Alma Mater Studiorum - Università di Bologna

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTAMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE- DEI

TESI DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE

Fault Diagnosis of a Variable Speed Wind Turbine via Support Vector Machines

CANDIDATO:	*RELATORE:*
Hamid Irandoost	*Prof. Andrea Tilli*
No. 0000729919	*CORRELATORE:*
	Christian Conficoni
	Alessandro Bosso

Anno Accademico: 2016/2107

Sessione III

Abstract

In recent years, wind energy is considered as the most practical substitute energy to replace the fossil fuels. Wind turbines are massive and installed in locations, where a non-planned maintenance is very costly. Therefore, a fault-tolerant control system that is able to maintain the wind turbine connected after the occurrence of certain faults can avoid major economic losses.

To keep the wind turbine operational or at least safe, in severe cases, a reliable fault diagnosis methodology has to be exploited. It must detect, in the required time, any deviation of the system behaviour from its ordinary case, identify the location and type of the fault and reconfigure the control system to accommodate the so-called discrepancy.

To achieve the above goals, a vast number of methods have been suggested by many researchers all around the world. In this thesis, the promising classification framework of the Support Vector Machines is applied to fault detection for variable speed turbines, highlighting its features. In this regard, different fault scenarios are imposed on a benchmark model of a horizontal-axis wind turbine to check the functionality of the mentioned fault detector and the control reconfiguration module.

Astratto

Negli ultimi anni, l'energia eolica viene considerata la forma rinnovabile più adatta per sostituire i combustibili fossili. Le grandi dimensioni delle turbine, e il fatto che siano spesso installate in luoghi non facilmente accessibili, rendono le operazioni di manutenzione non pianificata piuttosto costose. Pertanto, un sistema di controllo fault-tolerant in grado di mantenere la turbina eolica collegata dopo il verificarsi di determinati guasti può evitare gravi perdite economiche.

Al fine di mantenere la turbina eolica operativa o almeno sicura, nei casi più gravi, deve essere sfruttata una metodologia di diagnosi dei guasti affidabile. Si deve rilevare, nel tempo richiesto, qualsiasi deviazione di comportamento del sistema dal suo caso ordinario, identificare la posizione e il tipo del guasto, e riconfigurare il sistema di controllo per gestire la cosiddetta "discrepanza" (di comportamento del sistema).

Per raggiungere gli obiettivi di cui sopra, diversi metodi sono stati proposti in letteratura.. In questo lavoro di tesi viene applicato il framework di classificazione Support Vector Machines alla diagnosi di guasti e anomalie per turbine eoliche a velocità variabile. A questo proposito, diversi scenari di errore sono imposti su un modello di riferimento di una turbina eolica ad asse orizzontale per verificare la funzionalità del rilevatore di guasti menzionato, e il modulo di riconfigurazione del controllo.

Table of Contents

Introduction

Energy shortage, Due to the decrease of fossil fuel sources, and environmental pollution are two important issues of the human lives and social developments in these years. Traditional mineral energy sources such as coal, oil and gas will be used out in a few years and will cause serious environmental problems. Consequently, the renewable energy, especially wind and solar types have become very popular all over the world. Among the available options, wind energy is considered as the most practical substitute energy to replace the fossil fuel, due to its competitive cost, the maturity of technology and good infrastructure [13]. The future of wind energy passes through the installation of offshore wind farms. In such locations, maintenance costs in an unpredicted situation grow drastically. Hence, a fault-tolerant control system that keeps the wind turbine operational, even after any fault incidence, avoids huge economic impacts.

Wind turbines exhibit behaviours like wind-generated noise, nonlinear aerodynamics, vibration in the components that might potentially lead to a fault. Recently, Process Fault Diagnosis (PFD) systems have been developed to a high degree of quality in modern industry to avoid severe damages. The objective of fault detection is to generate auxiliary signals based on measurements and rules, which highlight whether the considered system is running in nominal or faulty conditions. Fault isolation is then performed to classify the detected irregularities and also to determine the origin of the observed abnormal status. To be more precise, a fault diagnosis procedure is typically divided into three tasks; 1) fault detection indicates the occurrence of a fault in a monitored system; 2) fault isolation establishes the type and/or location of the fault; 3) fault identification determines the magnitude of the fault. After a fault has been detected and diagnosed, it is required in some applications that the fault be self-corrected, usually through controller reconfiguration. This is usually referred to as fault accommodation. [5]

During the last two decades, there have been significant research activities in the design and analysis of fault diagnosis and accommodation schemes. Several researchers have also investigated wind turbine fault diagnostics. The main challenges of fault detection and isolation (FDI) design, that will be discussed in the second chapter, for wind turbines include i) the aerodynamic rotor torque is not measured and ii) the wind speed is only measured at the hub position with high noise. [7]

Sensors and actuators are key components in a wind turbine control system. A faulty sensor or actuator may cause performance degradation, process shutdown, or even a fatal accident. Early FDI can provide required data to the control system and, therefore, helps reduce the cost of wind energy and increase wind power penetration into electrical grids. Existing FDI techniques can be broadly classified into two major categories, including model-based and signal-based methods [6]. To use any model-based technique, it is necessary to obtain a dynamic model of the wind turbine [1] (i.e. based on differential or difference equations). The system model could be mathematical or knowledge-based. Faults are detected based on the residual, generated by model parameter estimation [10] or state variables. Some methods, ranging from parity equations [3], observers [11] or Kalman filters [12] have been already suggested as possible model-based techniques for fault diagnosis of wind turbines. For signal-based FDI, mathematical or statistical operations are usually performed on the measurements, hence diagnosis is carried out exploiting a-priori fixed rules (not based on dynamic models) to distinguish between healthy and faulty conditions. Among the signal-based methods, Pattern Recognition techniques can be applied to the measurements to extract the information about the behaviours in healthy and faulty conditions. As a result, they do not require any a-priori defined rule or dynamic model of the process and solely employ the historical data to design a fault diagnosis model to lay down the current process operating condition [6]. Furthermore, these models permit dealing with the inherent nonlinearities of most chemical processes with no extra cost. In these methods, diverse operating conditions including normal and abnormal ones are considered as patterns. Then, the resulting classifier is used to examine the online measurement data and to convert them to a known class label for abnormal or normal, so that the existing system condition is categorized [2].

Many methodologies based on the so-called pattern recognition techniques have been implemented. Amongst these methods, Support Vector Machines (SVM) is suggested to achieve high generalization capability and to deal with problems with a few samples and high input features, while conventional Artificial Neural Networks (ANN) require a large number of samples for training, which are not necessarily obtainable every time [7]. However, artificial neural networks, including SVM, as signal-based PFD methods have demonstrated to be good classifiers for some of their salient features such as minimization of structural risk, generalization capability, noise tolerance and dealing with nonlinearity [14].

Support vector machine is a binary classification algorithm on the foundation of statistical learning theory introduced by Vapnik in the late 1960s which soon became a powerful tool in the classification area [8]. This method is used to separate data of two classes. In this method, classification is done via a constrained optimization problem with a convex objective function. Number of the constraints depend on the size of learning data. The solution is finding an optimal hyperplane that separates points lying on opposite classes while the separation margin is maximized. SVM-based classifiers has shown to be more effective in numerous practical applications than other pattern classifiers. However, their applications to process engineering problems are still rare [15].

In the current study, SVM as a fault diagnosis method, along with the fault tolerant control techniques are used in order to detect and accommodate the faults in a horizontal-axis, variable-speed wind Turbine. Thus, a benchmark model of a wind turbine that is developed by Odgaard et al [1] has been taken into account. In order to achieve high reliability in the fault detection system, physical redundancy as well as the control reconfiguration method has been considered.

In the first chapter, a full description of a wind turbine components and model is provided. Second chapter devotes to the definition of fault diagnosis and fault tolerant control schemes. The theory of the support vector machines are explained with details in the third chapter. In the fourth part of this dissertation, the application of SVM as a fault diagnosis method on a wind turbine [2] is discussed and supportive results to the high performance of this methodology are presented. In this regard, references [1], [2], define the wind turbine model characteristics, SVM methodology and requirements. A model that fully couples these two parts can be obtained in [15]. Being modular, the wind turbine parameters and SVM design can be updated according to the system characteristics, input and requirements. They are explained in the related chapters.

References

[1] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines—A benchmark model," in Proc. 7th IFAC Symp. Fault Detection, Supervis., Safety Tech. Process., Barcelona, Spain, Jun. /Jul. 2009, pp. 155–160.

[2] N. Laouti, N. Sheibat-Othman, and S. Othman, "Support Vector Machines for Fault Detection in Wind Turbines" in Proc. 18th IFAC world congress, Volume 44, Issue 1, Pages 7067-7072, January 2011.

[3] Christian Dobrila, Rasmus Stefansen; Fault Tolerant Wind Turbine Control; Master's thesis, Technical University of Denmark, Kgl. Lyngby, Denmark, 2007.

[4] Barry Dolan; Wind Turbine Modelling, Control and Fault Detection; Technical University of Denmark, Kongens Lyngby 2010, IMM-M.Sc.-2010-66

[5] A. Paoli, A. Tilli; "Diagnosis and Control, A General Overview" – LM Automation Engineering – DEI University of Bologna

[6] A. Paoli, A. Tilli; "Diagnosis and Control, Principles of Fault Detection and Isolation" – LM Automation Engineering – DEI University of Bologna

[7] S. Haykin; "Neural Networks. A Comprehensive Foundation"; Pearson Prentice Hall.

[8] V.N. Vapnik, "The Nature of Statistical Learning Theory", Springer

[9] K. E. Johnson, L. Y. Pao, M. J. Balas, L. J. Fingersh; "Control of Variable-Speed Wind Turbines, Standard & Adaptive Technics"

[10] Xiaodong Zhang, Qi Zhang, Marios M. Polycarpou, Songling Zhao, Riccardo Ferrari, Thomas Parisini ; Fault Detection and Isolation of the Wind Turbine Benchmark, an Estimation-Based Approach; Preprints of the 18th IFAC World Congress, Milano (Italy) August 28 - September 2, 2011

[11] P. F. Odgaard, J. Stoustrup, R. Nielsen, C. Damgaard; Observer based detection of sensor faults in wind turbines. In Proceeding of European Wind Energy Conference 2009, Marseille, France, March 2009. EWEA, EWEA.

[12] X. Wei, M. Verhaegen, T. van den Engelen; Sensor Fault Diagnosis of Wind Turbines for Fault Tolerant; Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008

[13] Liu Wenyi, Wang Zhenfeng, Han Jiguang, Wang Guangfeng; Wind turbine fault diagnosis method based on diagonal spectrum and clustering binary tree SVM ; Renewable Energy 50 (2013) 1e6

[14] Mehdi Namdari, Hooshang Jazayeri-Rad, Seyed-Jalaladdin Hashemi; Process Fault Diagnosis Using Support Vector Machines with a Genetic Algorithm based Parameter Tuning; Journal of Automation and Control, 2014, Vol. 2, No. 1, 1-7

[15] M. Berahman, A.A. Safavi, and M. Rostami Shahrbabaki; Fault detection in Kerman combined cycle power plant boilers by means of support vector machine classifier algorithms and PCA; 3rd International Conference on Control, Instrumentation, and Automation (ICCIA 2013), December 28-30, 2013, Tehran, Iran

[15] https://it.mathworks.com/matlabcentral/fileexchange/35130-award-winning-fdi-solution-in-wind-turbines

Chapter 1:

Wind Turbine

Wind turbines, as modern instances of energy generation cycle that reduces the reliance on fossil fuels, are the evolution of classic windmills that can be seen mainly in rural regions. Their operation is due to the kinetic energy of the wind, which pushes the blades and rotates a motor accordingly. Then, a generator converts the kinetic energy into electrical type for the consumer usage.

The Purpose of this chapter is to provide the detailed information on how a wind turbine can be modelled. In this regard, their historical evolution, the nature of their energy source that is wind, the components of a typical wind turbine as well as the related equations which govern its mathematical model are discussed.

1.1 Historical Development of Wind Turbines

Over a thousand years ago, windmills were in operations in Persia and China. "Post mills" appeared in Europe in the twelfth century, and by the end of the thirteenth century the "tower mill", on which only the timber cap rotated rather than the whole body of the mill, had been introduced. In the United States, the development of the "water-pumping windmill" was a major factor in allowing the farming and ranching of vast areas in the middle of the nineteenth century.

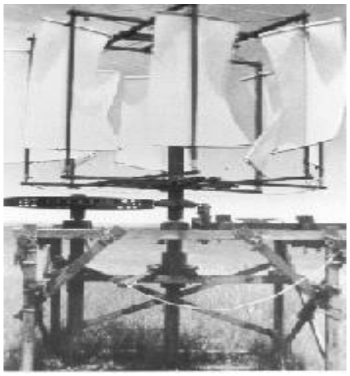


Figure 1.1.a

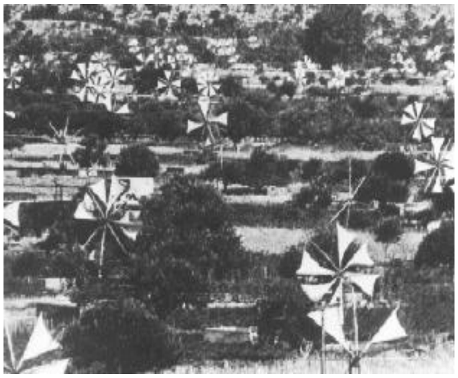


Figure 1.2.b

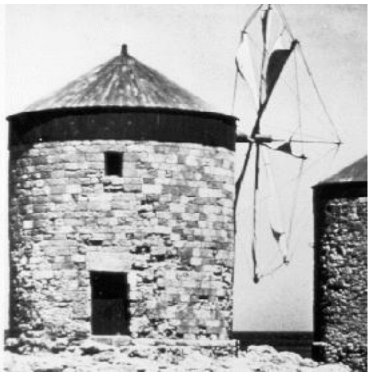


Figure 1.3.c

Figure 1.4.a: A 19th-century American knock-off of the Persian Panemone that probably made a wonderful clothes dryer

Figure 1.11.b: Water Pumping Sail wing Machines on the Island of Crete

Figure 1.1.c: An early sail-wing horizontal-axis mill on the Mediterranean coast

Of the 200,000 windmills existing in Europe in the middle of the nineteenth century, only one in ten remained after 100 years. The old windmills have been replaced by steam and internal combustion engines. However, since the end of the last century, the number of wind turbines (WT) is growing gradually and began to take a significant role in the power generation system of many countries. [8]

There are two primary designs for modern wind turbines: horizontal-axis and vertical-axis. Vertical-axis wind turbines (VAWTs) are pretty rare. The only one currently in commercial production is the Darrieus turbine that is shown in figure 1.2.

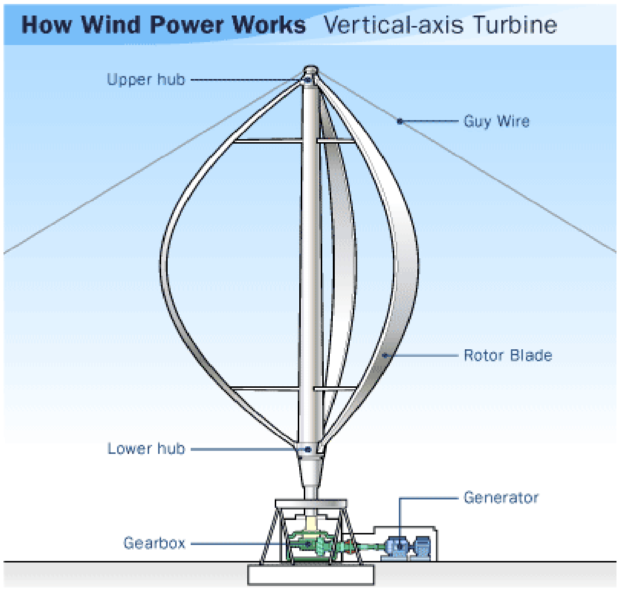


Figure 1.5: Darrieus-design VAWT

In a VAWT, the shaft is mounted on a vertical axis, perpendicular to the ground. VAWTs, unlike their horizontal-axis counterparts, are always aligned with the wind, so there is no necessary adjustment when the wind direction changes. A VAWT cannot start rotating all on its own and needs a boost from its electrical system to get started. Instead of a tower, it is typically supported by guy wires[1], so the rotor elevation is lower. Lower elevation leads to slower wind rate due to the ground interference. Therefore, VAWTs are generally less efficient than HAWTs. On the upside, all equipment is at ground level for easy installation and servicing; but that means a larger footprint for the turbine, which is a considerable disadvantage in farming areas.

[1] A tensioned cable designed to add stability to a free-standing structure.

VAWTs may be used for small-scale applications, like pumping water in rural areas, but all commercially produced, utility-scale wind turbines are horizontal-axis wind turbines (HAWTs).

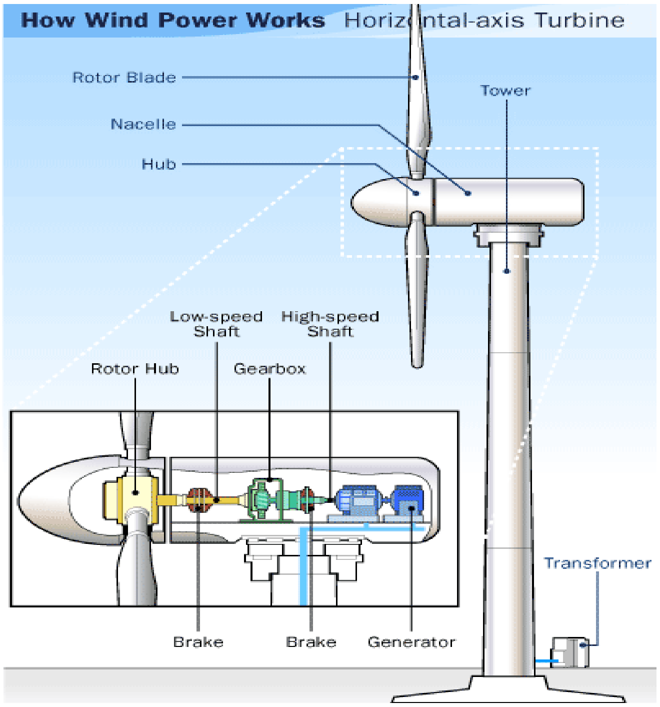


Figure 1.6: Horizontal Axis Wind Turbine (HAWT)

As implied by its name, the HAWT shaft is mounted horizontally, parallel to the ground. HAWTs have to align themselves constantly with the wind by employing a yaw-adjustment mechanism in order to capture the most wind energy available. HAWTs use a tower to lift the turbine components to an optimum elevation for more wind speed. Meanwhile, the blades can clear the ground and take up very little ground space since almost all of the components are up in the air.

1.2 The Nature of the Wind

The energy available in the wind varies as the cube of the wind speed. Therefore, an understanding of the characteristics of the wind resource is critical to all aspects of wind energy exploitation, from the identification of suitable sites and predictions of the economic viability of wind farm projects through to the design of wind turbines themselves, and understanding their effect on electricity distribution networks and consumers. [2]

From the point of view of wind energy, the most striking characteristic of the wind resource is its variability. The wind is highly variable, both geographically and temporally. Furthermore, this variability persists over a very wide range of scales, both in space and time. The importance of this is amplified by the cubic relationship to available energy. Although the availability of accurate historical records is a restriction, there is evidence that the wind speed at any particular location may be subject to very slow long-term variations.

For all wind turbines, wind power is proportional to wind speed cubed [8]. Wind energy is the kinetic form of the moving air with mass m *and velocity* v*, that is:*

$$\mathrm{E_{kin}} = \frac{1}{2}\mathrm{mv^2} \qquad (1.1)$$

The air mass m *can be determined by the air density* ρ *and the air volume* V*.*

$$\mathrm{m} = \rho\mathrm{V} \qquad (1.2)$$

Then, replacing the equation (1.2) in (1.1), we obtain:

$$\mathrm{E_{kin,wind}} = \frac{1}{2}\mathrm{V}\rho\mathrm{v^2} \qquad (1.3)$$

In a short time section $\Delta\mathrm{t}$*, the air particles travel the distance* $x = v.\Delta\mathrm{t}$ *. Multiplying the distance with the rotor area of the wind turbine* A*, results in the air volume* $\Delta\mathrm{V}$*, that is:*

$$\Delta\mathrm{V} = \mathrm{A.v.}\Delta\mathrm{t} \qquad (1.4)$$

It drives the wind turbine for a short interval. Considering the definition of "power", that is, energy divided by time, the wind power is given as:

$$P_{\text{wind}} = \frac{E_{\text{kin,wind}}}{\Delta t} = \frac{\Delta V \rho v^2}{2\Delta t} = \frac{\rho A v^3}{2} \qquad (1.5)$$

To sum up, the wind power increases with the cube of the wind speed. For instance, doubling the wind speed gives eight times the wind power. Hence, the selection of a "windy" location is very important for the wind turbine installation.

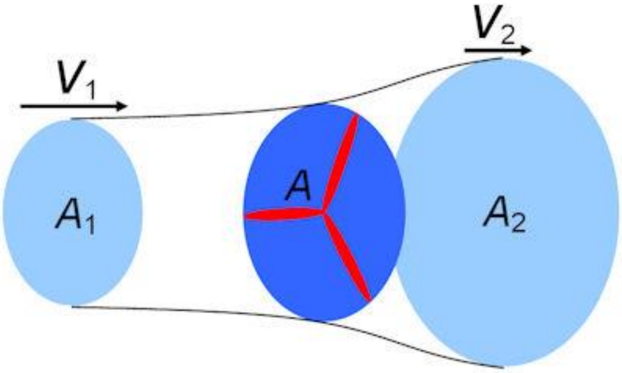


Figure 1.7: Covered area by air in the upstream and downstream of a wind turbine

The wind speed of a wind turbine reduces while passing through the blades. As the mass flow is continuous ($m_1 = m_2$), the area A_1 in the upstream is smaller than the area A_2 in the downstream. The effective power is the difference between these two wind powers.

$$P_{\text{eff}} = P_1 - P_2 = \frac{\Delta V \rho}{2\Delta t}(v_1^2 - v_2^2) = \frac{\rho A}{4}(v_1 + v_2)(v_1^2 - v_2^2) \qquad (1.6)$$

If the speed difference between these two sections is zero, net efficiency becomes zero. In contrary, if this difference tends to infinity, the air flow through the rotor is hindered too much. The power coefficient C_P characterizes the relative drawing power:

$$C_P = \frac{P_{\text{eff}}}{P_{\text{wind}}} = \frac{(v_1+v_2)(v_1^2-v_2^2)}{2v_1^3} = \frac{(1+x)(1-x^2)}{2} \qquad (1.7)$$

To derive the above equation, it was assumed that $A_1 v_1 = A_2 v_2 = A\frac{(v_1+v_2)}{2}$ *. Let's define the ratio* $x = \frac{v_2}{v_1}$ *on the right-hand side of the equation. The value of* x *that gives the maximum value of* C_P *can be found by setting the derivative with respect to* x *has to be zero. This gives a maximum* C_P *when* $x = \frac{1}{3}$*. Maximum drawing power is then obtained for* $v_2 = \frac{v_1}{3}$*, and the ideal power coefficient is given by:*

$$\mathrm{C_P} = \frac{\mathrm{P_{eff}}}{\mathrm{P_{wind}}} = \frac{16}{27} = 59.3\% \qquad (1.8)$$

According to Betz's law, no turbine can capture more than 59.3% of the kinetic energy in wind. This factor is known as Betz's coefficient. Practical utility-scale wind turbines attain a peak at 75% to 80% of the Betz limit.

