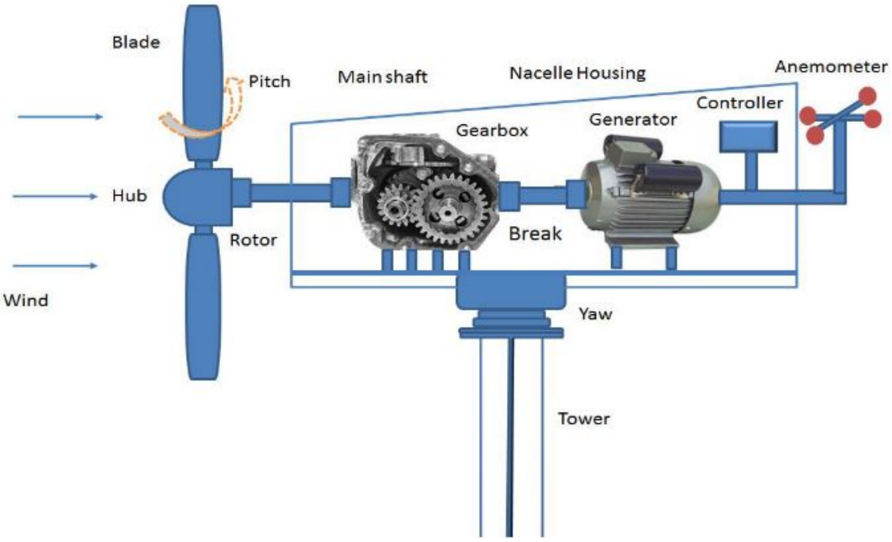


Figure 2.2: A typical schematic of a wind turbine.

A wind turbine benchmark model was proposed in [21], based on a three-blade horizontal wind turbine driven by variable speeds, with a full converter coupling and a rated power of 4.8 MW. It can be modelled in several subsystems as follows:

- Blade and Pitch Systems.
- Drive Train
- Generator and Converter
- Controller

The signal relation between these subsystems can be shown in Fig. 2.3 [21]. As illustrated in this figure, each of the subsystems may require some signals from other subsystems and supply a feedback to them. From the left side, the velocity of the wind (v_w) is working as the input of the Blade and Pitch system; so, it generates the required rotor torque (τ_r) for the Drive Train subsystem. By receiving the feedback of generator torque (τ_g), it also produces two states of the system, ω_r and ω_g, which are the rotor speed and generator speed, respectively. The Generator and Converter subsystem, receives the desired generator torque, $\tau_{g,r}$ and produce the generator power, P_g. The responsibility of the controller here, is to generate the desired pitch angle (β_r) and the desired generator torque, $\tau_{g,r}$, for related subsystems, by getting feedback from the other three subsystems.

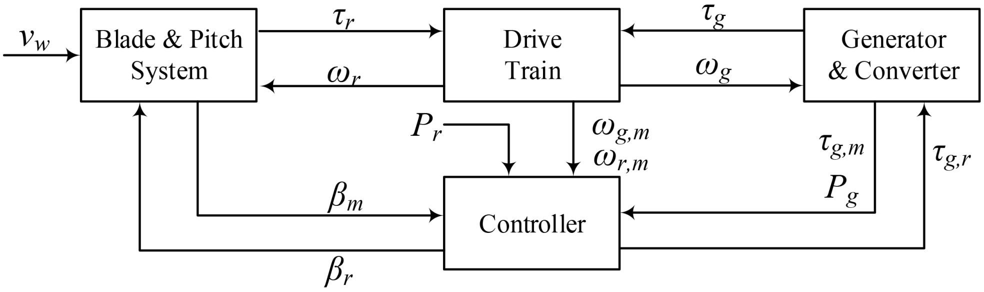


FIGURE 2.3: Wind turbine subsystems.

In the following sections, different subsystems, various parameters, their relating physical equations and the wind model are fully explained [21].

2.1.1 Wind Model

The wind can be generally modelled in four parts: the mean wind (slow wind variations) $v_m(t)$, a stochastic part $v_s(t)$, the wind shear $v_{ws}(t)$ (which is the effect of wind energy lost at the surface of the earth, commonly resulting in an increasing wind speed as the distance to earth surface increases), and the tower shadow $v_{ts}(t)$. The combined wind model is given by:

$$v_w(t) = v_m(t) + v_s(t) + v_{ws}(t) + v_{ts}(t). \tag{2.1.1}$$

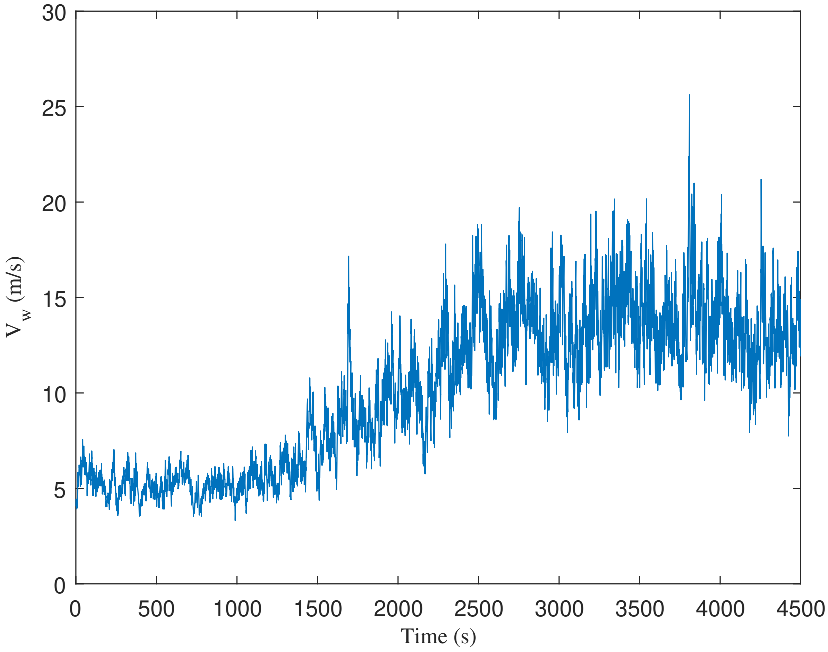


FIGURE 2.4: The variation of wind in the model.

The variation of the wind velocity in this benchmark can be seen in Fig. 2.4. As it can be seen in this figure, the velocity of the wind is varied between 4-20 m/s, with some spike of 25 m/s. As discussed in [21], it is an acceptable range, in which, the generator can operate normally.

2.1.2 Blade and Pitch System

This model is comprised of two parts: the aerodynamic model and pitch model.

2.1.2.1 Aerodynamic Model

The aero torque applied to the rotor is modelled as below:

$$\tau_r(t) = \frac{1}{2}\rho\pi R^3 C_q(\lambda(t), \beta(t)) v_w^2, \tag{2.1.2}$$

where ρ is the air density, R is the radius of the rotor, C_q is the torque applied to the rotor coefficient which is a function of the pitch angle β and tip-speed-ratio λ

given by:

$$\lambda(t) = \frac{R\omega_r(t)}{v_w(t)}, \tag{2.1.3}$$

where ω_r is the turbine rotor angular speed.

2.1.2.2 Pitch System Model

The pitch system considered in this model is hydraulic. The closed-loop dynamic of the pitch system is described by a second-order system [21]:

$$\frac{\beta(s)}{\beta_r(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega n s + \omega_n^2} \tag{2.1.4}$$

where β_r is the pitch reference.

2.1.3 Drive Train Model

The drive train model is a two-mass model provided by [21]:

$$\begin{bmatrix} \dot{\omega}_r(t) \\ \dot{\omega}_g(t) \\ \dot{\theta}_\delta(t) \end{bmatrix} = A_{DT} \begin{bmatrix} \omega_r(t) \\ \omega_g(t) \\ \theta_\delta(t) \end{bmatrix} + B_{DT} \begin{bmatrix} \tau_r(t) \\ \tau_g(t) \end{bmatrix}, \tag{2.1.5}$$

where τ_g is the generator torque, ω_g is the generator rotating speed and θ_δ is the torsion angle of the drive train. The state-space matrices are:

$$A_{DT} = \begin{bmatrix} -\frac{B_{dt}+B_r}{J_r} & \frac{B_{dt}}{N_g J_r} & -\frac{K_{dt}}{J_r} \\ \frac{\eta_{dt} B_{dt}}{N_g J_g} & \frac{-\frac{\eta_{dt} B_{dt}}{N_g^2} - B_g}{J_g} & \frac{\eta_{dt} K_{dt}}{N_g J_g} \\ 1 & -\frac{1}{N_g} & 0 \end{bmatrix}, B_{dt} = \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \\ 0 & 0 \end{bmatrix},$$

where J_r and J_g are the rotor and generator moment of inertia, B_r and B_g are the rotor and generator external damping, B_{DT} is the torsion damping coefficient, N_g and η_{DT} are the gear ratio and efficiency of drive train, and K_{DT} is the torsion stiffness.

2.1.4 Generator and Converter Model

On a system level of the wind turbine, the generator and converter dynamics can be modelled as a first-order transfer function [21]:

$$\frac{\tau_g(s)}{\tau_{g,r}(s)} = \frac{\alpha_{gc}}{s + \alpha_{gc}} \tag{2.1.6}$$

where α_{gc} is the generator and converter model parameter. The power produced by the generator is given by:

$$P_g = \eta_g \omega_g(t) \tau_g(t) \tag{2.1.7}$$

where η_g is the efficiency of the generator.

2.1.5 Controller

A simple control scheme, such as a PID controller is often used in wind turbine systems. More details and numerical values of the control parameters can be found in [21].

2.1.6 Overall Model

The overall benchmark model can be obtained by integrating these subsystems [22]:

$$
\begin{aligned}
\dot{x} &= A(x)x + Bu, \\
y &= Cx,
\end{aligned}
\tag{2.1.8}
$$

where $x = [\omega_r \quad \omega_g \quad \theta_\delta \quad \dot{\beta} \quad \beta \quad \tau_g]^T$ is the state vector and $u = [\tau_{g,r} \quad \beta_r]^T$ is the control input from the pre-designed controller.

$$
\begin{bmatrix} \dot{\omega}_r \\ \dot{\omega}_g \\ \dot{\theta}_\delta \\ \ddot{\beta} \\ \dot{\beta} \\ \dot{\tau}_g \end{bmatrix} = A(x) \begin{bmatrix} \omega_r \\ \omega_g \\ \theta_\delta \\ \dot{\beta} \\ \beta \\ \tau_g \end{bmatrix}, + B \begin{bmatrix} \tau_{g,r} \\ \beta_r \end{bmatrix},
\tag{2.1.9}
$$

By substituting (2.1.3) - (2.1.7) in (2.1.9), the following state-space equation of the wind turbine can be obtained:

$$
\begin{bmatrix} \dot{\omega}_r \\ \dot{\omega}_g \\ \dot{\theta}_\delta \\ \ddot{\beta} \\ \dot{\beta} \\ \dot{\tau}_g \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 & -\dfrac{1}{J_g} \\ 1 & -\dfrac{1}{N_g} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\zeta\omega n & -\omega_n^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha_{gc} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_g \\ \theta_\delta \\ \dot{\beta} \\ \beta \\ \tau_g \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \omega_n \\ 0 & 0 \\ \alpha_{gc} & 0 \end{bmatrix} \begin{bmatrix} \tau_{g,r} \\ \beta_r \end{bmatrix},
\tag{2.1.10}
$$

where $A11 = -\frac{B_{dt}+B_r}{J_r} + \frac{1}{2J_r\lambda^2}\rho\pi R^5 C_q(\lambda,\beta)\omega_r$, $A_{12} = \frac{B_{dt}}{N_g J_r}$, $A_{13} = -\frac{K_{dt}}{J_r}$, $A_{21} = \frac{\eta_{dt}B_{dt}}{N_g J_g}$, $A_{22} = \frac{-\frac{\eta_{dt}B_{dt}}{N_g^2} - B_g}{J_g}$, and $A_{23} = \frac{\eta_{dt}K_{dt}}{N_g J_g}$. The physical meaning and the numerical quantity of each parameter can be found in Table 2.1 [21, 22].

Since just four out of six states of the system are measurable, output vector is defined as $y = [\omega_r \quad \omega_g \quad \beta \quad \tau_g]^T$. Therefore, the output matrix can be clearly written as below:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.1.11}$$

It is worth to mention that θ_δ and $\dot{\beta}$ are immeasurable and it is one of the challenges to design NN observer for wind turbines. In chapter 4, an algorithm is proposed to deal with this problem.

As it is clearly stated in $A11$, this parameter, which is part of the matrix A of the system, depends on the parameter $C_q(\lambda,\beta)$, defined as the torque applied to the rotor coefficient. As β is a state of the system, it is concluded that matrix A is depending on one of the state of the system. Therefore, it is not independent and the value of it depends on β. Hence, the equation (2.1.10) is nonlinear. Due to the nonlinearity of the model, the identification and observer designing is challenging. The novel ideas to relax this condition will be discussed thoroughly in chapters 3, 4 and 5.

Table 2.1: Numerical quantity and their physical meanings of wind turbine parameters.

Param.	**Physical Meaning**	**Value**
B_{dt}	Torsion Damping Coefficient	$775.49\frac{Nm.s}{rad}$
B_r	Rotor External Damping	$7.11\frac{Nm.s}{rad}$
J_r	Rotor Moment of Inertia	$55 \times 10^6 Kg.m^2$
ρ	Air Density	$1.225\frac{Kg}{m^3}$
R	Rotor Radius	$57.5m$
N_g	Gear Ratio	95
K_{dt}	Torsion Stiffness	$2.7 \times 10^9 \frac{Nm}{rad}$
J_g	Generator Moment of Inertia	$390Kg.m^2$
η_{dt}	Efficiency of Drive Train	0.97
B_g	Generator External Damping	$45.6\frac{Nm.s}{rad}$
ζ	Damping Ratio	0.6
ω_n	Natural Frequency	$11.11\frac{rad}{s}$
α_{gc}	Generator and Converter Parameter	$50\frac{rad}{s}$

2.1.7 Breakdown and Faults in a Wind Turbine

Similar to other complex industries, faults may occur in wind turbines due to age or unexpected events, which may cause breakdown and relatively high-cost maintenance. Faults in wind turbines can be both in electrical parts and mechanical ones. The nature of the faults can be based on environmental factors, such as high fluctuations of the wind, or based on physical aspects of the components, e.g aging, saturation, or thermal problems. The most occurred faults and their ratios are illustrated in Fig. 2.5 [23].

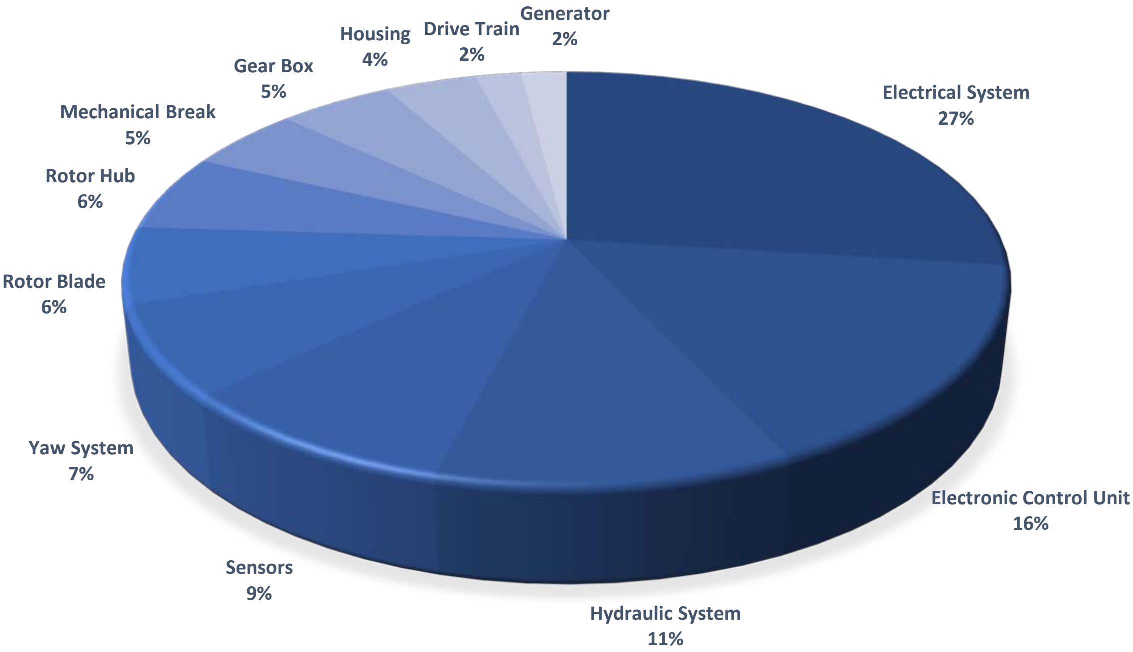


Figure 2.5: The ratio of occurred faults in wind turbines .

Apart from the faults with the basis of high wind fluctuations, many of the faults can be prevented or effectively decreased by a suitable monitoring system. If a typical and low-risk fault occurs, and it is not detected and resolved in proper time, it may lead to a severe failure and probable breakdown. Besides, substandard reliability directly decreases the availability of wind power in the grid. Based on the mentioned issues, fault detection plays an essential role in increasing the reliability of wind turbines.

2.2 Fault Diagnosis Techniques

Fault diagnosis can be categorized into three main methods: *model-based*, *signal-based*, and *knowledge-based* approaches following the classical literature survey[24], [25].

2.2.1 Model-Based Fault Diagnosis Approach

In model-based methods, the model of the system to be monitored should be available to the designers. The fault diagnosis is developed based on the model of systems, and the accuracy of the approach depends on the accuracy of physical parameters and the precision of mathematical modelling [26, 27]. Two cascade steps are usually applied in a typical model-based fault diagnosis: *Residual Generation*, and *Residual Evaluation*. In the step of residual generation, the output of the designed model is compared to the output of the real system as follows[28]:

$$r(t) = y - \hat{y} \tag{2.2.1}$$

where, y is the output of the real system and $\hat{y}$ is the output of the model. After this step, the step of residual evaluation should be designed. In this step, based on the expert decision, some thresholds are set to reflect the conditions of the system, which can be healthy or faulty. This step may also include some mathematical or statistical approaches to design a model-based fault observer. In [29], a hidden Markov model-based on scalar quantisation was proposed to solve the problem of accuracy and sensitivity of fault diagnosis in wind turbines. Performance degradation was addressed in [30] and an approach based on classifier adapting, and a regression model was proposed to cope with this problem. In [31], an interval-based observer with analytical redundancy relations was studied to design a fault diagnosis method. A method consists of a model-based fault detection technique, and the proportional-integral observer was presented in [32] to address the problem of noise filtering and to achieve a suitable convergence.

Moreover, many approaches were studied in order to reduce the effects of environmental disturbances and modelling error. [33] propounded a model-based fault diagnosis and detection in order to correct signals in uncertainty and actuator faults conditions. A sensor failure detection scheme was proposed in [34] using robust analytical performance. In [35], a fault detection and isolation method was presented to attenuate the disturbance's effects based on $H\infty$ performance index and LMI approach. The problem of uncertain parameters in a class of nonlinear systems was studied in [36] and an adaptive sliding mode observer was proposed in order to approximate the fault signals. A similar approach was also studied in [37] in order to solve the problem of oscillatory failure case in an actuator. Although these approaches have a lot of advantageous, the main disadvantageous of them are their inability to deal with immeasurable states.

2.2.2 Signal-Based Fault Diagnosis Approach

In signal-based methods, the input-output model is not necessary to be available. However, the measured signals become essential, and the decision on fault diagnosis is made based on these signals, and their attributes [38]. Knowing about the featured signals and how an occurred fault has a reflection on the output signal, needs a technical aspect of view.

In general, the signals that are used for fault diagnosis can be analysed in time-domain or frequency-domain. In time-domain fault diagnosis, some time-domain features, such as slope, and root mean square are analysed [39, 40]. In frequency-domain fault diagnosis, the frequency-based parameters are studied to carry out a signal-based fault diagnosis approach. One of the examples of this category is using wavelet for denoising of the vibration of the machinery systems [41]. There are also some researches on having Fourier Transform in signal-based fault diagnosis [42–44].

Moreover, signal-based fault diagnosis is vastly studied in the application of wind turbines. In [45], fault diagnosis in wind turbine planetary gearboxes was analysed, and an approach based on automatic sparse representation was proposed for detecting weak transients. [46] addressed a multiscale filtering construction approach to solve the fault diagnosis problem under speed-varying and noisy conditions in wind turbine gearboxes. In [47], a fault diagnosis method was suggested in order to investigate bearing faults in wind turbine generators using variable shaft rotating frequencies. [48] studied a diagnostic method based on Park‘s vector phase angle in the application of permanent-magnet synchronous generator drives of a wind turbine. In [49], a fault diagnosis approach based on time-frequency maps was considered in order to solve rotor asymmetry faults in the wind turbine generator unit. Two fault classification methods were proposed in [50, 51] based on Principle Component Analysis technique, to deal with the problem of sinusoidal fault and actuator effectiveness loss in a wind turbine benchmark.

2.2.3 Knowledge-Based Fault Diagnosis Approach

Alternatively, knowledge-based methods are particularly suitable for the cases with a large amount of historical data, and where the explicit relationships of the system dynamics are challenging to derive. From this aspect, knowledge-based fault diagnosis is called *data-driven* approach [25]. Based on the fact that it is complicated to model fault dynamics for a system in general and a wind turbine in particular, it is very challenging to study fault diagnosis in the aforementioned complex system. Therefore, data-driven methods can be beneficial in industrial area [52]. [53] addressed a data-driven method in order to monitor nonlinear systems using available measurements. Recent results in key-performance-indicator oriented prognosis and diagnosis with a Matlab toolbox Db-kit were reported in [54]. A data-driven fault diagnosis approach was studied in [55], developed an extreme learning machine in order to cope with the sensor fault problem.

Computational methods, such as Fuzzy [56, 57], Support Vector Machine (SVM) [58, 59], Long Short-term Memory network (LSTM) [60, 61], Artificial Neural Networks (ANN) [62], and Convolutional Neural Networks (CNN) [63] have been widely used in the field of fault diagnosis and detection due to their spectacular power to predict unknown parameters and identify the nonlinear systems. For instance, in [64], a fault detection method based on LSTM was proposed in a wind turbine benchmark. An LSTM based fault diagnosis was studied by [65] by analysing frequency data. [66] proposed a fault diagnosis and isolation approach in order to handle uncertain models and noisy signals, using a fuzzy method in wind turbine systems. [10], and [22] addressed a robust method based on Takagi-Sugeno Fuzzy systems for the problem of unknown fault diagnosis. The problem of fault classification for vibration signals was studied in [67] and a novel solution based on Fuzzy and SVM were proposed. By using the current signal of the generator, a fault identification method was considered based on SVM in order to classify different faults occurred in the generator. Neuro-Fuzzy fault diagnosis approach for the problem of bearing failures in wind turbines was suggested in [68]. In [69], a CNN based approach was prposed for fault classification in a wind turbine.

2.2.4 Hybrid Fault Diagnosis Approach

The hybrid method is defined as using the combination of any of the previously mentioned methods. In practice, many approaches can be categorized as a hybrid method, since typically there is some information (e.g. the subsystems, some of the parameters, states, outputs) available to the designers. Therefore, they can use them to model a better and accurate technique. As for the examples of hybrid methods, in [70], by using SVM and stochastic subspace identification and signal processing methods, a novel hybrid fault diagnosis approach was proposed. By integrating Fuzzy method and Wavelet transform, [71] suggested a fault diagnosis approach to deal with the problem of bearing faults. For the problem of gearbox breakdowns, a method based on SVM and Wavelet transform was studied in [72]. In [73], a novel fault diagnosis method was proposed based on ANN and Wavelet transform to solve the problem of identifying single and double components in a generator.

In Chapter 3 and Chapter 4, two hybrid fault diagnosis methods based on ANN and the model of the system are presented. However, as the main contribution of this thesis is based on ANN and machine learning methods, the comprehensive explanation will be given, in section 2.3.

2.3 Machine Learning

Machine Learning (ML) is growing rapidly in the field of data-driven methods as a powerful means to handle a large amount of data in a complicated system. In this part, we will provide an overview of the concept of Artificial Neural Networks, its applications in fault diagnosis in wind turbine systems, deep learning and specifically Convolutional Neural Networks.

2.3.1 Artificial Neural Networks

Artificial Neural Networks (ANN) is widely applicable in the area of mapping nonlinear functions and complex systems. It is worth to mention that, multilayer neural networks as one of the most effective computational intelligence (CI) approaches, has received exceptional attention due to its ability as universal approximator [74], in identification and modelling of industrial systems.

System identification in grey-box modelling is very influential in understanding the behaviour of the system in tackling of the unpredicted faults. One of the capable tools in modelling and identification of the nonlinear functions is multilayer perceptron (MLP) neural networks. The schematic of a fully connected MLP is presented in Fig. 2.6. As it can be seen in this figure, a typical MLP contains an input layer, which can be the states of the system ($z_i : i = 1, ..., n$), a hidden layer and an output layer ($t_i : i = 1, ..., p$), that designed to be an approximation of the system output. The $\hat{V}$ and $\hat{W}$ are weight matrices for the hidden and output layers. This network is called fully connected, since each neuron in hidden and output layers is connected to every neuron in the previous layer.

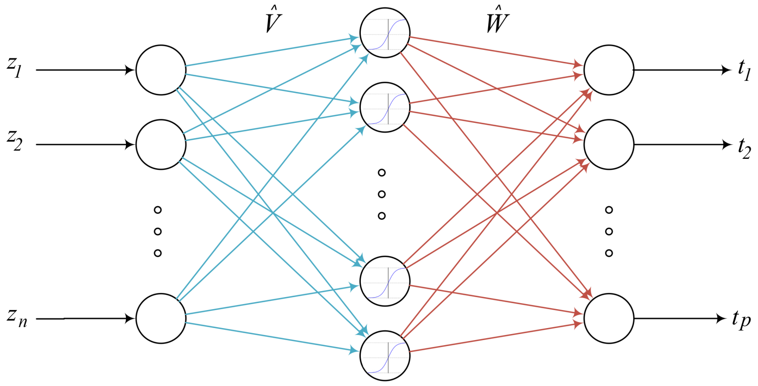


FIGURE 2.6: A Three-layer Fully Connected Neural Network.

One of the most widespread approaches to update the weights in an MLP is *Back Propagation Neural Networks* (BPNN) algorithm. This method is composed of three main steps: initializing, feed-forward, and backward. The last two steps recurred in each iteration until the error is less than a predefined value.

In *initializing*, all the wights are initialized by unsupervised approach, e.g. random quantities. In *feed-forward*, input vector $Z = [z_1, z_2, ..., z_n]^T$ goes through the network as feed-forward and with the previous weight matrices, the output vector $T = [t_1, t_2, ..., t_m]^T$ is achieved follows:

$$T = \hat{W}(\sigma(\hat{V}Z)), \tag{2.3.1}$$

In the last part of feed-forward step, the output of the network is compared to the output of the system and the error vector is obtained.

In *backward* step, the error vector is applied to train the weights $\hat{W}$ and $\hat{V}$ using an updating rule. The main task of each BPNN designing is to derive an equation to update the training weights in each iteration based on the defined cost function.

2.3.2 Applications of ANN in Fault Diagnosis of Wind Turbines

There are many researches on using ANN with the application of wind turbine systems in the literature. In [75], power curve modelling was studied in a wind turbine benchmark and an ANN method was suggested for parameters estimation. In [76], by having standard deviation, the previous output power, and the wind speed average, an ANN nonlinear model of the wind turbine was developed in order to estimate the output power in future. [77] investigated using the experimental results to train ANN in order to confront the problem of parameters finding of a counter-rotation wind turbine. In [78], an ANN concept was studied in order to estimate the imbalance faults in a wind turbine. In [28], a data-driven residual-based fault detection is investigated by using ANN in wind turbine benchmark. The robust fault estimation to prevent environmental disturbances in a wind turbine system is proposed in [79, 80] by applying ANN with linear matrix inequalities approach. Despite every useful aspect of ANN and also ML, some knowledge of the system should be covered in order to design the ML/ANN techniques effectively.

2.3.3 Deep Learning

One of the subcategories of ML, which is highly developed in recent years for coping the problem of black-box models and large systems is deep learning (DL). DL can deal with raw data in a vast amount and provide feature extraction automatically [81, 82]. Moreover, DL can guarantee an assuring and very effective solution in fault diagnosis [83]. Many DL methods have been developed for fault diagnosis in literature review, such as sparse filtering [84], convolutional deep belief networks [85], and convolutional neural network [86].

2.3.3.1 Convolutional neural network

CNN is categorised among the multi-layer perceptrons ANN, which was first developed to model the behaviour of visual cortex [87]. CNN is a potent method in both classification and regression problems. However, it is somehow different from a fully connected neural network (FCNN) in which each neuron is connected to all neurons in the previous layer. However, in CNN, apart from a layer of FCNN, there are other layers in which each neuron is connected to a small part of neurons in the previous layer [88].

In a typical CNN, one can see three main layers: 1) convolutional layer, 2) pooling layer and 3) fully connected layer [88]. In each convolutional layer, there are plenty of kernels, which behave as filters to extract the features of the input data. In pooling layer, down-sampling is applied in order to reduce the dimension of features from the previous layer. In a fully connected layer, an FCNN is trained in order to classify the scores and data in supervised learning [88]. In Fig. 2.7, a simple form of CNN consists of three layers is depicted.