A second wind turbine located too close in the direction of the wind, but behind the first one would be driven only by slower air. Thus, wind farms in the prevailing wind direction need a minimum distance of approximately eight times the rotor diameter. To have an idea, the usual diameters of ordinary wind turbines are 50 or 126 meters with an installed capacity of 1MW or 5MW, respectively. The latter is mainly used offshore.

1.3 Wind Turbine Components

Wind turbine harnesses the power in wind to generate electricity. Simply stated, a wind turbine works the opposite of a fan. Instead of using electricity to make wind, like a fan, a wind turbine uses wind to produce electricity. The energy in the wind rotates two or three propeller-like blades around a rotor. The rotor is connected to the main shaft, which turns a generator to make electrical energy. This illustration depicts a detailed view of the inside of a wind turbine, its components, and their functionality. [5], [6]

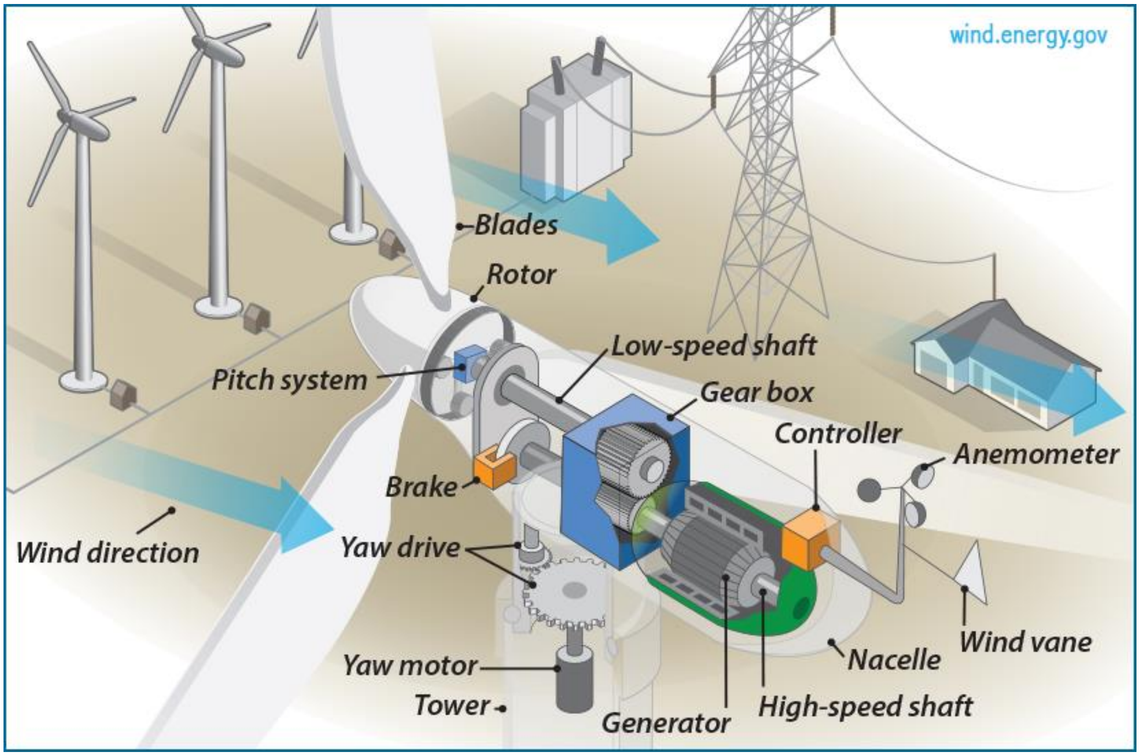


Figure 1.8: Wind Turbine Components

1.3.1 Tower

It supports the structure of the turbine. The duty of the tower is to lift the hub off the ground. An increase in height leads to more uniform and higher mean wind speed. Taller towers enable turbines to capture more energy and generate more electricity. In addition, the ground clearance let the blades to be longer, thus increasing the aerodynamic torque.

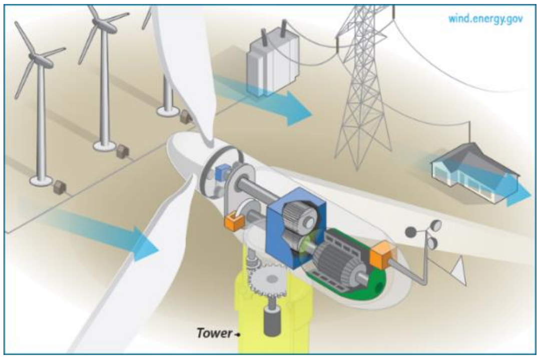


Figure 1.9: Tower

1.3.2 Wind direction

It determines the design of the turbine. Upwind turbines, like the one shown in figure 1.7, face into the wind while a downwind turbine faces away.

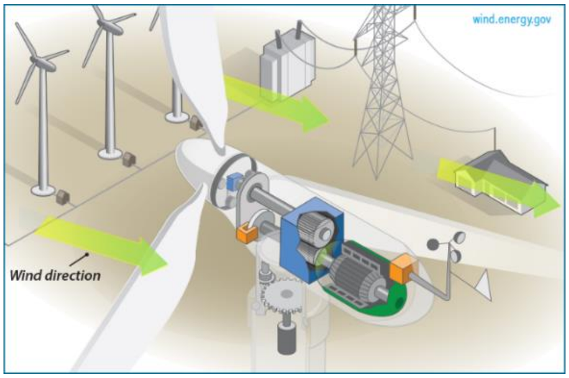


Figure 1.10: Wind Direction

1.3.3 Nacelle

A Nacelle sits atop the tower and holds all the turbine machinery including the generator, high-speed and low-speed shafts, gearbox, controller, and brake. It must be able to turn to follow the wind direction. Thus, via bearings, it is connected to the tower.

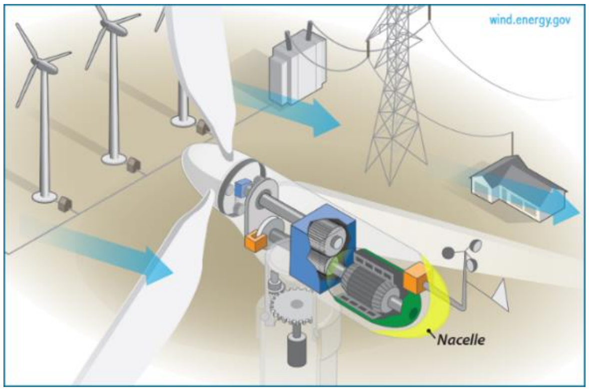


Figure 1.11: Nacelle

1.3.4 Yaw drive

Yaw is the rotation angle of the nacelle around its vertical axis. Efficient yaw control is essential to assure that a wind turbine always faces directly into the wind. Yaw drive orients upwind turbines to keep them facing towards the wind when its direction varies. Downwind turbines do not require a yaw drive as the wind manually blows the rotor away from it.

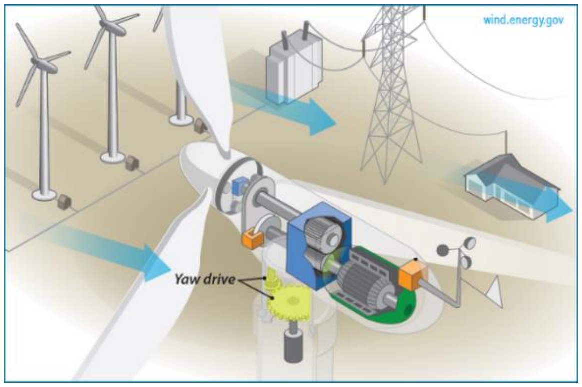


Figure 1.12: Yaw Drive

1.3.5 Yaw motor

As described above, the rotor should always face the wind to generate as much electricity as possible. To achieve it, the yaw motor powers the yaw drive to orient the nacelle so that the rotor faces the wind.

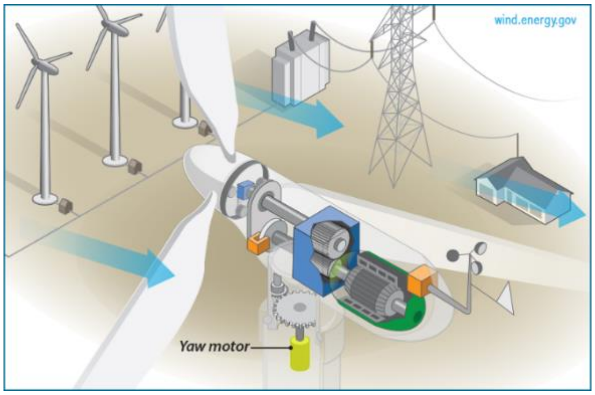


Figure 1.13: Yaw Motor

1.3.6 Wind vane

A wind vane always positions itself according to the wind direction. At the foot of the wind vane, there is a small sensor that measures the wind direction and notifies the yaw drive to rotate the nacelle properly.

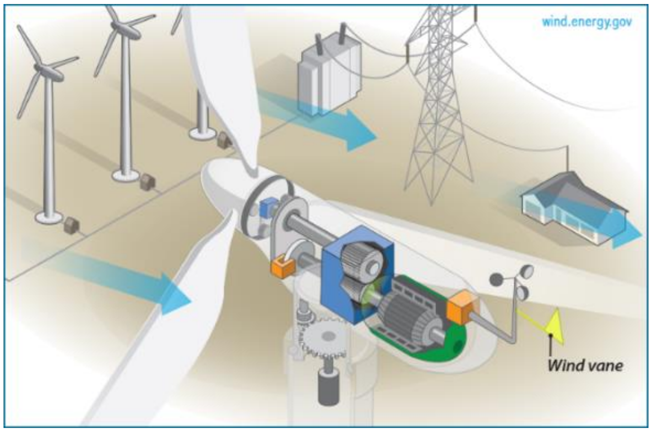


Figure 1.14: Wind Vane

1.3.7 Anemometer

The anemometer measures the wind speed and communicates the wind turbine controller when there is enough wind that it would be profitable to consume power to make the wind turbine rotate (yaw) toward the wind direction and start running.

It is important to know the amount of wind. If it is too strong, the wind turbine might break. This is why the wind turbine is brought to a shut-down, while the wind speed exceeds 25 meters per second. When it drops, the anemometer informs the controller to restart the turbine.

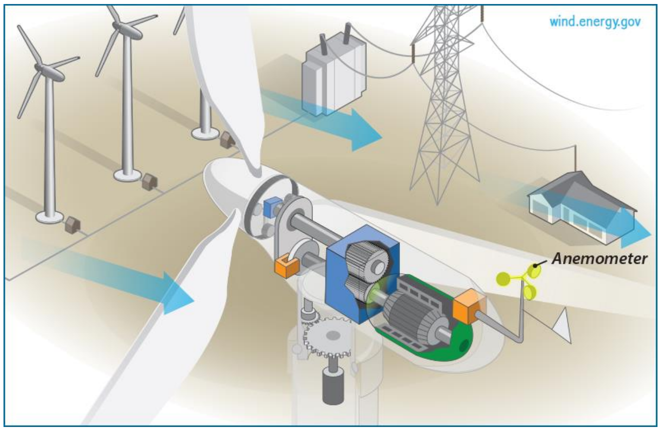


Figure 1.15: Anemometer

1.3.8 Drive Train

Power, in a typical wind turbine drive train, is transmitted from the rotor to the generator via a system containing a rotor shaft, shaft-hub clamping lock, multiplicating gearbox, and flexible coupling. It is transferred through the rotor shaft (connected to the multiplicating gearbox), hub joint and further by the flexible coupling and composite sleeve.

In general, the drive train consists of the following main components:

1. *Main shaft with bedding*
2. *Gear box (direct drive turbines have none)*
3. *Brake(s) and coupling*
4. *Generator.*

There are many ways to arrange these components that vary from one manufacturer to another. Certification providers have specifications for load profiles, noise levels and oscillation response of these components that are very important, as they are subjected to tremendous loads.

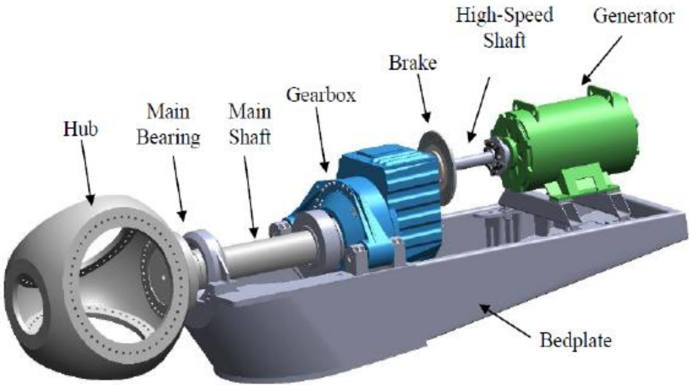


Figure 1.16: Drive Train with Gearbox

1.3.9 Main (Low-speed) shaft

The main shaft is a firm disc that links the rotor to the gearbox. The rotor is bolted to one side of it, while the gearbox is placed at the other end. It is also named the low-speed shaft and rotates at about 30-60 rpm.

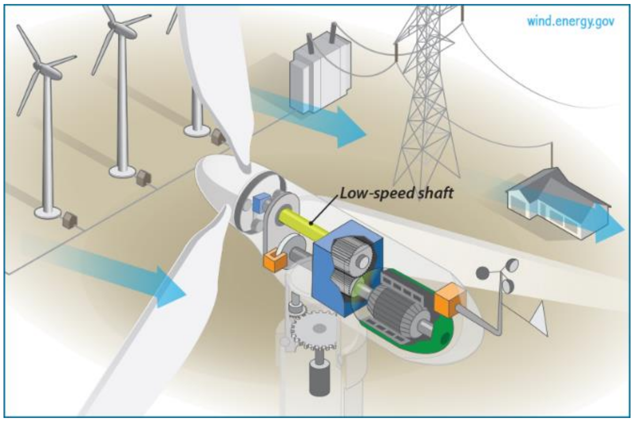


Figure 1.17: Low Speed Shaft

1.3.10 Gear box

A gearbox is basically used in a wind turbine to boost the rotational speed from the low-speed rotor to a higher-speed electrical generator. To be more precise, it connects the low-speed shaft with the input from the rotor of about 30-60 rpm to the high-speed shaft that delivers the output of about 1000-1800 rpm to the generator.

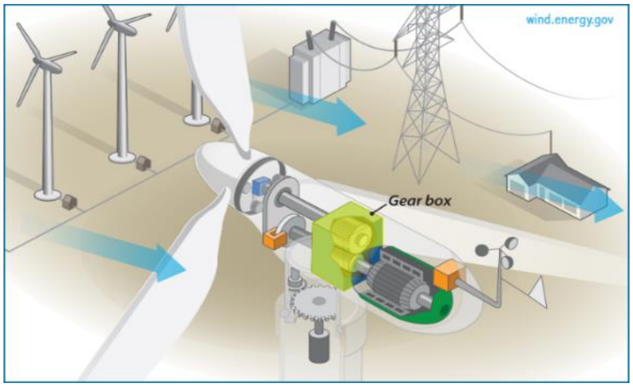


Figure 1.18: Gearbox

The gearbox is a heavy and costly component of a wind turbine. Engineers are exploring "direct-drive" generators that operate at lower rotational speeds and need no gearboxes.

The drive system of modern wind energy converters is based on a simple principle: "fewer rotating components reduce mechanical stress (fewer wearing parts) and simultaneously increase the technical service life of the equipment (no gear oil change).

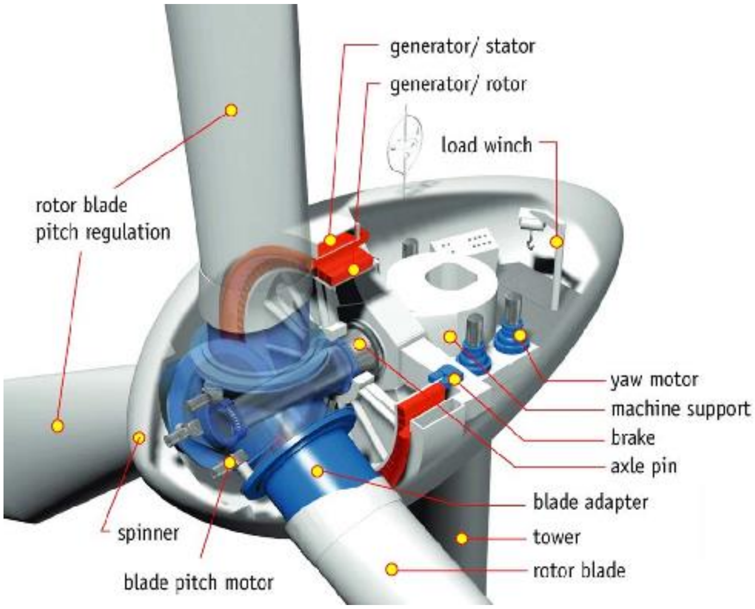


Figure 1.19: Direct Drive Train

1.3.11 Brake

A wind turbine must have two independent braking systems. However, the IEC 61400-1 requirement does not specify what kind of braking systems must have been employed. It only requires the protection system to remain effective even after the failure of any non-safe-life[1] *protection system component. [7]*

The aerodynamic brake, that is more benign than the mechanical type. It is always used in normal shut-downs by pitching the blade (for example, to align the blade chord with the wind direction) or pitching the blade tip.

The mechanical brake consists of a steel disc that can be positioned on the rotor shaft (low-speed shaft) or between the gearbox and the generator (high-speed shaft). It is only used in emergencies, in case of the blade tip brake failure.

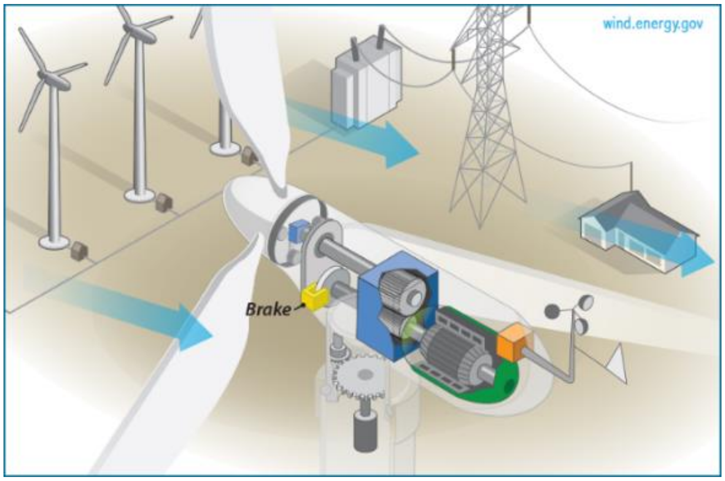


Figure 1.20: Mechanical Brake

1.3.12 Generator

The wind turbine generator converts mechanical energy into the electrical type. Wind turbine generators are a bit different, compared to other generating units you might have ordinarily found attached to the electrical grid. One reason is that the generator has to work with a power source (the wind turbine rotor) which supplies very fluctuating mechanical power (torque).

[1] The safe-life design technique is employed in critical systems which are either very difficult to repair or may cause severe damage to life and property.

A generator needs to be cooled down while working. On most turbines this is carried out by encapsulating the generator in a duct, using a huge fan for air-cooling, but a few manufacturers use water-cooled generators. The latter may be built more compactly, which is followed by some electrical efficiency advantages. However, a radiator in the nacelle is required to get rid of the heat absorbed by the liquid.

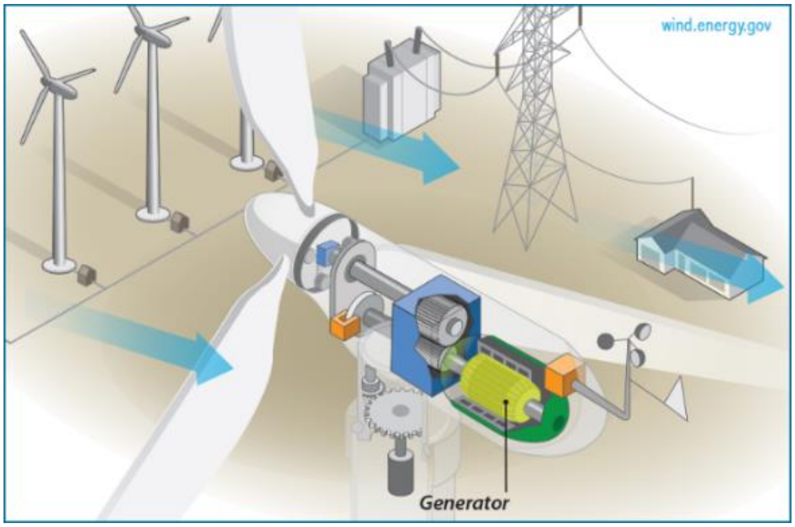


Figure 1.21: Generator

1.3.13 Small (High-speed) shaft

The small shaft drives the generator by connecting it to the gearbox. This shaft does not require to transmit as much turning force as the main shaft does. That is the reason for its thinner body which let it revolve very quickly (1500 rpm).

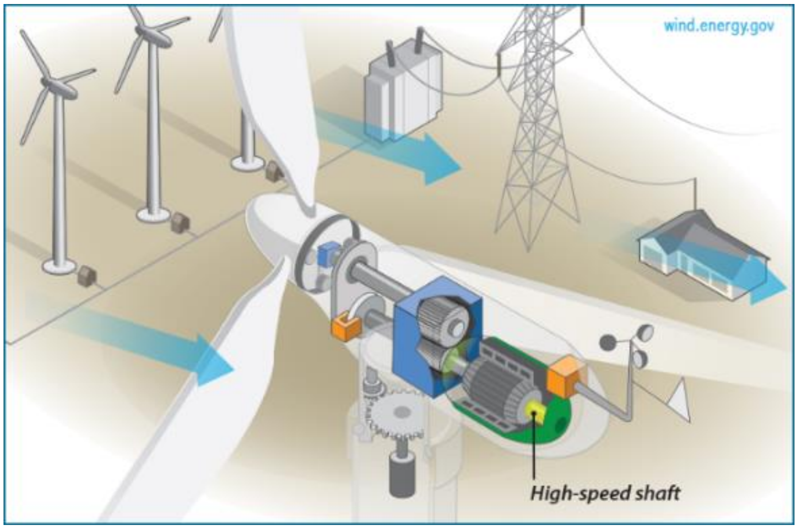


Figure 1.22: High Speed Shaft

1.3.14 Rotor

The rotor can be entitled as the heart of a wind turbine. It consists of multiple rotor blades attached to a hub. The rotor converts the wind energy into a rotation.

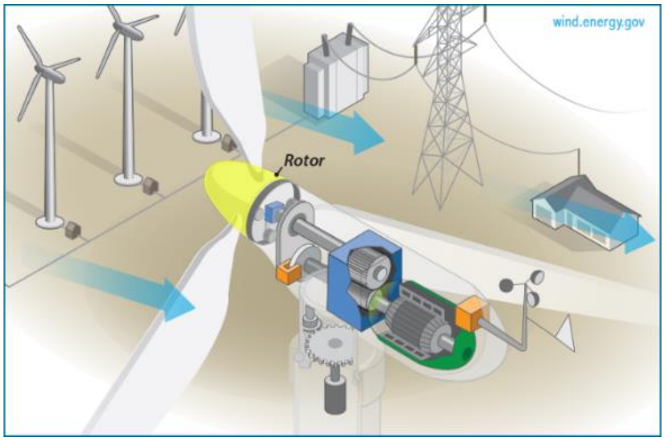


Figure 1.23: Rotor

1.3.15 Hub

The hub is the central part of the rotor to which the rotor blades are attached. It directs the energy from the rotor blades onto the generator. If the wind turbines have a gearbox, the hub is linked to the slowly-rotating gearbox shaft, converting the wind energy into the rotation. If the turbine has a direct drive, the hub passes the energy directly to the generator.

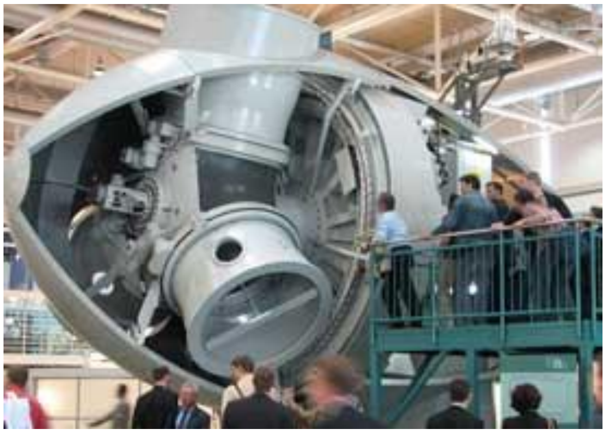


Figure 1.24: Hub

1.3.16 Blades

Rotor blades extract energy out of the wind. They capture the wind and convert its motive energy into the rotation of the hub. Rotor blades utilize the same "lift" principle, that is, below the *wing, the stream of air creates overpressure, while above the wing is a vacuum. These forces make the rotor rotate.*

Today, most rotors have three blades with a horizontal axis, a diameter of between 40 and 90 meters. Over time, it was found that three-blade rotors are the most efficient for power generation by immense wind turbines. In addition, the usage of three rotor blades allows for a better distribution of mass, which makes rotation smoother and a calmer appearance.

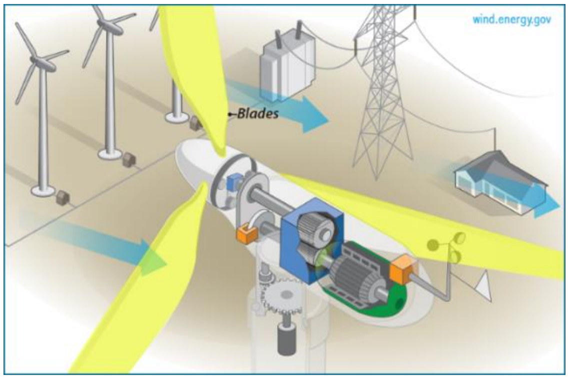


Figure 1.25: Rotor Blades

1.3.17 Controller

The wind turbine is controlled by several computers that keep an eye on whether or not the wind turbine works as it should. Monitoring task apart, they make the turbine work automatically by a proper control action. They will be discussed in details in section 2.4. Every time a reconfiguration in required to adjust the turbine, it is the controller that takes its responsibility.

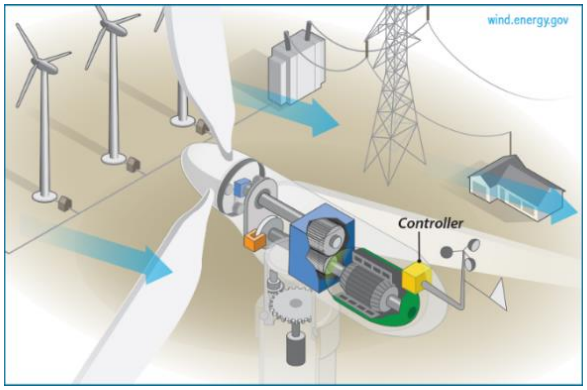


Figure 1.26: Controller

The wind turbine control system starts up the machine at wind speeds of about 8 to 16 miles per hour (mph) and shuts off the machine at about 55 mph, because they may be damaged by the high winds.

1.3.18 Pitch System

Wind turbines have to be designed in a way to produce electrical energy as cheaply as possible. They are therefore designed so that they yield maximum output. In case of stronger winds, it is necessary to waste a part of the surplus energy in order to avoid damaging the equipment. Hence, all wind turbines are designed with some sort of power control. In modern Turbines, there are two different approaches to accomplish this safely, including Pitch Controlled Wind Turbines and Stall Controlled Wind Turbines. [6]

On a pitch controlled wind turbine, the controller checks the power output of the turbine several times per second. When the power output exceeds a threshold, it sends an order to the blade pitch mechanism which immediately pitches (turns) the rotor blades slightly out of the wind to control the rotor speed. Conversely, the blades are turned back into the wind, whenever the wind drops again.

Designing a pitch controlled wind turbine requires some clever engineering to make sure that the rotor blades pitch exactly the required amount. Every moment the wind varies, the controller pitches the blades a few degrees to maximize the output by maintaining the rotor blades at the optimum angle. The pitch mechanism is usually operated using hydraulics.

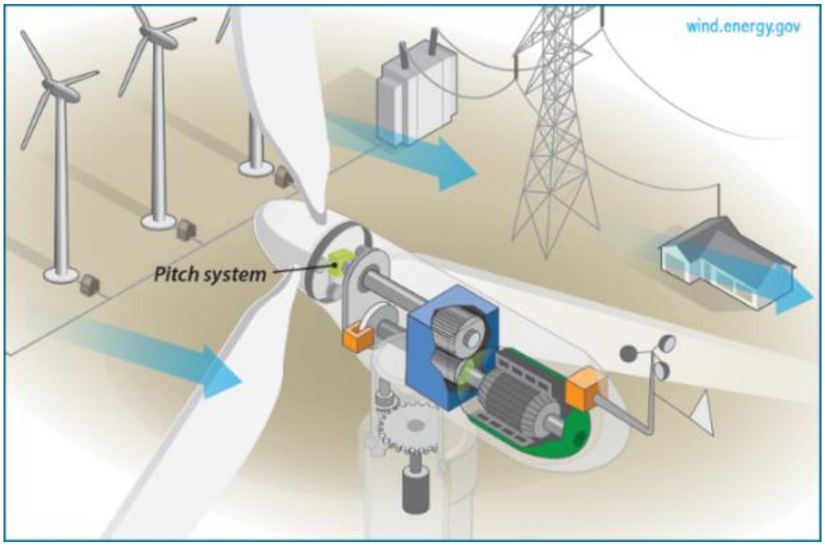


Figure 1.27: Pitch System

1.4 Wind Turbine Model

The configuration of a wind generation system is shown in figure 1.25. A wind turbine produces electricity by using the power of the wind that drives an electrical generator. Wind passes over the blades, generating lift and exerting a turning force. The rotating blades turn a shaft inside the nacelle, which goes into a gearbox. The gearbox or the drive train system increases the rotational speed from the rotor to a level which is appropriate for the generator.

The energy conversion from wind to mechanical energy in terms of a rotating shaft can be controlled by changing the aerodynamics of the turbine through pitching the blades or by controlling the rotational speed of the turbine relative to the wind speed.

The generator converts the rotational energy into electricity using magnetic fields. To be more precise, the mechanical energy is converted into the electrical form by a generator fully coupled to a converter. The converter is used to set the generator torque, which consequently can be used to control the rotational speed of the generator as well as the rotor.

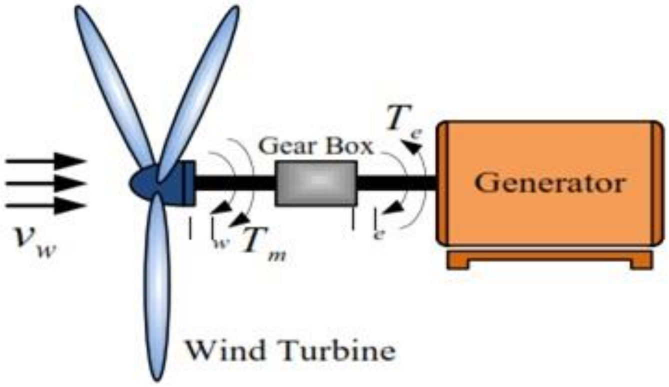


Figure 1.28 : Wind generation system configuration

In this section the different model parts are presented. The parts are presented in the following order: Wind model, Blade and Pitch model, Drive train model, Generator/Converter model, Controller and parameters of the models. The model is presented in terms of equations and a Simulink model is then considered as a rough estimation of the overall system behavior.

The dynamics of the wind turbine can be illustrated using a block diagram of the figure 1.26, which describes the interactions of the various sub-models. [1]

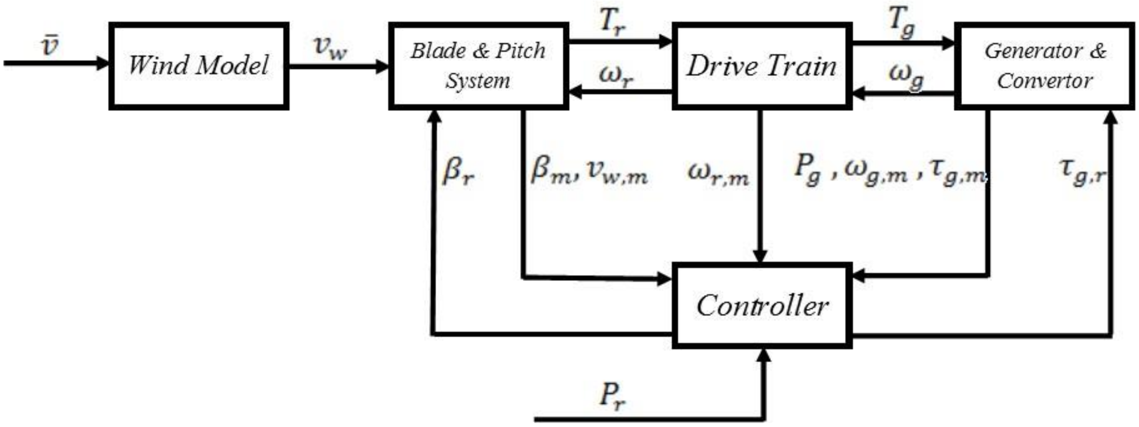


Figure 1.29: Overview of the benchmark model structure

1.4.1 Wind Model

Being non-controllable, the wind speed is referred to as the exogenous input. As described before, it is a chaotic process and susceptible to climatic influences. Therefore, a complete wind spectrum would span several decades. Diurnal and seasonal variations have to be observed as a result of weather changes. However, these are slow-changing variations compared to turbulences that take place on the scale of minutes or even seconds. As a result, in order to express a wind speed v_ω , a mean speed $\overline{v}$ and a turbulent wind speed $\tilde{v}$ has to be taken into account.

In this study, in order to generate comparable test results of detection and accommodation schemes, a predefined wind sequence is proposed in different tests. Vector v_ω is provided as a defined test sequence of the wind. However, for more details about how to obtain a reliable wind model, refer to [3], [8].

1.4.2 Blade and Pitch Model

This model is composed of the Aerodynamic model and the Pitch System model that will be explained in the following sections.

1.4.2.1 Aerodynamics of Horizontal-axis Wind Turbines

The aerodynamics description is involved with the information on how the kinetic energy of the wind is passed onto the shaft through the blades. In this part, equations governing the extraction of kinetic energy from the wind and the subsequent conversion to mechanical torque are described. [2]

A wind turbine is a device for drawing kinetic energy from the wind. By extracting some of its kinetic energy the wind must slow down, but only that mass of air which passes through the rotor disc is affected. Assuming that, the affected mass of air remains separated from the air which does not pass through the rotor disc and does not slow down a boundary surface, can be drawn containing the affected air mass and this boundary can be extended upstream as well as downstream forming a long stream-tube of circular cross-section (figure 1.4). No air flows across the boundary and so the mass flow rate of the air flowing along the stream-tube will be the same for all stream-wise positions along the stream-tube. Because the air within the stream-tube slows down but does not become compressed, the cross-sectional area of the stream-tube must expand to accommodate the slower moving air.

Although kinetic energy is extracted from the airflow, a sudden step change in its velocity is neither possible nor desirable because of the enormous accelerations and forces this would require. All wind turbines, whatever their design, operate in a way that the pressure energy can be extracted in a step-like manner.

The presence of the turbine causes the approaching air, upstream, steadily to slow down so that when the air arrives at the rotor disc its velocity is already lower than the free-stream wind speed. The stream-tube expands as a result of the slowing down and, because no work has yet been done on, or by, the air its static pressure rises to absorb the decrease in kinetic energy.

As the air passes through the rotor disc, there is a drop in static pressure such that, on leaving, the air is below the atmospheric pressure level. The air then proceeds downstream with reduced speed and static pressure; this region of the flow is called the wake. Eventually, far downstream, the static pressure in the wake must return to the atmospheric level for equilibrium to be achieved. The rise in static pressure is at the expense of the kinetic energy and so causes a *further slowing down of the wind. Thus, between the far upstream and far wake conditions, no change in static pressure exists but there is a reduction in kinetic energy.*

Wind Turbine Performance curves

The performance of a wind turbine can be characterized by the manner in which the three main indicators (power, torque and thrust) vary with wind speed. The power determines the amount of energy captured by the rotor, the torque developed determines the size of the gearbox and must be matched by whatever generator is being driven by the rotor. The rotor thrust has great influence on the structural design of the tower. It is usually convenient to express the performance by means of non-dimensional, characteristic performance curves from which the actual performance can be determined, regardless of how the turbine is operated, e.g., at constant rotational speed or some regime of variable rotor speed. Assuming that the aerodynamic performance of the rotor blades does not deteriorate the non-dimensional aerodynamic performance of the rotor depend upon the tip speed ratio and, if appropriate, the pitch setting of the blades. It is usual, therefore, to display the power, torque and thrust coefficients as functions of tip speed ratio. [2]

$C_P - \lambda$ ***Performance curve***

Once the blade has been designed for optimum operation at a specific design tip speed ratio (defined as the ratio between the blade tip speed and the wind speed), the performance of the

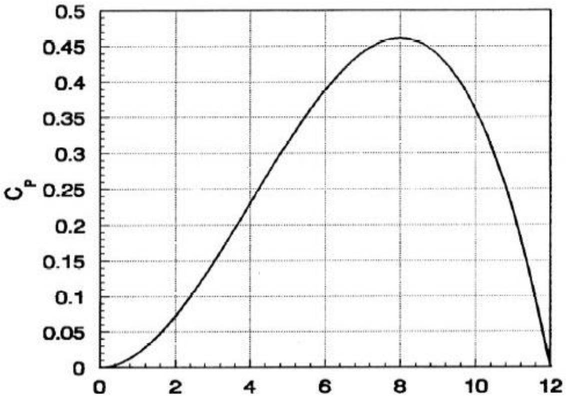


Figure 1.30-a: Representative C_P-λ Curve

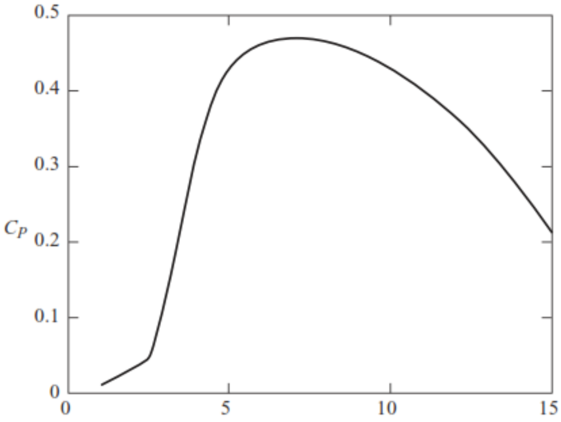


Figure 1.27-b: C_P-λ Performance curve for a Modern Three-Bladed Wind Turbine

rotor over all expected tip speed ratios needs to be determined. For each tip speed ratio, the aerodynamic conditions at each blade section need to be defined. From these, the performance of the total rotor can be determined. The results are usually presented as a graph of power coefficient C_P versus the tip speed ratio λ. This graph is called the $C_P - \lambda$ curve.

The logic behind the shape of this curve is plausible, considering that, at λ = 0 the rotor does not rotate and hence cannot draw power from the wind, while at very high λ (here at λ=12) the rotor runs so fast, that it is seen as a completely blocked disc by the wind. Wind flows around this "solid" disc, so there is no mass transport through the rotor, and thus, no possibility to extract energy from a moving mass. Somewhere between λ=0 and λ=12, there will be an optimum value (here at λ=8) for which the maximum power is extracted. This will be the condition, in which the mean velocity at the rotor disc is almost 2/3rd of the wind speed, according to the Betz law.

$C_Q - \lambda$ Performance curve

The torque coefficient is derived from the power coefficient, simply by dividing it by the tip speed ratio. As a result, it gives no additional information about the turbine performance. The principal use of the $C_Q - \lambda$ curve is for torque assessment purposes when the rotor is connected to a gearbox and generator. [2]

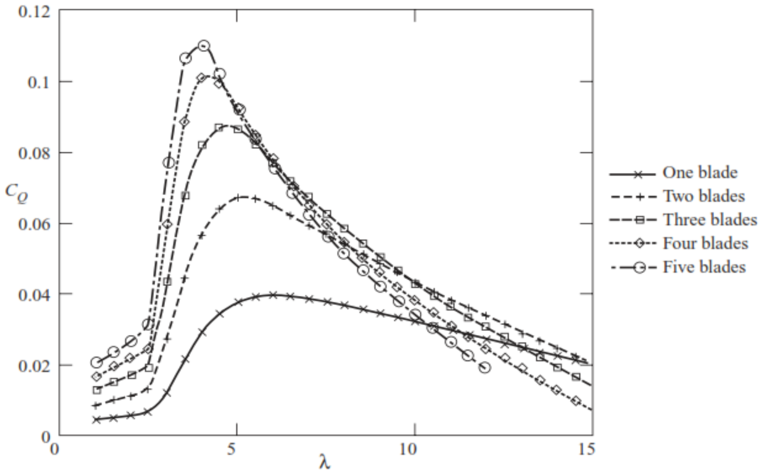


Figure 1.31: The effect of Solidity on Torque

Figure 1.28 shows how the torque developed by a turbine rises with increasing solidity. For modern high-speed turbines, designed for electricity generation, as low a torque as possible is desirable in order to reduce gearbox costs. On the other hand, a high-solidity multi-bladed turbine rotates slowly and has a very high starting torque coefficient.

The peak of the torque curve takes place at a lower tip speed ratio than the peak of the power curve. For the highest solidity, shown in figure 1.28 the peak of the curve occurs, while the blade is stalled.

$C_T - \lambda$ *Performance curve*

The thrust force on the rotor is directly applied to the tower on which the rotor is supported and so considerably influences the structural design of the tower. Generally, the thrust on the rotor rises with increasing solidity. [2]

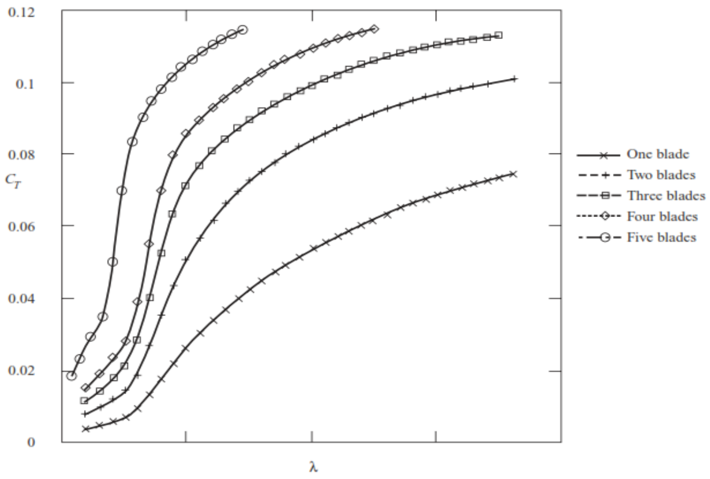


Figure 1.32: The effect of Solidity on Thrust

For the fault diagnosis study, a simulator that provides a rough model of a variable-speed three-bladed HAWT, within a uniform wind field is considered. This model has been applied in the benchmark model by Odgaard [1] as described in the "Introduction".

Considering the briefly-described performance curves; the rotor is a uniform wind extractor transforming kinetic energy of the wind to mechanical type at the shaft and then, into *electrical power by the generator. The power obtained by the rotor can be characterized by the following equation.*

$$P_R = \frac{\rho A_r C_P(\lambda(t),\beta(t)).\vartheta_\omega(t)^3}{2} \qquad (1.9)$$

where $A_r = \pi R^2$ *is the swept area by the rotor,* ρ *is the density of air and* C_P *is the power coefficient.*

The two most important forces experienced by the rotor are thrust and torque. The rotor torque is a result of the lift force induced on the blades by the oncoming wind. Since ideally speaking $P_r = \tau_R \omega_r$*, the rotor torque can be expressed as:*

$$\tau_R = \frac{\rho A_r C_P(\lambda(t),\beta(t)).\vartheta_\omega(t)^3}{2\omega_r} \qquad (1.10)$$

As the torque coefficient C_q *is obtained directly from the power coefficient* C_P*, the aerodynamic torque,* τ_R *acted upon the blades can be represented by:*

$$\tau_R = \frac{\rho \pi R^3 . C_q(\lambda(t),\beta(t)).\vartheta_\omega(t)^2}{2} \qquad (1.11)$$

The thrust force is a by-product of the interaction between the blades and the wind. This force is transferred from the blades to the nacelle and tower and is given as:

$$F_T = \frac{\rho A_r C_T(\lambda(t),\beta(t)).\vartheta_\omega(t)^3}{2} \qquad (1.12)$$

Both C_P *and* C_T *depend on the tip-speed ratio* λ *and the pitch angle* β*. The tip speed ratio will be defined as:*

$$\lambda \equiv \frac{\omega_r R}{\vartheta_\omega} \qquad (1.13)$$

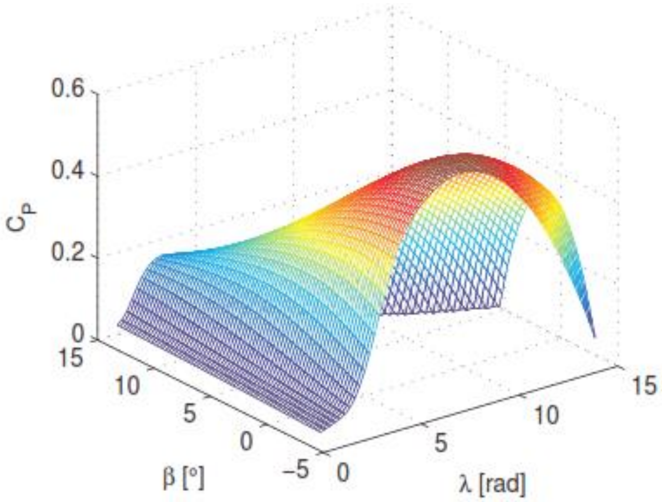
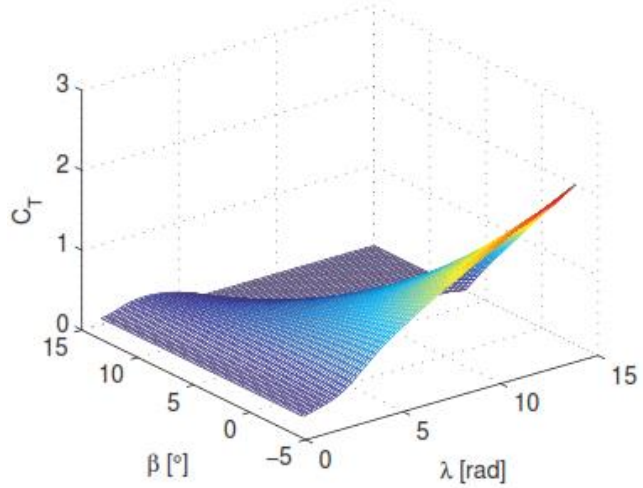


Figure 1.33: Power efficiency coefficient C_P given the tip speed ratio λ and the pitch angle β



.