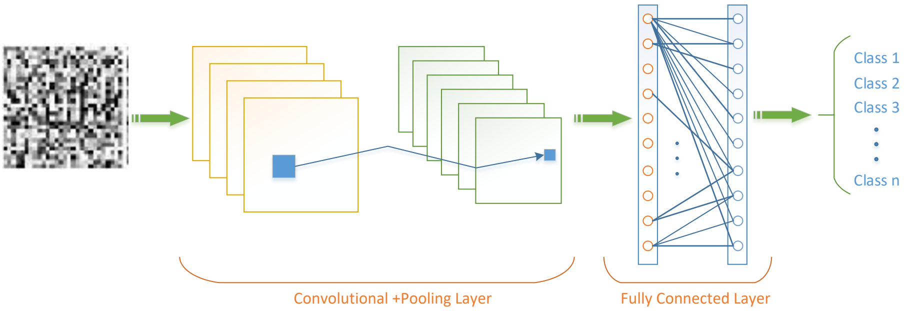


FIGURE 2.7: A simplified CNN consists of three main layers, used for classification.

Before continuing to the application of CNN, there are some primary concepts in this method, which are worth explaining.

Kernel: They are defined as the filters in the convolutional layer, which applied to a sample matrix of input data and produce a convolved result of it. For instance, if there are kernels of the size $K \times K$, and there is an input data of $N \times N$, the output matrix of the convolved result of kernel and the image will be $((N-K)/S+1) \times ((N-K)/S+1)$ [15], in which S is the stride of moving of kernel in input data. The visual representation of this concept can be seen in Fig. 2.8.

ReLU: This function is used in the output of the convolutional layer as an activation function of each neuron. ReLU can be defined as [15]:

$$Z = max(0, T), \tag{2.3.2}$$

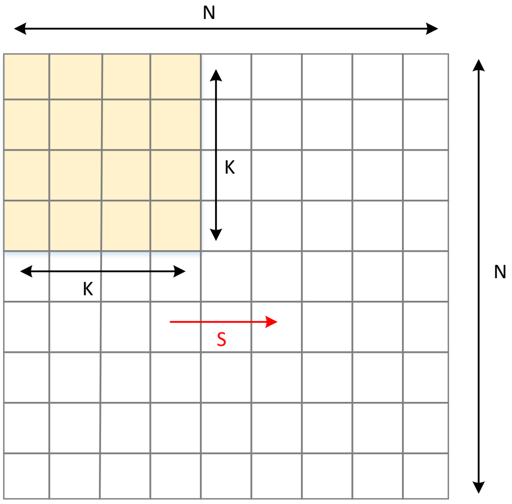


FIGURE 2.8: The kernel and its stride in convolutional layer.

in which, Z is the output of ReLU, and T is the input of it. The advantage of ReLU compared to the sigmoid function is that it is faster and requires less computational efforts.

Max Pooling: This layer is categorised in the pooling layer, which is used to downsample the previous layer. In max pooling, in a $M \times M$ sample, the maximum cell is chosen and routed to the next layer. This layer helps the network omit the weakest feature and express the strongest one. The simple form of max pooling layer of 2 × 2 is illustrated in Fig. 2.9.

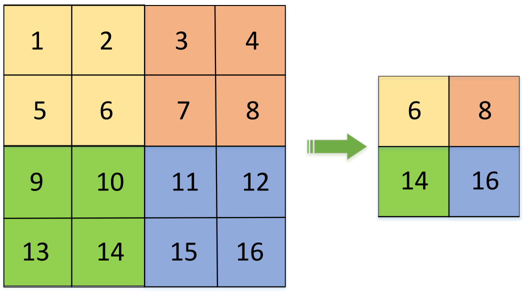


FIGURE 2.9: The max pooling layer.

Softmax Layer: This layer is brought to CNN as an activation function for the layer of FCNN. This layer can help the network choose a preferred class more smoothly and accurately [15]. The jth output of this layer can be found as follows:

$$\sigma(j) = \frac{\exp(z_j)}{\Sigma_{p=1}^{P} exp(z_p))} \tag{2.3.3}$$

in which, z_j is the output of jth neuron of FCNN, and P is the number of neurons in the last layer of FCNN.

2.3.3.2 Applications of CNN

Convolutional neural network (CNN), as one of the most capable of DL methods, has also been growing in recent years. It has shown great advantages in fault diagnosis in the area of 2-D format images in biology [89] and also in mapping [90]. In addition, there are fruitful researches in the area of classification for the problem of bearing faults [91–93]. CNN can also be helpful for fault diagnosis in time-series format data [94]. In [88], time-series format data is converted to 2-D images. Then, a CNN structure is developed in order to have a fault diagnosis method in the application of motor bearing.

2.4 Summary

In this chapter, an introduction to ANN and CNN was explained as well as a literature review on fault diagnosis and its categories. Since the studied benchmark in this thesis is a 4.8 MW wind turbine, the model and the dynamics of the benchmark was also investigated. As in many industrial systems, wind turbines included, there is a considerable amount of data stored via SCADA systems, it is very reasonable to use this data in order to investigate faults. Therefore, data-driven fault diagnosis can be a reasonable solution to this problem.

In the next chapters, we are going to study some fault diagnosis approaches in order to cope with the critical problems in big industrial models, such as actuator faults, sensor faults, noises, disturbances and unmodelled parameters. Then, the proposed methods are applied to wind turbine benchmark to be evaluated in performance.

Chapter 3

Grey-box Model Identification and Fault Detection Using Artificial Neural Networks

In this chapter, a model identification method based on ANN for wind turbine dynamics is studied. The goal is to obtain a neural network based model to be substituted for the real dynamics of the wind turbine due to the lack of complete knowledge about the dynamics and the existence of nonlinearities.

Due to the fact that the wind turbine has a nonlinear dynamics with partially measured states, ANN cannot be applied directly. The reason is that two-layered ANN uses all states and the matrices of the systems to calculate the output. To cope with this problem, in section 3.1, first, a Luenberger observer is designed to estimate the states (both measured and unmeasured ones), and then, for the nonlinear part, a multi-input multi-output (MIMO) backpropagation neural-network-based observer is proposed. By having an ANN model as the reference, a fault detection method is studied based on the residual of the system. In section 3.3, this algorithm is evaluated in simulation on the wind turbine benchmark, and the results approve satisfactory performance of the proposed approach. The main contribution in this chapter, is to define the problem in the way that can be observed with neural network observer, and to implement a fault detection technique to identify even small anomalies.

3.1 Model Identification and Fault Detection method

3.1.1 Model Identification

Consider a nonlinear MIMO system which has available states as follows:

$$\begin{aligned} \dot{x}(t) &= f(x,u), \\ y(t) &= Cx(t), \end{aligned} \tag{3.1.1}$$

in which the $x \in R^n$ is the state vector of the system, $u \in R^m$ is the input signal, $y \in R^p$ and f is an unknown nonlinear function. The main goal is to design a model identification system to minimize the output error, defined as follows:

$$\widetilde{x} = x - \hat{x}, \tag{3.1.2}$$

in which $\hat{x}$ is the estimated state vector and $\widetilde{x}$ is the estimation error. In multilayer neural network identification, all states of the system should be available. However, in some cases, similar to wind turbines (to be explained in section V), not all states are measurable. In this case, it would better to first design an observer to estimate the unmeasurable states and after that, train an ANN to identify the model and make the system stable by decreasing the absolute value of error in (3.1.2).

In order to achieve this goal, two assumptions are made: the first one is that the nonlinear system is observable and the second one is that the states of the system are bounded in L_∞ [95].

Now, by adding and subtracting the term $\bar{A}x$, system (3.1.1) becomes

$$\begin{aligned} \dot{x}(t) &= \bar{A}x + g(x,u), \\ y(t) &= Cx(t), \end{aligned} \tag{3.1.3}$$

where $g(x,u) = f(x,u) - \bar{A}x$, $\bar{A}$ is a Hurwitz matrix and the pair of $(C,\bar{A})$ is observable. An important factor to design a neuro-observer is to design a conventional observer to estimate the states and train a neural network to identify the nonlinearity. By applying a Luenberger observer [96], the observer model of the system (3.1.3) can be defined as follows:

$$\begin{aligned} \dot{\hat{x}}(t) &= \bar{A}\hat{x} + \hat{g}(\hat{x},u) + G(y - C\hat{x}), \\ \hat{y}(t) &= C\hat{x}(t), \end{aligned} \tag{3.1.4}$$

where $G \in R^{n\times m}$ is selected so that $\bar{A} - GC$ is Hurwitz. As far as establishing the observability criteria of pair $(C,\bar{A})$, the Luenberger gain is feasible [96]. The block diagram of the observer can be seen in Fig. 3.1. In this figure, $M(s) = (sI - \bar{A})^{-1}$. As it was discussed in section 2.3.1, a MLP neural network has the capability of identifying the nonlinear function $g(x,u)$ due to the universal approximator theorem. Therefore, the below equation can be written:

$$g(x,u) = W\sigma(V\bar{x}) + \epsilon(x) \tag{3.1.5}$$

As explained before, W and V are weight matrices, $\bar{x} = [x \quad u]^T$ is the input of the neural network and $\epsilon(x)$ is the bounded approximation error. Therefore, the function g can be estimated by

$$\hat{g}(\hat{x}, u) = \hat{W}\sigma(\hat{V}\hat{\bar{x}}) \tag{3.1.6}$$

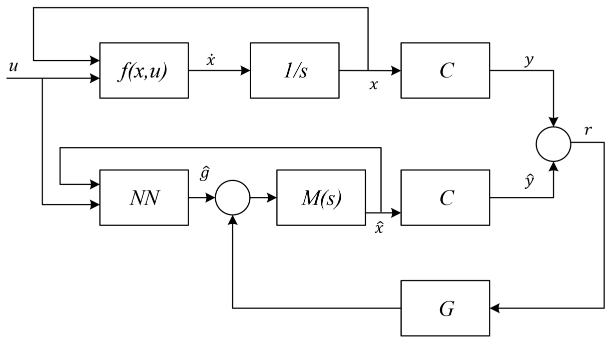


FIGURE 3.1: Block diagram of neural network observer.

The proposed observer is then given by:

$$\begin{aligned} \dot{\hat{x}}(t) &= \bar{A}\hat{x} + \hat{W}\sigma(\hat{V}\hat{\bar{x}}) + G(y - C\hat{x}), \\ \hat{y}(t) &= C\hat{x}(t), \end{aligned} \tag{3.1.7}$$

From (3.1.2) and (3.1.7) one has

$$\begin{aligned} \dot{\widetilde{x}}(t) &= \bar{A}x + W\sigma(V\bar{x}) - \bar{A}\hat{x} - \hat{W}\sigma(\hat{V}\hat{\bar{x}}) \\ &\quad - G(Cx - C\hat{x}) + \epsilon(x), \\ \widetilde{y}(t) &= C\widetilde{x}(t), \end{aligned} \tag{3.1.8}$$

where $\widetilde{y}(t) = y - \hat{y}$. By adding $W\sigma(\hat{V}\hat{\bar{x}})$ to and subtracting from (3.1.8), the error dynamics can be written as below:

$$\begin{aligned} \dot{\widetilde{x}}(t) &= A_c\widetilde{x} + \tilde{W}\sigma(\hat{V}\hat{\bar{x}}) + w(t) \\ \widetilde{y}(t) &= C\widetilde{x}(t), \end{aligned} \tag{3.1.9}$$

where $A_c = \bar{A} - GC$, $\tilde{W} = W - \hat{W}$, and $w(t) = W(\sigma(V\bar{x}) - \sigma(\hat{V}\hat{\bar{x}})) + \epsilon(x)$ is a bounded disturbance error.

After introducing the structure of the neural network observer, the next step is defining an updating rule to train W and V.

Theorem 1: Consider the dynamical model of (3.1.1) and observer of (3.1.8). If the network weights are updated as

$$\begin{aligned} \dot{\hat{W}} &= -\eta_1(\widetilde{y}^T C A_c^{-1})^T(\sigma(\hat{V}\hat{\bar{x}}))^T - \rho_1||\widetilde{y}||\hat{W}, \\ \dot{\hat{V}} &= -\eta_2(\widetilde{y}^T C A_c^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\bar{x}})))^T sgn(\hat{\bar{x}})^T \\ &\quad - \rho_2||\widetilde{y}||\hat{V}, \end{aligned} \tag{3.1.10}$$

where $\Lambda(\hat{V}\hat{\bar{x}}) = diag\left\{\sigma_i^2(\hat{V}_i\hat{\bar{x}})\right\}$, and $i = 1, 2, ..., m$, then, $\widetilde{x}, \tilde{W}, \widetilde{V}, \widetilde{y} \in L_\infty$ are the estimation error, training weights error, and output error which are all bounded. η_1 and η_2 are positive learning rate and ρ_1 and ρ_2 are small positive numbers. The proof of this theorem can be found in [95].

3.1.2 Fault Detection

After training the ANN with the updating rule of (3.1.10), the model can be used as a reference model in order to detect the occurred faults in the system by using the residual as follows:

$$r(t) = y - \hat{y} \tag{3.1.11}$$

By investigating the above equaion, if $||r_t|| > \lambda_T$ (λ_T is a pre-selected threshold value), it indicates a fault happening. Otherwise, the system is healthy.

3.2 Validating on Wind Turbine

The first step to simulate such a problem is to investigate the Luenberger gain in equation (3.1.4). In order to do that, $\bar{A}$ is considered as:

$$\bar{A} = \begin{bmatrix} -10 & 0 & 4 & 0 & 3 & 0 \\ 5 & -12 & 0 & 0 & 0 & 1 \\ 10 & 2 & -20 & 3 & 4 & 5 \\ 3 & 0 & 0 & -21 & 0 & 0 \\ 0 & 2 & 0 & 0 & -5 & 0 \\ 6 & 0 & 8 & 1 & 0 & -5 \end{bmatrix},$$

There are infinite values to address $\bar{A}$. One of the easy way to choose this matrix is putting big negative values in the main diagonal of the matrix and then calculate the other values based on them. The important criteria in choosing this matrix is that it should be Hurwitz and pair of $(C, \bar{A})$ is observable. By choosing G in the form of (3.2.1), the matrix $A_c = \bar{A} - GC$ becomes Hurwitz, too.

$$G = \begin{bmatrix} -8 & 0 & 3 & 0 \\ 5 & -8 & 0 & 1 \\ 10 & 2 & 4 & 5 \\ 3 & 0 & 0 & 0 \\ 0 & 2 & 5 & 0 \\ 6 & 0 & 0 & 10 \end{bmatrix}, \tag{3.2.1}$$

Choosing η_1 and η_2 is very challenging. In fact, choosing any hyper-parameters in any neural network can be a difficult and consuming task and mainly they are chosen based on try and error and experts knowledge. It is recommended to start with small values and change them step by step. In this simulation, by choosing

$\eta_1 = 2\times10^7$ and $\eta_2 = 2\times10^{-7}$, the comparative results in Fig. 3.2, 3.3, 3.4, and 3.5 can be obtained. As it is obvious, the output estimations of the neural networks can follow the main system outputs accurately in 4500s. In all four outputs, the estimation errors converge to zero. The details of the related simulation can be found in Appendix A.

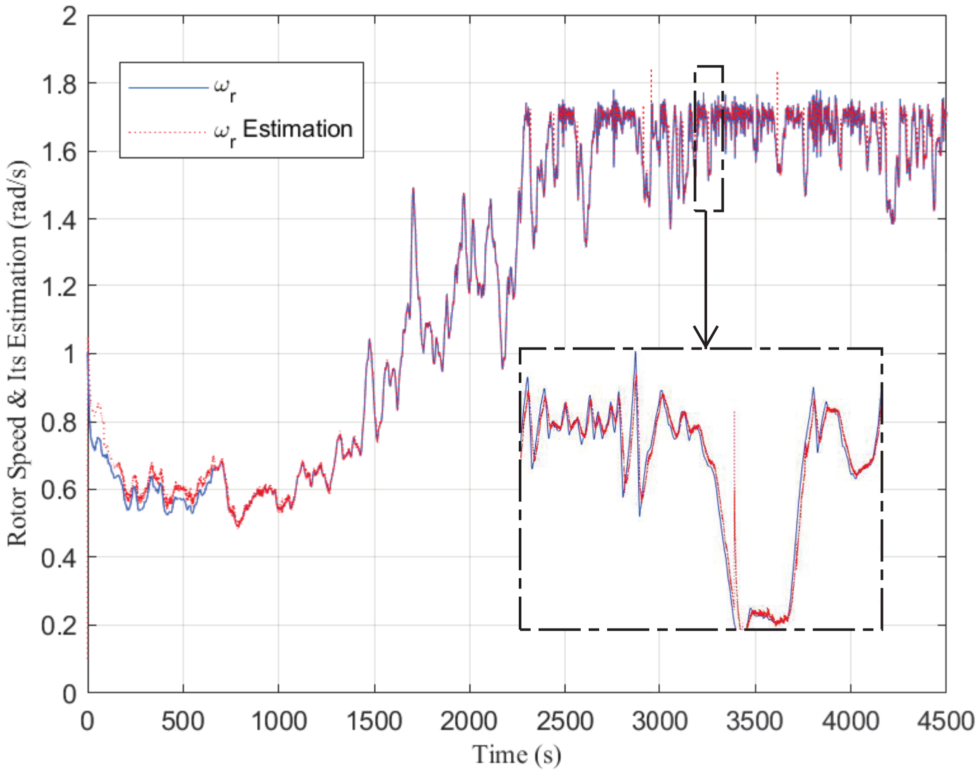


FIGURE 3.2: Rotor speed signal and its estimation.

3.3 Fault Detection for Wind Turbine

In the next step, to study the effectiveness of the ANN reference model in detecting faults, a case study consisting of 2% loss in actuator of generator torque reference is considered. This fault happened between the time 2500 to 3500s. One can see the effect of this fault in the fourth output of the system τ_g in Fig. 3.6. It is not distinguishable by comparing the healthy torque and faulty one. By using the approach of (3.1.11), the residual signal can be obtained and shown in Fig. 3.7.

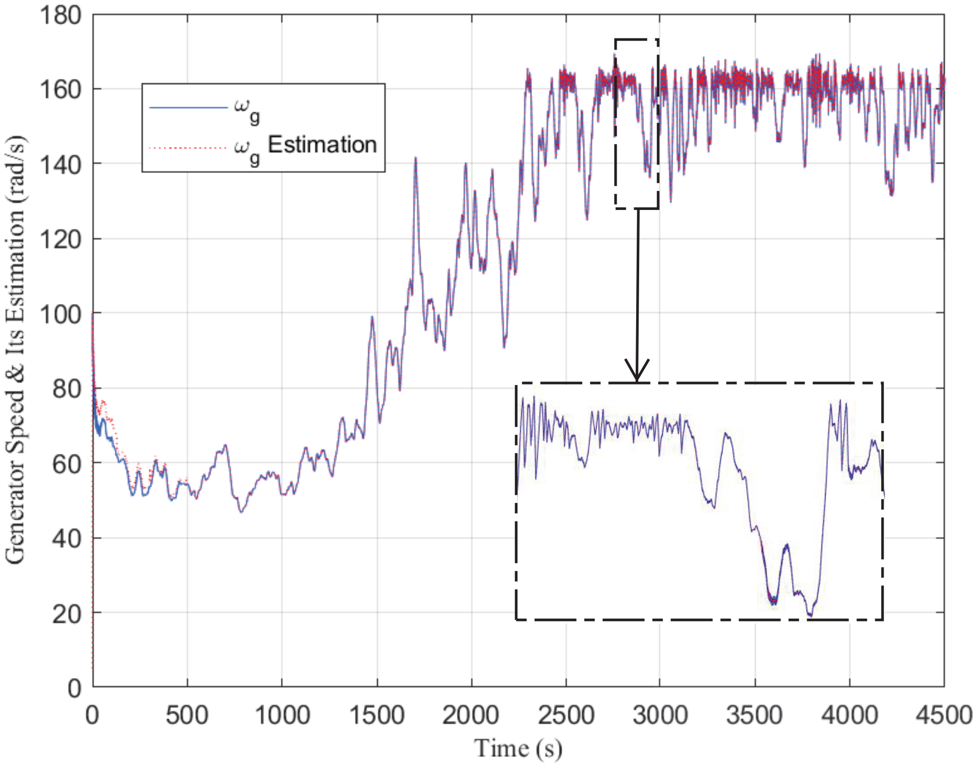


FIGURE 3.3: Generator speed signal and its estimation.

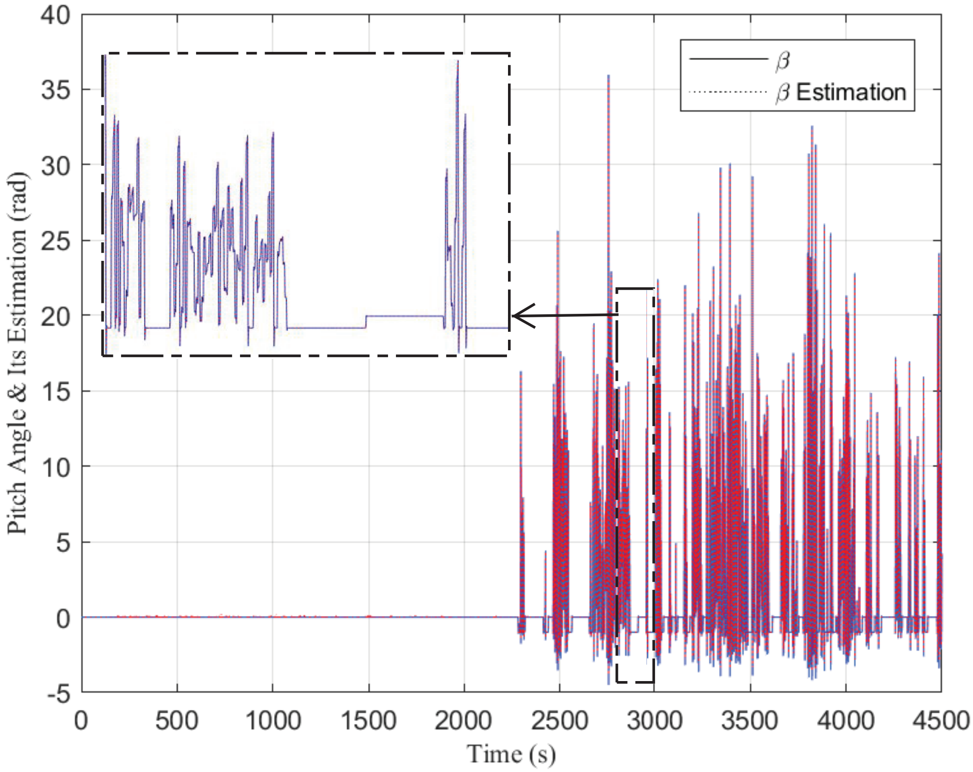


FIGURE 3.4: Pitch angle signal and its estimation.

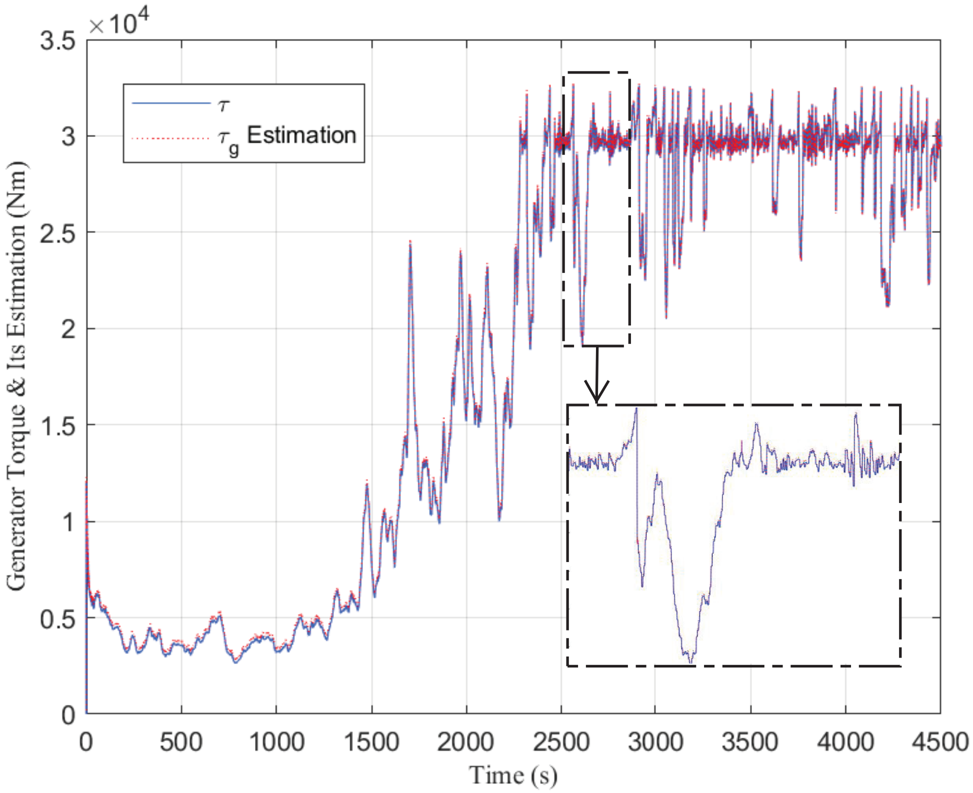


FIGURE 3.5: Generator torque signal and its estimation.

As one can see, the residual can successfully detect the fault even if the percentage of fault is so small.

3.4 Summary and Conclusions

In this chapter, the model identification and fault detection approach based on artificial neural networks has been proposed. As in some systems such as wind turbines, not all the states can be measured, neural networks observer cannot work correctly. Therefore, Luenberger observer jointly with neural networks identification method is proposed to deal with the issue of nonlinearity and being unmeasurable, which is the novelty of this algorithm. Based on the trained model, residual-based fault detection is investigated. The algorithm is simulated on the wind turbine benchmark. A case study is carried out for 2% actuator fault. The results of both sections, identification and fault detection, validate the effectiveness of the proposed algorithm.

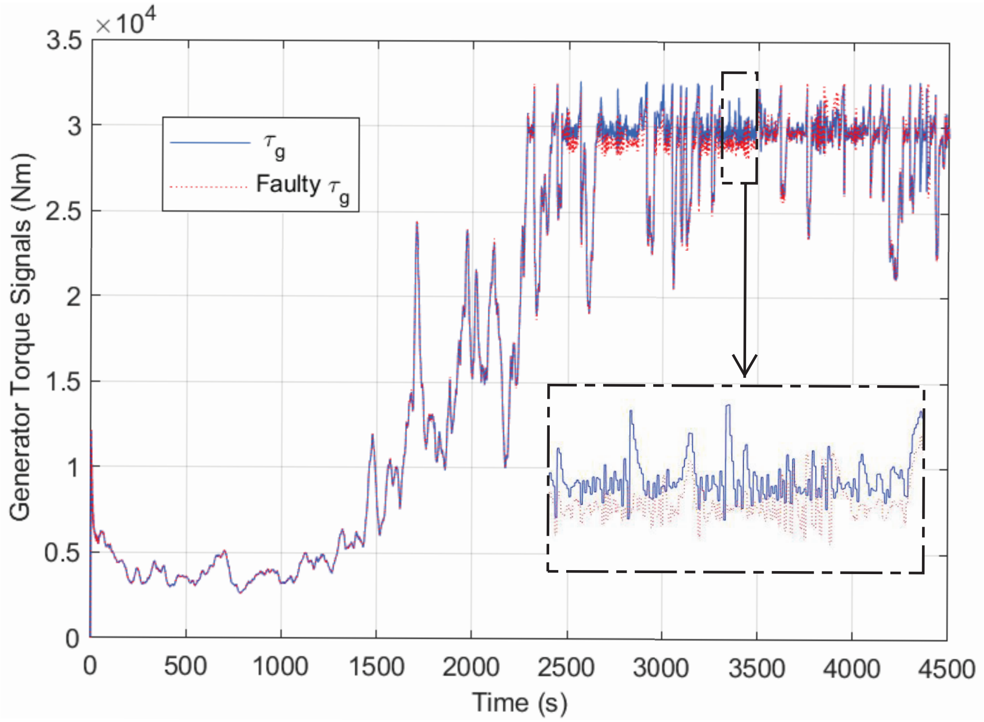


FIGURE 3.6: The healthy and faulty signal of τ_g.

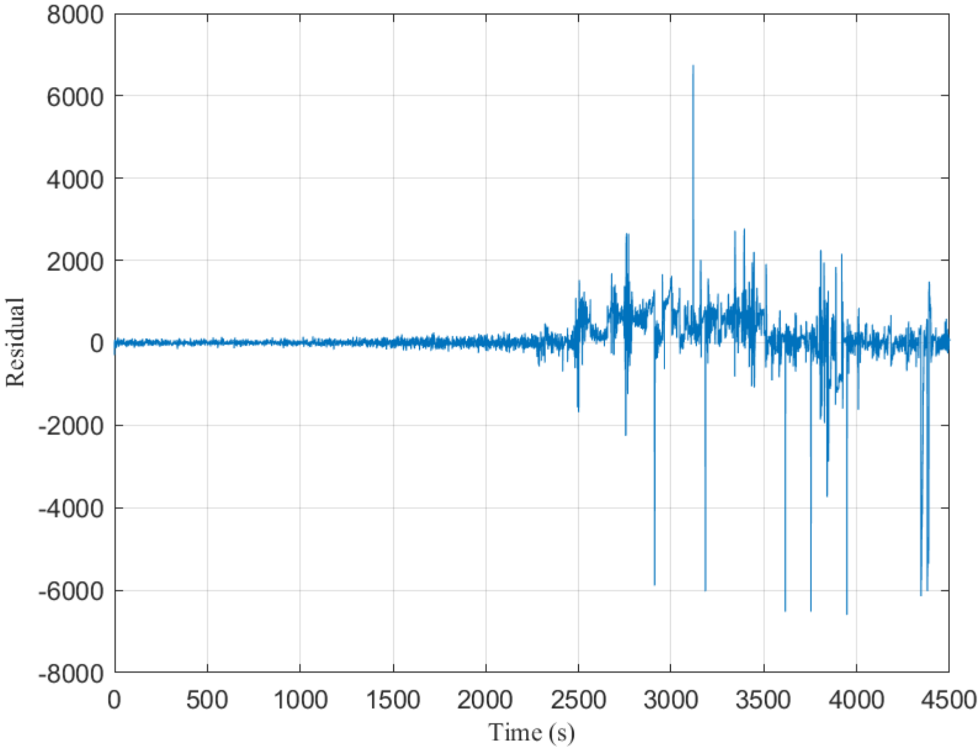


FIGURE 3.7: The norm of residual.