Figure 1.34: Thrust coefficient C_T given the tip speed ratio λ and the pitch angle β

Considering that the three blades might have different β values, a simple way to model the aerodynamic torque can be obtained by:

$$\tau_\omega(t) = \sum_{1 \le i \le 3} \frac{\rho \pi R^3 . C_q\big(\lambda(t), \beta(t)\big) . \vartheta_\omega(t)^2}{6} \qquad (1.14)$$

This model is valid for a small difference between β values. However, it has been examined that this simple model has similar behaviour as a more detailed model. [1]

1.4.2.2 Pitch System Model

In order to vary the angle of each blade toward the wind individually, one pitch actuator is used for each blade. The pitch actuators are placed in the hub of the wind turbine. They are driven by hydraulic servo systems, each of which can be characterized by a linear second order model [1].

$$\ddot{\beta} = \omega_n^2 \beta_{ref} - 2\dot{\beta}\omega_n \xi - \beta \omega_n^2 \tag{1.15}$$

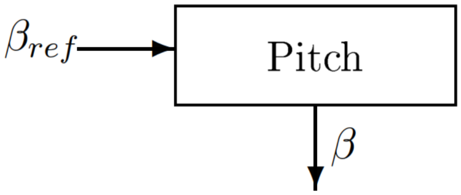


Figure 1.35: Pitch System Block

It can be shown both in the state space or Transfer Function form to be applied to the wind turbine simulator. [4]

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\omega_n \xi \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \beta_{ref} \tag{1.16}$$

$$\frac{\beta}{\beta_{ref}} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{1.17}$$

1.4.3 Drive Train Model

In this section, the two-mass drive train model, exploited in this project, is presented. Aerodynamic torque is transferred through the drive train to the generator via the gearbox. The gearbox scales up the rotational speed by a factor, known as the gearbox ratio, to the amount that is required by the generator. [3]

The dynamics of the drive train are described by the following three differential equations.

Drive Train Differential Equations:
$$\begin{cases} J_r\dot{\omega}_r = \ \tau_r - K_{dt}\theta_\Delta - B_{dt}\dot{\theta}_\Delta \\ J_g N_g \dot{\omega}_g = \ -\tau_g N_g + K_{dt}\theta_\Delta + B_{dt}\dot{\theta}_\Delta \\ \dot{\theta}_\Delta = \omega_r - \frac{\omega_g}{N_g} \end{cases} \qquad (1.18)$$

Considering the Schematic drawing shown in figure 1.33, the low-speed shaft has connected the rotor with the speed ω_r and the torque τ_r acting on the shaft. The Inertia of the rotor and the low-speed shaft is J_r.

K_{dt} is the spring stiffness coefficient and B_{dt} is the viscous damping parameter of the massless viscously damped rotational spring. The gearbox ratio is characterized by N_g.

At the right side, the Inertia of the gearbox, high-speed shaft and the generator is assumed as J_g. Generator Torque and its rotational speed are shown by τ_g, ω_g respectively.

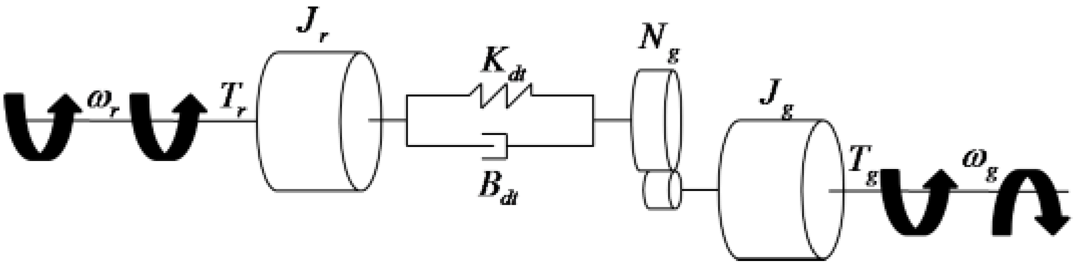


Figure 1.36: Schematic drawing of a flexible drive train

Let's define $\theta_\Delta(t)$ as the torsion angle of the drive train and η_{dt} for the drive-train efficiency. By ignoring both viscous frictions of low-speed and high-speed shafts (B_r, B_g), the state space model can be represented by:

$$\begin{bmatrix} \dot{\boldsymbol{\omega}}_{\boldsymbol{r}}(\boldsymbol{t}) \\ \boldsymbol{\omega_g(t)} \\ \boldsymbol{\theta_\Delta(t)} \end{bmatrix} = \begin{bmatrix} -\frac{B_{dt}}{J_r} & \frac{B_{dt}}{J_r N_g} & -\frac{K_{dt}}{J_r} \\ \frac{\eta_{dt}B_{dt}}{J_g N_g} & -\frac{\eta_{dt}B_{dt}}{J_g {N_g}^2} & \frac{\eta_{dt}K_{dt}}{J_g N_g} \\ 1 & -\frac{1}{N_g} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega_r(t)} \\ \boldsymbol{\omega_g(t)} \\ \boldsymbol{\theta_\Delta(t)} \end{bmatrix} + \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau_r(t)} \\ \boldsymbol{\tau_g(t)} \end{bmatrix} \qquad (1.19)$$

1.4.4 Generator and Converter Model

Given the delivered torque τ_g *on the generator side of the drive-train and the generator efficiency* η_g*, the electrical power can be obtained from:*

$$P_g = \eta_g\ \omega_g(t)\tau_g(t) \tag{1.20}$$

The generator torque cannot be changed instantaneously. Therefore, the relationship between the requested torque and the actual torque is represented by a first order transfer function, with frequency α_{gc} *.*

$$\frac{\tau_g(s)}{\tau_{g,r}(s)} = \frac{\alpha_{gc}}{s+\alpha_{gc}} \tag{1.21}$$

In reality, the power P_g *produced by the generator is controlled by adjusting the flow of current in the rotor of the generator, which adjusts the* $P_{g,r}$*, and as a consequence the torque* τ_g *changes. However, in this model,* P_g *will be controlled by varying* $\tau_{g,r}(s)$ *directly. [3]*

1.4.5 Wind Turbine Control system concept

The generated electrical power P_g and generator speed ω_g are the system outputs. The controllable inputs are the reference pitch angles $\beta_{i,ref}$ and generator torque $\tau_{g,r}$, which are used to control the output based on certain conditions which are primarily dependent upon the wind speed $\vartheta_{\omega}(t)$. The pitch angle of the blades varies by three pitch actuators, whilst the generator torque changes by a converter. The aerodynamic thrust experienced by the blades causes the tower to sway, and thus the wind speed seen by the rotor is obtained by adding the tower velocity to the wind velocity.

In the most general sense, a wind turbine control system contains a couple of sensors and actuators and a system consisting of hardware and software which processes the input signals from the sensors and generates output signals for the actuators. [2]

There might be various sensors, including an anemometer, a wind vane, an electrical power sensor, various limit switches, vibration sensors, temperature and oil level sensors, hydraulic pressure sensors, operator switches and push buttons.

In this study, considering the redundancy that will be discussed in chapter two and to decrease the complexity of the controller, it is assumed that the system is equipped with duplicate sensors in the following subsystems. Refer to Odgaard et al. [1], there are six sensors to measure the three pitch positions (β_{k,m_i} , $k = 1,2,3$; $i = 1,2$), four sensors for the generator and rotor speed measurements (ω_{g,m_i} , ω_{r,m_i} ; $i = 1,2$). It gives a total number of ten sensors, all subjected to two different kinds of faults, including fixed value and gain factor. To be more precise, twenty faults might potentially take place.

The actuators might include a hydraulic or electric pitch actuator, sometimes a generator torque controller, generator contactors, switches for activating shaft brakes, yaw motors, etc. In this project, the considered control actuators are the three pitch systems and the converter. They allow respectively pitching the blades and setting the generator torque to control the rotational speed of the generator and the rotor.

The system processes the inputs to generate outputs usually consists of a computer or microprocessor-based controller which carries out the normal control functions needed to operate the turbine, supplemented by a highly reliable hardwired safety system. The safety system must be capable of overriding the ordinary controller to bring the turbine to a safe state, while a serious problem appears.

In this study, considering a nonlinear model of the wind turbine and noisy measurements, a switching control structure has been deployed. Assuming that the wind turbine operates in different regions, a switching strategy for the control system is designed that switches between various modes. In the following, all the details of a HAWT turbine controller are explained more deeply.

1.4.5.1 Wind Turbine Control Regions

Variable-speed wind turbines have four main regions of operation. A stopped turbine that is just starting up is considered to be operating in region one[1]*. Region two is an operational mode with the objective of maximizing wind energy capture*[2]*. In the third region, the optimized produced power is kept constant*[3]*. In region four, which encompasses high wind speeds*[4]*, the turbine must limit the captured wind power so that safe electrical and mechanical loads are not exceeded [3].*

[1] *Startup: In this region, the rotor speed is held constant at its minimum value. Pitch angle is usually held constant at zero. As the wind speed increases, the aerodynamic efficiency and therefore the power coefficient rises towards its maximum value. Upon reaching its maximum value, the control strategy switches to region II.*

[2] *Power optimization: By regulating the generator torque, the rotor speed increases with the wind speed to hold the tip speed ratio constant, at its optimal value. The pitch angle is held at zero degrees. In this way the maximum power coefficient is achieved for all wind speeds within this region. When the rotor speed reaches its rated value, controller mode is switched to satisfy the objectives of region III.*

[3] *Power reference following: This region exists since variable speed machines cannot achieve rated power at rated rotor speed. The rotor speed is held constant at its rated value as the wind speed increases. It makes the power coefficient and the tip speed ratio to fall down. Despite this, the output power continues to rise towards rated power as the increasing wind speed compensates for the decreasing efficiency. Upon reaching rated power, control switches to region IV.*

[4] *High wind speed: As the wind speed increases towards the cut-out value, the pitch of the blades increases. This reduces the power coefficient and in turn the power output is held at its rated value.*

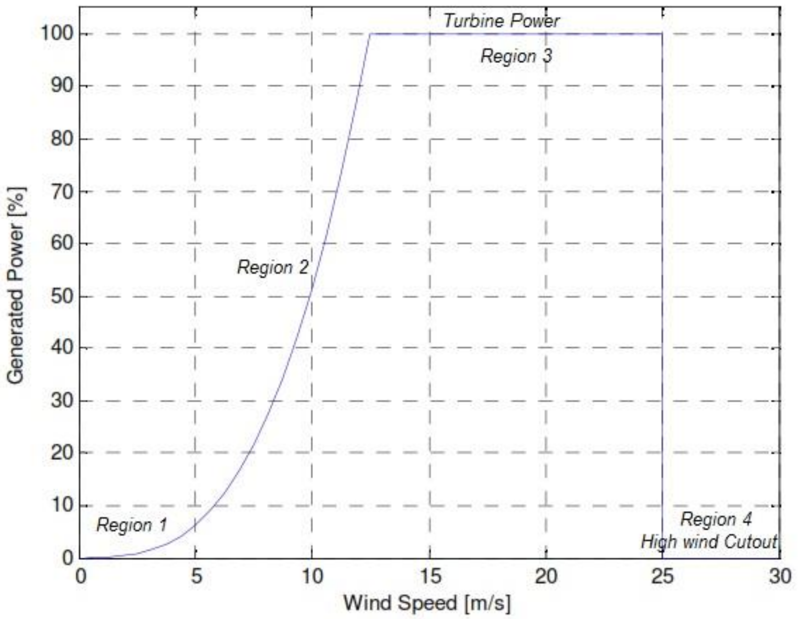


Figure 1.34: Reference power curve for a variable speed wind turbine

In this study, the focus is on the wind turbine normal operation zones, which are the second and the third regions [1]. For practical reasons, the transition between these zones must be done as smoothly as possible that means they might be divided into some other sub-regions.

Assuming that the wind turbine operates in region two and three, the control system switches between two different modes. In the first mode, power optimization can be achieved by setting the pitch angle to zero until the nominal power value is reached, while the target of the second mode is following the power reference in high wind speed. In this case, the generator torque is set to the rated value and blades are pitched to control the turbine angular speed, so that the power does not exceed the nominal one. In the following, all the details of a HAWT turbine controller are explained more deeply.

1.4.5.2 Wind Turbine Controller

To make the variable speed wind turbines operate in region two, the control objective is to maximize the captured energy. This means the turbine has to operate at the peak of the $C_P(\lambda, \beta)$ curve of the rotor. The power coefficient C_P, as described in section 1.4.2.1, is a function of the *tip-speed ratio λ and the blade pitch β. Given R, the radius of the blades, the wind speed v_ω and the angular rotor speed ω_r, The TSR λ is defined as:*

$$\lambda = \frac{\omega_r.\mathrm{R}}{v_\omega} \tag{1.22}$$

Power coefficient C_P increases the rotor power P. Therefore, operation at its maximum value $C_{P_{max}}$ is desirable. Note that C_P can be negative, which corresponds to operating the generator as a motor while drawing power from the utility grid. The C_P curve depends on the condition of the blades, as well. For example, icing or residue buildup on the blades pushes the C_P surface downward, causing the captured energy to reduce. In this project, we assumed that the blades are clean.

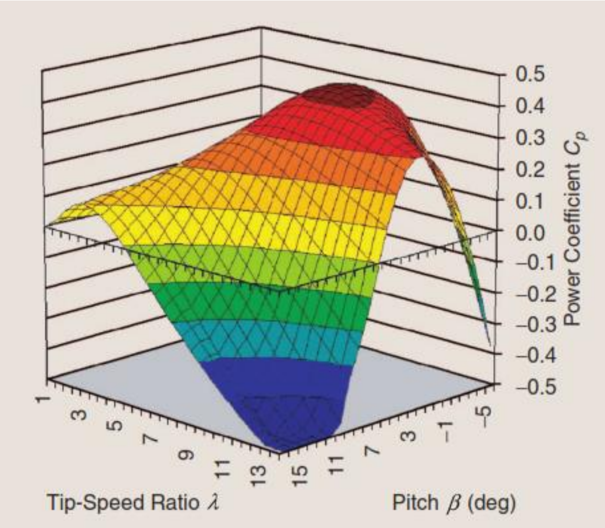


Figure 1.35: Power coefficient curve versus tip speed ratio and pitch for an industrial variable speed wind turbine

The standard control law for operation of variable-speed turbines in region two is to let the control torque (reference torque to the converter) $\tau_{g,r}$ be given by:

$$\tau_{g,r} = K_{opt}.\ {\omega_r}^2 \tag{1.23}$$

Where the gain K_{opt} is obtained from:

$$K_{opt} = \frac{1}{2}\rho A R^3 \frac{C_{P_{max}}}{\lambda_{opt}} \tag{1.24}$$

As explained above, the optimum value of the tip speed ratio λ_{opt} is obtained at the peak of the $C_P(\lambda, \beta)$ curve, which is the maximum power coefficient. ρ is the air density and A is considered as the area swept by the turbine blades.

Apart from the converter reference signal, the pitch reference angle is held at zero degrees, in which the maximum wind speed is absorbed by the blades within this region.

$$\beta_{ref} = 0 \tag{1.25}$$

While the rated power is achieved ($P_g[n] \geq P_r[n]$), by setting the converter torque at the reference value $\tau_{g,r}$, the controller switches to the mode corresponds to the region three. In this zone the control objective is to follow the power reference, P_r. This can be accomplished by controlling the blade angles β_k, in a manner that the power coefficient C_P and the tip speed ratio decrease to keep the generated power constant while the wind speed increases. In an industrial control scheme a PI controller is used to keep ω_r at the rated value by changing β_r.

To conclude about the control law, let's redefine mode one as the power optimization region and the second mode as the power reference following region. The control system switches to the second mode when the condition (1.26) holds and switches back to the first mode while the condition (1.27) becomes valid, in which the rated speed of the generator decreases to some extent and pass a small ω_Δ threshold.

$$P_g[n] \geq P_r[n] \quad \wedge \quad \omega_g[n] \geq \omega_{nom} \tag{1.26}$$

$$\omega_g[n] < \omega_{nom} - \ \omega_\Delta \tag{1.27}$$

The control signal (Converter reference signal) in zone two is obtained through the equations (1.23), (1.24) and (1.25). The so-called PI controller that is activated within the third region is as follows:

$$\beta_r[n] = \beta_r[n-1] + k_p e[n] + (k_i.T_s - k_p)e[n-1] \tag{1.28}$$

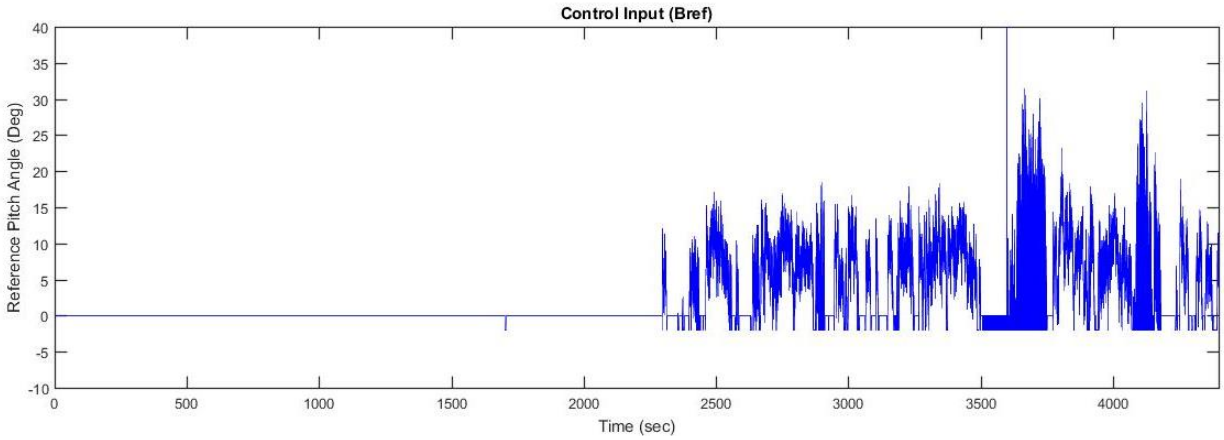


Figure 1.36: Reference Pitch angle in both regions varying with time and imposed to some faults

While the rotor speed has to reach to its nominal value by eliminating the following error through changing the pitch reference.

$$e[n] = \omega_r[n] - \omega_{nom} \tag{1.29}$$

In this case, the converter reference is used to suppress fast disturbances according to the equation (1.30):

$$\tau_{g,r} = \frac{P_r[n]}{\omega_g[n]} \tag{1.30}$$

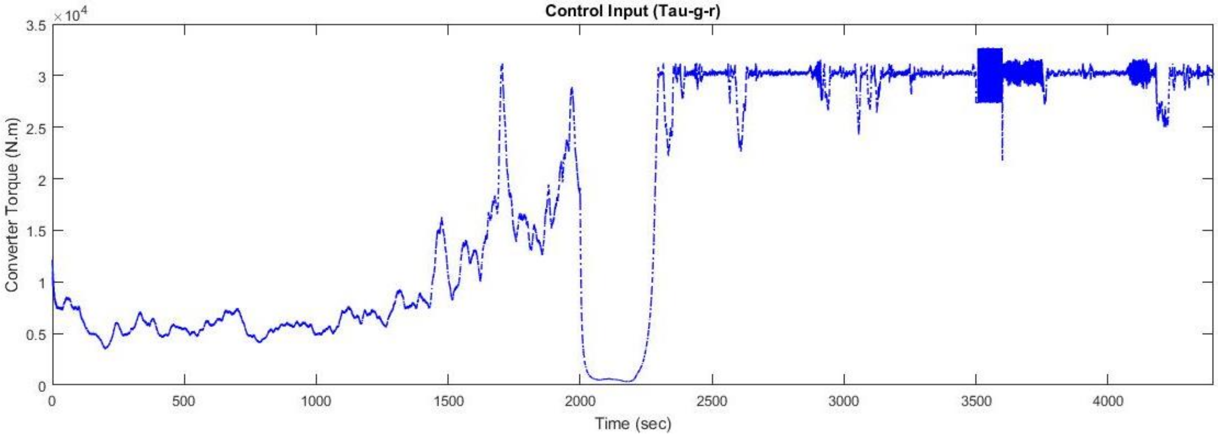


Figure 1.37: Converter Torque reference varying with time

References

[1] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines–A benchmark model," in Proc. 7th IFAC Symp. Fault Detection, Supervis., Safety Tech. Process., Barcelona, Spain, Jun. /Jul. 2009, pp. 155–160.

[2] T. Burton, D. Sharpe, N. Jenkins, E Bossanyi; Wind Energy Handbook; JOHN WILEY & SONS, LTD

[3] Barry Dolan; Wind Turbine Modelling, Control and Fault Detection; Technical University of Denmark, Kongens Lyngby 2010, IMM-M.Sc.-2010-66

[4] Christian Dobrila, Rasmus Stefansen; Fault Tolerant Wind Turbine Control; Master's thesis, Technical University of Denmark, Kgl. Lyngby, Denmark, 2007.

[5] https://energy.gov/eere/wind/how-do-wind-turbines-work

[6] http://xn--drmstrre-64ad.dk/wp-content/wind/miller/windpower%20web/en/kids/index.htm

[7] https://isaacbrana.wordpress.com/2010/07/12/brake-systems-in-wind-turbines/

[8] http://home.uni-leipzig.de/energy/ef/15.htm

Chapter 2:
Fault Diagnosis

Modern society depends strongly upon the availability and correct function of complex technological processes. Technological systems like large power distribution networks, mobile communications and manufacturing systems are vulnerable to faults which might potentially lead to economic impacts or even a damage to the environment and the human operators. As a consequence, the detection and the handling of faults play an increasing role in modern technology, where many highly automated components interact in a complex way such that a fault in a single component may cause the malfunction of the whole system.

In this chapter, the term fault and its resources, as well as its difference with failure will be defined. Then, the fault-tolerant approach that prevents a fault to become a failure is explained. At the last part of this chapter, fault locations and scenarios that are imposed on the wind turbine for diagnosis purposes are introduced.

2.1 Fault

A fault is an event that changes the behaviour of a technological system, such that the system does no longer satisfy its purpose. This event might take place due to an internal event in the system, like component ageing and wear-out, a change in the environment or a wrong control action by a human operator. As a consequence, it leads to a degraded system performance or even loss of the system function.