Chapter 4

Robust Neural Network Fault Estimation Approach for Nonlinear Dynamic Systems

In this chapter, a robust fault estimation approach is proposed for multi-input and multi-output nonlinear dynamic systems on the basis of back propagation neural networks. The augmented system approach, input-to-state stability theory, linear matrix inequality optimization, and neural network training/learning are integrated so that a robust simultaneous estimate of the system states and faults are achieved.

Based on the facts that there are always unexpected faults and disturbances in industrial systems, it is very challenging to design an observer to tackle this problem and guarantee the robustness of the system in conjunction with thoroughly estimating the occurred faults.

Augmented system methods achieve many advantages in estimating states and faults simultaneously, among distinct fault estimation approaches. The pioneering works on this topic can be found in [97–99]. Recently in [100], a discrete-time robust fault tolerant control approach is proposed based on linear matrix inequalities (LMI) techniques and augmented system approach in order to achieve input-to-states stability of the system subjected to input disturbances.

In this chapter, two different scenarios are considered: actuator fault estimation and sensor fault estimation. Each of these scenarios needs specific model description and fault diagnosis method. In this chapter, the goal is to achieve a fault estimator, which is robust against unknown inputs. As it is evident, unknown inputs such as modelling defects, perturbations, environmental disturbances and parameters uncertainties can influence the stability of the system. To achieve this goal, an augmented robust LMI optimization is proposed for each scenario based on back propagation neural network (BPNN). The stability of the system is guaranteed via Lyapunov and input-to-state stability criteria.

The main contribution in this chapter is proposing the novel robust fault estimation algorithm to not only cope with nonlinear dynamics, but also can handle the disturbances. The most important part of this novel contribution, is providing the mathematical proof for the stability of the system.

4.1 Neuro-Robust Actuator Fault Estimation

Consider a nonlinear multi-input and multi-output (MIMO) system with consideration of actuator faults and disturbances, as follows:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) + B_f f_a(t) + B_d d(t), \\ y(t) &= Cx(t), \end{aligned} \tag{4.1.1}$$

in which the $x \in R^n$ is the state vector of the system, $u \in R^m$ is the input signal, $y \in R^p$ is the output and $f(x(t), u(t)) \in R^n$ is an unknown nonlinear function. $f_a(t) \in R^m$ is the occurred actuator fault and B_f is the related fault matrix. $d(t)$ is continuously differentiable and bounded disturbance and B_d is the distribution matrix. As it was mentioned earlier in section 3.1.1, in multi-layer ANN mapping, all states of the model should be available. Nevertheless, in some cases, like wind turbines, not all states are measurable. State estimation in a system plays an important role in detecting and diagnosing the faults and monitoring the process more vividly. Based on this fact, it would better to have an augmented system not only to estimate the states but also identify the unanticipated faults at the same time.

In order to obtain this aim, two assumptions are considered: first, the nonlinear model is observable. Second, the states are bounded in L_∞ [95].

Now, by adding and subtracting the term Ax, system (4.1.1) becomes:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + g(x(t), u(t)) + B_f f_a(t) + B_d d(t), \\ y(t) &= Cx(t), \end{aligned} \tag{4.1.2}$$

where $g(x(t), u(t)) = f(x(t), u(t)) - Ax(t)$, A is an arbitrary Hurwitz matrix, which has been chosen in the way that the pair of (C, A) is observable. The main reason to decouple the system into linear and nonlinear blocks is to be able to design a robust observer based on the LMI for the linear part while the nonlinear block error is augmented into the disturbance vector. This vector plays the role of exogenous input in the process of formulating the LMI. By this approach, the nonlinear observability criteria are relaxed into linear observability criteria. More explanation will be brought in Theorem 4.1 later on.

As it is evident that the dynamics of the fault is unknown, there are some methods to deal with this problem. For instance, in [22], it is assumed that the second-order derivative of the occurred fault is zero. However, in this research, this condition will be relaxed by considering the novel equation of (4.1.3) which is correct in all situations instead of previous forms.

$$\dot{f}_a = \dot{f}_a - f_a + f_a. \tag{4.1.3}$$

It is worth to mention that f_a and $\dot{f}_a$ should be continuously differentiable and bounded. By augmenting (4.1.3) and (4.1.2), the model of the system can be written as:

$$\begin{aligned} \dot{X}(t) &= \bar{A}X(t) + G + \bar{B}_d\bar{d}, \\ y &= \bar{C}X(t), \end{aligned} \tag{4.1.4}$$

in which, $\bar{A} = \begin{bmatrix} A & B_f \\ 0 & -I \end{bmatrix}$, $X(t) = [x(t) \quad f_a]^T$, $G = [g(x(t), u) \quad 0]^T$, $\bar{B}_d = \begin{bmatrix} B_d & 0 & 0 \\ 0 & I & I \end{bmatrix}$, $\bar{d} = [d \quad \dot{f}_a \quad f_a]^T$, and $\bar{C} = [C \quad 0]$. The main goal in this approach is to design a model identification system to minimize the augmented state error vector in (4.1.5):

$$\widetilde{X}(t) = X(t) - \hat{X}(t), \tag{4.1.5}$$

where $\hat{X}$ is the estimated state vector and $\widetilde{X}$ is the estimation error vector.

[95] proposed a model to design a neural network observer (NNO) by decoupling systems into linear and nonlinear blocks. By using this model and modifying it by adding faults and disturbances, the NNO model can be seen in Fig. 4.1.

In this model, NNO is designed to estimate the nonlinear block of the model, G, and a robust observer (described in the equation (4.1.6)) is designed to cope with the disturbance and unexpected fault, which are augmented in $\bar{d}$. Moreover,

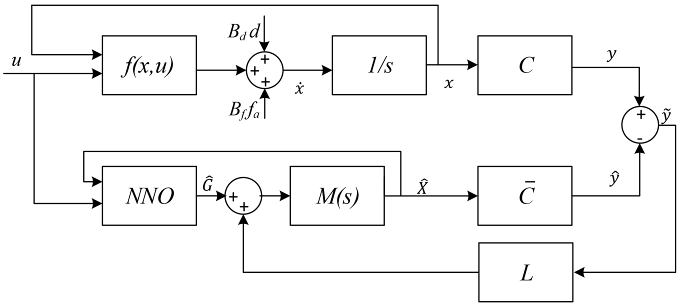


FIGURE 4.1: The scheme of ANN based observer.

$M(s) = (sI - \bar{A})^{-1}$ and $\hat{y}$ is the estimation of y. The upper part is the main model with its inputs, while the lower part is the estimation system. The input and the output of the main model are applied to the estimation system. The whole system is not closed loop since no signal from the estimation part is entering the main model via feedback.

The observer model of the system (4.1.2) can be defined as follows:

$$\begin{aligned} \dot{\hat{X}}(t) &= \bar{A}\hat{X}(t) + \hat{G} + L(y - \bar{C}\hat{X}(t)), \\ \hat{y}(t) &= \bar{C}\hat{X}(t), \end{aligned} \tag{4.1.6}$$

where $L \in R^{(n+m)\times p}$ is selected so that the augmented system becomes robust against the disturbance term of $\bar{d}$. Moreover, $\hat{G}$ is the output of the neural networks of NNO and the estimation of G. As it was discussed in (2.3.1), $\hat{G}$ can be written as:

$$\hat{G} = \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t)), \tag{4.1.7}$$

in which, $\hat{\hat{X}}(t) = [\hat{X}(t) \quad u]^T$ is the input of NNO.

By substituting (4.1.7) into (4.1.6), the following observer equation can be obtained:

$$\begin{aligned} \dot{\hat{X}}(t) &= (\bar{A} - L\bar{C})\hat{X}(t) + \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t)) + L\bar{C}X(t), \\ \hat{y}(t) &= \bar{C}\hat{X}(t), \end{aligned} \tag{4.1.8}$$

Procedure 1. For achieving the goal, the following procedure is applied:

i By having the idea of BPNN that was introduced in section II, in feed-forward step after having the output of NNO, a robust observer gain is designed via LMI in order to reduce the influence of the unmodeled dynamics and disturbances. In addition, the stability of the system is guaranteed through Lyapunov function (to be addressed in Theorem 4.1). The output error is calculated at the end of this step.

ii In the next step, the backward step of BPNN is applied, and the updating rules for the weights of the NNO are obtained (to be presented in Theorem 4.2) via the predefined cost function and the output error, which is assessed in the earlier step.

By considering the error function of (4.1.5), the error dynamics can be written as follows:

$$\begin{aligned}\dot{\tilde{X}}(t) =& \dot{X}(t) - \dot{\hat{X}}(t) \\ =& \bar{A}X(t) - \bar{A}\hat{X}(t) + G - \hat{W}\sigma(\hat{V}\hat{\bar{X}}(t)) \\ & - L(\bar{C}X(t) - \bar{C}\hat{X}(t)) + \bar{B}_d\bar{d}.\end{aligned} \tag{4.1.9}$$

By substituting $\tilde{G} = G - \hat{W}\sigma(\hat{V}\hat{\bar{X}}(t))$, which is the error of nonlinear function estimation, equation (4.1.9) is given as below:

$$\dot{\tilde{X}}(t) = \tilde{A}\tilde{X}(t) + \tilde{G} + \bar{B}_d\bar{d}, \tag{4.1.10}$$

in which $\tilde{A} = \bar{A} - L\bar{C}$. (4.1.10) can be further simplified as:

$$\dot{\tilde{X}}(t) = \tilde{A}\tilde{X}(t) + \tilde{B}_1\tilde{F}, \tag{4.1.11}$$

in which $\tilde{F} = [\tilde{G} \quad \bar{d}]^T$. Now, the system has the state vector of $\widetilde{X}(t)$ and the exogenous input of $\tilde{F}$.

Lemma 1 [101]: Consider $f(x,u)$ is continuously differential function and globally Lipschitz in (x,u). If $\dot{x} = f(x,0)$ has a globally exponentially stable equilibrium point at the origin, then the system $\dot{x} = f(x,u)$ is input-to-state stable.

Before presenting the main result of Theorem 4.1, we firstly give the definitions of the robust performance index and associated Hamiltonian function as follows [102]:

$$J_{11} = ||\widetilde{X}(t)||_{Tf} - \gamma^2 ||\tilde{F}||_{Tf} < 0. \tag{4.1.12}$$

The associated *Hamiltonian* function is defined as:

$$J_{12} = \int_0^{Tf} (\frac{dV(\widetilde{X}(t))}{dt} + \widetilde{X}^T(t)\widetilde{X}(t) - \gamma^2 \widetilde{F}^T \widetilde{F})\, dt \tag{4.1.13}$$

Theorem 4.1. *There exists robust observer (4.1.8) for the augmented system of (4.1.4), so that: (i) the estimation error dynamics in (4.1.11) is input-to-state stable; (ii) the estimation error satisfy the robust performance index (4.1.12), if there are a positive definite matrix P and a matrix Q so that*

$$\begin{bmatrix} P\bar{A} + \bar{A}^T P - Q\bar{C} - \bar{C}^T Q^T + I & P\tilde{B}_1 \\ \tilde{B}_1^T P & -\gamma^2 I \end{bmatrix} < 0 \tag{4.1.14}$$

in which $\tilde{B}_1 = [I \quad \bar{B}_d]$. Then, the observer gain is calculated as $L = P^{-1}Q$.

Proof: The proof of this theorem is divided into two parts: (i) input-to-state stability, and (ii) robust performance index.

(i) **Proof of the input-to-state stability**. For any $\tilde{X}_1(t)$, $\tilde{X}_2(t)$, $\tilde{F}_1$, and $\tilde{F}_2$, we can have:

$$\begin{aligned} &|h(\tilde{X}_1(t), \tilde{F}_1) - h(\tilde{X}_2(t), \tilde{F}_2)| \\ &\qquad = |\tilde{A}(\tilde{X}_1 - \tilde{X}_2) + \tilde{B}_1(\tilde{F}_1 - \tilde{F}_2)| \\ &\qquad \leq \alpha|\tilde{X}_1(t) - \tilde{X}_2(t)| + \beta|(\tilde{F}_1 - \tilde{F}_2)|, \end{aligned} \tag{4.1.15}$$

where $\alpha = ||\tilde{A}||$, and $\beta = ||\tilde{B}_1||$. As a result, $h(\tilde{X}(t), \tilde{F})$ is globally Lipschitz in $(\tilde{X}(t), \tilde{F})$. It is evident that $h(\tilde{X}(t), \tilde{F})$ is continuously differentiable.

Since the matrix $\tilde{A}$ is Hurwitz, the unforced system $\dot{\tilde{X}}(t) = \tilde{A}\tilde{X}(t) = h(\tilde{X}(t), 0)$ is globally exponentially stable at the origin. Therefore, by using Lemma 1, we can conclude that the estimation error dynamics $\dot{\tilde{X}}(t) = h(\tilde{X}(t), \tilde{F})$ is input-to-state stable.

(i) **Proof of robust performance index**. One can take the Lyapunov candidate of $V(\tilde{X})$ as follows:

$$V(\tilde{X}(t)) = \tilde{X}(t)^T P \tilde{X}(t), \tag{4.1.16}$$

in which P is positive definite symmetric matrix. By having derivative of (4.1.16), one can have:

$$\begin{aligned} \dot{V}(\tilde{X}(t)) =& \tilde{X}(t)^T P \dot{\tilde{X}}(t) + \dot{\tilde{X}}(t)^T P \tilde{X}(t) \\ =& \tilde{X}(t)^T P(\tilde{A}\tilde{X}(t) + \tilde{B}_1\tilde{F}) \\ &+ (\tilde{A}\tilde{X}(t) + \tilde{B}_1\tilde{F})^T P \tilde{X}(t). \end{aligned} \tag{4.1.17}$$

Therefore, by substituting (4.1.17) into (4.1.13), J_{12} can be obtained as:

$$\begin{aligned} J_{12} = \int_0^{Tf} & (\tilde{X}(t)^T P(\tilde{A}\tilde{X}(t) + \tilde{B}_1\tilde{F}) \\ & + (\tilde{A}\tilde{X}(t) + \tilde{B}_1\tilde{F})^T P\tilde{X}(t) \\ & + \widetilde{X}^T(t)\widetilde{X}(t) - \gamma^2\widetilde{F}^T\widetilde{F})\, dt. \end{aligned} \tag{4.1.18}$$

By extracting the vector block of $Z = [\tilde{X}(t) \quad \tilde{F}]^T$ and using *Schur Complement*, (4.1.18) can be rewritten as:

$$J_{12} = \int_0^{Tf} Z^T R Z \; dt, \tag{4.1.19}$$

in which,

$$R = \begin{bmatrix} P\tilde{A} + \tilde{A}^T P + I & P\tilde{B}_1 \\ \tilde{B}_1^T P & -\gamma^2 I \end{bmatrix} \tag{4.1.20}$$

Consequently, for having $J_{12} < 0$, R should be negative definite. By substituting $\tilde{A} = \bar{A} - L\bar{C}$, $R < 0$ is equivalent to the following LMI:

$$\begin{bmatrix} P\bar{A} + \bar{A}^T P - Q\bar{C} - \bar{C}^T Q^T + I & P\tilde{B}_1 \\ \tilde{B}_1^T P & -\gamma^2 I \end{bmatrix} < 0$$

where $Q = PL$. As a result, the condition (4.1.14) implies $R < 0$, then $J_{12} < 0$. It is noticed that $V(\tilde{X}(t)) \geq 0$, and from (4.1.13) and $J_{12} < 0$, the robust performance index (4.1.12) can thus be obtained. Therefore, the gain matrix of $L = P^{-1}Q$ can be calculated $\therefore$.

Now, by calculating the output error vector of $\widetilde{y} = y - \hat{y}$, the feed-forward step of designing is finished. The next step is to design a neural network and propose an updating rule for weight matrices by using the output error $\widetilde{y}$.

As it was discussed in [103], based on the *Universal Approximator* theorem, a multi-layer neural network (MLP) with three layers and updating rule of BPNN has the capability of identifying any nonlinear function. Therefore, as it was brought earlier, for estimating nonlinear function of G, equation (4.1.7) with the estimated weight matrices of $\hat{W}$ and $\hat{V}$ is considered. Moreover, the basic updating rules in BPNN are as follows:

$$\begin{aligned} \dot{\hat{W}} &= -\eta_1(\frac{\partial J_2}{\partial \hat{W}}) - \rho_1||\widetilde{y}||\hat{W}, \\ \dot{\hat{V}} &= -\eta_2(\frac{\partial J_2}{\partial \hat{V}}) - \rho_2||\widetilde{y}||\hat{V}, \end{aligned} \tag{4.1.21}$$

in which, J_2 is the cost function of the system that should be minimized. For finding an updating rule to minimize the cost function, the following theorem is discussed.

Theorem 4.2. *Given the nonlinear model of (4.1.1) and the observer scheme of Figure 4.1 with observer equation of (4.1.8). If the ANN weights are trained as*

$$\begin{aligned} \dot{\hat{W}} =& - \eta_1(\widetilde{y}^T C\tilde{A}^{-1})^T(\sigma(\hat{V}\hat{\bar{X}}))^T - \rho_1||\widetilde{y}||\hat{W}, \\ \dot{\hat{V}} =& - \eta_2(\widetilde{y}^T C\tilde{A}^{-1}\hat{W}(I - \Lambda(\hat{V}\hat{\bar{X}})))^T\hat{\bar{X}}^T \\ & - \rho_2||\widetilde{y}||\hat{V}, \end{aligned} \tag{4.1.22}$$

where $\Lambda(\hat{V}\hat{\bar{x}}) = diag\left\{\sigma_i^2(\hat{V}_i\hat{\bar{X}})\right\}$, *and* $i = 1, 2, ..., m$, *then,* $\widetilde{X}, \tilde{W}, \widetilde{V}, \widetilde{y} \in L_\infty$ *are the estimation error, the weights error, and the output error which are all bounded.* η_1 *and* η_2 *are positive learning rate and* ρ_1 *and* ρ_2 *are small positive numbers.*

Proof: By defining cost function $J_2 = \frac{1}{2}(\widetilde{y}^T\widetilde{y})$ and using the basic updating rule for BPNN that is introduced in (4.1.21), it is obvious that the only terms that should be changed into simpler terms are $\frac{\partial J_2}{\partial \hat{W}}$ and $\frac{\partial J_2}{\partial \hat{V}}$. In order to solve this issue, two terms are introduced as below:

$$\begin{aligned} net_{\hat{W}} &= \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t)), \\ net_{\hat{V}} &= \hat{V}\hat{\hat{X}}(t). \end{aligned} \tag{4.1.23}$$

Now, $\dfrac{\partial J_2}{\partial \hat{W}}$ and $\dfrac{\partial J_2}{\partial \hat{V}}$ can be decoupled into four partial derivatives of:

$$\begin{aligned} \frac{\partial J_2}{\partial \hat{W}} &= \frac{\partial J_2}{\partial \widetilde{y}} \times \frac{\partial \widetilde{y}}{\partial \hat{X}} \times \frac{\partial \hat{X}}{\partial net_{\hat{W}}} \times \frac{\partial net_{\hat{W}}}{\partial \hat{W}} \\ \frac{\partial J_2}{\partial \hat{V}} &= \frac{\partial J_2}{\partial \widetilde{y}} \times \frac{\partial \widetilde{y}}{\partial \hat{X}} \times \frac{\partial \hat{X}}{\partial net_{\hat{V}}} \times \frac{\partial net_{\hat{V}}}{\partial \hat{V}} \end{aligned} \tag{4.1.24}$$

By using the cost function equation, one can get:

$$\frac{\partial J_2}{\partial \widetilde{y}} = \widetilde{y}^T. \tag{4.1.25}$$

By having $\widetilde{y} = \bar{C}(X(t) - \hat{X}(t))$, it can be obtained that:

$$\frac{\partial \widetilde{y}}{\partial \hat{X}} = -\bar{C}. \tag{4.1.26}$$

For the third term of each equation in (4.1.24), by considering (4.1.8) and (4.1.23), following equations can be achieved:

$$\begin{aligned} \frac{\partial \dot{\hat{X}}}{\partial net_{\hat{W}}} &= (\bar{A} - L\bar{C})\frac{\partial \hat{X}}{\partial net_{\hat{W}}} + I, \\ \frac{\partial \dot{\hat{X}}}{\partial net_{\hat{V}}} &= (\bar{A} - L\bar{C})\frac{\partial \hat{X}}{\partial net_{\hat{V}}} + \hat{W}^T \frac{\partial(\sigma(\hat{V}\hat{\hat{X}}(t)))}{\partial net_{\hat{V}}}. \end{aligned} \tag{4.1.27}$$

By having partial derivatives of a vector on a vector, and by considering the derivative of sigmoid function, one can have:

$$\begin{aligned}\frac{\partial(\sigma(\hat{V}\hat{\hat{X}}(t)))}{\partial net_{\hat{V}}} &= \frac{\partial(\sigma(\hat{V}\hat{\hat{X}}(t)))}{\partial(\hat{V}\hat{\hat{X}}(t))} \\ &= \begin{bmatrix} 1-\sigma^2(\hat{V}_1\hat{\hat{X}}(t) & & 0 \\ & \ddots & \\ 0 & & 1-\sigma^2(\hat{V}_n\hat{\hat{X}}(t) \end{bmatrix}\end{aligned}$$

in which, $\hat{V}_i \quad i = 1, ..., n$ is the ith row of the weight matrix $\hat{V}$. Therefore, the above equation can be written as belows:

$$\frac{\partial(\sigma(\hat{V}\hat{\hat{X}}(t)))}{\partial net_{\hat{V}}} = 1 - \Lambda(\hat{V}\hat{\hat{X}}(t)), \tag{4.1.28}$$

in which, $\Lambda(\hat{V}\hat{\hat{X}}(t)) = diag[\sigma^2(\hat{V}_i\hat{\hat{X}}(t)] \quad i = 1, ..., n.$

In (4.1.27), static approximation of the gradient can be assumed due to the fact that the network converges relatively fast [95]. Therefore, $\dfrac{\partial\dot{\hat{X}}}{\partial net_{\hat{W}}} = 0$ and $\dfrac{\partial\dot{\hat{X}}}{\partial net_{\hat{V}}} = 0$. Based on this assumption, (4.1.27) can be written as:

$$\begin{aligned} 0 &= (\bar{A} - L\bar{C})\frac{\partial\hat{X}}{\partial net_{\hat{W}}} + I, \\ 0 &= (\bar{A} - L\bar{C})\frac{\partial\hat{X}}{\partial net_{\hat{V}}} + \hat{W}^T\frac{\partial(\sigma(\hat{V}\hat{\hat{X}}(t)))}{\partial net_{\hat{V}}}. \end{aligned}$$

Consequently, by using above equation and (4.1.28), following equations can be obtained:

$$\begin{aligned} \frac{\partial\hat{X}}{\partial net_{\hat{W}}} &= -(\bar{A} - L\bar{C})^{-1}, \\ \frac{\partial\hat{X}}{\partial net_{\hat{V}}} &= -(\bar{A} - L\bar{C})^{-1}\hat{W}^T(1 - \Lambda(\hat{V}\hat{\hat{X}}(t))), \end{aligned} \tag{4.1.29}$$

By considering the definition of (4.1.23), the forth term of equations of (4.1.24) are achieved:

$$\begin{aligned} \frac{\partial net_{\hat{W}}}{\partial \hat{W}} &= \sigma(\hat{V}\hat{\hat{X}}(t)), \\ \frac{\partial net_{\hat{V}}}{\partial \hat{V}} &= \hat{\hat{X}}(t). \end{aligned} \tag{4.1.30}$$

Now, by substituting (4.1.25), (4.1.26), (4.1.29) and (4.1.30) in (4.1.24), the following equation is obtained:

$$\begin{aligned} \frac{\partial J_2}{\partial \hat{W}} &= (\widetilde{y}^T C\tilde{A}^{-1})^T(\sigma(\hat{V}\hat{\hat{X}}))^T \\ \frac{\partial J_2}{\partial \hat{V}} &= (\widetilde{y}^T C\tilde{A}^{-1}\hat{W}(I-\Lambda(\hat{V}\hat{\hat{X}})))^T\hat{\hat{X}}^T \end{aligned} \tag{4.1.31}$$

By replacing (4.1.31) into the updating rule of (4.1.21), the equations of (4.1.22) can be obtained and based on the BPNN approach and universal approximator theorem, the neural network whose weight matrices are updated based on (4.1.22) is stable ∴.

Procedure 2. For designing the robust neural network fault estimator, the following procedure is noted:

i Select the matrix A so that pair of (C, A) is observable.

ii Construct the augmented system in the form of (4.1.11).

iii Solve the LMI (4.1.14) to achieve the matrices P and Q in order to have $L = P^{-1}Q$.

iv Consider a three-layer back propagation neural network with initial random weights.

v Update the weight matrices W and V using (4.1.22).

vi Obtain the augmented state of $\hat{X}$ and compare it to the real value of X.

The flowchart of the algorithm is depicted in Figure 4.2. In this flowchart, T_s is the initial time of simulation and T_{Final} is the end of it.

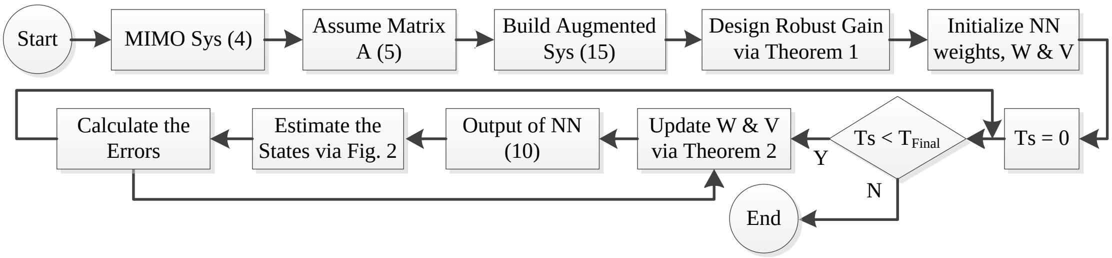


FIGURE 4.2: Flowchart of the Combined Algorithm.

4.1.1 Actuator Fault Estimation for 4.8 MW Wind Turbine Benchmark

In this part, the proposed robust fault estimation based on ANN is simulated for the wind turbine model, which was introduced in section 2.1.

B1. Luenberger NN observer for WT

Before validating the proposed algorithm in wind turbine benchmark, we test the system by an approach based on Luenberger observer and ANN without considering fault estimation capability [28]. For having such observer, the model of (4.1.32) is considered. Detailed information on the steps of designing neural network Luenberger observer can be found in [28].

$$\begin{aligned} \dot{x}(t) &= Ax + g(x,u), \\ y(t) &= Cx(t), \end{aligned} \tag{4.1.32}$$

However, as the scenario here is faulty system, the input actuator faults are considered to be 20% effectiveness loss on $\tau_{g,r}$ occurred in $t = 2500s - 3500s$ and 20% effectiveness loss on β_r occurred in $t = 3000s - 4000s$. No disturbances are considered for this problem. The expectation is that two faulty states of τ_g and β are estimated accurately. The results of this observer can be seen in Figs. 4.3, and 4.4 from which we can see the system states cannot be estimated well. As a result, the algorithm in [28] can track the healthy system states rather than faulty system states, without the capability to track the faulty signals.

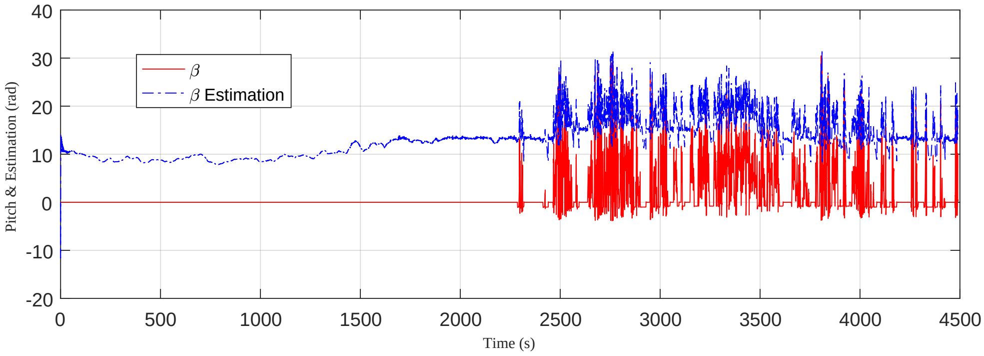


FIGURE 4.3: Pitch angle signal and its estimation using Luenbeger observer.

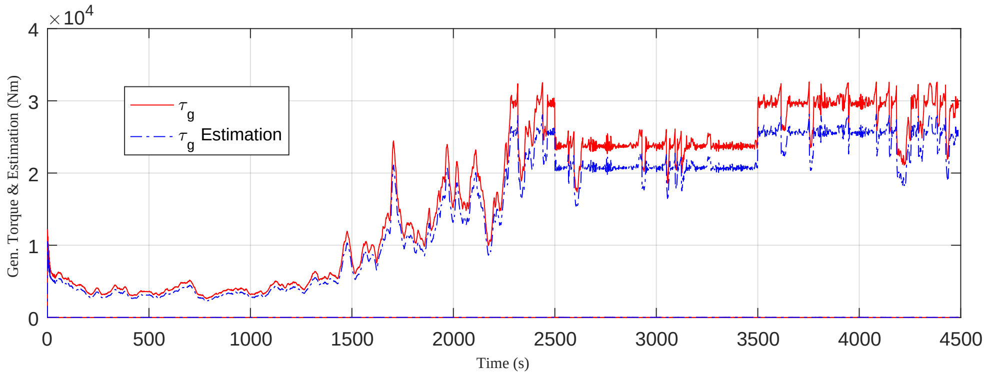


FIGURE 4.4: Generator torque signal and its estimation using Luenbeger observer.