In technical automatic control systems, faults may happen in sensors, actuators, the components of a plant or within hardware or software of the control unit. But they have to be found as quickly as possible. Then, some decisions have to be made to stop the propagation of their effects into the whole system. Such decisions make the system fault tolerant. Being fault tolerant, after a short time of degraded performance caused by the appearance of a fault, the system function recovers in a way to satisfy the system objectives. In another word, a fault is an event that can be worked around. However, in the absence of such mechanism, a fault might lead to a failure which is an irrecoverable event. The term failure refers to the inability of the system or a component to accomplish its function. As a result, it has to be shut off for safety and maintenance services. Note that the failure chance increases by the system complexity.

Faults can be classified as "deterministic" or "random" type. The former occurs whenever certain conditions, usually design mistakes, take place; while the latter happens according to a certain probability, mainly because of system components ageing and wear out or random external events. To avoid them, their probability must be kept under a given target. Deterministic faults have to be avoided along the system design and development. However, it does not necessarily mean that design mistakes do not lead to random faults. Considering its cost, there are different statistical tests on prototypes to discover and prevent them at the development stage.

2.2 Fault Diagnosis

As far as the open loop system is concerned, the system under consideration can be decomposed into three parts including sensors, actuators, and system dynamics.

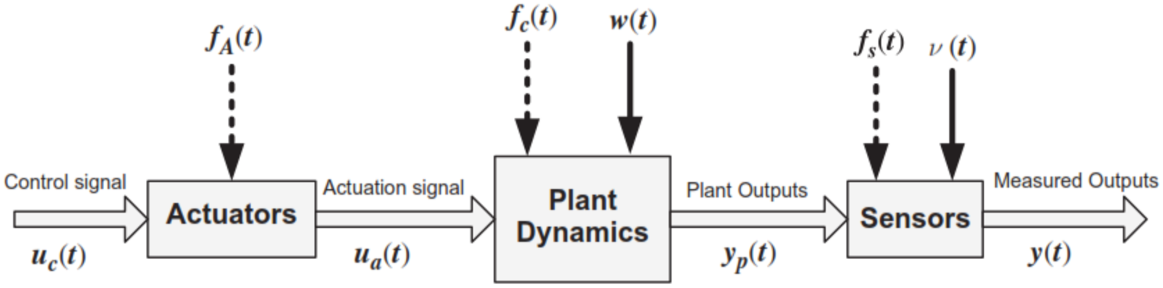


Figure 2.1: Decomposition of the open loop system components and possible faults

As can be observed from figure 2.1, faults may occur in any of the three components of the open loop system. Furthermore, the plant dynamics and the sensor measurements are always affected by external system disturbances and measurement noises. A reliable fault diagnosis system should be able to distinguish faults from system disturbances and measurement noise. More precisely, the fault diagnosis system must be robust to these uncertainties while remaining sensitive to faults.

Sensors convey information about system behaviour and its internal states. Therefore, sensor faults may cause substantial performance degradation of all decision-making systems or processes that depend on data integrity. For example, in a feedback control system, the presence of sensor faults may deteriorate state estimations and consequently result in inefficient control.

Common sensor faults include bias, drift, performance degradation or loss of accuracy, sensor freezing and calibration error. Figure 2.2 depicts the effect of the above faults on system measurements.

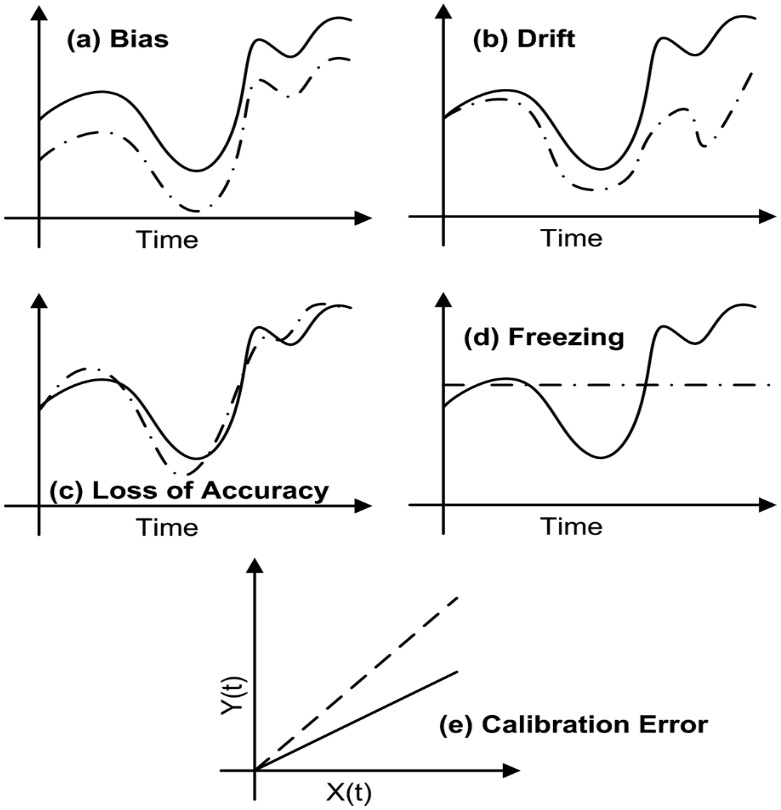


Figure 2.2: The effect of various sensor faults on system measurements

Actuator faults are usually dependent on the actuator type. Actuators transform control signals to proper actuation signals such as torque and force that drive the system. Therefore, consequences of the presence of faults in system actuators interrupt the influence of the controller on the plant and may vary from higher energy consumption (due to an incipient fault) to total loss of control (due to total failure of an actuator).

The component faults are usually represented as cases, where some undesired changes in the system operation make the nominal dynamic equation of the system invalid. These faults are also dependent on the system being monitored.

Fault diagnosis is the problem of autonomously detecting the presence, isolating the location, and identifying the type and severity of any of the three aforementioned faults in a system.

2.3 Fault Tolerant Control

Fault tolerant control has to prevent a fault leading to a failure. It is achieved by a proper interaction between a system plant and its control system.

The conventional approach to design a fault tolerant control system contains different steps through separate modules; modelling of the plant system, design of the controller, FDI (Fault Detection and Isolation) scheme, and a method for reconfiguring the control system. These stages can be performed separately or by applying a combined method. In the FDI step, the existence of a fault has to be detected and its type to be identified. In the control reconfiguration stage, the controller has to be adapted to the faulty situation, so that the overall system continues to satisfy its goal.

The motivation of applying such a method is the essence of increasing the reliability[1]*, availability*[2]*, safety*[3] *and dependability*[4] *of the technological system under observation and control. To be precise, with a fault-tolerant controller the overall system remains operational and safe*[5]*, even after the appearance of a fault. [2]*

Figure 2.3 provides an intuitive idea about the different regions of a plant behaviour with respect to the parameters y_1, y_2 that describe the system performance and the duty of the fault tolerant control system. In the region of required performance, the system satisfies its function and the controller makes the nominal system remain in this region despite disturbances and uncertainties. The region of degraded performance shows where the faulty system is allowed to remain. The fault tolerant controller should be able to initiate recovery actions that prevent a

[1] *The probability that a system performs its intended function for a specified period of time under normal conditions. It evaluate the frequency of a fault, although they cannot say anything about the current fault status.*

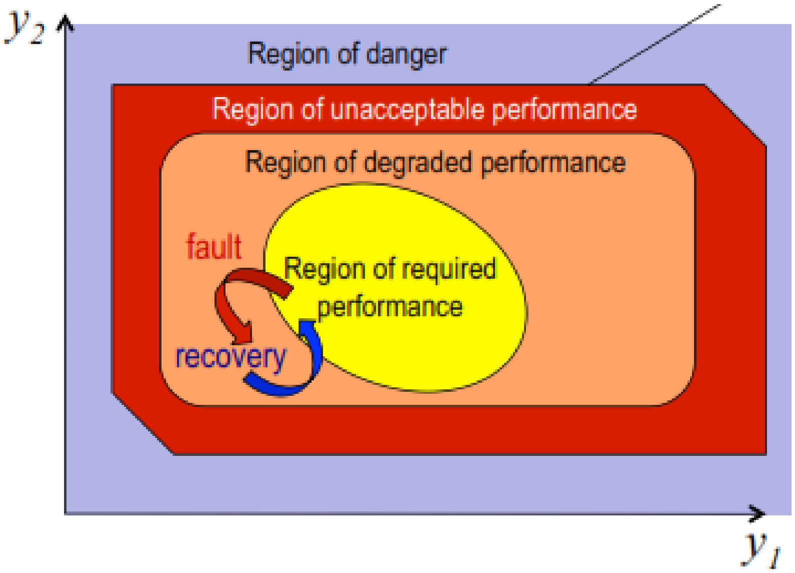
[2] *The probability of a system to be operational when needed. It also depends on the maintenance policies that are applied to the system components.*

[3] *Describes the absence of hazard even under fault/failure conditions. A safety system takes the technological system into a safe state (sail-safe) that prevents any danger for people and environment.*

[4] *A dependable system is a fail-safe one with high availability and reliability.*

[5] *In some references, "Fault Tolerance" is intended as the capability of providing fail-safe behavior, instead of fail-operational.*

further degradation and move the system back into the region of required performance. At the threshold between the two regions, the supervision system is invoked to diagnose the fault and adjusts the controller to the new situation.



y_2
Region of danger
Region of unacceptable performance
Region of degraded performance
fault
Region of required performance
recovery
y_1

Figure 2.3: Different regions of a plant system function

The region of unacceptable performance should be avoided by means of a fault tolerant controller. It lies between the region of acceptable performance, in which the system should remain and the region of danger, which the system should never reach by means of a safety system that is invoked if the outer threshold of the region of unacceptable performance is exceeded to avoid danger for the system and its environment. As described before, the safety system and the fault tolerant controller work in separate regions of the signal space and satisfy complementary aims.

The conventional architecture of fault tolerant control consists of two main blocks. Firstly, the diagnosis block that uses the measured inputs and outputs and tests their consistency. The result is a characterization of the fault with sufficient accuracy for the second block that is the controller re-design. In this part, the information is used to adjust the controller to the faulty situation. It might result in a new control configuration or updated parameters of the same controller.

It is worth mentioning that the existence of a fault tolerant controller is not a property of the controller or the control re-design method, but a property of the plant subject to faults. Note that fault tolerant control is not the unique approach to fault tolerance. One of the commonly-used methods, particularly in industrial plants, is component redundancy. Sometimes, these physical duplications are avoided by analytic redundancy [3]. Fault tolerant control is a technique to reduce such duplications, when possible.

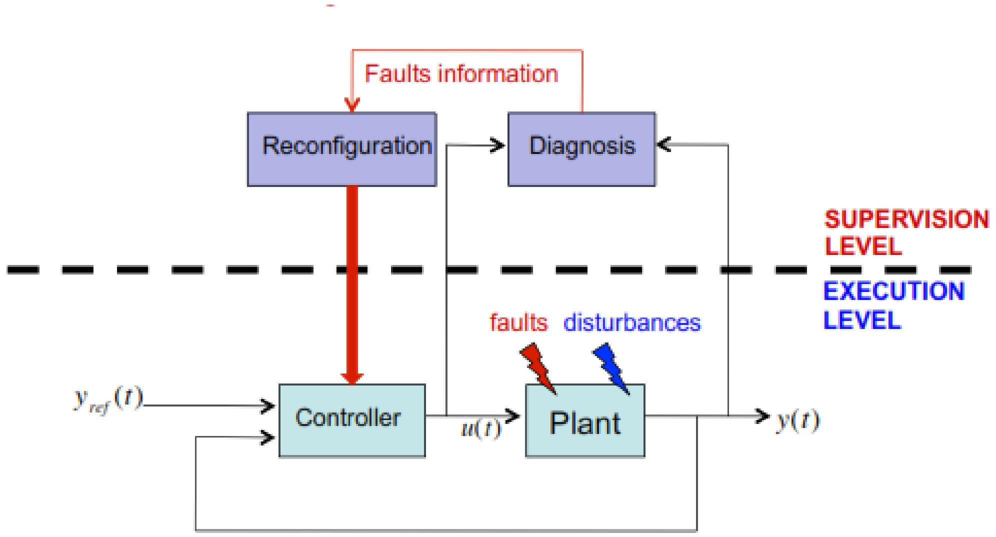


Figure 2.4: Fault Tolerant Control Architecture

To sum up, no fully-guaranteed fault tolerance (fail-operational) or fail-safe behaviour can be obtained. For each system, it is necessary to understand what faults and failures are critical for the system availability and safety. Then, the designer has to decide about the level of fault tolerance to provide for the system with respect to the environmental impacts and costs. In the end, it is his task to find the best solution to satisfy the goals by applying the proper diagnosis algorithms, component redundancies, safety systems and FTC. In the next chapter, the diagnosis algorithm that is applied in this project is deeply explained.

2.4 Fault Scenarios

As previously mentioned in chapter 2.1, faults may occur in different parts of an entire system. In the benchmark model, a number of faults are considered as some test signals, being presented at predefined times. They are covering various kinds of possible faults in a wind turbine. They take place in various test series at different operational points of the turbine, with a time-varying wind input.

In this section, these different kinds of faults and their degree of severity are described. Some are very serious and should result in a fast safe shut down of the wind turbine and others are less severe in a way that the controller can be accommodated to handle these faults without any damage to the system or the environment. [1]

2.4.1 Sensor Faults

The measurement system in a HAWT is a complex system with many variables being monitored continuously to make sure that the wind turbine is within its operating range, and thus it is safe to continue its energy production without any risk of damaging vital parts of the turbine. These sensors carry some data that has to be reliable. Any deviation from the real information, because of their malfunction, may lead to a controller action or a system shut down. In order to increase the reliability of the system, as described in section 2.3, physical redundancy has been considered as a pre-processing countermeasure, as well as the analytical method, explained in chapter 3.

In the benchmark model, the main focus is on monitoring the blades status, rotor and generator speed measurements. The former are either electrical or mechanical faults in the blade position sensors and can result in either a fixed value or a gain factor on the measurements. They are the first type of faults introduced to the system. The latter are speed measurements, implemented mainly by encoders. Such faults can be either due to electrical and mechanical reasons, leading to a fixed value or a gain factor on the measurements. Rotor and generator measurement faults are considered as the second and the third types, respectively.

2.4.2 Actuator Faults

Actuator faults that might happen in a variable speed wind turbine can be divided into two major types; converter malfunctions and pitch actuator faults.

The conversion of an input AC power at a given frequency and voltage to an output power at different frequency and voltage can be obtained with static circuits called power converters, containing controllable power electronic devices. The power converter in a variable speed wind turbine is an essential assembly that feeds the active and reactive power into the grid. In modern wind turbines with variable-speed generator configurations, that are Doubly Fed Induction Generator or Synchronous Generator, back to back power converters are introduced to control the generators speed and torque. The power converter malfunction will deviate the operation of a wind turbine and generate disturbance to the Grid. There are several types of faults in power converters, mainly appeared in the internal converter control loop, like open-switch or short-switch of the IGBT or DC-link fault. Such faults are considered as an offset and named as type (4).

Apart from the converter system malfunctions, three pitch actuators operate asynchronously, leading to an unbalanced load. Consequently, the pitch system actuator includes two types of faults. The first one is a pressure drop in the hydraulic supply system that may appear by a leakage in a hose or a blocked pump. The pressure drop causes the oscillatory unbalanced load, leading to the damage of either the blade or the gearbox. This abrupt change in the dynamics of the pitch system has been taken into account as (5a) fault type.

The second type of pitch actuator faults (5b) is for the high air content in the oil. Oils are not completely free of air. Whether during operation or storage in barrels, oils are constantly in an exchange process with their air-containing environment, leading to air dispersions inside the oil. Even if the oil is free of air bubbles, it might have a proportion of dissolved air. As long as the air remains dissolved in the oil, this generally causes no particular issue. However, free air bubbles may lead to serious disruptions in equipment operation, including cavitation, increased oxidation tendency, decreased oil pump capacity and shortened oil life. Considering that air is much more compressible than oil, it is impossible to control the varying air content level in the hydraulic oil. This issue will change the dynamics of the hydraulic actuator.

2.4.3 System Faults

As described in chapter 1.3.8, the drive-train system transmits the power from the rotor to the generator. The connection between the rotor shaft and the low-speed shaft of the gearbox is performed by the friction hub-shaft joint, having a friction coefficient, in total that varies from one manufacturer to another. This friction coefficient is imposed to change by time, leading to the increased level of vibrations of the drive-train. This very slow trend changes the so-called coefficient and thus the dynamics of the system. Regarding its correlation with the rotor and generator speed, it is considered as a "changed dynamics" fault type by number *(6).*

No	***Location***	***Type***	***Symbol***	***Severity***	***Consequence***	***Development Time***	$T_D^{desired}$
1a	*Sensor*	*Fixed Value*	$\Delta\beta_{1,m1}, \Delta\beta_{2,m1}, \Delta\beta_{3,m1}$ $\Delta\beta_{1,m2}, \Delta\beta_{2,m3}, \Delta\beta_{3,m2}$	*Low*	*False Measurement, System Reconfiguration*	*Medium*	$< 10T_s$
1b	*Sensor*	*Gain Factor*	$\Delta\beta_{1,m1}, \Delta\beta_{2,m1}, \Delta\beta_{3,m1}$ $\Delta\beta_{1,m2}, \Delta\beta_{2,m3}, \Delta\beta_{3,m2}$	*Low*	*False Measurement, System Reconfiguration*	*Medium*	
2a	*Sensor*	*Fixed Value*	$\Delta\omega_{r,m1}, \Delta\omega_{r,m2}$	*Low*	*False Measurement, System Reconfiguration*	*Medium*	
2b	*Sensor*	*Gain Factor*	$\Delta\omega_{r,m1}, \Delta\omega_{r,m2}$	*Low*	*False Measurement, System Reconfiguration*	*Medium*	
3a	*Sensor*	*Fixed Value*	$\Delta\omega_{g,m1}, \Delta\omega_{g,m2}$	*Low*	*False Measurement, System Reconfiguration*	*Medium*	
3b	*Sensor*	*Gain Factor*	$\Delta\omega_{g,m1}, \Delta\omega_{g,m2}$	*Low*	*False Measurement, System Reconfiguration*	*Medium*	
4	*Actuator*	*Offset*	$\Delta\tau_g$	*High*	*Slow Torque Control, Indicate Serious Problems*	*Fast*	$< 5T_s$
5a	*Actuator*	*Changed Dynamic*	$\Delta\beta_1, \Delta\beta_2, \Delta\beta_3$ *Hydraulic Pressure drop*	*High*	*Slow Control Action, Pump issues or leakage*	*Medium*	$< 8T_s$
5b	*Actuator*	*Changed Dynamic*	$\Delta\beta_1, \Delta\beta_2, \Delta\beta_3$ *Air Dispersion in oil*	*Medium*	*Slow Control Action, High Air in Oil*	*Slow*	$< 600T_s$
6	*System*	*Changed Dynamic*	$\Delta\omega_r, \Delta\omega_g$	*Medium*	*Increased Vibrations in the Drive Train*	*Very slow*	*free*

Table 2. 1: Considered Faults and their features

The total number of faults to be supervised is therefore 28, including twenty sensor faults, seven actuator faults and a system fault. However, it can be seen that some faults are similar, particularly those related to the sensors. As a result, if one is able to detect a fault from each of the blade position sensors, then under the same conditions, it is possible to detect faults on sensor measuring positions of other blades. Looking at this viewpoint, it can be seen that ten different kinds of faults have been introduced distinctly, as classified in table 2.1. To be noted, the process has other measuring sensors, for instance, for the wind speed that is not observed in this study.

2.4.4 **FDI Requirements**

A dynamical system behavior, subjected to some faults, deviates from the healthy situation. The task of the FDI system using an I/O pair is solving the problem in three main steps:

1. *Fault Detection which decides whether or not a fault has occurred. This step determines the time at which the system is subject to some faults.*
2. *Fault Isolation that finds in what component a fault has taken place. This step determines the location of the fault.*
3. *Fault identification and estimation that identifies the fault and estimates its magnitude. This step determines the kind of fault and its severity.*

To handle all the above faults, they have been prioritized according to some parameters. The main factors, while facing a risk that might cause a hazard is its probability and the severity of the consequences. In another word, the dimensions of a fault are defined by its frequency and consequences. The former can be defined by the time of development, while the latter is characterized by the magnitude of the impacts. Taking a look back to the table 2.1, all the faults, their severity and consequences are listed.

Considering physical redundancy of the measurements, sensor faults are not that critical and harmful to the system. Hence, they are of lower priority with respect to their low severity and medium time of development. Strictly speaking, no sensor fault could be a problem if it is detected immediately, and thus the sensor system is reconfigured. According to the FDI requirements, the *desired detection time can be no more than ten times of the sampling time ($T_D < 10T_s$). Note that, the sampling time of the control system is equal to 0.01 seconds ($T_s = 0.01\ s$).*

Regarding the above-mentioned constraints, control actions have to be implemented as early as possible. It means that the severity of actuators, especially the torque controller is high.

The converter that sets the generator torque to control the rotational speed of the generator and the rotor might have an offset that should be detected rapidly. Abrupt pressure drop in the hydraulic pumps of the pitch system has to be handled quickly, as well. However, the air content in the oil is a gradual trend and does not lead to a fast dynamic change. To satisfy the FDI requirements, the converter faults should be detected in $T_D < 5T_s$, while the pump pressure drop is a bit more tolerable, equal to $T_D < 8T_s$. For the pitch system faults due to the air content in oil, the maximum acceptable detection time is $T_D < 600T_s$.

The fault related to the dynamic changes of the drive train is an indication of the system ageing. Although it is very slow and slight, in case of being neglected at all, it can result in a total breakdown of the drive train. Ultimately, detection of the drive train friction rise should just achieve freely.

Apart from the detection time, there are other constraints over the number of false[1] *and missed*[2] *detections. The number of false detections should be kept low and the mean time between false detections have to be larger than 100000 samples (1000 sec). In case of a false positive detection, the detection should be off after three samples. In contrary, no missed detection is acceptable. In other words, all the faults should be detected.*

Finally, there are some issues that cause uncertainties and the FDI system has to be robust in this regard. One of the major problems in the wind turbine control system is that the wind turbine is driven by a disturbance, the wind. However, as already have been discussed in chapter 1.2, it can be measured to some degree, but only with a poor measurement resulting in a high noise

[1] A false positive error, commonly called a false detection, is a result that indicates a given condition exists, while it does not. It is also named as the type I error in the statistical test theory.

[2] A false negative error, referred to as type II error or missed detection, is a test result which wrongly indicates that a particular condition or attribute is absent.

level on this measurement, as well as a large risk of an offset of the wind measurement, it can be calibrated, but it should be considered in the FDI system. Nonlinearities in the aerodynamics of the turbine as well as the switching control structure, are the other sources of uncertainty. The FDI system should as well be robust towards uncertainties in this aerodynamic model, partly because it is difficult to exactly measure the mappings on the specific turbines, they might as well change with time due to debris build up.

2.4.5 Fault Accommodation Requirements

This benchmark model contains both kinds of fault for which the control system should be reconfigured to continue power generation in its optimal condition, as well as very severe faults which should result in a safe, fast shutdown of the wind turbine. The latter includes the control actuator faults related to the converter (fault 4) and the pressure drop on the hydraulic pumps that lead to an unbalanced load over the blades or the gearbox (fault 5a).