B2. NN fault estimator for WT

First step to validate robust neural network fault estimator is to obtain the robust LMI gain in theorem 4.1. In order to get this gain, A is assumed as:

$$A = \begin{bmatrix} -20 & 3 & 4 & 2 & 3 & 0 \\ 5 & -30 & 4 & 3 & 6 & 1 \\ 10 & 2 & -20 & 3 & 4 & 5 \\ 3 & 17 & 2 & -21 & 11 & 9 \\ 9 & 12 & 2 & 0 & -25 & 4 \\ 6 & 20 & 8 & 1 & 0 & -35 \end{bmatrix},$$

The important criteria in choosing A is that it should be Hurwitz and the pair of (C, A) is observable. B_f is chosen equal to B in equation (2.1.11) to fully cover the actuators of the system. As it is explained in previous chapter, putting large negative values in the main diagonal of the matrix would be very helpful to find the suitable Hurwitz matrix. Therefore, $\bar{A}$ in (4.1.4) can be written as:

$$\bar{A} = \begin{bmatrix} -20 & 3 & 4 & 2 & 3 & 0 & 0 & 0 \\ 5 & -30 & 4 & 3 & 6 & 1 & 0 & 0 \\ 10 & 2 & -20 & 3 & 4 & 5 & 0 & 0 \\ 3 & 17 & 2 & -21 & 11 & 9 & 0 & \omega_n \\ 9 & 12 & 2 & 0 & -25 & 4 & 0 & 0 \\ 6 & 20 & 8 & 1 & 0 & -35 & \alpha_{gc} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

By considering $B_d = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]^T$, one can write:

$$\bar{B}_d = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T,$$

Now, by considering $\gamma = 0.1$, LMI of (4.1.14) can be obtained through LMI solver in MATLAB. Then, L can be achieved via theorem 4.1 and be shown as:

$$L = 10^3 \times \begin{bmatrix} 51.91 & -0.67 & -0.67 & -0.67 \\ -2.51 & 43.75 & -0.37 & -0.36 \\ 370.24 & 397.04 & 367.59 & 456.04 \\ 137.05 & 163.01 & 118.87 & 133.62 \\ -2.52 & -0.38 & 43.75 & -0.39 \\ -2.54 & -0.40 & -0.41 & 43.70 \\ 304.20 & 318.73 & 318.71 & 573.21 \\ -0.02 & -0.02 & -0.02 & -0.02 \end{bmatrix}$$

As in Luenberger observer problem, the input actuator faults are 20% reduction on $\tau_{g,r}$ occurred in $t = 2500s - 3500s$ and 20% reduction on β_r occurred in $t = 3000s - 4000s$.

For the next step, ANN should be set with updating rule of (4.1.22). The ANN training data comes from the benchmark introduced in section IV-Part A. The input of model, which is $u = [\tau_{g,r} \quad \beta_r]^T$ goes directly to ANN model. However, only the error vector of the output of the system is applied to ANN for learning process. By choosing 20 neurons in hidden layers, $\eta_1 = 500$ and $\eta_2 = 500$, the results in Figures 4.5, 4.6, 4.7, and 4.8 can be compared. It is worth to mention that choosing the correct value for η_1 and η_2 can be very challenging. It is recommended to start with small values.

In these four plots, four measurable states of wind turbine model, e.g. ω_r, ω_g, β and τ_g are depicted (the red solid line) in comparison to the related output of robust neural network state estimator (the blue dash line). As it can be clearly seen, the estimations getting from the robust ANN algorithm can follow the outputs of the main system accurately. The estimation errors converge to zero in all outputs.

In addition, in Figures 4.9 and 4.10, the two unmeasured states, e.g. θ_δ, and $\dot{\beta}$, are exhibited. The comparison of the red solid line (which is the output of the system) and the blue dash line (which is the estimation) can explain the effectiveness of the robust neural network algorithm in estimating the unmeasured states.

Now, by considering faults as described earlier on two inputs of the main system, $\tau_{g,r}$, and β_r, the influence of fault on β_r can not be easily seen in the state β (Figure 4.7). The healthy signal, which is green dash line is not so different with the red solid line, which is faulty signal. However, by comparing the healthy signal and faulty one in Figure 4.8, the effect of fault on $\tau_{g,r}$ is completely recognizable on the state τ_g. By the way, without considering that it is recognizable in the output or not, the robust neural network algorithm can precisely estimate the occurred faults. The results are also well-illustrated in Figures 4.11 and 4.12.

Moreover, one can see the RMS value of the estimation errors (RMSE) and normalized estimation error (NRMSE) in Table 4.1, which is calculated based on (4.1.33) and (4.1.34).

$$RMSE = \sqrt{\frac{1}{T_f - T_s} \int_{T_f}^{T_s} (x_i - \hat{x}_i)^2}, \tag{4.1.33}$$

in which, T_s is the start time, T_f is the final time, x_i is the ith parameter, and $\hat{x}_i$ is the estimation of the ith parameter. In addition, the normalized RMSE (NRMSE) can be intoduced as belows:

$$NRMSE = \frac{RMSE}{max(x_i) - min(x_i)}, \tag{4.1.34}$$

in which, $max(x_i)$ is the maximum of x_i and $min(x_i)$ is the minimum of x_i. As one can see in Table 4.1, the RMSE and NRMSE for each states and faults are very small, relatively.

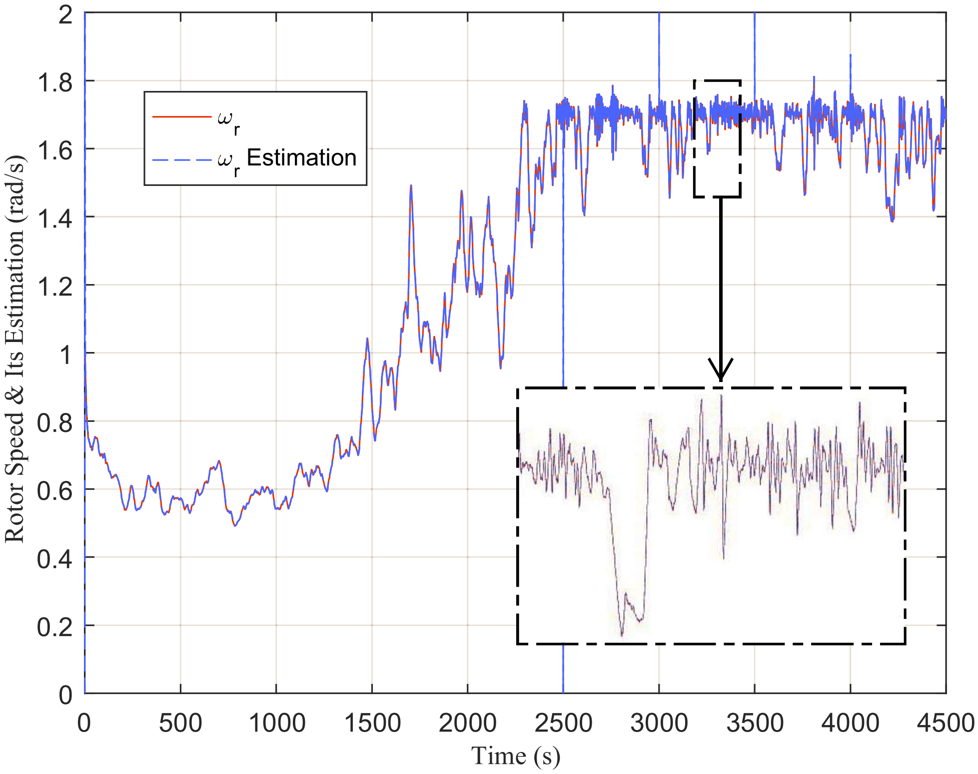


Figure 4.5: Rotor speed signal and its estimation.

B3. Some discussions on the proposed algorithm

It is well illustrated in the literature that neural networks are powerful in estimating complex nonlinear models. However, there are some difficulties related to the simulation. The very challenging point is that due to the big value of the signals, large matrices, and computational cost, the training process is quite

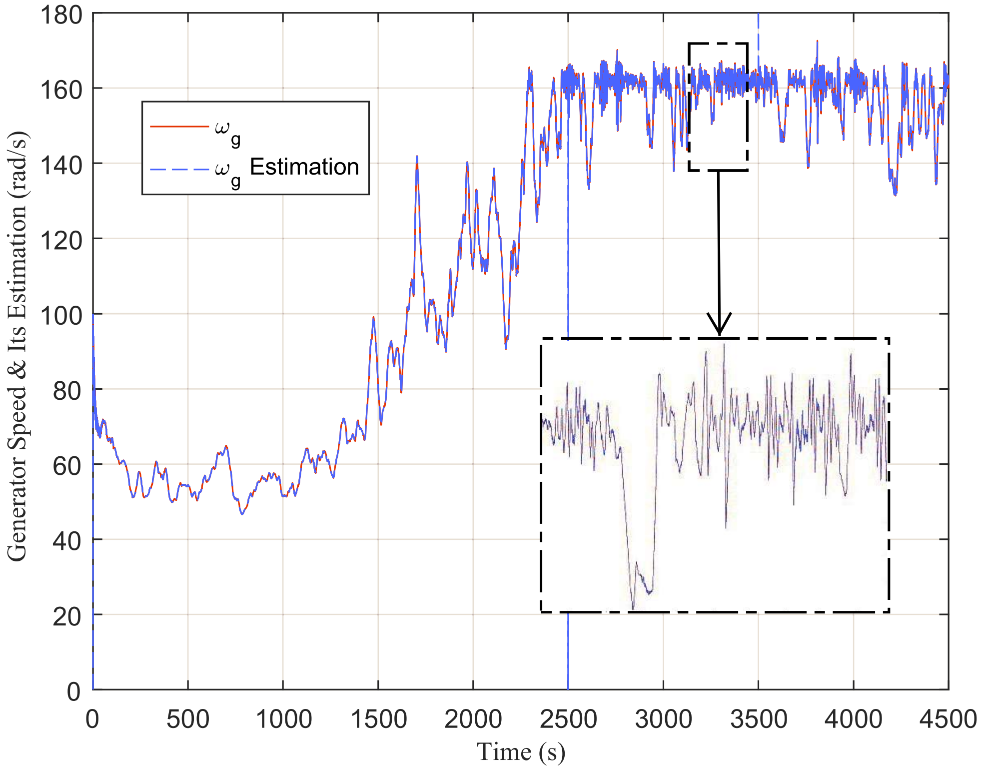


FIGURE 4.6: Generator speed signal and its estimation.

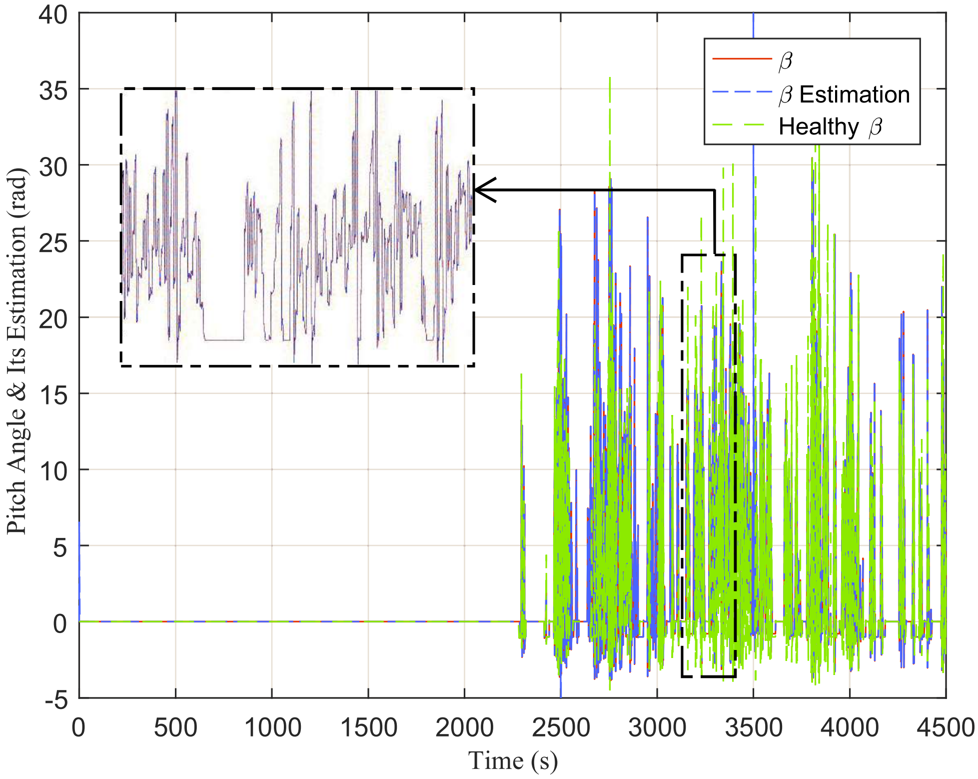


FIGURE 4.7: Pitch angle signal and its estimation.

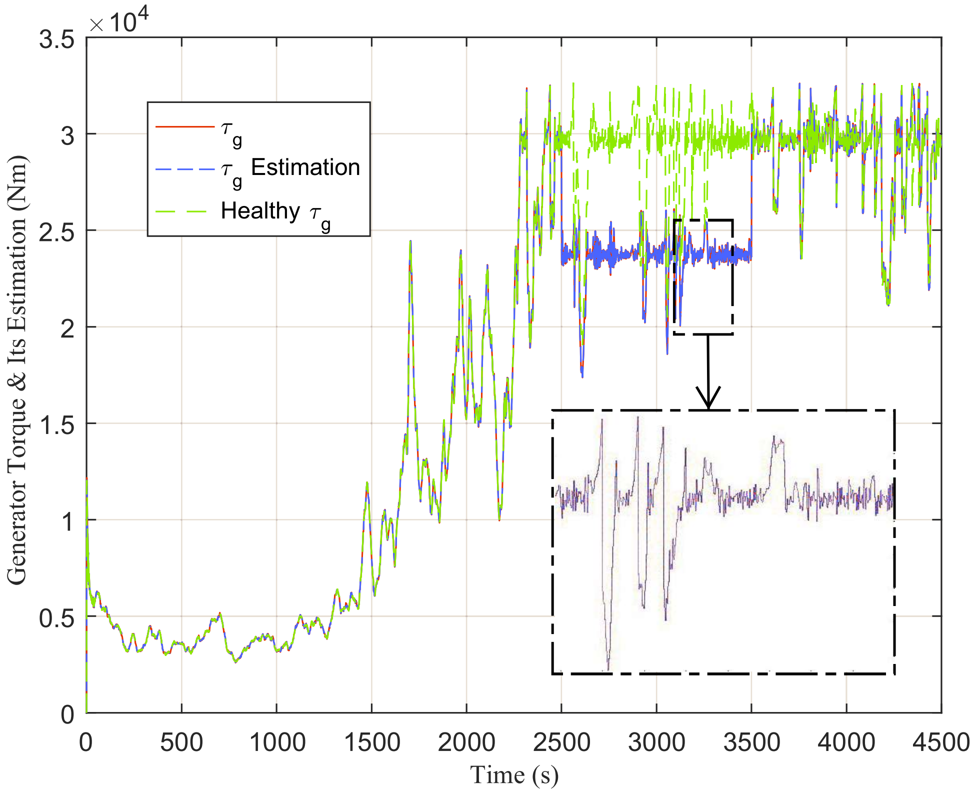


FIGURE 4.8: Generator torque signal and its estimation.

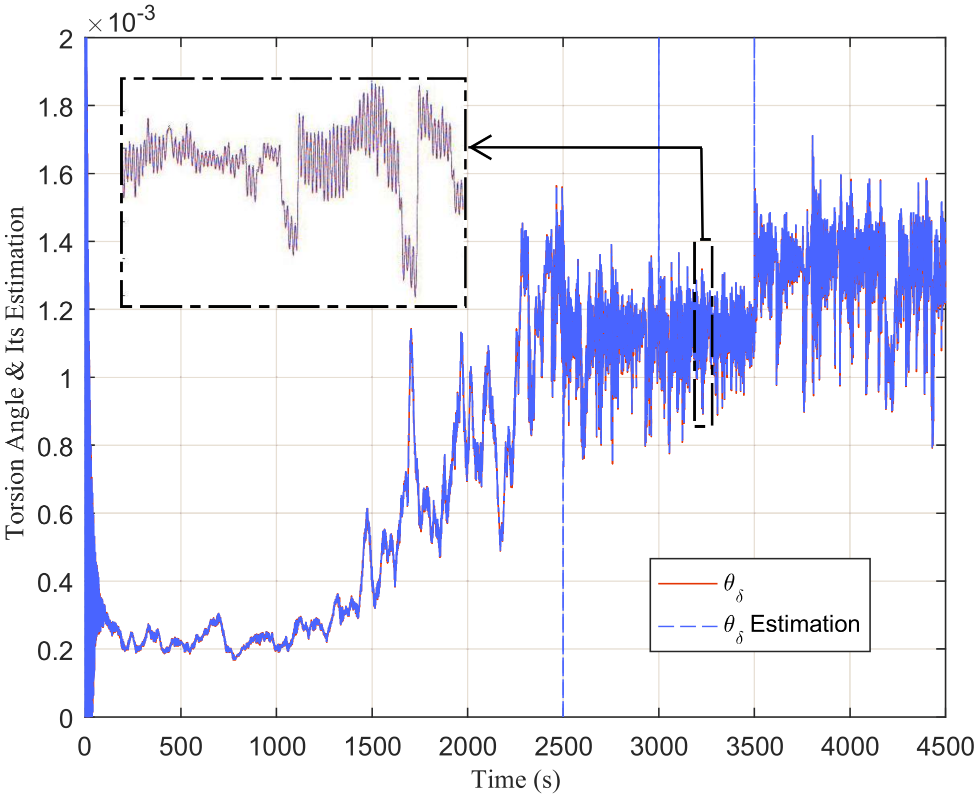


FIGURE 4.9: Torsion angle signal and its estimation.

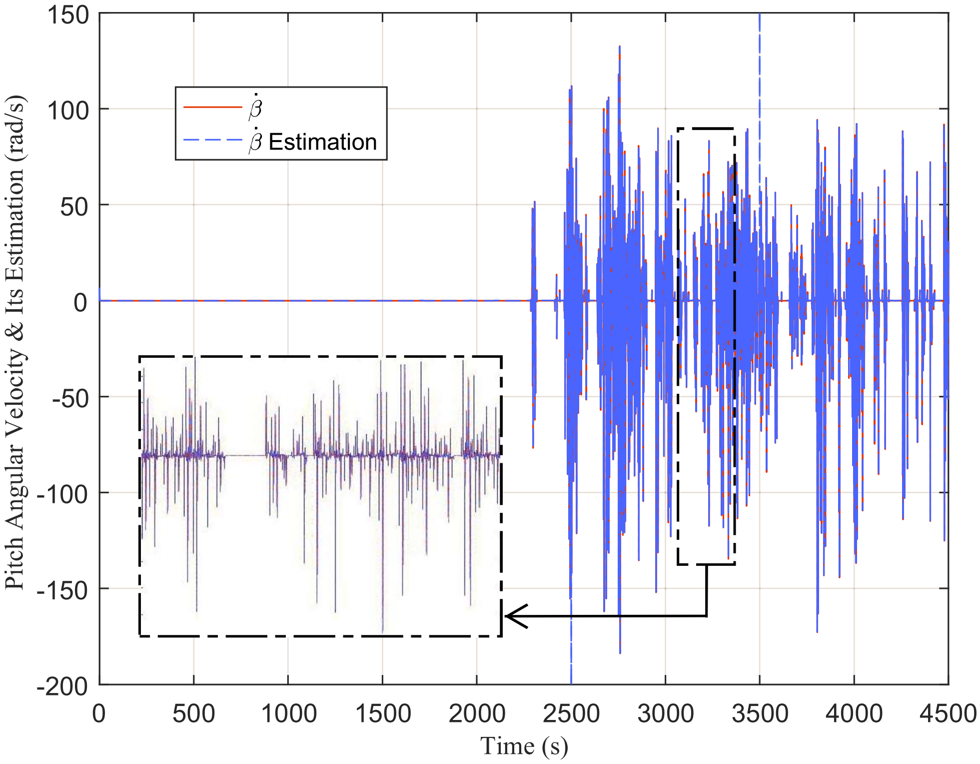


FIGURE 4.10: Pitch angular velocity signal and its estimation.

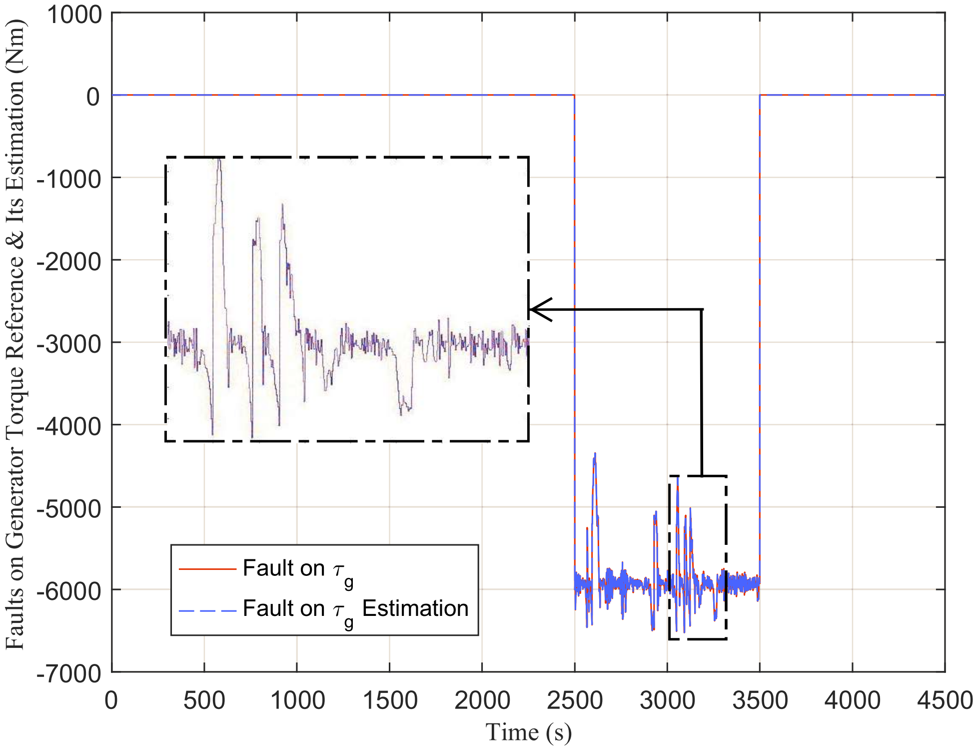


FIGURE 4.11: 20% Faults on reference of generator torque actuator and its estimation.

time-consuming. The other important issue is the solver steps in Matlab. By increasing the step size of the solver, one can get faster training results. However,

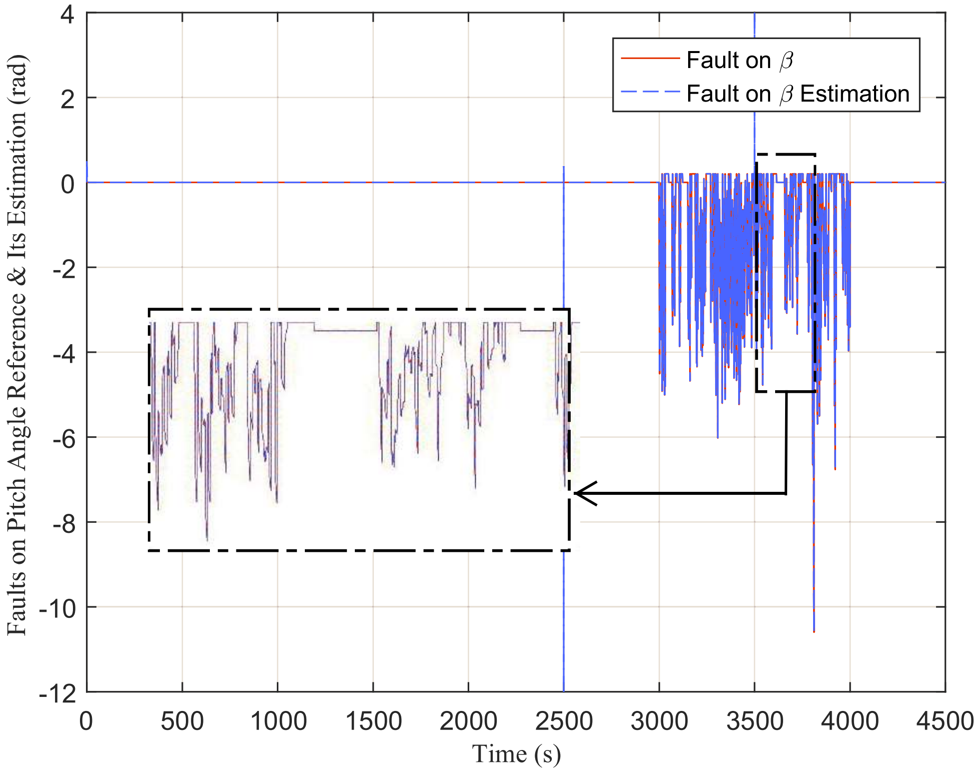


FIGURE 4.12: 20% Faults on reference of pitch angle actuator and its estimation.

TABLE 4.1: RMSE value of each states and faults.

States	**Range**	**RMSE**	**NRMSE (%)**
ω_r	$0 : 1.82\ rad/s$	0.0176	0.98
ω_g	$0 : 180\ rad/s$	1.1782	0.65
β	$-3.85 : 30.50\ rad$	0.1982	0.57
τ_g	$0 : 32600\ Nm$	1.71	0.0052
θ_δ	$0 : 0.0017$	0.000041	2.14
$\dot{\beta}$	$-183.7 : 132.5\ rad/s$	1.4	0.44
Faults on $\tau_{g,r}$	$-6526 : 0\ Nm$	42.87	0.65
Faults on β_r	$-10.60 : 0.2\ Nm$	0.067	0.62

it influences directly on the accuracy of the estimation performance. Having a trade-off between these items, an acceptable accuracy with satisfactory speed can be achieved.

4.2 Neuro-Robust Sensor Fault Estimation

Consider a nonlinear system, represented by the following equations:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) + B_d d(t), \\ y(t) &= Cx(t) + D_f f_s + D_s d_s, \end{aligned} \tag{4.2.1}$$

where $x \in R^n$ is the state vector of the system, $u \in R^m$ is the input signal, $y \in R^p$ is the output and $f(x(t), u(t)) \in R^n$ is an unknown nonlinear function. $f_s(t) \in R^p$ is the sensor fault and D_f is the related fault matrix. $d(t)$ is assumed as a continuously differentiable and bounded environmental disturbance signal and B_d is the distribution matrix. $d_s(t)$ is sensor noise and D_s in the related matrix.

For achieving a robust stable fault estimation, two assumptions are made: first, the nonlinear system is observable. Second, the states of the system should be bounded in L_∞ [79].

By having the term $\pm Ax$, (4.2.1) can be written as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + g(x(t), u(t)) + B_d d(t), \\ y(t) &= Cx(t) + D_f f_s + D_s d_s, \end{aligned} \tag{4.2.2}$$

where $g(x(t), u(t)) = f(x(t), u(t)) - Ax(t)$. The optional Hurwitz matrix A should be selected in the way that yields to the observability of the pair of (C, A).

As it was well-explained in (4.1.3), since the dynamics of the fault is not known, here, a general form is considered without special conditions [79]:

$$\dot{f}_s = \dot{f}_s - f_s + f_s. \tag{4.2.3}$$

Here, it is assumed that f_s and $\dot{f}_s$ are continuously differentiable and bounded. By considering the augmentation of (4.2.3) and (4.2.2), the model can be revised as follows:

$$\begin{aligned} \dot{X}(t) &= \bar{A}X(t) + G + \bar{B}_d\bar{d}, \\ y &= \bar{C}X(t) + D_s d_s, \end{aligned} \tag{4.2.4}$$

where, $X(t) = [x(t) \quad f_s]^T$, $\bar{d} = [d \quad \dot{f}_s \quad f_s]^T$, $G = [g(x(t), u) \quad 0]^T$, $\bar{A} = \begin{bmatrix} A & 0 \\ 0 & -I \end{bmatrix}$, $\bar{B}_d = \begin{bmatrix} B_d & 0 & 0 \\ 0 & I & I \end{bmatrix}$, and $\bar{C} = [C \quad D_f]$. Again, similarly to the previous section, also in this method, the aim is to design an estimator to minimize $\widetilde{X}(t)$ in (4.2.5):

$$\widetilde{X}(t) = X(t) - \hat{X}(t), \tag{4.2.5}$$

where $\hat{X}$ is the vector of estimated states and $\widetilde{X}$ is the vector of estimation errors.

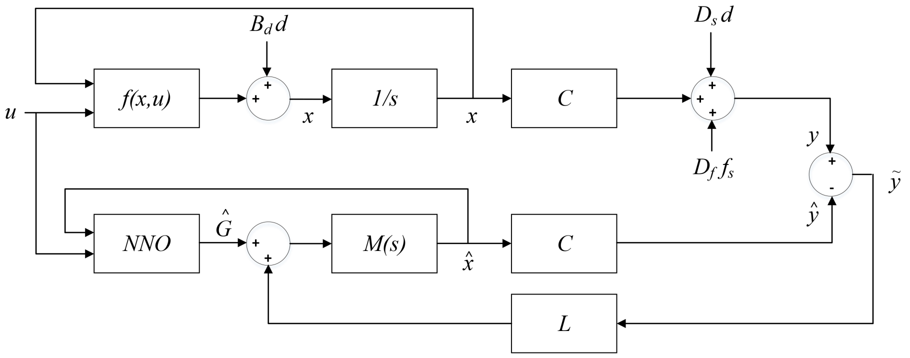


FIGURE 4.13: The model of the observer based on ANN.

As it can be seen in Fig. 4.13, the block NNO is a *Neural Network Observer* and it should be formulated to estimate the nonlinear term of (4.2.4), G, and the block L is a robust observer (designed in Theorem 4.3). Its main task is to eliminate the effects of the disturbance and unmodeled dynamics. In addition, $\hat{y}$ is the estimation of the output vector and $M(s)$ can be obtained as $(sI - \bar{A})^{-1}$.

The upper part of the diagram is the main system of (4.2.1), while the estimation system is illustrated in the lower part.

The observer model can be described as:

$$\begin{aligned} \dot{\hat{X}}(t) &= \bar{A}\hat{X}(t) + \hat{G} + L(y - \bar{C}\hat{X}(t)), \\ \hat{y}(t) &= \bar{C}\hat{X}(t), \end{aligned} \tag{4.2.6}$$

where $L \in R^{(n+p)\times p}$. As it is explained before, $\hat{G}$ is the output of NNO and can be calculated as follows:

$$\hat{G} = \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t)), \tag{4.2.7}$$

where, $\hat{\hat{X}}(t) = [\hat{X}(t) \quad u]^T$ is the input of the observer.

The following observer equation can be written by replacing (4.2.7) into (4.2.6):

$$\begin{aligned} \dot{\hat{X}}(t) &= (\bar{A} - L\bar{C})\hat{X}(t) + \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t)) + L\bar{C}X(t) + LD_sd_s, \\ \hat{y}(t) &= \bar{C}\hat{X}(t), \end{aligned} \tag{4.2.8}$$

By using the error vector in (4.2.5), the dynamics of the output error can be obtained as:

$$\begin{aligned} \dot{\tilde{X}}(t) =& \dot{X}(t) - \dot{\hat{X}}(t) \\ =& \bar{A}X(t) - \bar{A}\hat{X}(t) + G - \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t)) \\ & - L(\bar{C}X(t) - \bar{C}\hat{X}(t)) + \bar{B}_d\bar{d} - LD_sd_s. \end{aligned} \tag{4.2.9}$$

By replacing $\widetilde{G} = G - \hat{W}\sigma(\hat{V}\hat{\hat{X}}(t))$, equation (4.2.9) can be calculated as below:

$$\dot{\widetilde{X}}(t) = \widetilde{A}\widetilde{X}(t) + \widetilde{G} + \bar{B}_d\bar{d} - LD_sd_s, \tag{4.2.10}$$

in which $\tilde{A} = \bar{A} - L\bar{C}$. (4.2.10) can be rewritten as:

$$\dot{\widetilde{X}}(t) = \widetilde{A}\widetilde{X}(t) + \widetilde{D}d_s + \widetilde{B}\widetilde{F}, \tag{4.2.11}$$

in which $\widetilde{F} = [\widetilde{G} \quad \bar{d}]^T$, $\widetilde{D} = -LD_s$, and $\widetilde{B} = [I \quad \bar{B}_d]$. Moreover, output error can be derived as:

$$\begin{aligned} \widetilde{y} &= y - \hat{y} \\ &= \bar{C}\widetilde{X} + D_s d_s. \end{aligned} \tag{4.2.12}$$

In this system, state vector is $\widetilde{X}(t)$ and the exogenous input is $\widetilde{F}$ and d_s. By obtaining (4.2.11), Lemma 1 in section 4.1 should be considered.

Based on [102], the robust performance index can be written as below:

$$J_{11} = ||\widetilde{X}(t)||_{Tf} - \gamma_1^2||\widetilde{F}||_{Tf} - \gamma_2^2||d_s||_{Tf} < 0. \tag{4.2.13}$$

Moreover, the associated *Hamiltonian* function can be defined as follows:

$$\begin{aligned} J_{12} = \int_0^{Tf} (&\frac{dV(\widetilde{X}(t))}{dt} + \widetilde{y}^T(t)\widetilde{y}(t) \\ &- \gamma_1^2\widetilde{F}^T\widetilde{F} + \gamma_2^2 d_s^T d_s)\, dt \end{aligned} \tag{4.2.14}$$

Theorem 4.3. *The robust observer (4.2.8) can be achieved for the model of (4.2.4), so that: (i) the estimation error dynamics of (4.2.11) is prove to have ISS; (ii) the estimation error satisfies the robust performance index (4.2.13), if there are a positive definite matrix P and a matrix Q so that*

$$\begin{bmatrix} \Sigma & -QD_s + \bar{C}^T D_s & P\widetilde{B} \\ -D_s^T Q^T + D_s^T \bar{C} & D_s^T D_s - \gamma_2^2 I & 0 \\ \widetilde{B}^T P & 0 & -\gamma_1^2 I \end{bmatrix} < 0 \tag{4.2.15}$$

in which $\Sigma = P\bar{A} + \bar{A}^T P - Q\bar{C} - \bar{C}^T Q^T + \bar{C}^T\bar{C}$. *Then, the observer gain can be obtained as* $L = P^{-1}Q$.

Proof: Theorem 4.3 can be proved in two sections: (i) ISS, and (ii) robust performance index.

(i) **The ISS**. For any $\widetilde{X}_1(t)$, $\widetilde{X}_2(t)$, $\widetilde{F}_1$, $\widetilde{F}_2$, d_{s1}, and d_{s2} we can get:

$$\begin{aligned}|h(\widetilde{X}_1(t), &\widetilde{F}_1, d_{s1}) - h(\widetilde{X}_2(t), \widetilde{F}_2, d_{s1})| \\ &= |\widetilde{A}(\widetilde{X}_1 - \widetilde{X}_2) + \tilde{B}(\widetilde{F}_1 - \widetilde{F}_2) + \widetilde{D}(d_{s1} - d_{s2})| \\ &\leq \alpha|\widetilde{X}_1(t) - \widetilde{X}_2(t)| + \beta|\widetilde{F}_1 - \widetilde{F}_2| + \delta|d_{s1} - d_{s2}|,\end{aligned} \tag{4.2.16}$$

in which, $\alpha = ||\tilde{A}||$, $\beta = ||\tilde{B}_1||$, and $\delta = ||\widetilde{D}||$. Consequently, $h(\widetilde{X}(t), \widetilde{F}, d_s)$ is globally Lipschitz in $(\widetilde{X}(t), \widetilde{F}, d_s)$. Therefore, $h(\widetilde{X}(t), \widetilde{F}, d_s)$ can be confirmed as continuously differentiable function.

Considering the matrix $\widetilde{A}$ being Hurwitz, the unforced system $\dot{\widetilde{X}}(t) = \widetilde{A}\widetilde{X}(t) = h(\widetilde{X}(t), 0, 0)$ is globally exponentially stable at the origin. As a result, by means of Lemma 1, it is well-founded that the estimation error dynamics $\dot{\widetilde{X}}(t) = h(\widetilde{X}(t), \widetilde{F}, d_s)$ is input-to-state stable.

(ii) **Robust performance index**. The Lyapunov candidate of $V(\widetilde{X})$ is taken as below:

$$V(\widetilde{X}(t)) = \widetilde{X}(t)^T P \widetilde{X}(t), \tag{4.2.17}$$

in which P is positive definite symmetric matrix. Derivative of (4.2.17) can be written as:

$$\begin{aligned}\dot{V}(\widetilde{X}(t)) =& \widetilde{X}(t)^T P \dot{\widetilde{X}}(t) + \dot{\widetilde{X}}(t)^T P \widetilde{X}(t) \\ =& \widetilde{X}(t)^T P(\widetilde{A}\widetilde{X}(t) + \widetilde{D}d_s + \widetilde{B}\widetilde{F}) \\ &+ (\widetilde{A}\widetilde{X}(t) + \widetilde{D}d_s + \widetilde{B}\widetilde{F})^T P \widetilde{X}(t).\end{aligned} \tag{4.2.18}$$

By substituting (4.2.18) into (4.2.14), J_{12} can be derived as:

$$\begin{aligned} J_{12} = \int_0^{Tf} & (\widetilde{X}(t)^T P(\widetilde{A}\widetilde{X}(t) + \widetilde{D}d_s + \widetilde{B}\widetilde{F}) \\ & + (\widetilde{A}\widetilde{X}(t) + \widetilde{D}d_s + \widetilde{B}\widetilde{F})^T P\widetilde{X}(t) \\ & + \widetilde{y}^T(t)\widetilde{y}(t) - \gamma_1^2\widetilde{F}^T\widetilde{F} + \gamma_2^2 d_s^T d_s)\, dt \end{aligned} \tag{4.2.19}$$

Considering the vector of $Z = [\widetilde{X}(t) \quad d_s \quad \tilde{F}]^T$ and refering to *Schur Complement*, (4.2.19) can be obtained as:

$$J_{12} = \int_0^{Tf} Z^T R Z \; dt, \tag{4.2.20}$$

in which,

$$R = \begin{bmatrix} P\widetilde{A} + \widetilde{A}^T P + \bar{C}^T\bar{C} & P\widetilde{D} + \bar{C}^T D_s & P\widetilde{B} \\ \widetilde{D}^T P + D_s^T\bar{C} & D_s^T D_s - \gamma_2 I & 0 \\ \widetilde{B}^T P & 0 & -\gamma^2 I \end{bmatrix} \tag{4.2.21}$$

Therefore, for achieving $J_{12} < 0$, R should be negative definite. By replacing $\widetilde{A} = \bar{A} - L\bar{C}$, and $\widetilde{D} = -LD_s$, $R < 0$ is lead to (4.2.15), where $Q = PL$. It is worth to mentioned that $V(\tilde{X}(t)) \geq 0$, and from (4.2.14) and $J_{12} < 0$, (4.2.13) can be maintained. Hence, the robust gain of $L = P^{-1}Q$ can be obtained $\therefore$.

The next step to obtain a neural network robust observer is proposing an updating rule for weight matrices, $\hat{W}$ and $\hat{V}$, by utilizing the output error $\widetilde{y}$.

Theorem 4.4. *Considering the nonlinear system of (4.2.1) and the observer model of Figure 4.13 with observer model of (4.2.8). If the weights of the network are* *updated as:*

$$\begin{aligned}\dot{\hat{W}} =& -\eta_1(\widetilde{y}^T C\tilde{A}^{-1})^T(\sigma(\hat{V}\hat{\hat{X}}))^T - \rho_1||\widetilde{y}||\hat{W},\\ \dot{\hat{V}} =& -\eta_2(\widetilde{y}^T C\tilde{A}^{-1}\hat{W}(I-\Lambda(\hat{V}\hat{\hat{X}})))^T\hat{\hat{X}}^T\\ &- \rho_2||\widetilde{y}||\hat{V},\end{aligned} \tag{4.2.22}$$

where $\Lambda(\hat{V}\hat{\hat{x}}) = diag\left\{\sigma_i^2(\hat{V}_i\hat{\hat{X}})\right\}$, $i = 1, 2, ..., m$, η_1 *and* η_2 *are positive learning rate and* ρ_1 *and* ρ_2 *should have positive values. Then, It can be proved that* $\widetilde{X}, \tilde{W}, \widetilde{V}, \widetilde{y} \in L_\infty$, *which are the estimation error, the weights error, and the output error, respectively, are all bounded.*

Proof: The proof of this theorem is thoroughly explained in [79] $\therefore$

Procedure 1. For obtaining the aim of this algorithm, the procedure is completely similar to the one in actuator fault estimation in section 4.1, which was explained earlier in this chapter.

4.2.1 Simulation and Results

In this part, the proposed robust neural network sensor fault estimation is simulated in the wind turbine benchmark.

Based on the approach explained in Theorem 4.3, the first step to design an observer is to calculate LMI gain. To calculate L, A can be considered as a

Hurwitz matrix:

$$A = 1000 \times \begin{bmatrix} -40 & 3 & 4 & 2 & 3 & 1 \\ 5 & -30 & 4 & 3 & 6 & 1 \\ 10 & 2 & -20 & 3 & 4 & 5 \\ 3 & 10 & 2- & 21 & 3 & 1 \\ 9 & 12 & 2 & 0 & -25 & 4 \\ 6 & 9 & 8 & 1 & 0 & -35 \end{bmatrix}$$

Again, as explained before, obtaining A can be very challenging and it affects the stability of the system. It is recommended to put large negative poles on the main diagonal of it. B_d is as $[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]^T$ to consider the environmental disturbances affect all of the states equally. So, $\bar{B}_d$ and then $\widetilde{B}$ can be consequently calculated. D_f and D_s are considered as $I_{4\times 4}$ to evaluate faults on all four sensors. Now, by applying the above variables into Theorem 1, L can be computed as:

$$L = 1000 \times \begin{bmatrix} 1276 & 1196 & 534 & 1472 \\ -324 & 595 & -113 & 444 \\ 194 & 745 & 359 & 731 \\ -7574 & -3300 & -698 & -2603 \\ -1441 & -683 & -1253 & -521 \\ 1186 & 1003 & 239 & 2179 \\ -2710 & -1417 & -7377 & -1280 \\ 2483 & 2413 & 1601 & 2109 \\ 1154 & 3939 & -38 & 2774 \\ 5521 & 4255 & 2847 & 4040 \end{bmatrix}$$

Two different scenarios are applied to this benchmark. The first one, which is shown in Table 4.2, considered the 20% reduction on the performance in each sensor, in different timing. The results of this scenario are illustrated in Figs.

4.14-4.17.

Table 4.2: Sensor faults for the first scenario.

Fault	Timing
ω_r	1500 $\leq$ t $\leq$ 2000
ω_g	3000 $\leq$ t $\leq$ 3200
β	2500 $\leq$ t $\leq$ 2700
τ_g	3000 $\leq$ t $\leq$ 4000

As it can be seen in Figs. 4.14-4.17, that the four outputs of the system are affected by sensor faults, described in Table 4.2. In part (a) of each figure, one of the outputs of the system is illustrated and it can be easily seen that the observer can estimate the states accurately. In addition, in part (b) of each diagram, the occurred faults are plotted and it can be also noted that the observer estimates the related fault precisely, even tough it is relatively large.

The simulation is also repeated with different percentages of faults with different timing. The details of each sensor fault are shown in Table 4.3. The results of this scenario are illustrated in Figs. 4.18-4.21

Even though the faults are more complicated and there are more than one fault at a time, it is obvious that the observer converges to the outputs and the faults. Therefore, the goal of the approach, which is estimating the occurred faults is achieved.

Table 4.3: Sensor Faults for the Second Scenario

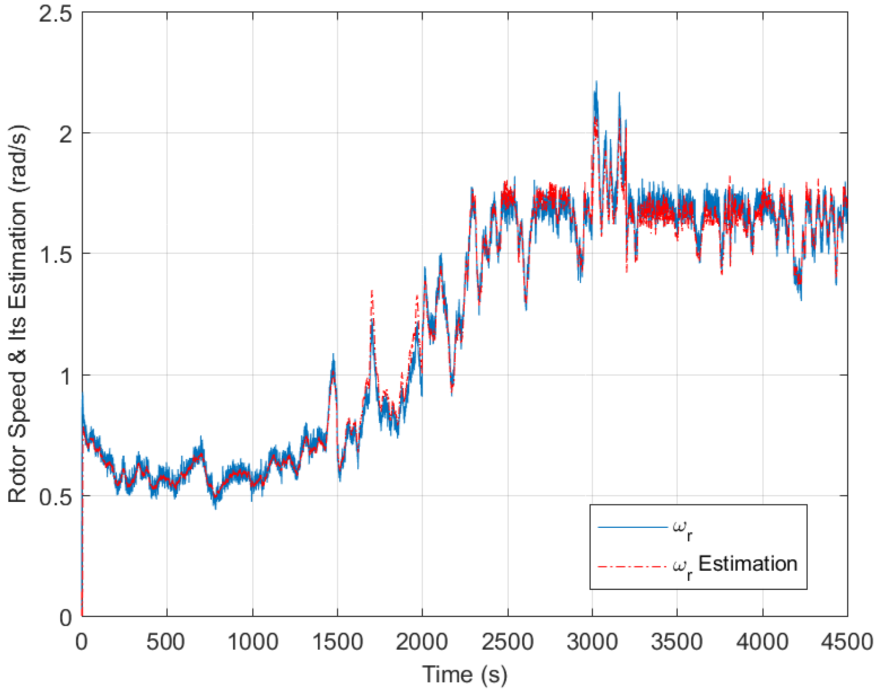
Fault	Timing	Percent
ω_r	1200 $\leq$ t $\leq$ 2500	-30%
ω_g	2300 $\leq$ t $\leq$ 3500	-10%
β	3500 $\leq$ t $\leq$ 4000	-15%
τ_g	3000 $\leq$ t $\leq$ 3800	-25%

4.3 Summary and Conclusions

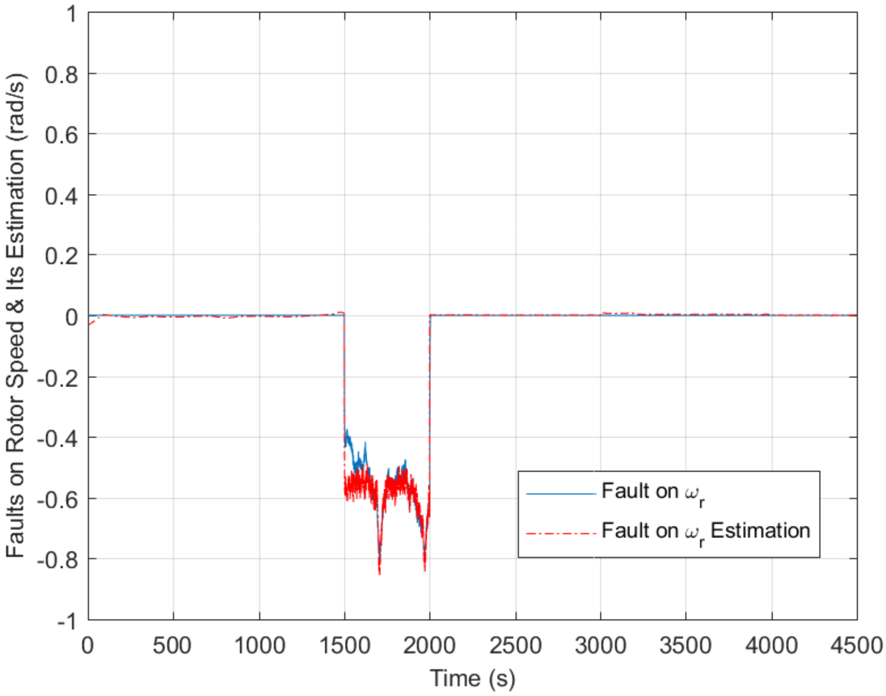
In this chapter, a novel robust fault estimation approach has been proposed based on artificial neural networks. The first difficulty to confront is the unmeasurable states in MIMO systems such as wind turbines. To cope with this problem, the dynamical model is decoupled into a linear and a nonlinear block. For the nonlinear one, a fully connected ANN is developed to identify the nonlinearities. For relaxing the conditions on fault modelling, a model is proposed, and a robust LMI is studied to deal with unmodeled faults and disturbances using input-to-state stability lemma. Two different scenarios, including occurred faults on actuators and the other one on sensors, were investigated.

The approach is validated on the wind turbine benchmark. A case study is investigated for 20% loss of performance on each actuator. The results demonstrate the effectiveness of the proposed algorithm in the first part. The faults are estimated successfully and the outputs of the observer converge to the real output, simultaneously.

For the part related to the sensor faults, two different case studies were investigated for each sensor. The results validated the effectiveness of the studied method. The sensor faults were estimated accurately and the observer's outputs converge to the model output, synchronously.

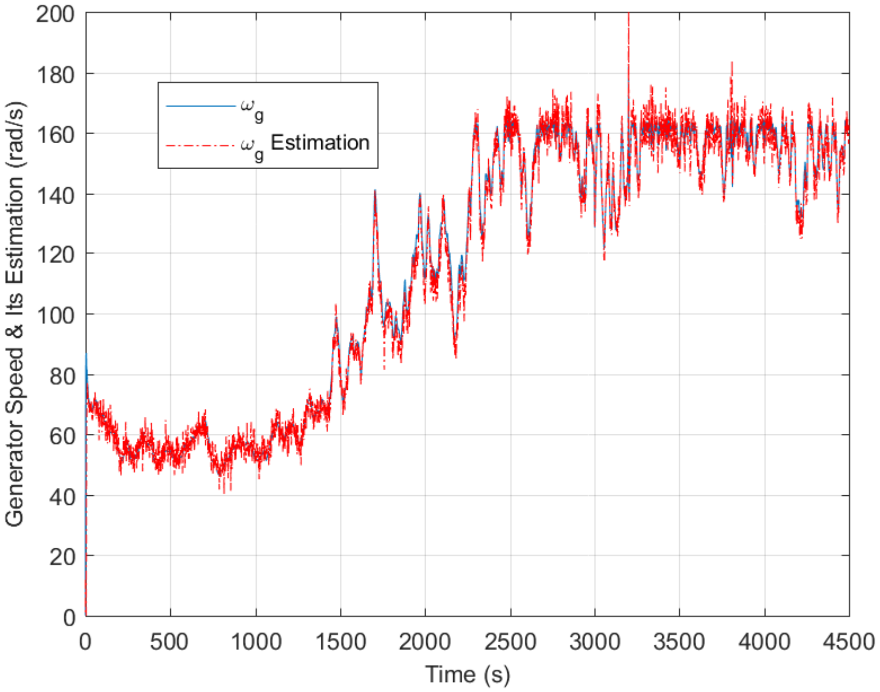


(a) Rotor speed and its estimation.

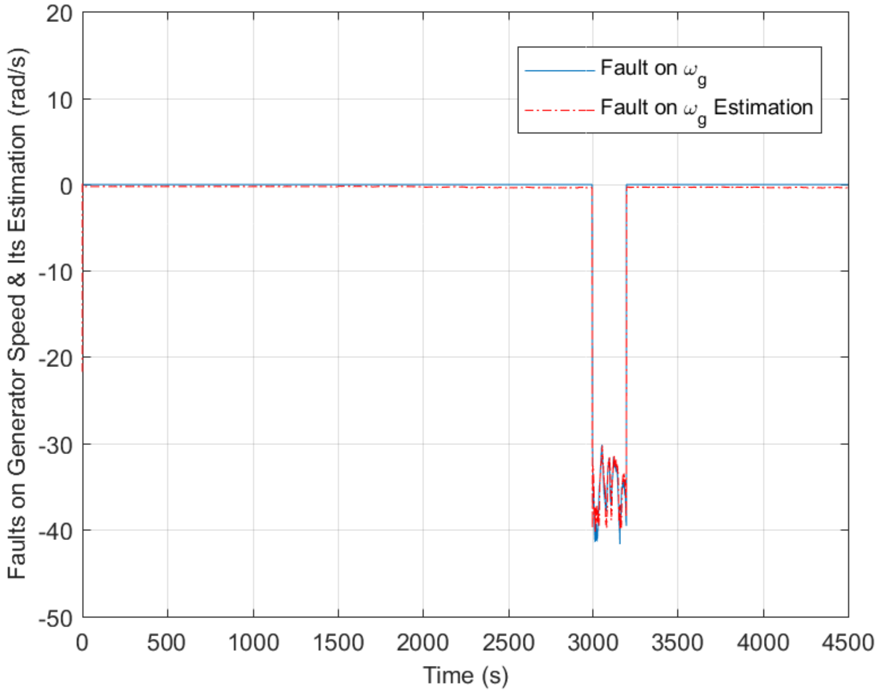


(b) Fault on rotor speed and its estimation.

FIGURE 4.14: Comparison of occurred fault on ω_r sensor in the first scenario.

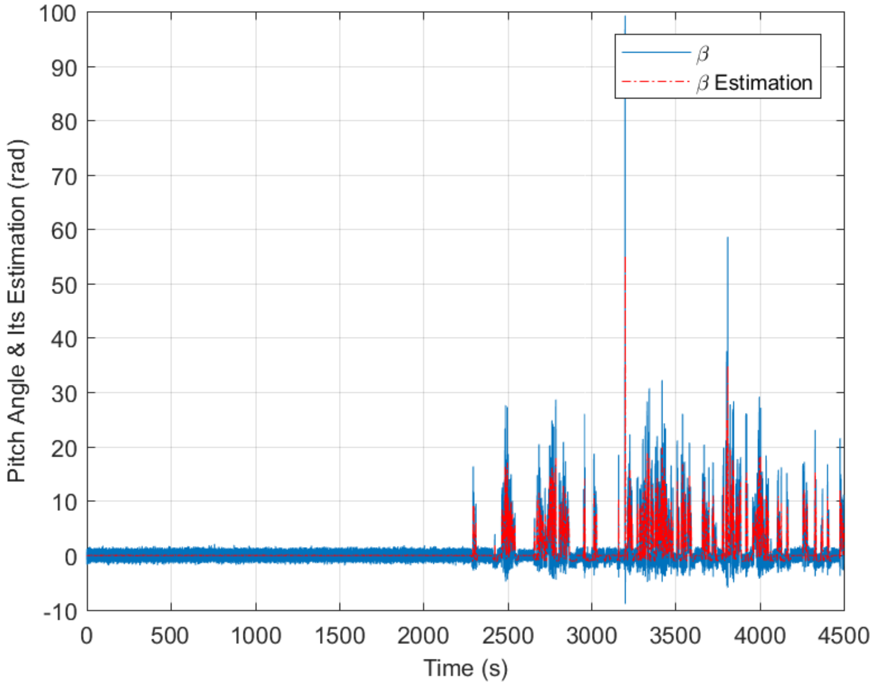


(a) Generator speed and its estimation.

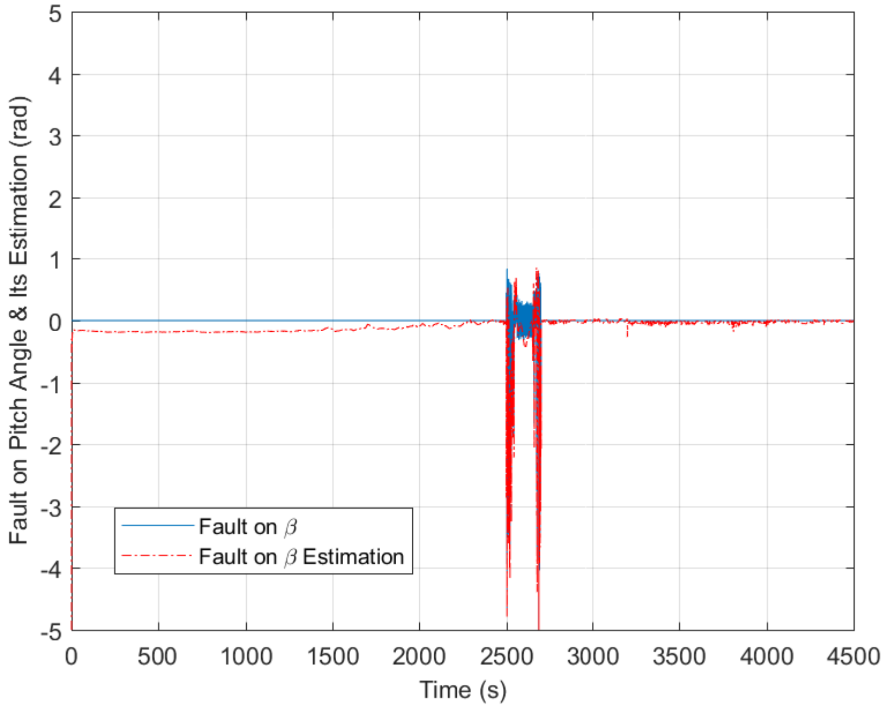


(b) Fault on generator speed and its estimation.

FIGURE 4.15: Comparison of occurred fault on ω_g sensor in the first scenario.

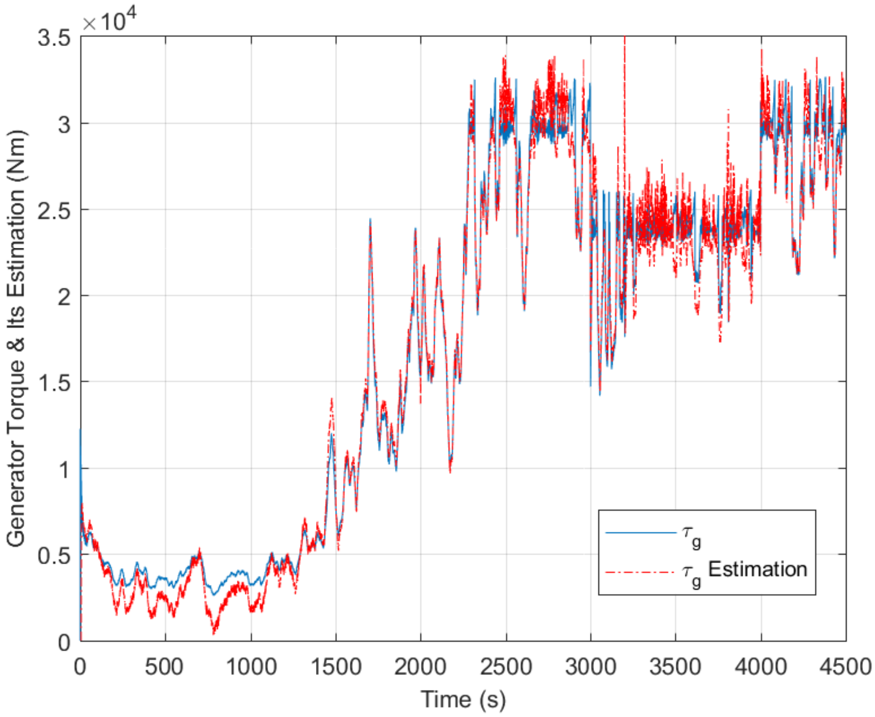


(a) Pitch angle and its estimation.