The rest of the faults should be accommodated in a way, by which the turbine continues its safe operation. In all cases, detection of faults, as well as the automatic control action should be reported to the system operator.

Considering the physical redundancy, in case of only one sensor fault, system performance should not decrease. In other cases, some decreases in the system performance could be expected[1]. Finally, large transients when accommodating the fault should be avoided.

[1] Fail Operational

References

[1] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines–A benchmark model," in Proc. 7th IFAC Symp. Fault Detection, Supervis., Safety Tech. Process., Barcelona, Spain, Jun. /Jul. 2009, pp. 155–160.

[2] A. Paoli, A. Tilli; Diagnosis and Control, A General Overview – LM Automation Engineering – DEI University of Bologna"

[3] A. Paoli, A. Tilli; Diagnosis and Control, Principles of Fault Detection and Isolation – LM Automation Engineering – DEI University of Bologna"

[4] K. E. Johnson, L. Y. Pao, M. J. Balas, L. J. Fingersh; "Control of Variable-Speed Wind Turbines, Standard & Adaptive Technics"

[5] Barry Dolan; Wind Turbine Modelling, Control and Fault Detection; Technical University of Denmark, Kongens Lyngby 2010, IMM-M.Sc.-2010-66

Chapter 3:
Support Vector Machines

Nowadays, research into fault diagnosis in industrial machines, such as wind turbines, is of great interest. This work presents a multi-sensory system for fault detection in a variable-speed wind turbine, combined with a data-mining solution for the classification of its healthy-faulty states. Support vector machines (SVMs) is selected for this classification task.

This chapter starts with an introduction to the importance of the classification techniques in the modern world. Then, the theory of support vector machines, as a high-quality classifier, is developed in both linear and nonlinear cases. SVM design steps are explained in details, since we are going to apply it for the fault detection of the previously described wind turbine.

3.1 Classification Analysis

In diverse scientific and technical fields, we eventually face sets of data with small or large sizes. However, the most important matter is not just obtaining such a database, but having the ability to obtain a higher level of knowledge according to that database. Many scientists and scholars in different fields of science, along with creativity and expertise, have a common and significant characteristic, which is the ability to provide interpretations and conclusions from the observations and collected data.

In various scientific disciplines, all the theories obtained by researchers share a common component, i.e. the acquisition of rules require a concise explanation of available data. This operation and a series of related works is the topic of one of the subcategories of computer science and statistics, called Data Mining. The overall process of knowledge extraction from data has a more general concept known as Knowledge Discovery from Data or KDD.

Classification is one of the data mining techniques that classify unstructured data into the structured groups of data. Providing intelligent decision making, it helps the user for knowledge discovery and future plan. There are two phases in a classification task. The First step is a learning process in which a training data sets are analyzed to create some rules or patterns and the second phase is the data evaluation or test.

There are two approaches to the learning procedure; supervised and unsupervised. In supervised learning algorithms, training the classifier is based on some observations or instances of the input and target data whose category membership is known, in advance. But in unsupervised learning techniques, known as clustering, sets of possible classes are partitioned into groups based on data similarity. Then, different labels are assigned to the so-called clusters. Here, SVM classifier is applied as a mean for fault detection in the wind turbine, by exploiting a supervised learning algorithm that will be discussed in the following sections.

3.2 Support Vector Machines

Support Vector Machine is a supervised method that analyzes data and recognizes patterns used for classification. Given a training set and the data needs to be classified into two classes, an SVM classifier builds a model that assigns the data into each of the categories.

The main idea of a support vector machine as a feed-forward network is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. The machine achieves this desirable property by following a principled approach rooted in the statistical learning theory introduced in 1964 by Vapnik and Chervonenkis. More precisely, the support vector machine is an approximate implementation of the method of structural risk minimization. This induction principle is based on the fact that the error rate[1] of a learning machine on test data (i.e., the generalization error rate) is bounded by the sum of the training-error rate[2] and a term that depends on the Vapnik-Chervonenkis (VC) dimension[3] [3]. Only recently, SVM was introduced as a machine learning algorithm for

[1] *The inaccuracy of the output predictions (classification) is called error. The expectation of the test set error is called Error-rate. It is normally measured on the testing set only. In this case, it may also be called the off-training-set error-rate or OTS error.*

[2] *If error is calculated on the training set, then it would be called the training error-rate. In another word, it is equal to the normalized sum of the absolute values of the errors on the training set.*

[3] *The VC Dimension of a model "f" is the maximum number of points (in our case, a given set of functions whose parameters are found by a training algorithm) that can be arranged so that "f" shatters them. A classification model f with some parameter vector θ is said to shatter a set of data points $\{x_1, x_2... x_n\}$ if, for all assignments of labels to those points, there exists a θ such that the model f makes no errors when evaluating that set of data points. (Example: with a line in R^2 you can shatter three points)*

classifying data of two different classes by Vapnik in 1995. Accordingly, the support vector machine can provide a good generalization performance on pattern classification problems despite the fact that it does not incorporate problem-domain knowledge. This attribute is unique to support vector machines. [2]

3.2.1 Binary Classification

Given Training Data (xi, yi) *for i=1,…,N , with* $x_i \epsilon R^d$ *and* $y_i \epsilon \{-1 , 1\}$ *, a binary classifier* $f(x_i)$ *is such that :*

$$f(x_i) = \begin{cases} \geq 0 & y_i = +1 \\ \leq 0 & y_i = -1 \end{cases} \tag{3.1}$$

Then, $y_i . f(x_i) > 0$ *is a correct classification.*

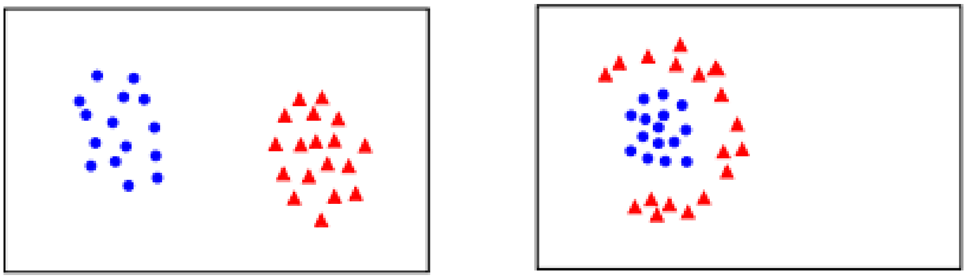


Figure 3.37: Binary Classification

Let's consider an example to understand these concepts [4]. We have a population composed of 50%-50% Males and Females. Using a sample of this population, we want to create some set of rules which will guide us the gender class for rest of the population. Using this algorithm, we intend to build a robot which can identify whether a person is a male or a female. It is a sample problem of supervised classification analysis. Using some set of rules, we will try to classify the population into two possible segments. For simplicity, assume that the two differentiating factors are: height of the individual and hair length. Following is a scatter plot of the sample.

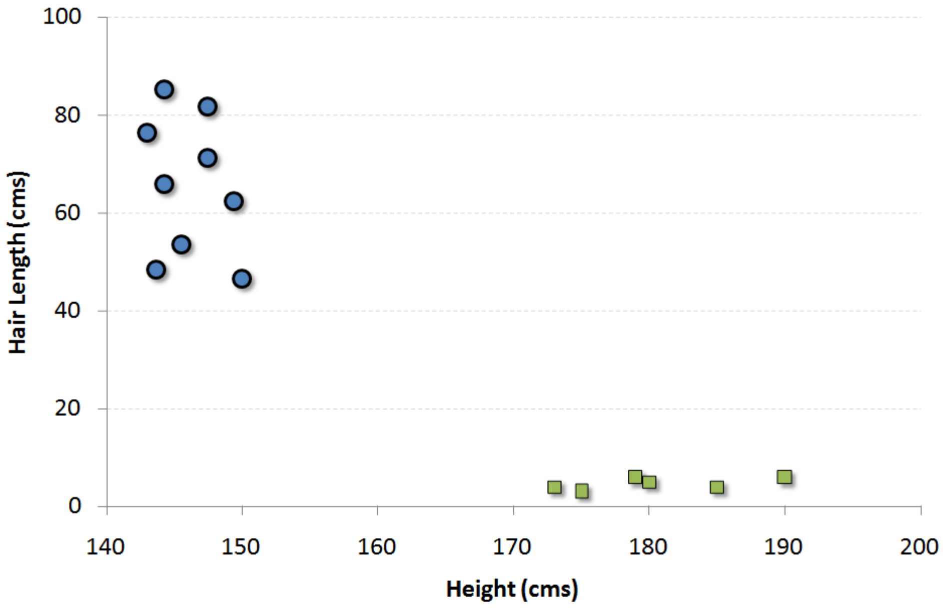


Figure 3.2: A dataset to be segregated by a binary classifier

The circles in the plot represent females and the squares represent males. A few expected insights from the graph are:

1. Males in our population have a higher average height.

2. Females in our population have longer scalp hairs.

If we were to see an individual with height 180 cms and hair length 4 cms, our best guess will be to classify this individual as a male. This is how we do a classification analysis.

3.2.2 Linear classifier

Support Vectors are simply the coordinates of individual observations. For instance, (45,150) in the previous example is a support vector which corresponds to a female. Support Vector Machine is a frontier which best segregates the male set from the females. In the above example, the two classes are well-separated from each other.

In general, there are linearly separable and not linearly separable sets of data:

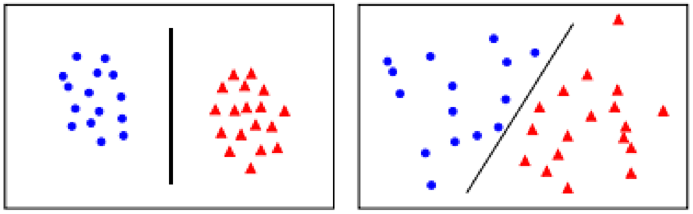


Figure 3.2.a: Linearly separable datasets

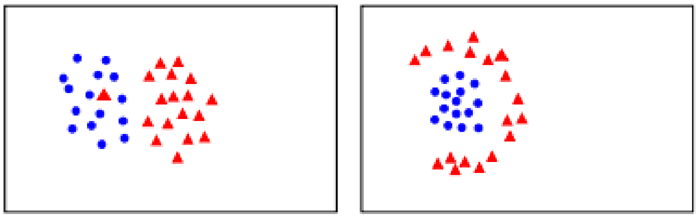


Figure 3.2.b: Nonlinearly separable datasets

There are many possible frontiers which can classify the linear dataset, in hand. Figure 3.3 shows the three possible frontiers. The easiest way to interpret the objective function in an SVM is to find the minimum distance of the frontier from closest support vector (this can belong to any class). For instance, the left line is the closest to the circles' distribution zone. As shown in the figure, the distance to the boundary is 2 units in this case. Once we have these distances for all the frontiers, we simply choose the one with the maximum distance (from the closest support vector). As can be seen, the middle out of the three shown frontiers is the farthest from its nearest support vector (i.e. 15 units).

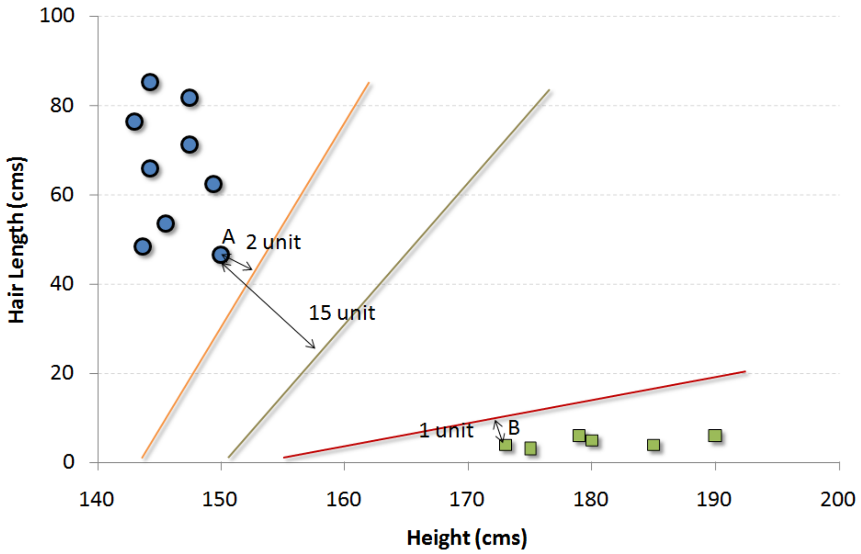


Figure 3.3: different linear classifiers according to their distance to the nearest support vector

A linear classifier has the form:

$$f(x) = w^T x + b \tag{3.2}$$

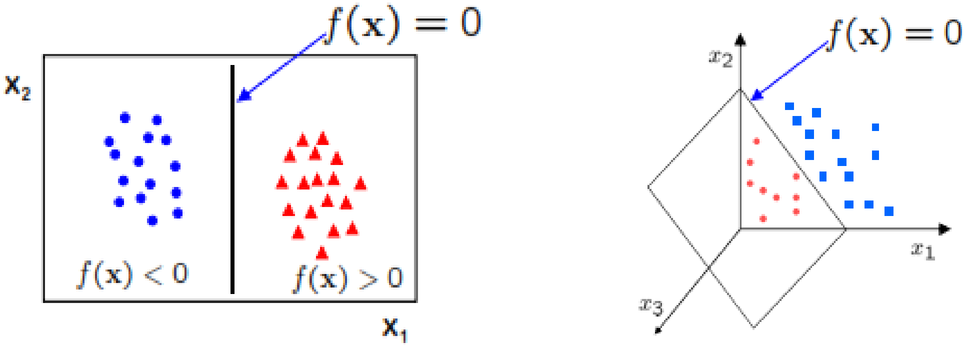


Figure 3.4: Linear classifiers in 2D and 3D space

Our first objective is to find the best w which gives the most stable line under perturbation of the inputs. Note that, in two-Dimensions, the discriminant is a line, while in 3D, it is a plane. In general, for an nD space, the classifier could be a hyperplane[1].

[1] Note that in section 3.4.2, by introducing kernel tricks for the nonlinear case, mapping to a hyperplane in nD dimensions is not necessarily required. It can belong to a higher (even infinite) dimensional space, connected to the original one with a proper (nonlinear) mapping.

For a given weight vector w and bias b, the separation between the hyperplane defined in equation (3.1) and the closest data point (support vector) is called the margin of separation, denoted by ρ. The goal of a support vector machine is to find the particular hyperplane for which the decision boundary ρ is maximized. Under this condition, the decision surface is referred to as the "Optimal Hyperplane" or the "Maximum Margin Solution".

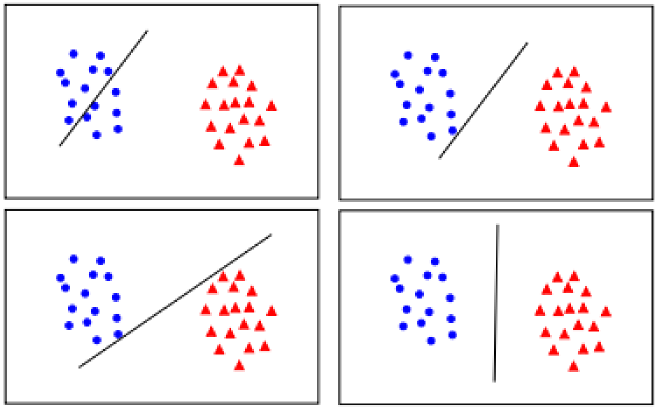


Figure 3.5: Different margins of separation

3.2.3 Nonlinear classifier

Our job was relatively easier finding the SVM in the linear case. There are cases where the distribution looked something like as follows (Not Linearly Separable)

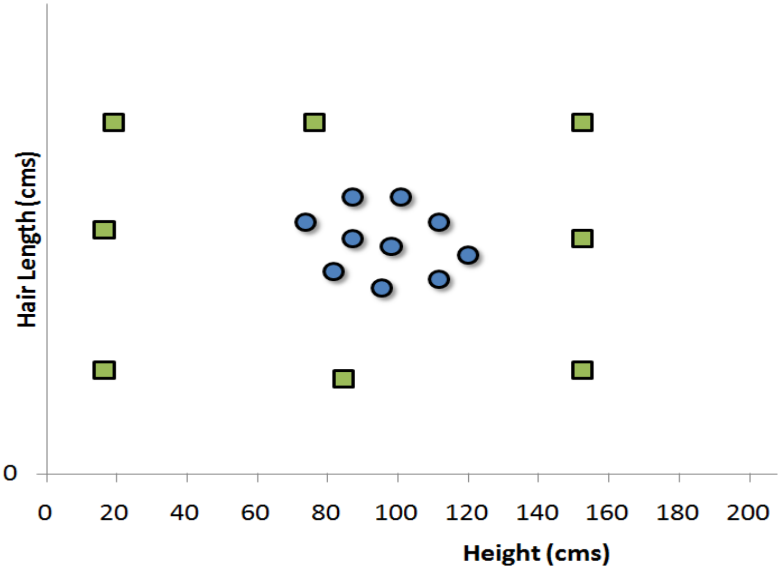


Figure 3.6: Not linearly separable dataset to be classified by a binary classifier

In such cases, there is no straight line frontier directly in current plane which can serve as the SVM. As an alternative, these data points need to be mapped to a higher dimension plane, so that they can be segregated from each other. Such cases will be covered in chapter 3.4. Just to have an attitude about them, it can be seen that some transformations will result into the following type of SVM, shown in figure 3.7.

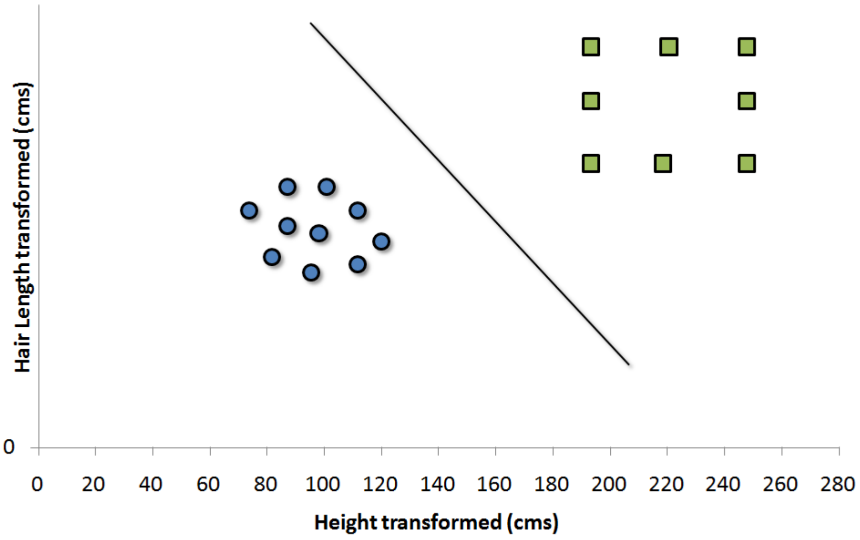


Figure 3.7: Transformation of non-linearly separable datasets to a higher dimension

Each of the squares in original distribution, in figure 3.4, is mapped on a transformed scale. In the new scale, different classes have clearly segregated. Many algorithms have been proposed to make these transformations. Some of which will be discussed in the following parts.

3.3 Optimal Hyperplane for Linearly Separable Patterns

In this section, we discuss about the construction of optimal hyperplane for the simple case of linearly separable patterns.

3.3.1 Linear SVM Formulation

- *Let* $x \epsilon \Re^n \ , \ y \epsilon \{1, -1\}$

  Objective Function $$f(x) = sign(< w, x > +b) \qquad (3.3)$$

  $w \epsilon \Re^n$

  $b \epsilon \Re$

  Hyperplane $$< w, x > +b = 0 \qquad (3.4)$$

  $$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b = 0$$

Our objective is finding the weight vector w and bias b such that the data could be linearly separated and the margin is then maximized.

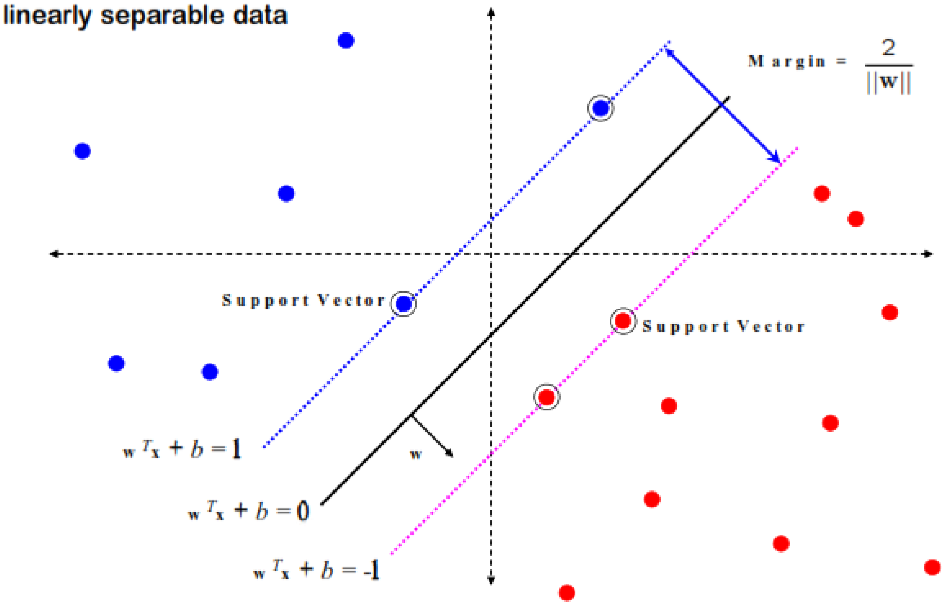


Figure 3.8: An optimal hyperplane for linearly separable patterns

Let training set $\{(x_i, y_i)\}\ i = 1, \cdots, n$ *while* $x_i \epsilon \Re^d$, $y_i \epsilon \{-1,1\}$ *be separated by a hyperplane with margin ρ. The margin is maximized while the discriminant lies between the two classes of data. Then, for each training example* (x_i, y_i)*:*

$$\begin{cases} w^T x_i + b \le -\frac{\rho}{2} & y_i = -1 \\ w^T x_i + b \ge \frac{\rho}{2} & y_i = 1 \end{cases} \tag{3.5}$$

The equation (3.5) can be rewritten as:

$$y_i(w^T x_i + b) \ge \frac{\rho}{2} \tag{3.6}$$

For every support vector x_s*, the above inequality becomes an equality. After rescaling w and b by* $\frac{\rho}{2}$ *in the equality, we obtain the algebraic distance from each* x_s *to the hyper plane, that is:*

$$r = \frac{y_s({w_0}^T x_s + b_0)}{\|w_0\|} = \frac{1}{\|w_0\|} \tag{3.7}$$

Then the margin can be expressed through (rescaled) w and b as:

$$\rho = 2r = \frac{2}{\|w_0\|} \tag{3.8}$$

Equation 3.8 states that maximizing the margin of separation between classes is equivalent to minimizing the Euclidean norm of the weight vector w*. Accordingly, the optimal hyperplane defined by equation 3.4 is unique in the sense that the optimum weight vector* w_0 *provides the maximum possible separation between positive and negative examples. This optimum condition is attained by minimizing the Euclidean norm of the weight vector* w*.*

As a result, $||w||^2$ *has to be minimized which leads to an easier optimization problem.*

$$\begin{cases} \min \|w\|^2 & where\ \|w\|^2 = w^T w \\ subject\ to & y_i < w, x > +b \ge 1 \quad for\ all\ i \end{cases} \tag{3.9}$$

This is a quadratic programming problem with inequality constraints that has to be solved through one of the optimization methods.