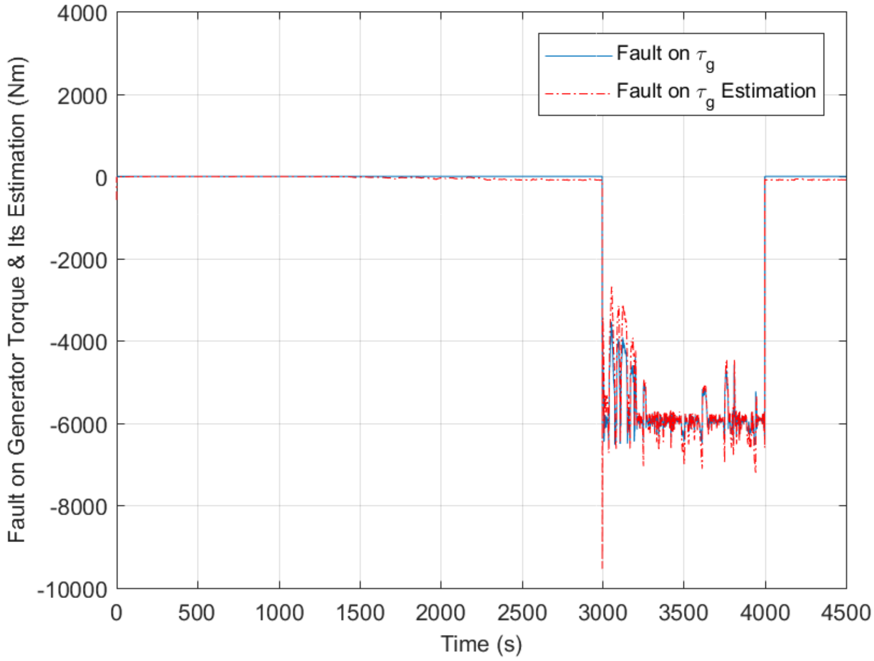


(b) Fault on pitch angle and its estimation.

FIGURE 4.16: Comparison of occurred fault on β sensor in the first scenario.

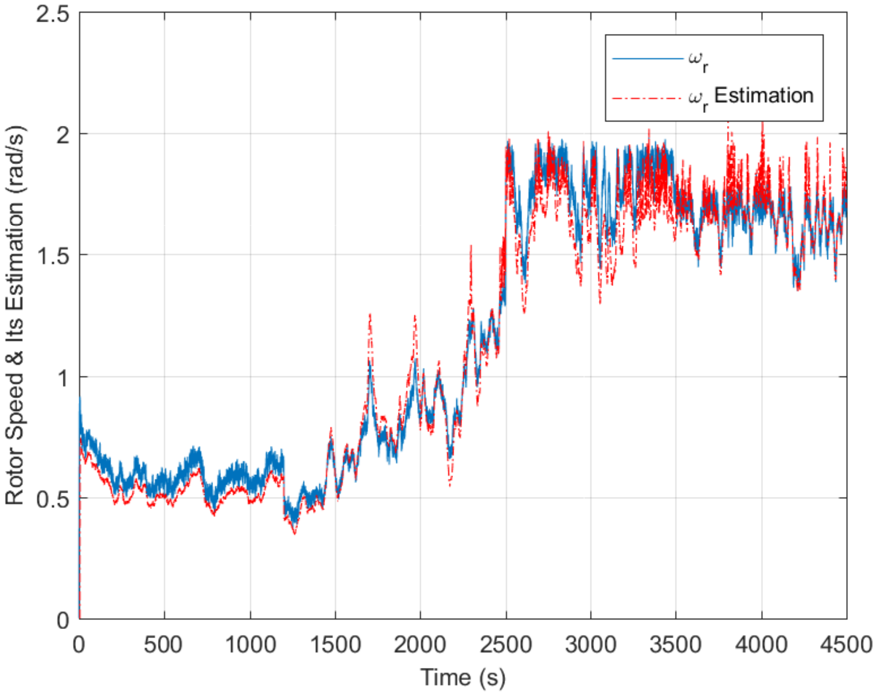


(a) Generator torque and its estimation.

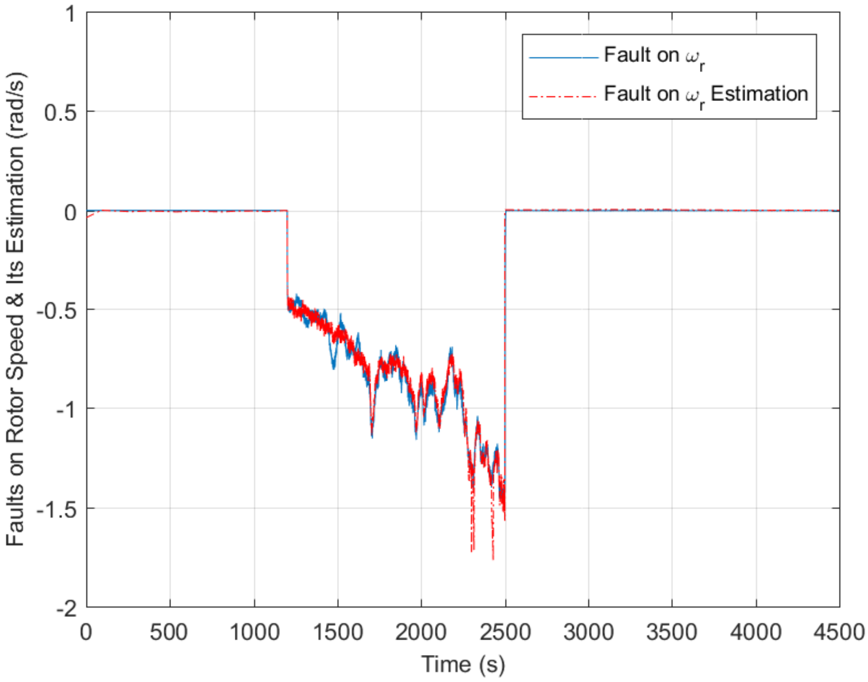


(b) Fault on generator torque and its estimation.

FIGURE 4.17: Comparison of occurred fault on τ_g sensor in the first scenario.

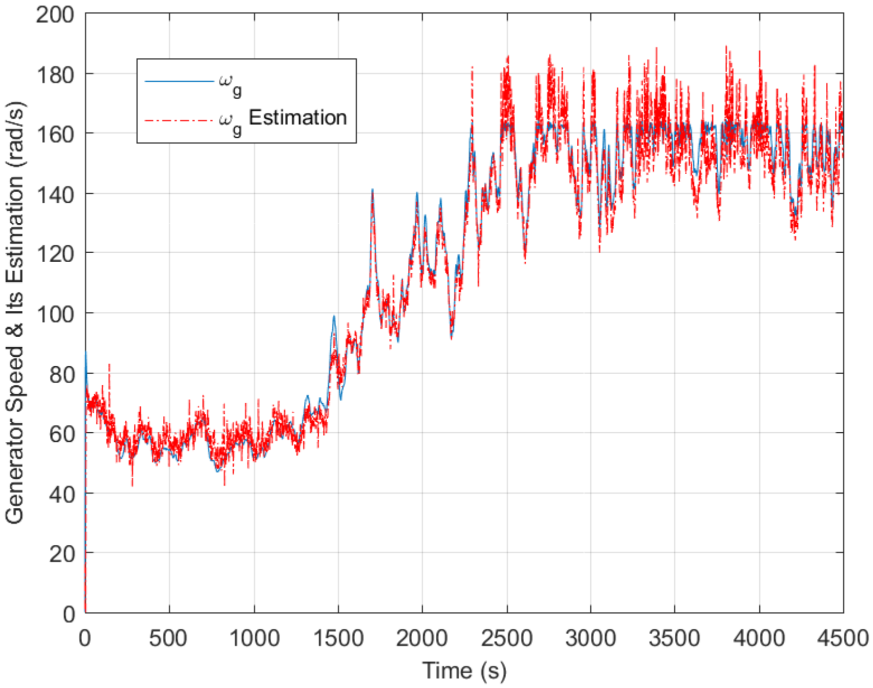


(a) Rotor speed and its estimation.

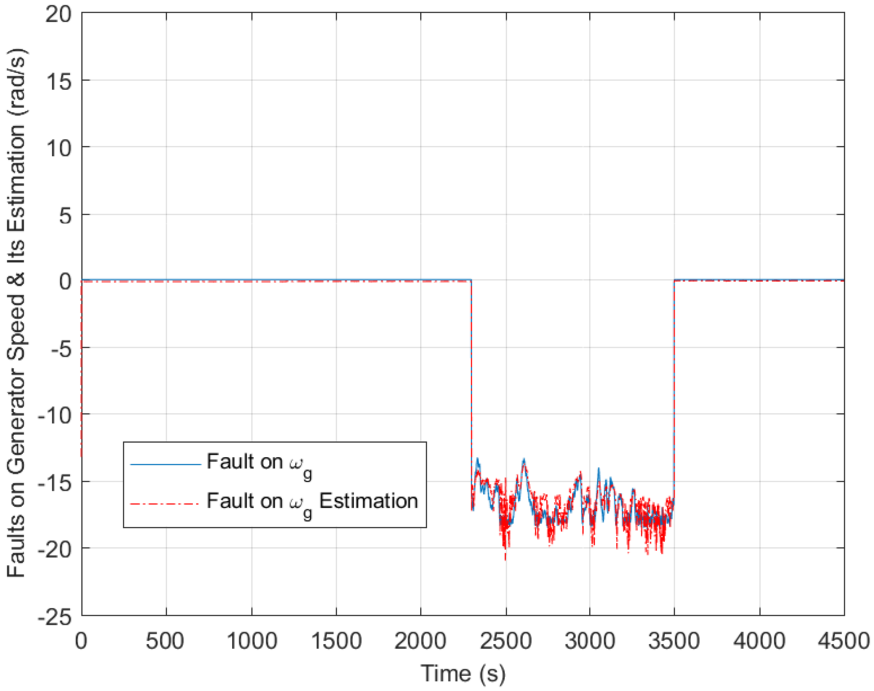


(b) Fault on rotor speed and its estimation.

FIGURE 4.18: Comparison of occurred fault on ω_r sensor in the second scenario.

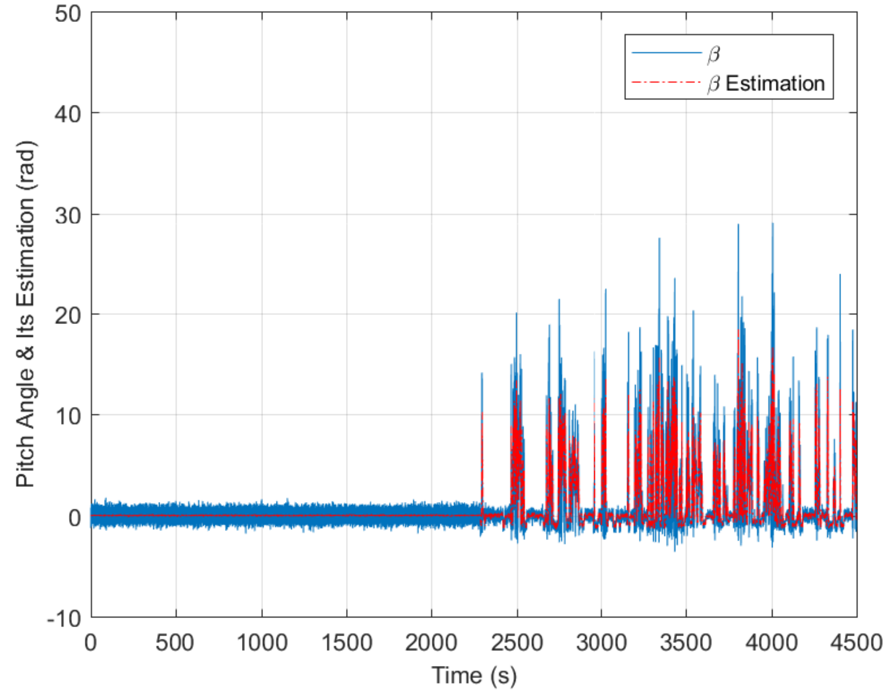


(a) Generator speed and its estimation.

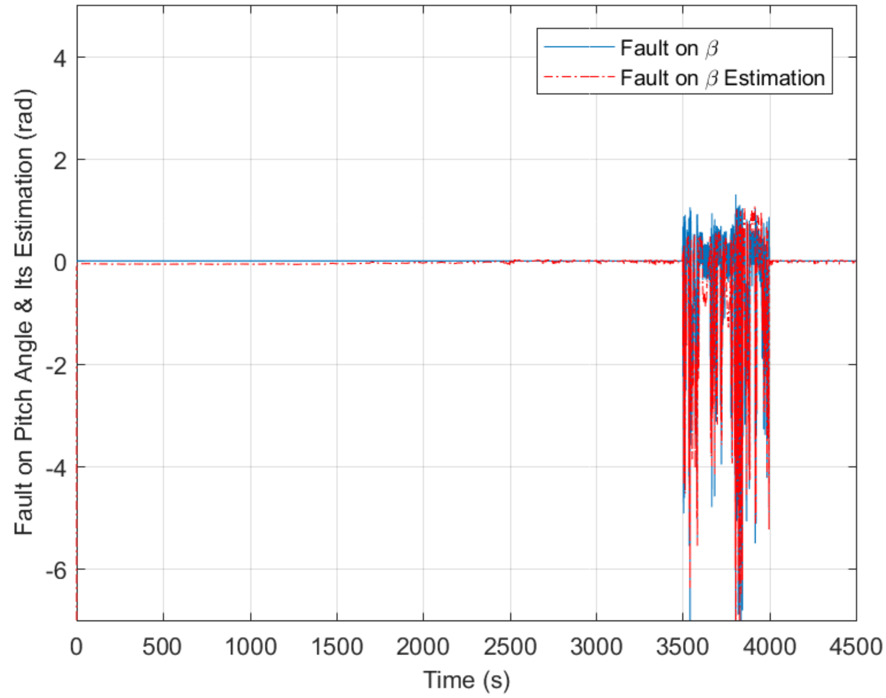


(b) Fault on generator speed and its estimation.

FIGURE 4.19: Comparison of occurred fault on ω_g sensor in the second scenario.

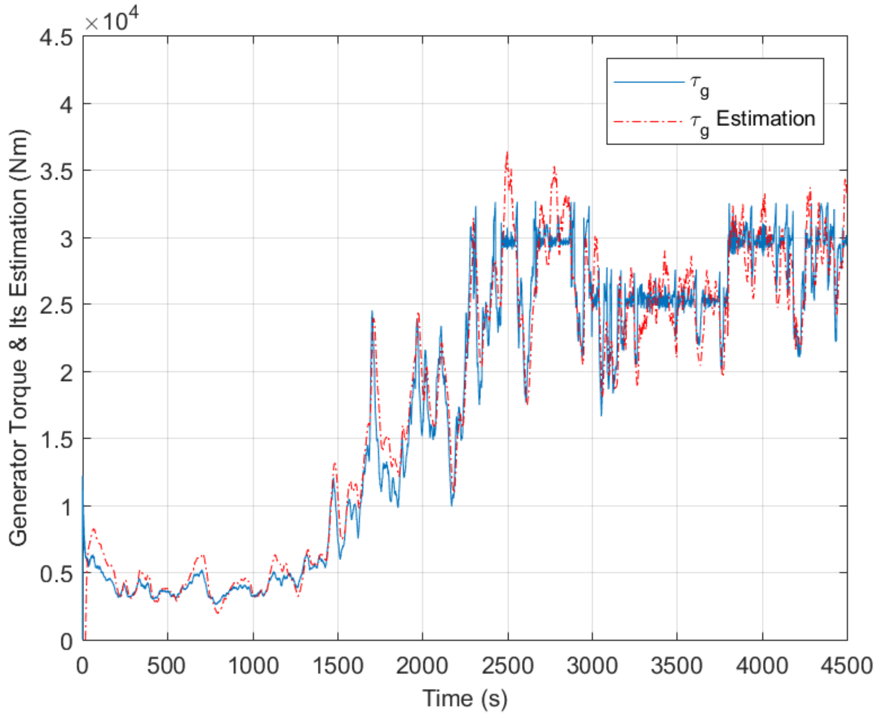


(a) Pitch angle and its estimation.

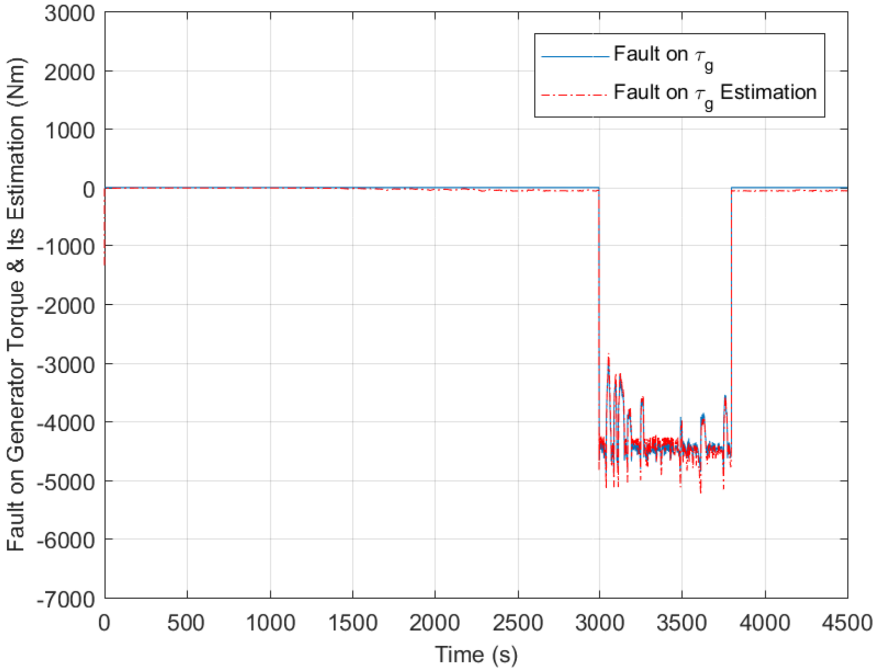


(b) Fault on pitch angle and its estimation.

FIGURE 4.20: Comparison of occurred fault on β sensor in the second scenario.



(a) Generator torque and its estimation.



(b) Fault on generator torque and its estimation.

FIGURE 4.21: Comparison of occurred fault on τ_g sensor in the second scenario.

Chapter 5

Time-series Deep Learning Fault Detection

In this chapter, the goal is to achieve an approach to diagnose even small anomalies in the output of the wind turbine system. To reach this goal, CNN structures are developed and trained based on the data from the benchmark, which contains sensor noises and actuator faults on generator torque and pitch angle. Three different scenarios are considered for this purpose. The first one includes fault detection based on just one actuator, and in the second one, the approach is developed for both actuators of the wind turbine system. In the last scenario, the faults on four sensors of the wind turbine are considered. In each scenario, various CNN structures are compared based on the accuracy criteria. Besides, some vital actions that are needed before CNN training is studied through this chapter.

5.1 Introduction to Deep Learning

As it has been discussed in section 2.3, the applications of machine learning and specifically deep learning are rapidly increasing due to their powerful ability to predict and categorise different data structures. The start of developing mathematical theory of these techniques has been backed to 1949, when Donald O. Hebb published his algorithm, named Hebbian Learning [104]; however, due to the lack of suitable technology, the implementation of them for real data has been quite recent. Today, AlexNet [105] and GoogleNet [106], as the two most powerful deep learning architecture, can handle more than 15 million image data for different applications of face recognition to model compression. In convolutional neural networks, as a class of deep learning fields, as fully explained in 2.3.3.1, three main layers are connected in series, convolutional layer, pooling layer, and fully connected neural networks. The details can be found in Figure 2.7.

5.2 Preprocess the Input Data

The first step in any deep learning problem is to understand what type of data you are working with. In a typical CNN, the input data is 2-D images. However, in this scenario, the data is time-series. One way to deal with this problem is to convert the raw data into 2-D images [88]. As it is clear in Figure 5.1, the time domain signal is queued into 2-D images row by row. As one can see in Figure 5.1, the time-series signal samples store in each pixel of the 2-D image and build a $n \times n$ matrix, in which $n = \sqrt{M}$. M is the number of samples in each record.

It is worth mentioning that the input images used in CNN normally have three colour channels as Red-Green-Blue (RGB). Here, for the simplicity, it is assumed that the time domain signal is converted to just one channel image; so, it is supposed to be greyscale. Therefore, at the end of this stage, each record is converted to $n \times n \times 1$, in which 1 is the quantity of the channels. In addition, the greyscale images are essentially 8 bits, so the output images have less visibility and accuracy than the input data. This issue causes no problem for this scenario,

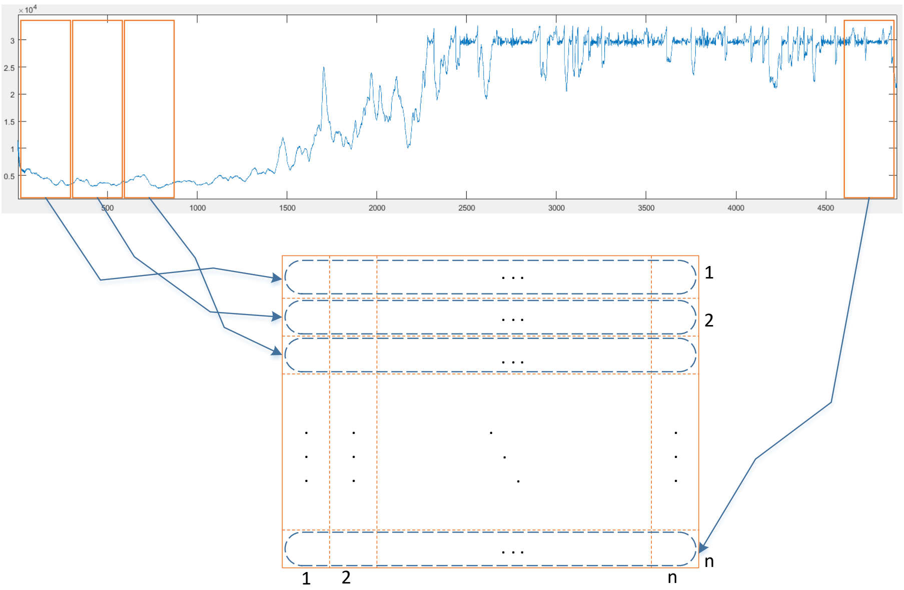


FIGURE 5.1: Fulfilling an image of $n \times n$ matrix with a time-series signal.

since the desired accuracy of 1% is satisfied. Another important thing is that CNN is usually used to extract information from images by considering the relationship of adjacent pixels. In this method, adjacent pixels may not have any reasonable relation. However, it is vital that in every record, the relation between two pixels is the same, except for the faulty one.

5.3 Scenario 1: One Actuator Fault

In this section, the first scenario, which includes one actuator fault, is studied based on the time-series data to image conversion in the previous section.

After finishing the previous conversion stage, a CNN structure should be trained to classify the processed input data. There are a lot of CNN structures in the literature, which are highly used for deep learning in image processing. Among these networks, some of them are quite famous and popular, such as AlexNet [105], and GoogLeNet [106]. In this section, two CNN structures are proposed to solve the problem of the classification of time-series signals. A comparison will be given in order to evaluate their performances.

5.3.1 CNN with One convolutional Layer

The schematic of the proposed structure can be found in Figure 5.2. As it is obvious in this figure, the input processed data is applied to a convolutional layer with the 20 kernels of 5 × 5. The quantity and size of the kernels can be varying. The output of this layer is conveyed through a ReLu activation function. A max pooling consists of pools of 3 × 3 is applied to extract the most reliable features. The next layer is an FCNN and then a softmax function, in order to classify the data as accurate as possible. For further information on the concepts and functions of ReLU, FCNN, Softmax, max pooling, and kernel, one can see 2.3.3.1.

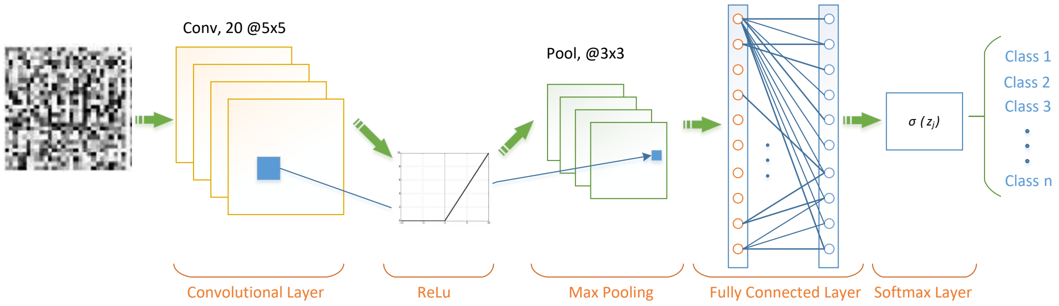


Figure 5.2: The schematic of proposed CNN with one convolutional layer.

5.3.2 CNN with Two convolutional Layer

This structure is similar to the previous one, with the difference of having another convolutional layer at the beginning. The reason for adding this layer is that it increases the nonlinearity of the network and helps to increase the chance of classification of nonlinear systems especially in noisy conditions [15]. The sequences of this structure can be seen in Figure 5.3.

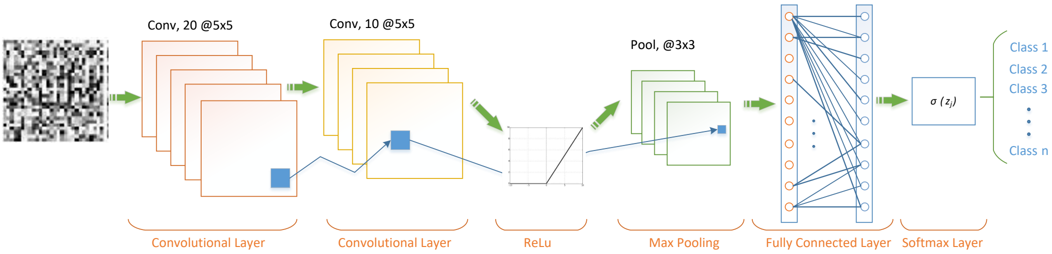


FIGURE 5.3: The schematic of proposed CNN with two convolutional layers.

5.3.3 Simulation

In this section, the proposed structures in Figures 5.3.1, and 5.3.2, are simulated with the data from a wind turbine benchmark, introduced in section 2.1. The internal model of the system is not important in this case, as the approach is fully data-driven. The only essential information about the benchmark is that one of its outputs is generator torque (τ_g). To become more similar to a real condition, it is assumed that there is a noisy sensor for gathering the data of τ_g with the variance of 0.3% and the mean value of 0. It is noted that the noisy data makes the classification harder for complex signals.

In addition, it is considered to have a 2 to 5% actuator effectiveness loss on generator torque reference. The low percentage of fault is assumed, since it is important to show that the algorithm can handle even small amounts of faults. This very low amount of fault along with sensor noise can cause even harder classification. In this condition, a dataset consists of 4000 records of τ_g are saved in the times-series signal format. The sampling time is 1 second and each record contains 4900 samples. In these records, 2000 records are faulty and the rest, which are 2000, are healthy. From each category, 80% is being separated randomly for the training dataset and 20% goes to the testing dataset. As it is obvious, the records for training and testing are completely different.

The next step is to preprocess the data as discussed in section 5.2. Each time-series signal is converted to a 2-D image. A random record from each category is shown in Figure 5.4. As it is visible that the difference between the faulty and healthy records are not distinguishable, neither in the time domain nor in the 2-D image. So, it is important to develop an approach to classify them.

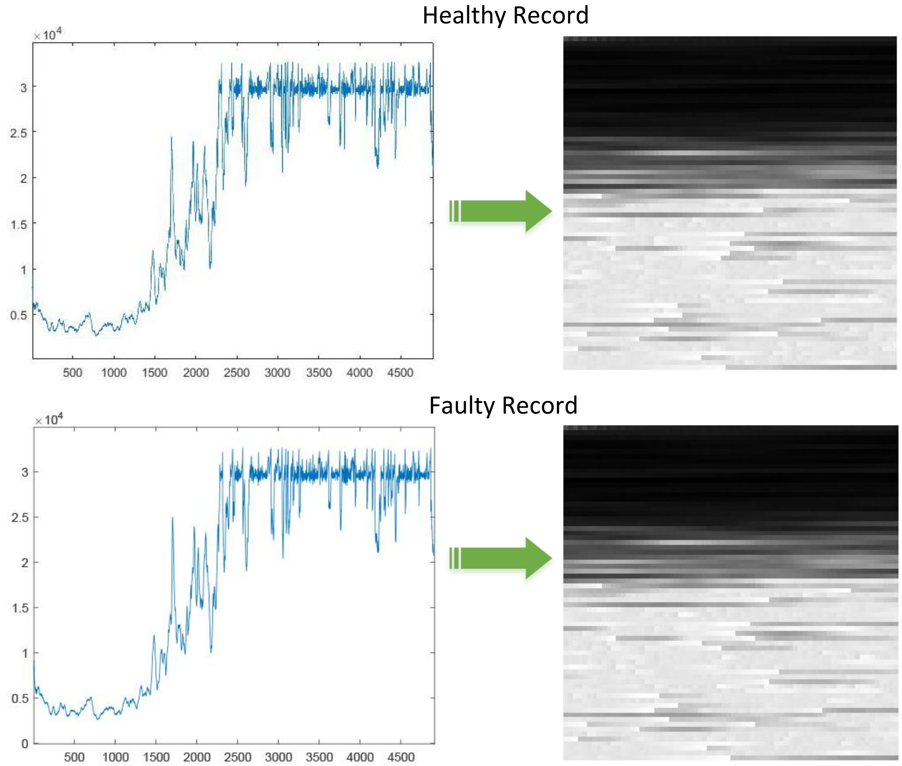
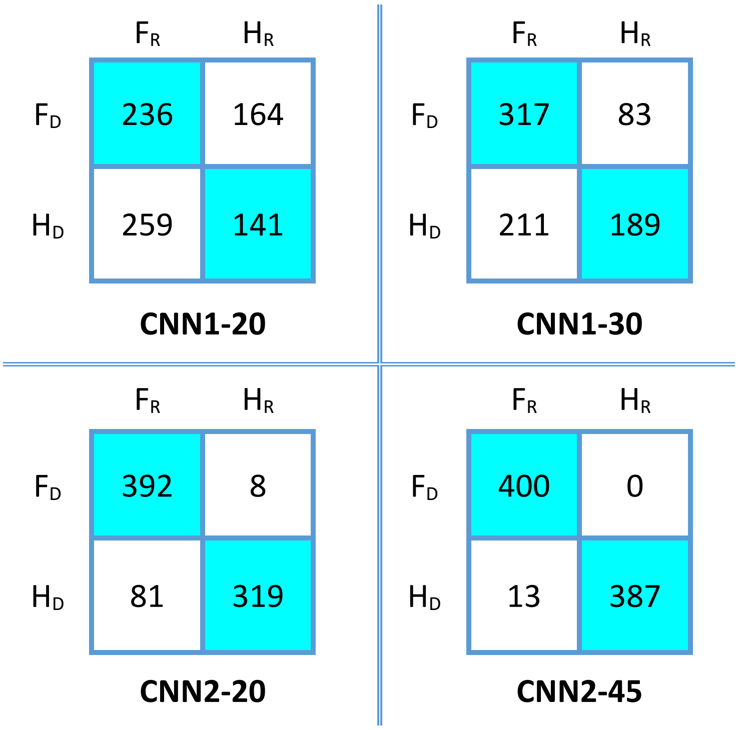


FIGURE 5.4: The conversion of raw data to 2-D gray scale image.