In the SVM case, the solution to the constrained optimization problem is determined by the saddle point of the Lagrangian function $L(w, b, \alpha)$*, which has to be minimized with respect to* w *and* b*. It also has to be maximized with respect to* α*. To relax the KKT condition, including the complementary slackness, equation 3.10 holds. All the mathematical steps of solving the constrained optimization problem by Lagrangian multipliers and the dual problem that leads to an easier solution is fully-described in the "Appendix".*

$$\alpha_i[y_i(wx_i \; + \; b) - 1] = 0 \tag{3.10}$$

That means among all the data points, only support vectors have non zero α_i*. Thus, it is only them that contribute in defining the decision boundary. In other words, the points that are far away from the decision boundary cannot be used to define the contours of that decision boundary. It is like the KNN, except that the process of figuring out which points actually matters is already done. The Lagrange multipliers* α_i *are called 'dual variables' and each training point has an associated dual variable.*

3.3.2 Linear SVM Formulation in the Dual Form

Assuming the duality problem in our SVM problem and given the training sample $\{(x_i, y_i)\}$ *for i=1… N; the objective function of the dual problem that needs to be maximized is*

$$\max \; w(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.11}$$

$$subject\ to \;\; \alpha_i \geq 0 \;\;, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \;, i = 1, \cdots, N \tag{3.12}$$

Where α_i *are non-negative Lagrangian multipliers and* $\sum_{i=1}^{N} \alpha_i y_i = 0$ *is zero by virtue of the optimality condition of equation (App.11) in the appendix.*

The function $w(\alpha)$ to be maximized depends only on the input patterns in the form of a set of dot products $\left\{x_j^T x_i\right\}$ for i=1… N. As the dot product is the projection of one vector to another, it can be considered as a notion of similarity.

To sum up, equation 3.12 mentions that we have to find all points and finding out which of them do matter (support vectors) for defining the decision boundary and then discover how they relate to each other in terms of their output labels with respect to how similar they are to one another.

Equations 3.11 and 3.12 can be solved by a quadratic programming (QP) tool. Solving this problem, a global maximum of α_i can always be found. w can be recovered by calculating the summation just over the support vectors, for which the corresponding Lagrange multiplier is non-zero (complementary slackness condition):

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i = \sum_{i \epsilon SV} \alpha_i y_i x_i \qquad (3.13)$$

When set (α^*, b^*) is obtained by solving the so-called QP, we are able to apply the SVM to classify the new sets of data. If x is a new set of data, its classifier is as follows:

$$sign[f(x, \alpha^*, b^*)] \qquad (3.14)$$

where:

$$f(x, \alpha^*, b^*) = w^* x + b^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i \, x + b^* = \sum_{i \epsilon SV} \alpha_i^* y_i x_i \, x + b^* \qquad (3.15)$$

The solution of the SVM, from the quadratic programming problem with linear inequality constraints has the nice property that the data enters only in the form of dot products in the equation. It means, given $\mathrm{x} = (x_1, x_1, \cdots, x_n)$ *and* $\mathrm{y} = (y_1, y_1, \cdots, y_n)$*; then the dot product of x and y is* $\mathrm{xy} = (x_1 y_1, x_2 y_2, \cdots, x_n y_n)$*. This is nice because it allows us to make nonlinear SVM without complicating the algorithm.*

3.4 Optimal Hyperplane for Non-Linearly Separable Patterns

The discussion thus far has focused on linearly separable patterns. In this section, we consider the more complicated case of non-separable patterns. Given such a set of training data, it is not possible to construct a separating hyperplane without encountering classification errors. Nevertheless, we would like to find an optimal hyperplane that minimizes the probability of classification error, averaged over the training set.

3.4.1 Soft Margin Hyper plane

The methodology described in section 3.3 provides a hard-margin SVM. Sometimes, applying this method may lead to such a narrow classifier that is shown in figure 3.9.a. As a countermeasure, we can get it as a classification error in order to widen the so-called hard margin and construct a second classifier with a wider margin of separation, shown in figure 3.9.b.

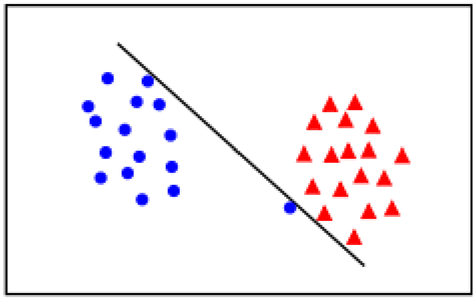


Figure 3.9.a: Narrow hard margin

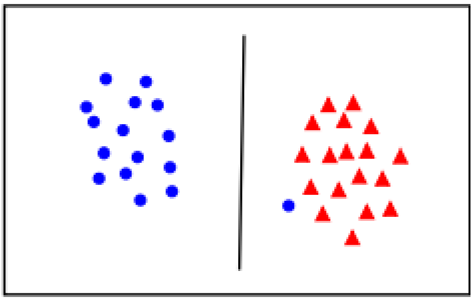


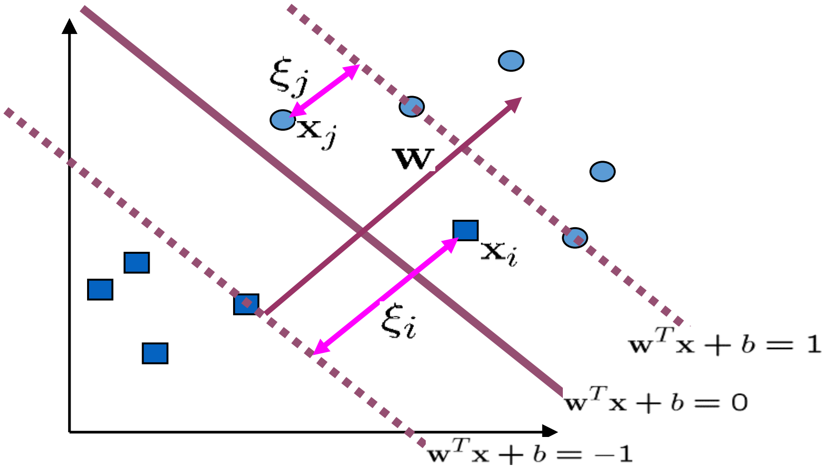
Figure 3.9.b: Violated soft margin

One strong assumption in the above procedure is that the sets of data are linearly separable. In practice, it is almost impossible to find such kind of data. One of the very first countermeasures is accepting some sorts of errors in our training data to find a more reasonable classifier. The margin of separation between classes is said to be soft, if a data point (x_i, y_i) *violates the following condition:*

$$y_i(w^T x_i + b) \geq 1 \qquad for\ \ i = 1, \dots, N \tag{3.16}$$

This violation takes place in one of two ways:

- *The data point* (x_i, y_i) *falls on the wrong side of the decision surface, as illustrated in figure 3.10 by the rectangle x*$_i$ *with error* ξ_i.
- *The data point* (x_i, y_i) *falls inside the region of separation but on the right side of the decision surface, as illustrated in figure 3.10 by the circle x*$_j$ *with error* ξ_j.



ξ_j
$\mathbf{x}_j$
$\mathbf{w}$
$\mathbf{x}_i$
ξ_i
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

Figure 3.10: Data points falls on the wrong side of the classifier x$_i$ *and inside the region of separation x*$_j$

In general, there is a trade-off between the margin and the number of mistakes in the training data.

In the above cases, by introducing a slack variable, we consider the effect of those errors in the objective function. As a result, the new optimization problem becomes:

$$\min_{w\epsilon\Re^d, \xi_i\epsilon\Re^+} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \qquad (3.17)$$

$$subject\ to\ \ y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \qquad for\ i = 1, \cdots, N \qquad (3.18)$$

While N is the number of training points and d is the dimension of feature vector x. The variables ξ_i *are data errors that are wrongly classified (either in the region of separation or on the wrong side of the classifier) by* $w^T x + b$ *and are called "Slack Variables". In the ideal case, the sum of slack variables is zero* $(\xi_i = 0)$.

Slack variables measure the deviation of a data point from the ideal condition of pattern separability. For $0 < \xi \leq 1$*, the data point falls inside the region of separation but on the correct side of the decision surface. This is called a* ***margin-violation****. For* $\xi > 1$*, it falls on the wrong side of the separating hyperplane and the point is* ***misclassified****.*

The parameter C controls the tradeoff between complexity of the machine and the number of non-separable points; it may therefore, be viewed as a form of a "regularization" parameter. It is a trade-off parameter between the error and the margin. Small C allows the constraints to be easily ignored, which means the margin is sufficiently large. In contrast, Large C makes constraints hard to ignore. It takes place while there is a narrow margin. Condition $C = \infty$ *enforces all constraints that implies a hard margin.*

The new problem is still a quadratic optimization problem and there is a unique minimum for it. Note, there is only one parameter, C.

The dual problem is finding α_i *that maximizes:*

$$\max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{3.19}$$

$$Subject\ to \quad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0\ , \qquad for\ i = 1, \cdots, N \tag{3.20}$$

This is very similar to the optimization problem in the linearly separable case, except that there is an upper bound C on α_i *now. Once again, a QP solver can be used to find* α_i*.*

Dual form only involves $\left(x_i^T\ x_j\right)$*. It Needs to learn d parameters for primal, and N for dual. If* $N \ll \mathrm{d}$ *, then it is more efficient to solve the problem for* α *rather than w.*

3.4.2 Nonlinear SVM Classifier

There are some cases that we cannot use a linear classifier to separate all sets of data. As described before, by introducing slack variables and solving the new optimization problem, we can handle classification over data with the following configurations:

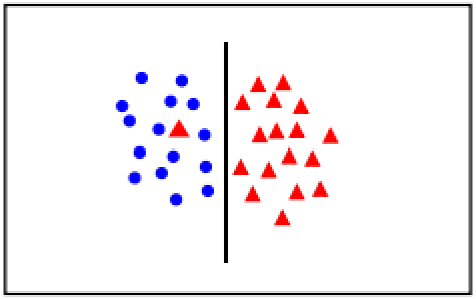


Figure 3.11: Linear classifier for non-separable sets, with accepting classification errors

The issue is more crucial when it is not appropriate to apply a linear classifier.

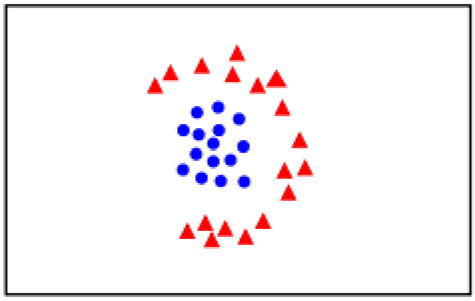


Figure 3.12: Non-separable sets where no linear classifier can be utilized

There are two solutions in this regard. The first one is mapping nonlinear data into "Polar Coordinates".

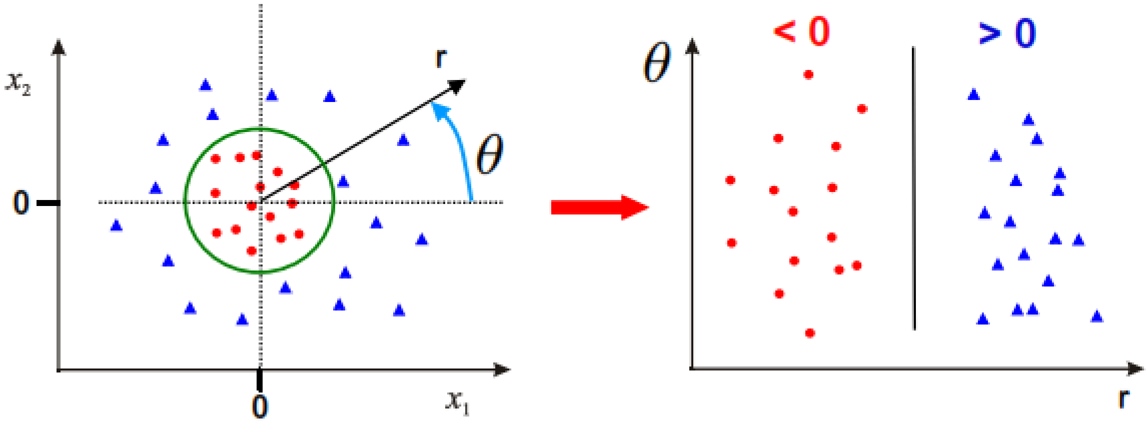


Figure 3.13: Data is linearly separable in polar coordinates

Although, data is acting nonlinearly in the original space, but it has a linear behaviour in the new space.

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} r \\ \theta \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

The second solution is mapping data into a higher dimension. Here is an example of 2D data that is mapped into a 3D space.

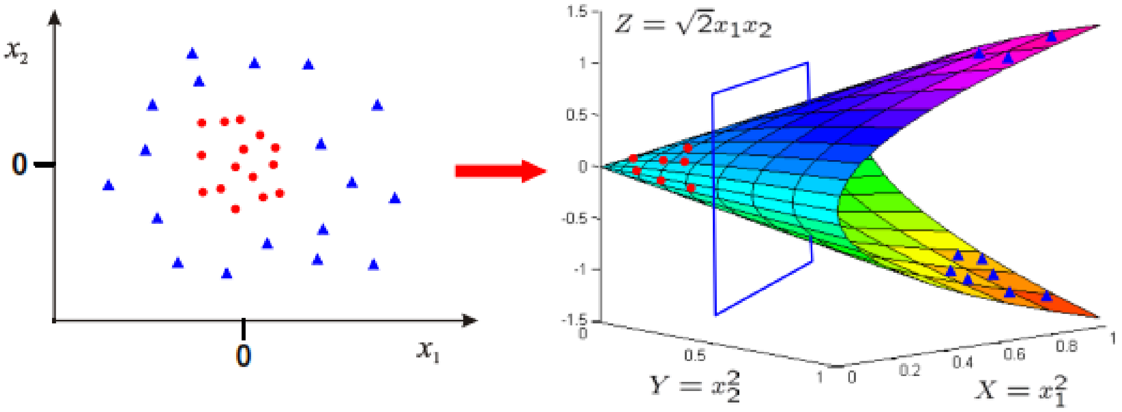


Figure 3.14: Data is linearly separable in 3D

As a result, by applying a linear classifier, the problem can be solved. In general, by applying a feature map in a higher dimension, a linear classifier can be obtained as shown in figure 3.15.

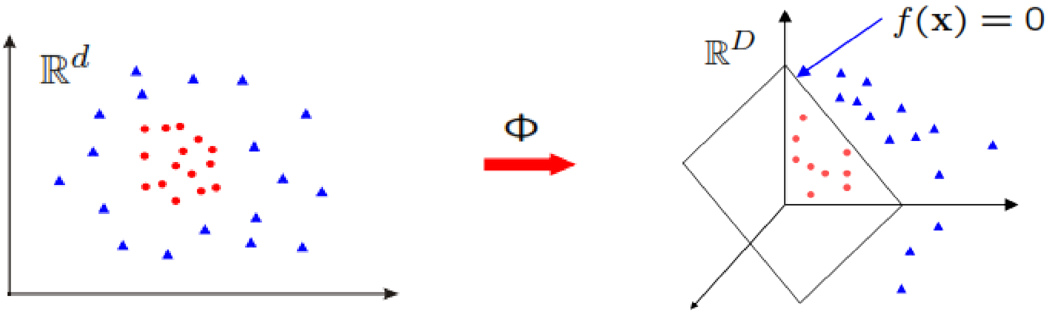


Figure 3.15: Learn a linear classifier in a higher Dimension, where D>d

$$\Phi : x \rightarrow \phi(x) \qquad \mathbb{R}^d \rightarrow \mathbb{R}^D$$

The new objective function becomes $f(x) = w^T\phi(x) + b$, *where linear classifier w has to be computed in* $\mathbb{R}^D$. $\phi(x)$ *is called a feature map.*

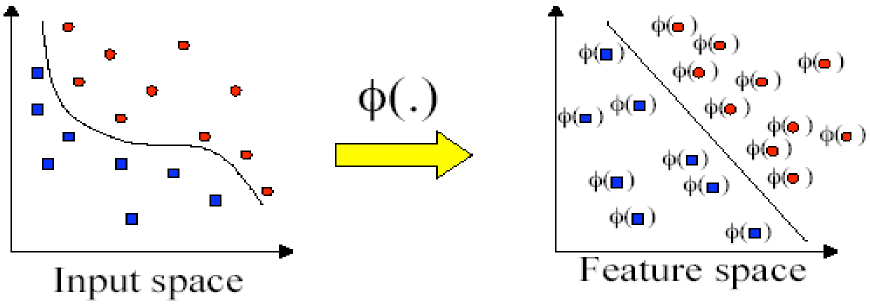


Figure 3.16: Transforming the data from the Input space to a Feature space by applying a Feature map $\phi(.)$

Basically, the idea of a support vector machine hinges on two mathematical operations:

1. *Nonlinear mapping of an input vector into a high-dimensional feature space that is hidden from both the input and output.*
2. *Construction of an optimal hyperplane for separating the features discovered in step 1.*

The first operation is performed in accordance with Cover's theorem on the separability of patterns. Consider an input space made up of nonlinearly separable patterns. Cover's theorem states that such a multidimensional space may be transformed into a new feature space where the patterns are linearly separable with high probability, while two conditions are satisfied. First, the transformation is nonlinear. Second, the dimensionality of the feature space is high enough. These two conditions are embodied in operation 1. Note, however, Cover's theorem does not discuss the optimality of the separating hyperplane. It is only by using an optimal separating hyperplane that the VC dimension is minimized and generalization is achieved.

This latter matter is where the second operation comes in. Specifically, operation two exploits the idea of building an optimal separating hyperplane in accordance with the theory described in section 3.4.1, but with a fundamental difference. The separating hyperplane is now *defined as a linear function of vectors drawn from the feature space rather than the original input space. Most importantly, construction of this hyperplane is performed in accordance with the principle of structural risk minimization that is rooted in VC dimension theory. The construction hinges on the evaluation of an inner product kernel.*

3.4.2.1 *Primal classifier in transformed feature space*

Here is a review of how to apply a primal classifier in transformed feature space.

$Classifier\ with\ w \epsilon \mathbb{R}^D$:

$$f(x) = w^T \phi(x) + b \qquad (3.21)$$

$Learning\ for\ w \epsilon \mathbb{R}^D$:

$$\min_{w \epsilon \mathbb{R}^D} \|w\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f(x_i)) \qquad (3.22)$$

The procedure is mapping x to $\phi(x)$ simply, where data is separable. Then, Solving the QP solver in high dimensional space $\mathbb{R}^D$. Note that if $D \gg d$, then there are many more parameters to learn for w. To avoid this, it is more convenient to solve the dual problem, as stated in section 3.3.1 about the outcomes of the equation (3.11).

3.4.2.2 *Dual classifier in transformed feature space*

$Classifier$:

$$f(x) = \sum_{i}^{N} \alpha_i y_i x_i^T x + b$$

$$\rightarrow f(x) = \sum_{i}^{N} \alpha_i y_i \phi(x_i)^T \phi(x) + b \qquad (3.23)$$

Learning:

$$\max_{\alpha_i \geq 0} \sum_{i}^{N} \alpha_i - \frac{1}{2} \sum_{jk}^{N} \alpha_j \alpha_k y_j y_k x_j^T x_k$$

$$\rightarrow \max_{\alpha_i \geq 0} \sum_{i} \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k \phi(x_j)^T \phi(x_k) \tag{3.24}$$

Subject to: $$0 \leq \alpha_i \leq C \quad for\ \forall i\,, \qquad \sum_{i} \alpha_i y_i = 0 \tag{3.25}$$

3.4.2.3 Kernel Trick

Note that $\phi(x)$ *only occurs in pairs* $\phi(x_j)^T \phi(x_k)$. *Once the scalar products are computed, only the N-dimensional vector α needs to be learnt; it is not necessary to learn in the D-dimensional space, as it is for the primal.*

Introducing the kernel function $\mathrm{k}(x_j, x_i) = \phi(x_j)^T \phi(x_i)$ *and replacing it in the dual problem, we update the formulation as follows*[1].

Classifier:

$$f(x) = \sum_{i}^{N} \alpha_i y_i k(x_i, x) + b \tag{3.26}$$

Learning:

$$\max_{\alpha_i \geq 0} \sum_{i} \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k k(x_j, x_k) \tag{3.27}$$

Subject to: $$0 \leq \alpha_i \leq C \quad for\ \forall i\,, \qquad \sum_{i} \alpha_i y_i = 0 \tag{3.28}$$

[1] Note that it is even not essential to know $\phi(x)$, but just $k(x_i, x)$, leaving the possibility to mapping into infinite dimensional spaces by some special transformations that will be introduced.

There are some special transformations, including Linear, Polynomial and Gaussian kernels. Here is an instance of a linear kernel.

$$\phi : \binom{x_1}{x_2} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \qquad\qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\mathrm{k}(\mathrm{x}, z) = \phi(\mathrm{x})^T\,\phi(\mathrm{z}) = \left(x_1^2\,,\,x_2^2\,,\sqrt{2}x_1x_2\right)\begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix}$$

$$= x_1^2z_1^2 + \,x_2^2\,z_2^2 + 2x_1x_2z_1z_2 = \,(x_1z_1 + x_2z_2)^2 = \,(x^Tz)^2$$

As can be seen in the above example, the classifier can be learnt and implemented without explicitly computing the function $\phi(x)$. To be more accurate, it can be learnt for high dimensional feature spaces, without actually having to map the points into the high dimensional space. All we required is the kernel function itself that is $\mathrm{k}(\mathrm{x}, z) = (x^Tz)^2$. To be noted, the complexity of the learning process depends on N (Typically it is $\mathrm{O}(N^3)$), but not on D.

These are some famous, commonly-used kernel functions.

- *Linear Kernels:*

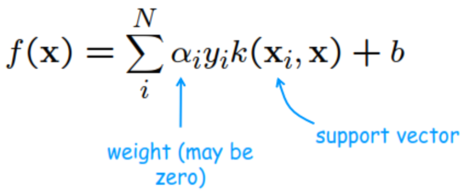
$$\mathrm{k}(x, x') = x^Tx' \tag{3.29}$$

- *Polynomial Kernels:*

$$\mathrm{k}(x, x') = (1 + x^Tx')^d \qquad for\ any\ d > 0 \tag{3.30}$$

- *Gaussian Kernels and RBF:*

Considering SVM classifier with a Gaussian kernel $\mathrm{k}(x, x')$:



$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

$$\mathrm{k}(x, x^{'}) = \mathrm{e}^{-\frac{\left\|x - x^{'}\right\|^2}{2\sigma^2}} \qquad for\ \ \sigma > 0 \tag{3.31}$$

We obtain a Radial Basis Function (RBF) as one of the most commonly-used classifiers.

$$f(x) = \sum_{i}^{N} \alpha_i y_i \mathrm{e}^{-\frac{\|x-x_i\|^2}{2\sigma^2}} + b \tag{3.32}$$

In Practice, a low degree polynomial kernel or RBF kernel with a reasonable width is a good initial try. Note that SVM with RBF kernel is closely related to RBF neural networks, with the centers of the radial basis functions automatically chosen for SVM.