Now, that the data is prepared, two CNN structures with a different number of kernels are trained. One important point is that the CNN is categorised in supervised learning. Therefore, the desired output should be prepared. Here based on 80 % of records, applied for the training stage, an output vector is created. Its cells are 0 and 1, for faulty input and healthy one, respectively.

The classification results for a test dataset of 800 records (20% of the records) are depicted in Figure 5.5. From 800 records, 400 of them are healthy and 400 belong to the faulty class. In this figure, CNN1-20 is the CNN structure with one convolutional layer and 20 kernels. CNN1-30 is the same structure, but with 30 kernels. CNN2-20 is the CNN structure with two convolutional layers and 20 kernels. At last, CNN2-45 is the same two-layer structure with 45 kernels. The F_D and H_D are the desired output for faulty and healthy records and the F_R and H_R are the result from the network as faulty and healthy. For example in the CNN2-20 can estimate just 392 out of 400 faulty records correctly. It classifies the other 8 records, as healthy which is incorrect. In addition, the results show that the classification in the CNN structure with two convolutional layers and 45 kernels are the most accurate one.



	F_R	H_R
F_D	236	164
H_D	259	141

CNN1-20

	F_R	H_R
F_D	317	83
H_D	211	189

CNN1-30

	F_R	H_R
F_D	392	8
H_D	81	319

CNN2-20

	F_R	H_R
F_D	400	0
H_D	13	387

CNN2-45

Figure 5.5: Classification of testing dataset of Scenario 1.

In Table 5.1, the accuracy of the considered CNN structures is compared. As one can see, CNN2 does show great improvement from CNN1. It means that adding one layer of convolution to the network can definitely increase the performance of the network in classification. The other important observation is that increasing the number of kernels, can effectively improve the accuracy of the network. In other words, each kernel represents one feature extractor. Therefore, it is completely logical to see increasing of features can lead to an increase in the accuracy.

Table 5.1: Comparison results between different CNN structures in Scenario 1.

Methods	**Accuracy (%)**
CNN1-20	47.12
CNN1-30	63.25
CNN2-20	88.87
CNN2-45	98.37

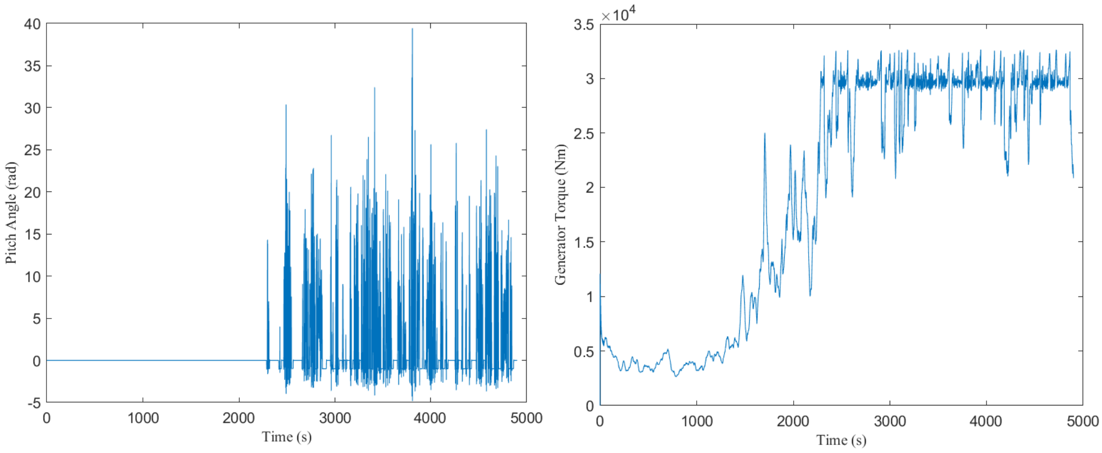
The other important factor is the computational time for each structure. It is plausible that CNN1 has lower computational costs than CNN2, due to one lesser convolutional layer. However, another factor is the quantity of the kernels. The more kernels involved in a layer of a network, the more time it spends to converge. In this problem, the time spending in CNN2-45 is about 4 times that required for the CNN1-30.

5.4 Scenario 2: Two Actuator Faults

In this section, the approach of CNN fault detection is developed into two actuator faults. For achieving this goal, first, we convert our time-series data into greyscale images as introduced in section 5.2 and then, apply the dataset to the previous structure CNN2-45. We also introduce another CNN structure, CNN3-32, to train with the dataset and compare the results.

5.4.1 Converting Two Signals into One Image

The healthy forms of the two actuators of the wind turbine benchmark are shown in Figure 5.6. As can be seen that, the natures of the two signals are completely different. Working with a signal like pitch angle, which variates harshly makes the classification much more difficult.



(a) Healthy Pitch Angle Signal.

(b) Healthy Generator Torque Signal.

FIGURE 5.6: Healthy form of the two actuators' signals.

Based on the idea of converting time-series sequences into images in section 5.2, we can convert each pair of actuators' signals into one image. The process underlying can be illustrated as Figure 5.7. Consequently, the image related to the healthy signals of Figure 5.6 can be illustrated in Figure 5.8.

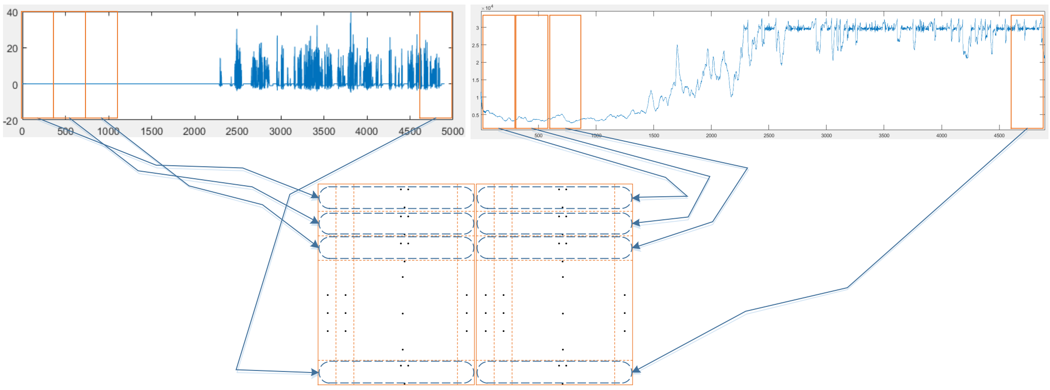


FIGURE 5.7: The process of converting a pair of signals into one image.

FIGURE 5.8: The converted image of a pair of healthy actuators' signals.

5.4.2 CNN Structure

This structure of CNN3-32 is similar to the ones, introduced in section 5.3, with the difference of having three convolutional layers. The reason for adding another layer is that, by adding a signal, which has high-frequency variation such as pitch angle, it is better to increase the nonlinearity of the model in order to cope with this problem. The sequences of this structure can be seen in Fig. 5.9.

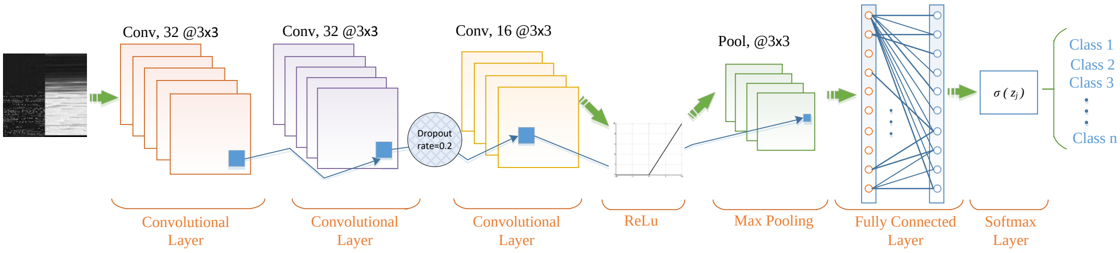


FIGURE 5.9: The schematic of proposed CNN with three convolutional layers.

As one can see from the above figure, another convolutional layer is added to the structure. The other layer, which is new in this figure, is Dropout. In the Dropout layer, some of the neurons are *droped out*, since they are very similar to the other ones. This layer is a simple solution to prevent CNN from overfitting [15].

5.4.3 Simulation

In this section, the proposed structures in sections 5.9, and 5.3.2, are trained and simulated with a dataset consists of 4000 records of two actuator signals, τ_g and

β. In addition, it is considered to have a Gaussian noise with a variance of 0.3% and the mean value of 0 on each signal.

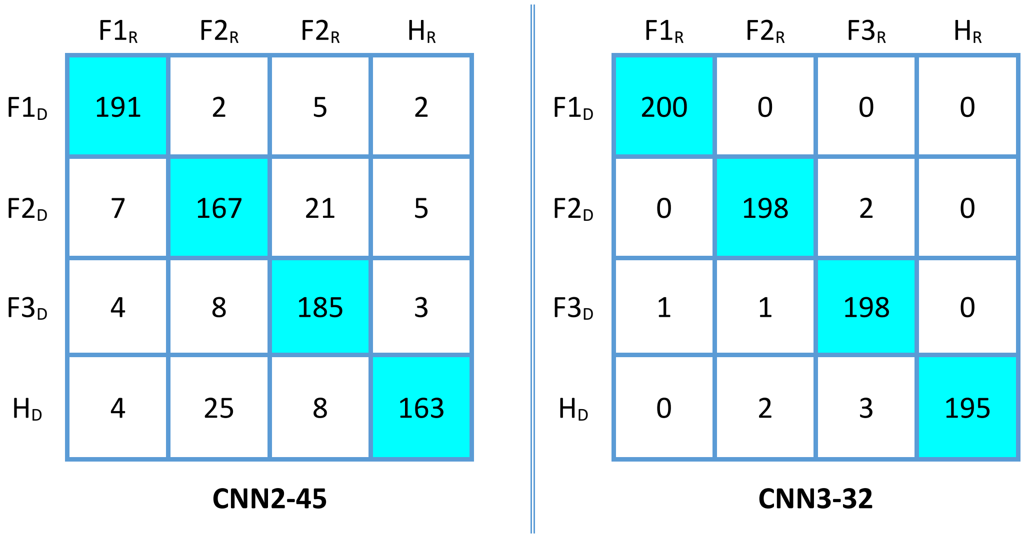
The faults considered on each actuator signal are 2 to 5% of effectiveness loss. The sampling time is 1 second and each record contains 5000 samples. In this scenario, it is considered to have four classes. The first one is when both of the signals are healthy (H). The second one is when the generator torque is faulty (F1). The third class is when the pitch angle contains a faulty interval (F3). The last class is when both of the signals are faulty (F3). It is noted that, in this scenario, in some of the records, generator torque fault and pitch angle fault happen at the same intervals. Each of the classes contains 1000 records. From each category, 80% is being separated randomly for the training dataset and 20% goes to the testing dataset. Again, similar to the previous scenario, the records for training and testing are completely different.

After converting each record to an image, as depicted in Figure 5.8, two CNN structures of CNN2-45 and CNN3-32 are trained by the prepared dataset. The results of 800 records (20% of the records), which belong to the testing part is brought in Figure 5.10. The accuracy of each structure is also compared in Table 5.2

Table 5.2: Comparison results between different CNN structures in Scenario 2.

Methods	**Accuracy (%)**
CNN2-45	88.25
CNN3-32	98.87

As it is obvious in Table 5.2, the accuracy of the CNN3-32 is much better than the previous structure. It is noted that although CNN2-45 has a great performance in



	$F1_R$	$F2_R$	$F2_R$	H_R
$F1_D$	191	2	5	2
$F2_D$	7	167	21	5
$F3_D$	4	8	185	3
H_D	4	25	8	163

CNN2-45

	$F1_R$	$F2_R$	$F3_R$	H_R
$F1_D$	200	0	0	0
$F2_D$	0	198	2	0
$F3_D$	1	1	198	0
H_D	0	2	3	195

CNN3-32

FIGURE 5.10: Classification of testing dataset of Scenario 2.

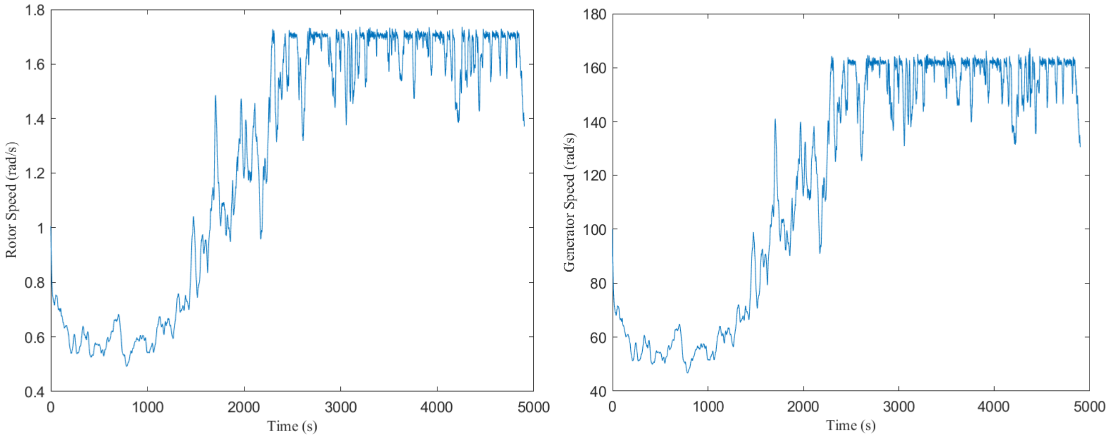
scenario 1, its performance degrades when having a signal with a high-frequency variation. It is also worth to mention that a more complicated structure and using the Dropout layer can increase the accuracy.

5.5 Scenario 3: Four Sensor Faults

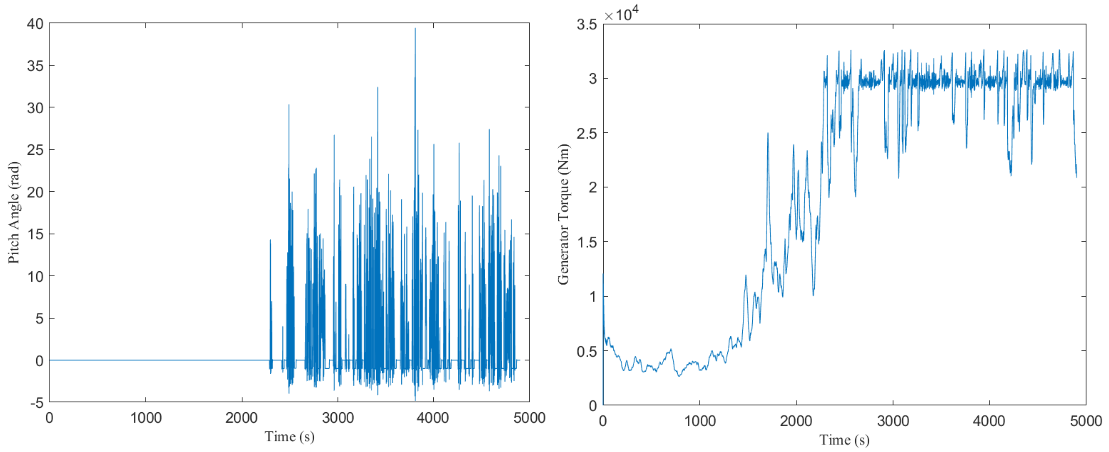
Following the previous sections, in this section, the CNN fault detection method is further developed into four sensor faults. The procedure for achieving this goal, is similar to the other two scenarios with little adjustments. All three CNN structures, introduced before, are tested in this scenario. In addition, a new and more complicated structure is also trained for this dataset, to be able to deal with more complex dataset. The results of them are compared at the end of this section.

5.5.1 Converting Four Signals into One Image

The healthy signals of the four sensors of the wind turbine benchmark can be seen in Fig. 5.11.



(a) Healthy Rotor Speed Signal. (b) Healthy Generator Speed Signal.



(c) Healthy Pitch Angle Signal. (d) Healthy Generator Torque Signal.

Figure 5.11: Healthy form of the four sensors' signals.

As introduced in section 5.2, we converted all of the output data into greyscale images. The format of converting can be seen in Fig. 5.12. The greyscale result for a sample healthy record is presented in Fig. 5.13.

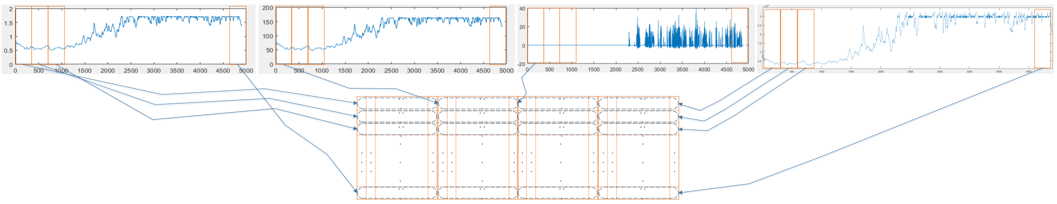


Figure 5.12: The process of converting four signals into one image.

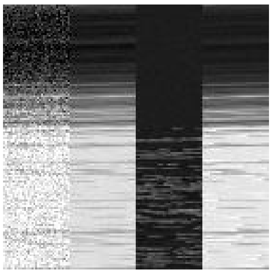


FIGURE 5.13: The converted image of four healthy sensors' signals.

5.5.2 CNN Structure

Based on the structure of CNN3-32, CNN4-128 structure is introduced by adding a fourth layer of convolution and increasing the number of kernels in the first layer. As it is mentioned earlier, increasing the nonlinearity of the model, helps the accuracy of training for a dataset with high-frequency variation. The sequences of CNN4-128 can be seen in Fig. 5.9.

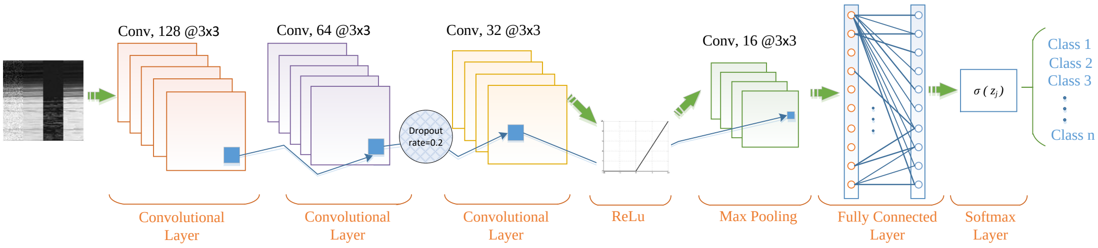


FIGURE 5.14: The schematic of proposed CNN with four convolutional layers.

As can be seen in this figure, another convolutional layer is added to the structure. Increasing the number of kernels can also help coping with the problem of high-frequency variation of the dataset.

5.5.3 Simulation

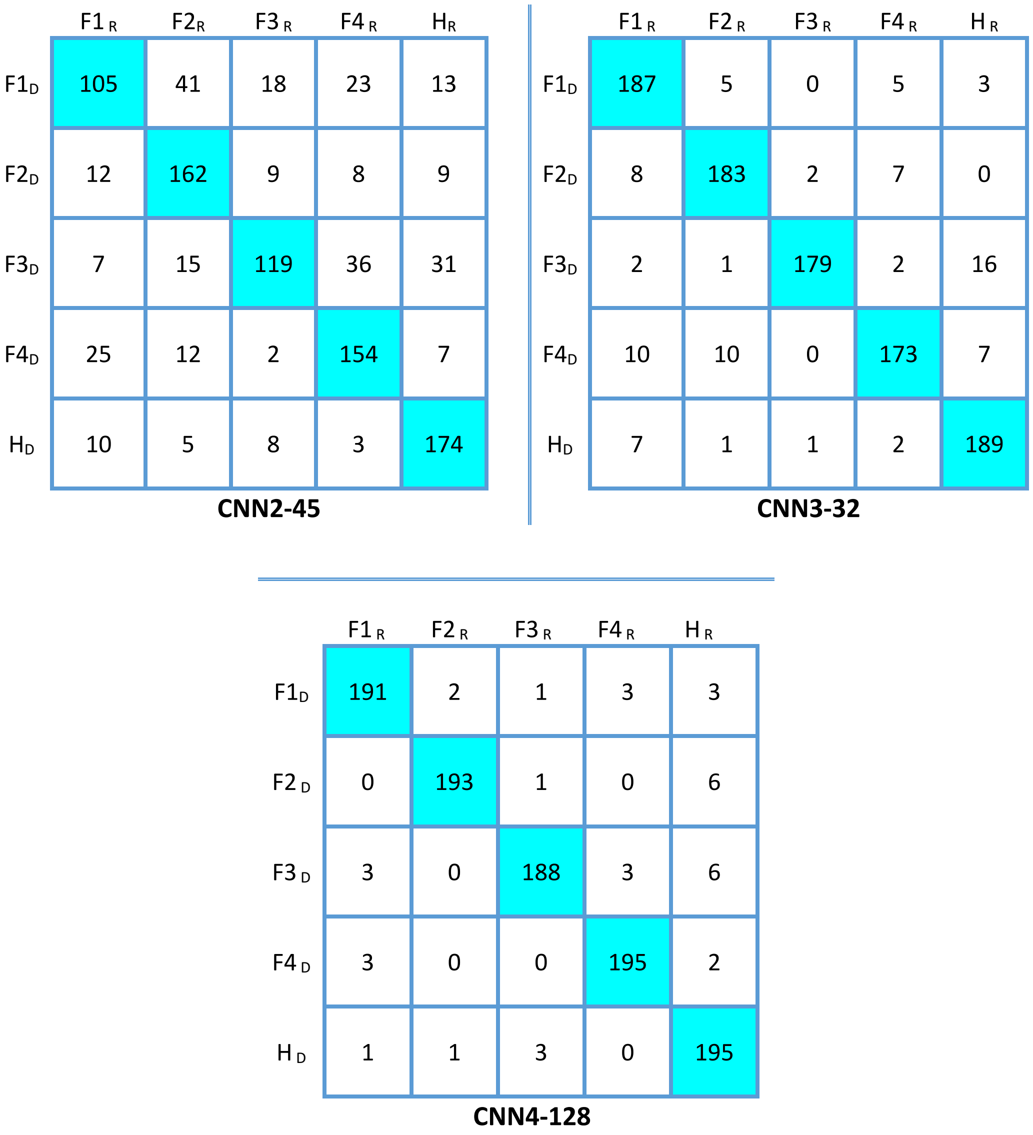
In this section, all of the former proposed structures along with the CNN4-128 are trained and simulated with a dataset consists of 5000 records of four sensor signals, ω_r, ω_g, β, and τ_g. Moreover, to have a more real data, it is considered to have a Gaussian noise with a variance of 0.3% and the mean value of 0 on each signal.

As the same as previous scenarios, the faults considered on each sensor signal are 2 to 5% of effectiveness loss. The sampling time is 1 second and each record contains 4900 samples. For this scenario, it is considered to have five classes. The first one is when all four signals are healthy (H). The second one is when the rotor speed (ω_r) is faulty (F1). The third class is when the generator speed (ω_g) is faulty (F2). The fourth class contains the fault on pitch angle (β) (F3). And, the last class is when there is a fault on generator torque (τ_g) (F4). It is noted that, in this scenario, it is assumed that there are no simultaneous faults on two or more sensors. Each of the classes contains 1000 records. From each category, 80% is being separated randomly for the training dataset and 20% goes to the testing dataset. Again, similar to the previous scenarios, the records for training and testing are completely different.

After converting each record to an image, as depicted in Fig. 5.13, three CNN structures of CNN2-45, CNN3-32, and CNN4-128 are trained by the prepared dataset. The results of 1000 records (20% of the records), which belong to the testing part is brought in Fig. 5.15. The accuracy of each structure is also compared in Table 5.3

TABLE 5.3: Comparison results between different CNN structures in Scenario 3.

Methods	**Accuracy (%)**
CNN2-45	71.4
CNN3-32	91.1
CNN4-128	96.2



	$F1_R$	$F2_R$	$F3_R$	$F4_R$	H_R
$F1_D$	105	41	18	23	13
$F2_D$	12	162	9	8	9
$F3_D$	7	15	119	36	31
$F4_D$	25	12	2	154	7
H_D	10	5	8	3	174

CNN2-45

	$F1_R$	$F2_R$	$F3_R$	$F4_R$	H_R
$F1_D$	187	5	0	5	3
$F2_D$	8	183	2	7	0
$F3_D$	2	1	179	2	16
$F4_D$	10	10	0	173	7
H_D	7	1	1	2	189

CNN3-32

	$F1_R$	$F2_R$	$F3_R$	$F4_R$	H_R
$F1_D$	191	2	1	3	3
$F2_D$	0	193	1	0	6
$F3_D$	3	0	188	3	6
$F4_D$	3	0	0	195	2
H_D	1	1	3	0	195

CNN4-128

FIGURE 5.15: Classification of testing dataset of Scenario 3.

As illustrated in Table 5.3, the accuracy of the CNN4-128 is much higher than the previous structures. It is noted that by complicating the input images, the necessity of complex structures is increasing. This means, more convolutional layers, more kernels, and also Dropout layers, are all helpful to reach the higher accuracy in fault detection. Another point here, is having a very perturbed signal as β makes the prediction very hard. So at the end, reaching higher accuracy, like 97% is very challenging. This problem is clearly distinguishable in fig. 5.14, where in the main diagonal of CNN4-128, the minimum value is related to predicting the category of β.

5.6 Summary and Conclusions

In this chapter, four CNN-based structures have been proposed in order to have a deep learning fault detection. The novel contribution in this chapter, is proposing a data-to-image conversion stage and proposing the suitable deep learning structure to handle this problem. For that, the first problem was that working with a time-series signal was so tricky. Therefore, a preprocessing stage was discussed to prepare the raw data into 2-D greyscale images. Then, two different scenarios were investigated. In the first scenario, two CNN structures were proposed and trained with a dataset in order to classify the faulty and healthy signals. As the data contains sensor noises, it was obvious that classification would become more difficult. However, the simulations showed that adding a convolutional layer to the model can increase the accuracy of the validation. It was also concluded, by increasing the number of kernels in each structure, the accuracy increased and reached to 98.37% with 45 kernels, while the computational costs are also increased.

In the second scenario, the proposed approach was developed to cope with two faulty signals, which might have faults in the same intervals. The simulation validated the effectiveness of adding another convolutional layer and also a Dropout layer by having 98.87% accuracy in a dataset of 4000 records.

In the last scenario, a CNN structure was studied to have a fault detection method for four sensor signals. As the simulation accurately showed, adding a layer of convolution had effects on increasing the accuracy to 96.2% in 5000 records.

Chapter 6

Conclusions and Future Works

6.1 Summary and Conclusions

In this thesis, the main focus is on developing fault diagnosis techniques based on Neural Networks and deep learning approaches. It is worth mentioning that to achieve this goal, several mathematical and computational methods are investigated. For instance, BPNN, Luenberger observer, residual calculation, augmented system, robust optimization, LMI, ISS, Lyapunov function, deep learning, and CNN. In the following, the main contributions of this thesis are discussed:

- Grey-box model identification and fault detection using artificial neural networks.

  Based on the fact that building a very accurate model of many industrial systems is challenging, it is beneficial to design an identification observer. In Chapter 3, an ANN based identification method along with Luenberger observer has been proposed to deal with the nonlinearity and unmeasurability of a wind turbine system. A residual fault detection has been applied based on this approach. The results of the applied case study, including 2% actuator fault, has shown the accuracy of the proposed algorithm.

- Robust neural network fault estimation approach for nonlinear dynamic systems.

  The other issues in a typical wind turbine are tackling with environmental disturbances and sensor noises along with unexpected errors in actuators and sensors. In Chapter 4, two novel robust neural network fault estimation methods have been proposed to deal with these problems. In these approaches, by applying the idea of BPNN to robust optimization criteria, different scenarios of faults in actuators and sensors have been studied. A very critical point in designing the mentioned fault estimation methods is their stability. To cover this point, a Lyapunov function has been proposed, and by applying input-to-state stability criteria, the stability of the system has been proved. Two different scenarios have been discussed, one related to actuator faults and the other related to sensor faults. Applied fault estimation methods on both scenarios has validated the effectiveness of the approaches.

- Time-series deep learning fault detection.

  Deep learning techniques can help to deal with a large amount of data in industrial systems with a lot of differnt applications. To benefit from this potential, scenarios of having faults on one or both of the actuators or one of the sensors in the wind turbine benchmark have been studied. A novel CNN based fault classification method has been proposed, and the pre-processing analysis has been carried out. The method has been applied to the aforementioned scenarios with different structures of CNN. The accuracy of the proposed method can be up to 98.87% in the available dataset.

6.2 Future Works

Based on this PhD thesis, it is suggested to carry out the following research topics:

- Robust neural network fault prognosis for nonlinear dynamics.

  Based on the fact that ANN and CNN have the potential of predicting the nonlinear models, it would be encouraging to develop prognostics algorithms for wind turbine systems by using these techniques.

- Time-series deep learning fault detection for physical environmental problems.

  Some physical environmental problems, such as temperature, humidity and vibrations, can influence both of actuators' performances. Fault classification can help find the cause of performance reduction, which can be very helpful in maintenance.

- Time-series deep learning fault detection for sensor losses.

  Due to the fact that sensors are crucial components in a stable control design, it is essential to make sure that they are working correctly. Therefore, a fault detection method to investigate the losses of the sensors seems quite useful.

- Time-series deep learning fault estimation for data losses.

  Based on the previously mentioned work, a fault estimation can be proposed based on deep learning techniques to have a regression for future output of each sensor in the occurrence of sensor losses. This idea helps to assure the control system works properly even though the sensors are faulty for a short period of time.

Appendix A

Simulation of Wind Turbine Benchmark in Matlab

The simulation of the wind turbine benchmark in this thesis have been done in Matlab. Different blocks of it can be seen in Figures A.1, A.2, A.3, A.4, and A.5. In all of these figures, the magenta terminals are the signals related to the states of the system, the cyan ones are the output of controller, and the green ones are the measured signals. In theory, the measured signals are the same as the states of the system. However, to make the simulation more compatible to the real world, sensor noises are added to the outputs of the system and the measured signals are provided.

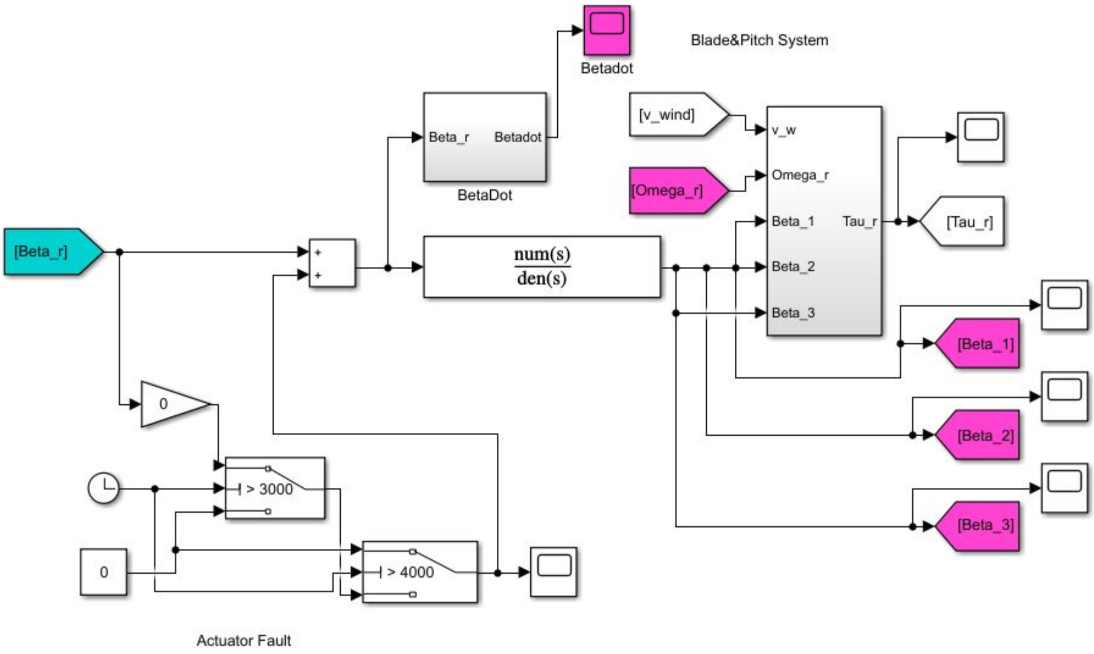


FIGURE A.1: Blade & Pitch System with Actuator Fault Blocks.

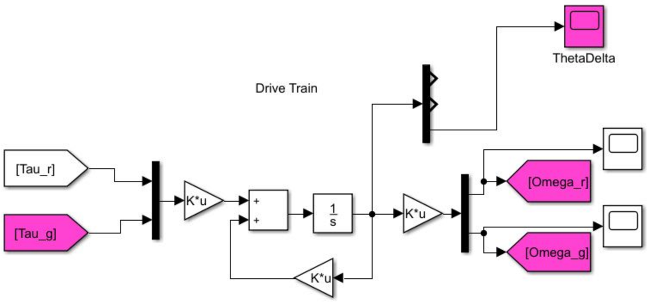


FIGURE A.2: Drive Train.

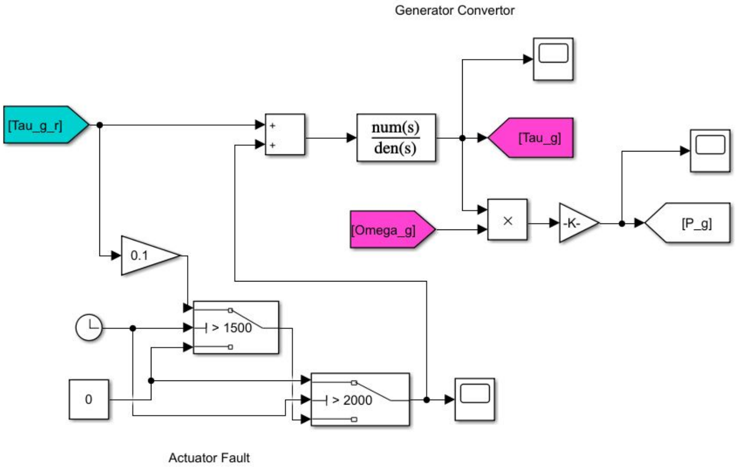


FIGURE A.3: Generator System with Actuator Fault Blocks.

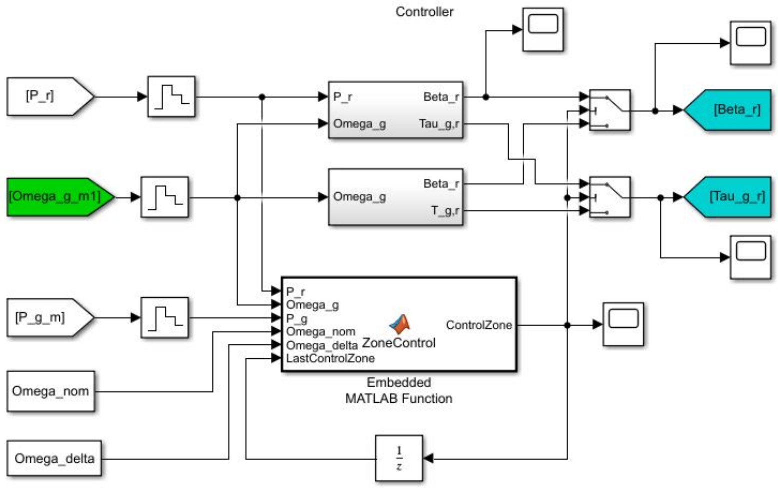


FIGURE A.4: Controller.

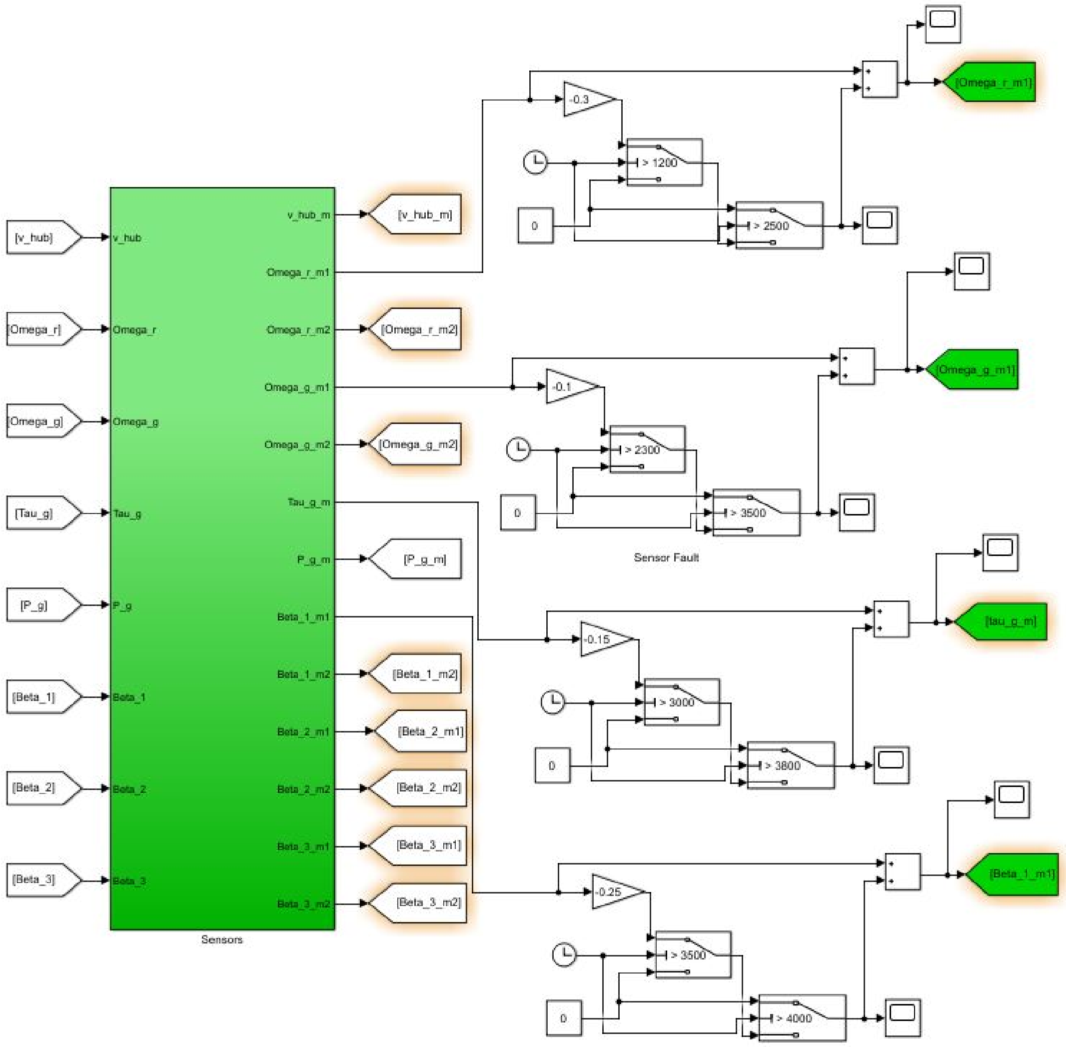


FIGURE A.5: Sensors with Fault Blocks.

References

[1] E. . I. S. Department for Business, “Digest of united kingdom energy statistics 2020,” *National Statistics Publication*, Jul. 2020.

[2] W. Qiao and D. Lu, “A survey on wind turbine condition monitoring and fault diagnosis—part i: Components and subsystems,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6536–6545, 2015.

[3] W. Qiao and D. Lu, “A survey on wind turbine condition monitoring and fault diagnosis—part ii: Signals and signal processing methods,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6546–6557, 2015.

[4] J. Ribrant and L. Bertling, “Survey of failures in wind power systems with focus on swedish wind power plants during 1997-2005,” in *2007 IEEE power engineering society general meeting*, pp. 1–8, IEEE, 2007.

[5] Y. Qiu, Y. Feng, P. Tavner, P. Richardson, G. Erdos, and B. Chen, “Wind turbine scada alarm analysis for improving reliability,” *Wind Energy*, vol. 15, no. 8, pp. 951–966, 2012.

[6] F. Cheng, Y. Peng, L. Qu, and W. Qiao, “Current-based fault detection and identification for wind turbine drivetrain gearboxes,” *IEEE Transactions on Industry Applications*, vol. 53, no. 2, pp. 878–887, 2016.

[7] J. Liniger, S. Stubkier, M. Soltani, and H. C. Pedersen, "Early detection of coil failure in solenoid valves," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 683–693, 2020.

[8] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using lstm networks and svm," *IEEE Access*, vol. 6, pp. 7675–7686, 2017.

[9] Z. Xu, Y. Ji, and D. Zhou, "A new real-time reliability prediction method for dynamic systems based on on-line fault prediction," *IEEE transactions on reliability*, vol. 58, no. 3, pp. 523–538, 2009.

[10] X. Liu, Z. Gao, and A. Zhang, "Observer-based fault estimation and tolerant control for stochastic takagi–sugeno fuzzy systems with brownian parameter perturbations," *Automatica*, vol. 102, pp. 137–149, 2019.

[11] V. John, K. Yoneda, Z. Liu, and S. Mita, "Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching," *IEEE transactions on computational imaging*, vol. 1, no. 3, pp. 159–173, 2015.

[12] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[13] Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for credit card fraud detection," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 569–579, 2020.

[14] Z. Zhang, L. Chen, Q. Liu, and P. Wang, "A fraud detection method for low-frequency transaction," *IEEE Access*, vol. 8, pp. 25210–25220, 2020.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[16] W. G. Hatcher and W. Yu, “A survey of deep learning: Platforms, applications and emerging research trends,” *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[17] E. Hau, *Wind turbines: fundamentals, technologies, application, economics.* Springer Science & Business Media, 2013.

[18] B. Chen, “Automated on-line fault prognosis for wind turbine monitoring using scada data,” *PhD Dissertation, Durham University*, 2014.

[19] F. D. Bianchi, H. De Battista, and R. J. Mantz, *Wind turbine control systems: principles, modelling and gain scheduling design.* Springer Science & Business Media, 2006.

[20] X. Liu, “Fault estimation and fault tolerant control with application to wind turbine systems,” *PhD Dissertation, University of Northumbria*, 2017.

[21] P. Odgaard, J. Stoustrup, , and M. Kinnaert, “Fault tolerant control of wind turbines: a benchmark model,” *IEEE transaction on control systems technology*, vol. 21, no. 4, pp. 1168–1182, 2013.

[22] X. Liu, Z. Gao, and M. Z. Chen, “Takagi–sugeno fuzzy model based fault estimation and signal compensation with application to wind turbines,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5678–5689, Jul. 2017.

[23] B. Hahn, M. Durstewitz, and K. Rohrig, “Reliability of wind turbines,” in *Wind energy*, pp. 329–332, Springer, 2007.

[24] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques part i: Fault diagnosis with model-based and signal based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, p. 3757–3767, Jun. 2015.

[25] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[26] J. Su and W.-H. Chen, "Model-based fault diagnosis system verification using reachability analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–10, Jul. 2017.

[27] J. Chen and R. Patton, "Robust model-based fault diagnosis for dynamic systems," *Kluwer Academic Publishers, Boston, USA*, 1999.

[28] R. Rahimilarki and Z. Gao, "Grey-box model identification and fault detection of wind turbines using artificial neural networks," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 647–652, IEEE, 2018.

[29] Y. Zhao, Y. Liu, and R. Wang, "Fuzzy scalar quantisation based on hidden markov model and application in fault diagnosis of wind turbine," *The Journal of Engineering*, vol. 2017, no. 14, pp. 2685–2689, May 2017.

[30] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.

[31] H. Sanchez, T. Escobet, V. Puig, and P. F. Odgaard, "Fault diagnosis of an advanced wind turbine benchmark using interval-based arrs and observers," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3783–3793, 2015.

[32] B. Shafai, C. Pi, and S. Nork, "Simultaneous disturbance attenuation and fault detection using proportional integral observers," in *Proceedings of*

the 2002 American Control Conference (IEEE Cat. No. CH37301), vol. 2, pp. 1647–1649, IEEE, 2002.

[33] H. Badihi, Y. Zhang, and H. Hong, "Wind turbine fault diagnosis and fault-tolerant torque load control against actuator faults," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1351–1372, 2015.

[34] L. Xu and H. E. Tseng, "Robust model-based fault detection for a roll stability control system," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 519–528, May 2007.

[35] W. Tang, Z. Wang, and Y. Shen, "Fault detection and isolation for discrete-time descriptor systems based on h/l-inf observer and zonotopic residual evaluation," *International Journal of Control*, vol. 93, no. 8, pp. 1867–1878, 2020.

[36] X.-G. Yan and C. Edwards, "Adaptive sliding-mode-observer-based fault reconstruction for nonlinear systems with parametric uncertainties," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 11, pp. 4029–4036, 2008.

[37] H. Alwi and C. Edwards, "An adaptive sliding mode differentiator for actuator oscillatory failure case reconstruction," *Automatica*, vol. 49, no. 2, pp. 642–651, 2013.

[38] H. Chen and S. Lu, "Fault diagnosis digital method for power transistors in power converters of switched reluctance motors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 749–763, Feb. 2013.

[39] N. Wassinger, E. Penovi, R. G. Retegui, and S. Maestri, "Open-circuit fault identification method for interleaved converters based on time-domain analysis of the state observer residual," *IEEE Transactions on Power Electronics*, vol. 34, no. 4, pp. 3740–3749, 2018.

[40] H. Chen and S. Lu, "Fault diagnosis digital method for power transistors in power converters of switched reluctance motors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 749–763, 2012.

[41] N. Su, X. Li, and Q. Zhang, "Fault diagnosis of rotating machinery based on wavelet domain denoising and metric distance," *IEEE Access*, vol. 7, pp. 73262–73270, 2019.

[42] J. Burriel-Valencia, R. Puche-Panadero, J. Martinez-Roman, A. Sapena-Bano, and M. Pineda-Sanchez, "Short-frequency fourier transform for fault diagnosis of induction machines working in transient regime," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 3, pp. 432–440, 2017.

[43] K. Satpathi, Y. M. Yeap, A. Ukil, and N. Geddada, "Short-time fourier transform based transient analysis of vsc interfaced point-to-point dc system," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4080–4091, 2017.

[44] Y. Fu, Z. Gao, Y. Liu, A. Zhang, and X. Yin, "Actuator and sensor fault classification for wind turbine systems based on fast fourier transform and uncorrelated multi-linear principal component analysis techniques," *Processes*, vol. 8, no. 9, p. 1066, 2020.

[45] Y. Qin, J. Zou, and F. Cao, "Adaptively detecting the transient feature of faulty wind turbine planetary gearboxes by the improved kurtosis and iterative thresholding algorithm," *IEEE Access*, vol. 6, pp. 14602–14612, Mar. 2018.

[46] J. Wang, F. Cheng, W. Qiao, and L. Qu, "Multiscale filtering reconstruction for wind turbine gearbox fault diagnosis under varying-speed and noisy

conditions," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4268–4278, May 2018.

[47] X. Gong and W. Qiao, "Bearing fault diagnosis for direct-drive wind turbines via current-demodulated signals," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3419–3428, 2013.

[48] N. M. Freire, J. O. Estima, and A. J. M. Cardoso, "Open-circuit fault diagnosis in pmsg drives for wind turbine applications," *IEEE Transactions on Industrial electronics*, vol. 60, no. 9, pp. 3957–3967, 2013.

[49] I. Zamudio-Ramirez, J. A. Antonino-Daviu, R. A. Osornio-Rios, R. de Jesus Romero-Troncoso, and H. Razik, "Detection of winding asymmetries in wound-rotor induction motors via transient analysis of the external magnetic field," *IEEE Transactions on Industrial Electronics*, 2019.

[50] Y. Fu, Y. Liu, A. Zhang, and Z. Gao, "Multiple actuator fault classification for wind turbine systems by integrating fast fourier transform (fft) and multilinear principal component analysis (mpca)," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 3761–3766, IEEE, 2019.

[51] Y. Fu, Y. Liu, and Z. Gao, "Fault classification in wind turbines using principal component analysis technique," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, pp. 1303–1308, IEEE, 2019.

[52] Y. Jiang, S. Yin, and O. Kaynak, "Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond," *IEEE Access*, vol. 6, pp. 47374–47384, Aug. 2018.

[53] S. Yin, C. Yang, J. Zhang, and Y. Jiang, "A data-driven learning approach for nonlinear process monitoring based on available sensing measurements,"

IEEE Transactions on Industrial Electronics, vol. 64, no. 1, pp. 643–653, Jan. 2017.

[54] Y. Jiang and S. Yin, “Recent advances in key-performance-indicator oriented prognosis and diagnosis with a matlab toolbox: Db-kit,” *IEEE Transactions on Industrial Informatics*, Oct. 2018.

[55] B. Gou, Y. Xu, Y. Xia, G. Wilson, and S. Liu, “An intelligent time-adaptive data-driven method for sensor fault diagnosis in induction motor drive system,” *IEEE Transactions on Industrial Electronics*, 2018.

[56] B. Xu, X. Yin, X. Yin, Y. Wang, and S. Pang, “Fault diagnosis of power systems based on temporal constrained fuzzy petri nets,” *IEEE Access*, vol. 7, pp. 101895–101904, 2019.

[57] H. Yan, Y. Xu, F. Cai, H. Zhang, W. Zhao, and C. Gerada, “Pwm-vsi fault diagnosis for a pmsm drive based on the fuzzy logic approach,” *IEEE Transactions on Power Electronics*, vol. 34, no. 1, pp. 759–768, 2018.

[58] H. Malik and S. Mishra, “Proximal support vector machine (psvm) based imbalance fault diagnosis of wind turbine using generator current signals,” *Energy Procedia*, vol. 90, pp. 593–603, 2016.

[59] Q. Shi and H. Zhang, “Fault diagnosis of an autonomous vehicle with an improved svm algorithm subject to unbalanced datasets,” *IEEE Transactions on Industrial Electronics*, 2020.

[60] R. Yang, M. Huang, Q. Lu, and M. Zhong, “Rotating machinery fault diagnosis using long-short-term memory recurrent neural network,” *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 228–232, 2018.

[61] L. Cao, J. Zhang, J. Wang, and Z. Qian, “Intelligent fault diagnosis of wind turbine gearbox based on long short-term memory networks,” in *2019 IEEE*

28th International Symposium on Industrial Electronics (ISIE), pp. 890–895, IEEE, 2019.

[62] G. Iannace, G. Ciaburro, and A. Trematerra, “Fault diagnosis for uav blades using artificial neural network,” *Robotics*, vol. 8, no. 3, p. 59, 2019.

[63] W. Huang, J. Cheng, Y. Yang, and G. Guo, “An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis,” *Neurocomputing*, vol. 359, pp. 77–92, 2019.

[64] M. Li, D. Yu, Z. Chen, K. Xiahou, T. Ji, and Q. Wu, “A data-driven residual-based method for fault diagnosis and isolation in wind turbines,” *IEEE Transactions on Sustainable Energy*, vol. 10, no. 2, pp. 895–904, 2018.

[65] J. Lei, C. Liu, and D. Jiang, “Fault diagnosis of wind turbine based on long short-term memory networks,” *Renewable energy*, vol. 133, pp. 422–432, 2019.

[66] S. Simani, S. Farsoni, and P. Castaldi, “Fault diagnosis of a wind turbine benchmark via identified fuzzy models,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3775–3782, Jun. 2015.

[67] J. Hang, J. Zhang, and M. Cheng, “Application of multi-class fuzzy support vector machine classifier for fault diagnosis of wind turbine,” *Fuzzy Sets and Systems*, vol. 297, pp. 128–140, 2016.

[68] M. Kordestani, M. Rezamand, R. Carriveau, D. S. Ting, and M. Saif, “Failure diagnosis of wind turbine bearing using feature extraction and a neuro-fuzzy inference system (anfis),” in *International Work-Conference on Artificial Neural Networks*, pp. 545–556, Springer, 2019.

[69] S. Afrasiabi, M. Afrasiabi, B. Parang, M. Mohammadi, M. M. Arefi, and M. Rastegar, "Wind turbine fault diagnosis with generative-temporal convolutional neural network," in *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–5, IEEE, 2019.

[70] H. Zhao, Y. Gao, H. Liu, and L. Li, "Fault diagnosis of wind turbine bearing based on stochastic subspace identification and multi-kernel support vector machine," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 350–356, 2019.

[71] W. Deng, S. Zhang, H. Zhao, and X. Yang, "A novel fault diagnosis method based on integrating empirical wavelet transform and fuzzy entropy for motor bearing," *IEEE Access*, vol. 6, pp. 35042–35056, 2018.

[72] J.-m. Zhang, Y.-q. Zhai, and S.-m. Wang, "Fault diagnosis of wind turbine gearbox based on the wavelet decomposition and least square support vector machine [j]," *Transducer and Microsystem Technologies*, vol. 1, 2011.

[73] J. Zhang, H. Sun, Z. Sun, W. Dong, and Y. Dong, "Fault diagnosis of wind turbine power converter considering wavelet transform, feature analysis, judgment and bp neural network," *IEEE Access*, vol. 7, pp. 179799–179809, 2019.

[74] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[75] F. Pelletier, C. Masson, and A. Tahan, "Wind turbine power curve modelling using artificial neural network," *Renewable Energy*, vol. 89, pp. 207–214, Apr. 2016.

[76] S. Kelouwani and K. Agbossou, “Nonlinear model identification of wind turbine with a neural network,” *IEEE Transaction on Energy Conversion*, vol. 19, no. 3, pp. 607–612, Sep. 2004.

[77] L. Romanski, J. Bieniek, P. Komarnicki, M. Debowski, and J. Detyna, “Estimation of operational parameters of the counter-rotating wind turbine with artificial neural networks,” *Archives of Civil and Mechanical Engineering*, vol. 17, no. 4, pp. 1019–1028, Sep. 2017.

[78] H. Malik and S. Mishra, “Artificial neural network and empirical mode decomposition based imbalance fault diagnosis of wind turbine using turbsim, fast and simulink,” *IET Renewable Power Generation*, vol. 11, no. 6, pp. 889–902, May 2017.

[79] R. Rahimilarki, Z. Gao, A. Zhang, and R. Binns, “Robust neural network fault estimation approach for nonlinear dynamic systems with applications to wind turbine systems,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6302–6312, 2019.

[80] R. Rahimilarki, Z. Gao, N. Jin, R. Binns, and A. Zhang, “Data-driven sensor fault estimation for the wind turbine systems,” in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pp. 1211–1216, IEEE, 2020.

[81] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[82] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[83] G. Jiang, H. He, J. Yan, and P. Xie, “Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3196–3207, 2019.

[84] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3137–3147, 2016.

[85] H. Shao, H. Jiang, H. Zhang, and T. Liang, "Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2727–2736, 2018.

[86] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, 2017.

[87] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[88] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[89] G. Liang, H. Hong, W. Xie, and L. Zheng, "Combining convolutional neural network with recursive neural network for blood cell image classification," *IEEE Access*, vol. 6, pp. 36188–36197, 2018.

[90] J. Wang, T. Zheng, P. Lei, and X. Bai, "Ground target classification in noisy sar images using convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4180–4192, 2018.

[91] W. Gong, H. Chen, Z. Zhang, M. Zhang, R. Wang, C. Guan, and Q. Wang, "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved cnn-svm and multichannel data fusion," *Sensors*, vol. 19, no. 7, p. 1693, 2019.

[92] D. Wang, Q. Guo, Y. Song, S. Gao, and Y. Li, "Application of multiscale learning neural network based on cnn in bearing fault diagnosis," *Journal of Signal Processing Systems*, vol. 91, no. 10, pp. 1205–1217, 2019.

[93] M. Xia, G. Han, Y. Zhang, J. Wan, *et al.*, "Intelligent fault diagnosis of rotor-bearing system under varying working conditions with modified transfer cnn and thermal images," *IEEE Transactions on Industrial Informatics*, 2020.

[94] L. Eren, T. Ince, and S. Kiranyaz, "A generic intelligent bearing fault diagnosis system using compact adaptive 1d cnn classifier," *Journal of Signal Processing Systems*, vol. 91, no. 2, pp. 179–189, 2019.

[95] F. Abdollahi, H. A. Talebi, and R. V. Patel, "A stable neural network-based observer with application to flexible-joint manipulators.," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 118–129, 2006.

[96] D. G. Luenberger, "Observing the state of a linear system," *IEEE transactions on military electronics*, vol. 8, no. 2, pp. 74–80, Apr. 1964.

[97] Z. Gao and H. Wang, "Descriptor observer approaches for multivariable systems with measurement noises and application in fault detection and diagnosis," *Systems & Control Letters*, vol. 55, no. 4, pp. 304–313, Apr. 2006.

[98] Z. Gao, S. X. Ding, and Y. Ma, "Robust fault estimation approach and its application in vehicle lateral dynamic systems," *Optimal Control Applications and Methods*, vol. 28, no. 3, pp. 143–156, May 2007.

[99] Z. Gao and S. X. Ding, “Sensor fault reconstruction and sensor compensation for a class of nonlinear state-space systems via a descriptor system approach,” *IET Control Theory & Applications*, vol. 1, no. 3, pp. 578–585, 2007.

[100] X. Liu, Z. Gao, and A. Zhang, “Robust fault tolerant control for discrete-time dynamic systems with applications to aero engineering systems,” *IEEE Access*, vol. 6, pp. 18832–18847, Mar. 2018.

[101] H. K. Khalil, *Nonlinear control.* Pearson New York, 2015.

[102] A. J. van der Schaft, “L2-gain analysis of nonlinear systems and nonlinear state-feedback h-inf control,” *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.

[103] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[104] D. O. Hebb, *The organisation of behaviour: a neuropsychological theory.* Science Editions New York, 1949.

[105] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[106] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.