3.5 Summary

Support Vector Machines are very powerful classification algorithms. When used in conjunction with random forest and other machine learning tools, they give a very different dimension to ensemble models. Hence, they become very crucial for cases where very high predictive power is required. Such algorithms are slightly harder to visualize because of the complexity of the formulation.

SVMs were originally proposed by Boser, Guyon and Vapnik in 1992 and gained increasing popularity in the late 1990s. SVMs are currently among the best performers for a number of classification tasks ranging from text to genomic data. They can be applied to complex data types beyond feature vectors (e.g. graphs, sequences, and relational data) by designing kernel functions for such data. However, tuning SVMs remains a black art. Selecting a specific kernel and parameters is usually done in a try-and-see manner.

3.5.1 SVM Features

- *SVM finds an optimal linear separator*
    - ✓ *They pick the hyperplane that maximises the margin*
    - ✓ *The optimal hyperplane turns out to be a linear combination of support vectors*

- *The kernel trick makes SVM as a non-linear learning algorithm*
  - ✓ *Transform nonlinear problems to a higher dimensional space, using kernel functions. Then, there is more chance that classes will be linearly separable in the transformed space.*

3.5.2 SVM Steps

- *Prepare the data matrix*
- *Select the kernel function to use*
- *Execute the training algorithm using a QP solver to obtain the* α_i *values*
- *Unseen data can be classified using the* α_i *values and the support vectors*

3.5.3 Pros and Cons

- *Strengths*
  - ✓ *Training is relatively easy*
  - ✓ *Good generalization in theory and practice*
  - ✓ *Work well with few training instances*
  - ✓ *Find globally best model, not local optimum, unlike in neural networks*
  - ✓ *It scales relatively well to high dimensional data*
  - ✓ *Tradeoff between classifier complexity and error can be controlled explicitly*
  - ✓ *Non-traditional data like strings and trees can be used as input to SVM, instead of feature vectors*
- *Weaknesses*
  - ✓ *Need to choose a “good” kernel function.*

References

[1] *N. Laouti, N. Sheibat-Othman, and S. Othman, “Support Vector Machines for Fault Detection in Wind Turbines” in Proc. 18th IFAC world congress, Volume 44, Issue 1, Pages 7067-7072, January 2011.*

[2] *S. Haykin; “Neural Networks. A Comprehensive Foundation”; Pearson Prentice Hall.*

[3] *V.N. Vapnik, “The Nature of Statistical Learning Theory”, Springer*

[4] *https://www.analyticsvidhya.com/blog/2014/10/support-vector-machine-simplified/*

Chapter 4:
SVM Application to Wind Turbines

In the previous parts, all the prerequisites to design a fault detector for a variable-speed wind turbine are presented. In chapter 1.4, a model of a wind turbine based on the dynamics of its subsystems, as well as the wind input with its uncertain characteristics are introduced.

As described in section 3.1, fault detection by SVM as a classification method is developed in two parts. First of all, a set of measurement data with and without fault is used to learn models for detection of each fault (using the given wind sequence as an input). The obtained models are then validated in a new fault scenario. Some predefined fault scenarios that are imposed on the system in different control zones have been introduced in section 2.4.

In this chapter, the results and the performance of the SVM under the so-called faulty conditions have been discussed. As mentioned in the introduction part, SVM design in [2] and a preliminary modular model in [3] is considered as the reference of this work. The wind input, parameters of the turbine model and controller and all the SVM characteristics can be customized, according to the user's need.

4.1 Applied SVM for Fault Detection in Wind Turbine

In the problem in hand, the kernel function $\mathrm{k}(x_i, \mathrm{x})$ *that maps the data into a different space, where a hyperplane can be used to do the separation, is an RBF with variance σ (equation 3.31) and the decision boundary is as follows:*

$$f(x) = sgn\left(\sum_{i=1}^{N} \alpha_i y_i k(x_i, y) + b\right) \tag{4.1}$$

$$w = \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \tag{4.2}$$

Where $\alpha_i \geq 0$ *are the Lagrange multipliers and* b *is the bias. Finally, the residual can be written in the form of the equation (4.3).*

$$f(x) = \sum_{i}^{N} \alpha_i y_i \mathrm{e}^{-\frac{\|x_i - y\|^2}{2\sigma^2}} + b \tag{4.3}$$

4.1.1 Phase 1 - SVM Learning

For any classification task, the first phase is the so-called learning stage. The key step in learning a new model for fault detection by SVM is defining a vector to be used for classification. This vector should contain the most related information on the behaviour of the system. It means that not only the measurement outputs but also the inputs, set points, a combination of those or variation of the outputs with time should be considered. In order to build a useful vector x, the process outputs for each fault should carefully be observed to propose a combination of data that provides a sufficiently high impact of the considered fault. [2]

To detect different kinds of faults in different locations (including sensors, actuators and systems), various vectors have to be proposed. Considering the redundancy of the sensors, their difference could potentially indicate a faulty case, as well as a zero-difference between successive samples of the same sensor that is an indication of a sensor malfunction.

As a result, the following vector was deployed [2] for the six sensors measuring the pitch positions ($\beta_{k,mi}$ *; k=1, 2, 3, i=1, 2), that handles the fault types 1a and 1b in table 2.1:*

$$x = \begin{bmatrix} \left| \beta_{k,m1}(t_j) - \beta_{k,m2}(t_j) \right| \\ \left|\beta_{k,m1}(t_j) - \beta_{k,m1}(t_{j-1})\right| \\ \left|\beta_{k,m2}(t_j) - \beta_{k,m2}(t_{j-1})\right| \end{bmatrix} \tag{4.4}$$

Where t_j *and* t_{j-1} *are the time instances at time j and j-1, respectively. The first line in x at time* t_j *detects differences between two redundant sensors of the same blade. The second and third arrays show the variation with time for each sensor at two successive instances. Note that absolute values are used for all cases in the vector x.*

In this strategy, when $\left|\beta_{k,mi}(t_j) - \beta_{k,mi}(t_{j-1})\right| = 0$*, this term is replaced by a large constant value (e.g. 5000) in order to intensify the difference between the fixed value fault and the healthy case. To be noted, these values in the healthy case oscillate in a range from* 1×10^{-2}*to 2. To reduce the sensitivity of pitch sensors to process disturbances or measurement noise, the measurements were filtered using a first order, low pass filter with the time constant* $\tau = 0.06s$.

As mentioned before, the applied Kernel for all of the faults was Gaussian. The variance corresponding to x in equation (4.4) is $\sigma = 10$.

Considering the above reasoning, the following vector x is introduced [2] for the learning phase of the sensor faults (fault types 2a, 2b, 3a and 3b) related to the generator and rotor speed measurements ($\omega_{P,mi} : P = g, r;\ i = 1{,}2$*):*

$$x = \begin{bmatrix} |\, \omega_{P,m1}(t_j) - \omega_{P,m2}(t_j) \,| \\ |\omega_{P,m1}(t_j) - \omega_{P,m1}(t_{j-1})| \\ |\omega_{P,m2}(t_j) - \omega_{P,m2}(t_{j-1})| \end{bmatrix} \tag{4.5}$$

The generator speed measurements ω_g *were filtered with* $\tau = 0.02s$ *, while the filter of the rotor speed* ω_r *has the time-constant* $\tau = 0.06s$ *before running the equation (4.5). The Gaussian variance is tuned at* $\sigma = 15$ *in order to increase the ability of detection. However, note that very high variance values might lead to false alarms.*

For faults 4 and 6, the following vector was proposed [2]:

$$x = \begin{bmatrix} |\, \omega_{g,m1}(t_j) - \omega_{g,m2}(t_j) \,| \\ |\tau_g^d(t_j) - \tau_g^m(t_j)| \\ \left|\lambda_2 \times \left|\omega_g^d(t_j) - \dfrac{\omega_{g,m1}(t_j) + \omega_{g,m2}(t_j)}{2}\right|\right| \end{bmatrix} \tag{4.6}$$

Where ω_g^d *is the desired generator speed, calculated from the desired generator torque* τ_g^d *obtained by the controller with equation (1.30), that is* $\frac{Desired\ Power}{Desired\ Torque}$*. The parameter* λ_2 *in the third component of x is used as a normalization term as well as taking the wind speed into account, as its value is proportional to the wind speed (*$\lambda_2 = 10^{-10} \times v_{wind}^6$*). The used values* $\omega_{g,mi}$ *are the filtered ones, as used in the vector (4.5). Note that* τ_g^d *was also filtered using a first order filter with a time constant* $\tau = 0.02s$*. The objective of this filter was to take into account the dynamic of the control system that is the required time for* τ_g^m *to attain* τ_g^d *according to the equation (1.21) as* *well as not rejecting the measurement noise or disturbances. The variance corresponding to x in (4.6) for the torque converter fault (type 4) is* $\sigma = 10$ *for and for the drive train dynamic change (type 6) is* $\sigma = 200$.

For the detection of actuator faults 5a and 5b, the following vector was used [2]. The corresponding variance of x in equation (4.7) is $\sigma = 10$*, while the filters are those implemented to the x vectors in (4.4) and (4.5).*

$$x = \begin{bmatrix} \left| \omega_{g,m1}(t_j) - \omega_{g,m2}(t_j) \right| \\ \left| \beta_{k,m1}(t_j) - \beta_{k,m2}(t_j) \right| \\ \left| \beta_{k,m1}(t_j) - \beta_{k,m1}(t_{j-1}) \right| \\ \left| \beta_{k,m2}(t_j) - \beta_{k,m2}(t_{j-1}) \right| \end{bmatrix} \tag{4.7}$$

Once the learning vectors are defined for each fault, different fault scenarios are then simulated and attributed the label $y = \{1,0\}$ *(with or without fault). The SVM learning algorithm uses the outputs (x) and the corresponding y values to identify a model as a function of the given data. Then, nine scenarios are considered for each fault type with different amplitudes that will be explained in the validation phase.*

To have an example, the identified parameters for the SVM related to the fault $\beta_{1,m1}$ *are the weight vector* $\omega_{11\times1}$ *and bias* b*, which get the following results:*

$$\omega = [-0.117 \quad -0.172 \quad -0.024 \quad -0.144 \quad -0.335 \quad 1.109 \quad -0.717 \quad 0.017 \quad -0.011 \quad 1.111 \quad -0.718]^T$$

$$b = -0.215$$

4.1.2 Phase 2 - SVM Validation

Once the classifier is learned, different fault scenarios with different amplitudes at diverse time instances have been tested to verify the performance of the classifier. The wind input sequence, figure 4.1 is the data extracted from a wind park according to Odgaard et al. [1].

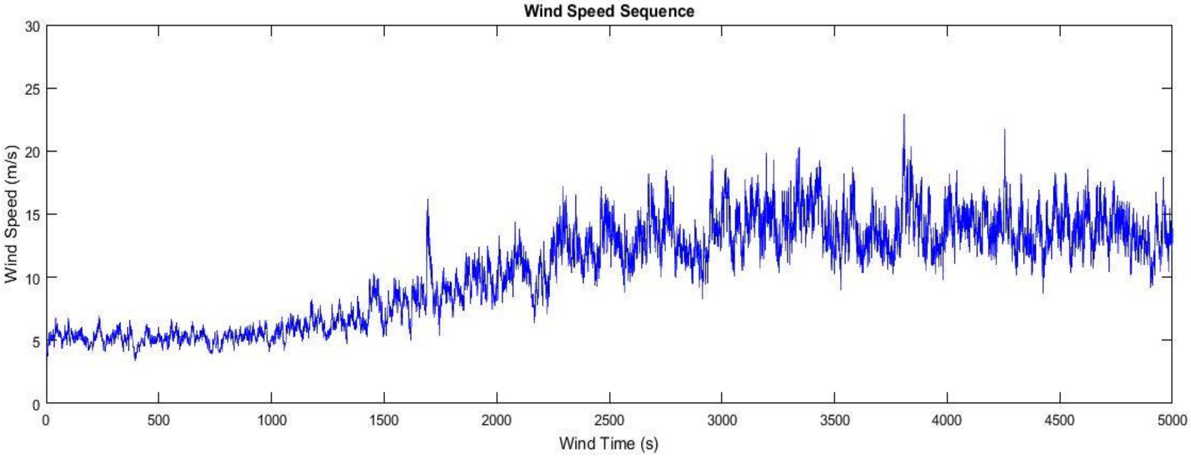


Figure 4.1: Test Wind Speed Sequence measured in a wind park

Refer to table 4.1, nine fault scenarios have been considered for the validation phase. It has been tried to cover various faulty situations in both control regions by modifying their magnitude and exposure time, to check the functionality of the SVM in response to each of them. In table 4.2, the time intervals, when any of the so-called faults are imposed to the wind turbine is listed. Note that, the identical models are used for all faults of the same type. Therefore, ten models were developed to cover all those, mentioned in table 2.1.

As can be seen, seven sensor faults, including four fixed-value and three gain-value (multiplicative type) at different locations have been considered. Such faults cover all sensor faults since the measurements are redundant and their models are identical. Therefore, the results can be generalized to all the other sensors with the same species. To be noted, all the measurements are exposed to Gaussian white noise, in order to be able to analyze the results in a more realistic manner.

There are three actuator faults, as described in section 2.4.2. An offset torque converter action (type 4) and two pitch system dynamic changes -abrupt (5a) and slow (5b) - have been taken into account.

Finally, according to section 2.4.3, a very slow drive-train system fault that appears as an indication of the system ageing has been introduced. It affects the rotor and generator speed directly.

No	**Location**	**Type**	**Involved Parameter**	**Sc. 1**	**Sc. 2**	**Sc. 3**	**Sc. 4**	**Sc. 5**	**Sc. 6**	**Sc. 7**	**Sc. 8**	**Sc. 9**
1a	*Sensor*	*Fixed Value*	$\beta_{1,m1}$	5	6	4	2	-5	1	24	-3	2
1a	*Sensor*	*Fixed Value*	$\beta_{3,m1}$	10	8	12	15	3	6	20	7	7
1b	*Sensor*	*Gain Factor*	$\beta_{2,m2}$	1.2	1.5	1.8	3	10	2	5	5	3
2a	*Sensor*	*Fixed Value*	$\omega_{r,m1}$	1.4	0.2	1.2	1.7	2.5	0.5	3.5	2	2
2b	*Sensor*	*Gain Factor*	$\omega_{r,m2}$	1.1	1.2	0.7	1.7	0.9	0.2	3	0.5	0.5
3a	*Sensor*	*Fixed Value*	$\omega_{g,m1}$	140	80	100	130	120	150	50	100	110
3b	*Sensor*	*Gain Factor*	$\omega_{g,m2}$	0.9	0.8	1.7	0.7	0.2	2.8	5	1.5	1.5
4	*Converter Actuator*	*Offset*	τ_g	100	100	700	-500	1500	2500	900	-1000	800
5a	*Pitch 2 Actuator*	*Changed Dynamic (Hydraulic ΔP)*	ξ_2	0.45	0.45	0.45	0.45	0.45	0.45	0.25	0.45	0.45
			ω_{n2}	5.73	5.73	5.73	5.73	5.73	5.73	8.73	5.73	5.73
5b	*Pitch 3 Actuator*	*Changed Dynamic (Air Content)*	ξ_3	0.9	0.9	0.9	0.9	0.9	0.9	0.95	0.9	0.9
			ω_{n3}	3.42	3.42	3.42	3.42	3.42	3.42	1.42	3.42	3.42
6	*Drive train System*	*Changed Dynamic*	η_{dt}	0.92	0.71	0.3	0.6	0.25	0.1	0.4	0.22	0.42

Table 4.1: Magnitude of the Faults Applied to the wind turbine in 9 different scenarios

Getting the sixth scenario as the default test set, the results are discussed in details in the following section.

No	***Type***	***Involved Parameter***	***Sc. 1***	***Sc. 2***	***Sc. 3***	***Sc. 4***	***Sc. 5***	***Sc. 6***	***Sc. 7***	***Sc. 8***	***Sc. 9***
1a	*Pulse*	$\beta_{1,m1}$	*[2000-2100]*	*[1000-1100]*	*[100-200]*	*[250-350]*	*[3000-3100]*	*[1500-1600]*	*[4200-4300]*	*[100-200]*	*[300-400]*
1a	*Pulse*	$\beta_{3,m1}$	*[2600-2700]*	*[1600-1700]*	*[900-1000]*	*[1000-1100]*	*[500-600]*	*[4000-4100]*	*[3700-3800]*	*[900-1000]*	*[700-800]*
1b	*Pulse*	$\beta_{2,m2}$	*[2300-2400]*	*[3900-4000]*	*[2500-2600]*	*[3200-3300]*	*[1300-1400]*	*[2600-2700]*	*[4000-4100]*	*[500-600]*	*[100-170]* *[2500-2570]*
2a	*Pulse*	$\omega_{r,m1}$	*[1500-1600]*	*[500-600]*	*[1200-1300]*	*[800-900]*	*[200-300]*	*[2900-3000]*	*[3500-3600]*	*[1200-1300-]*	*[1000-1100]*
2b	*Pulse*	$\omega_{r,m2}$	*[1000-1100]*	*[4000-4100]*	*[1700-1800]*	*[2200-2300]*	*[2000-2100]*	*[600-700]*	*[3000-3100]*	*[1700-1800]*	*[500-570]* *[2900-2970]*
3a	*Pulse*	$\omega_{g,m1}$	*[3600-3700]*	*[2600-2700]*	*[3900-4000]*	*[4200-4300]*	*[3300-3400]*	*[3500-3600]*	*[1000-1100]*	*[2200-2300]*	*[2200-2300]*
3b	*Pulse*	$\omega_{g,m2}$	*[1000-1100]*	*[4000-4100]*	*[1700-1800]*	*[2200-2300]*	*[2000-2100]*	*[600-700]*	*[3000-3100]*	*[1700-1800]*	*[500-570]* *[2900-2970]*
4	*Pulse*	τ_g	*[3800-3900]*	*[1800-1900]*	*[4200-4300]*	*[1200-1300]*	*[2400-2500]*	*[100-200]*	*[2100-2200]*	*[4200-4300]*	*[4200-4300]*
5a	*Pulse* *(Hydraulic* ΔP*)*	ξ_2 , ω_{n2}	*[2900-3000]*	*[2900-3000]*	*[3200-3300]*	*[2800-2900]*	*[3600-3700]*	*[2400-2500]*	*[2700-2800]*	*[2900-3000]*	*[1900-2000]*
5b	*Trapezoidal* *Pulse* *(Air Content)*	ξ_3 , ω_{n3}	*[3400-3500]*	*[3400-3500]*	*[3400-3500]*	*[3400-3500]*	*[3400-3500]*	*[3200-3300]*	*[2500-2600]*	*[3400-3500]*	*[1400-1500]*
6	*Pulse*	η_{dt}	*[4000-4200]*	*[3000-3200]*	*[2000-2000]*	*[3900-4100]*	*[2700-2900]*	*[2000-2200]*	*[1400-1600]*	*[300-500]*	*[3300-3500]*

Table 4.2: Fault Exposure Time-intervals to the wind turbine with 9 different scenarios

4.2 Simulation Results

As discussed in chapter 1.4.5, the generated power P_g and the generator speed ω_g are the outputs of a wind turbine, while the reference pitch angles $\beta_{i,ref}$ and generator torque $\tau_{g,r}$ are the controllable inputs. In this chapter, all the characteristics of the wind turbine benchmark model under the faulty situations have been studied. [2], [3]

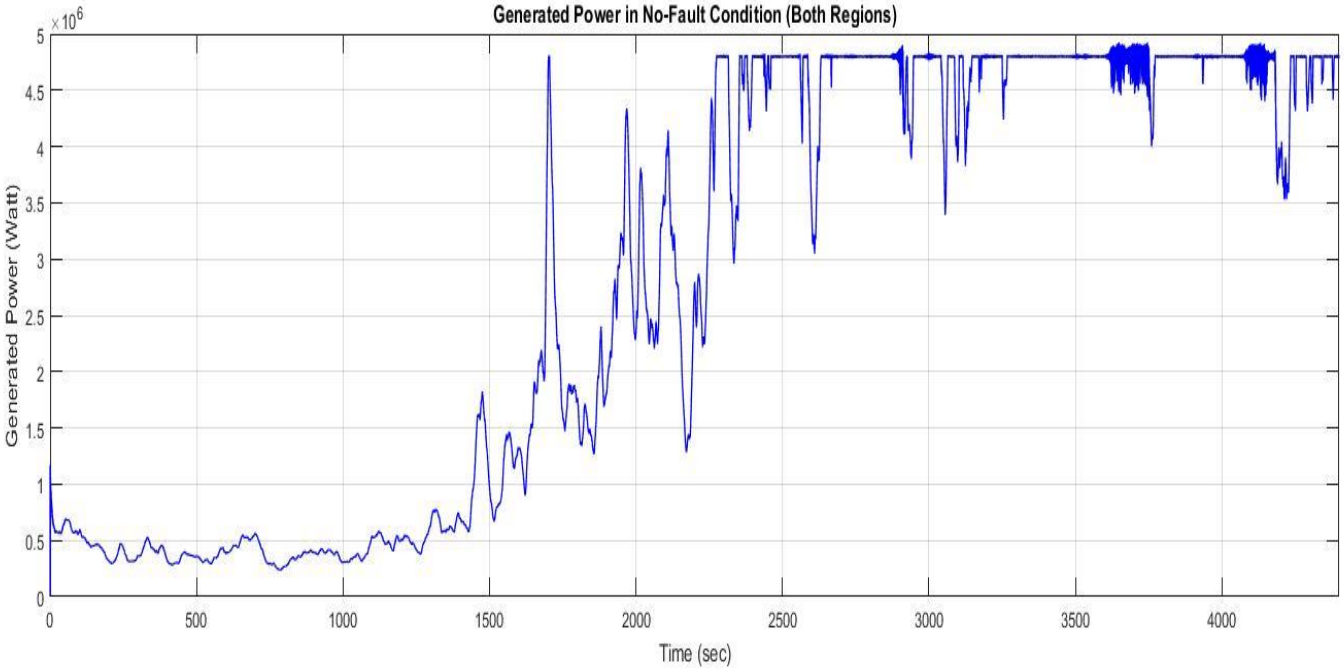


Figure 4.2: Generated Power in optimization and power reference following ($P_r = 4.8 \times 10^6$) regions in no- fault condition

As a reminder, figure 4.2 indicates the output power. It can be seen that in the left-side the power optimization is achieved by setting the pitch angle at zero (figure 1.36). As soon as the reference power and generator speed are attained, the controller switches to the second mode (equation 1.28), where the converter torque is set to equation (1.30). It is shown in figure 1.37.

4.2.1 Pitch sensor Faults

Our benchmark wind turbine has three blades and pitch systems. Each blade has two pitch sensors; each of which might be potentially involved with two types of faults. As a result, twelve fault scenarios might happen related to the pitch sensors.

4.2.1.1 Fixed Value

Based on the results of all the fault scenarios, the deployed learning vector x provides enough information for the SVM to simply detect fixed value faults (type 1b) for the pitch sensors in both control regions.

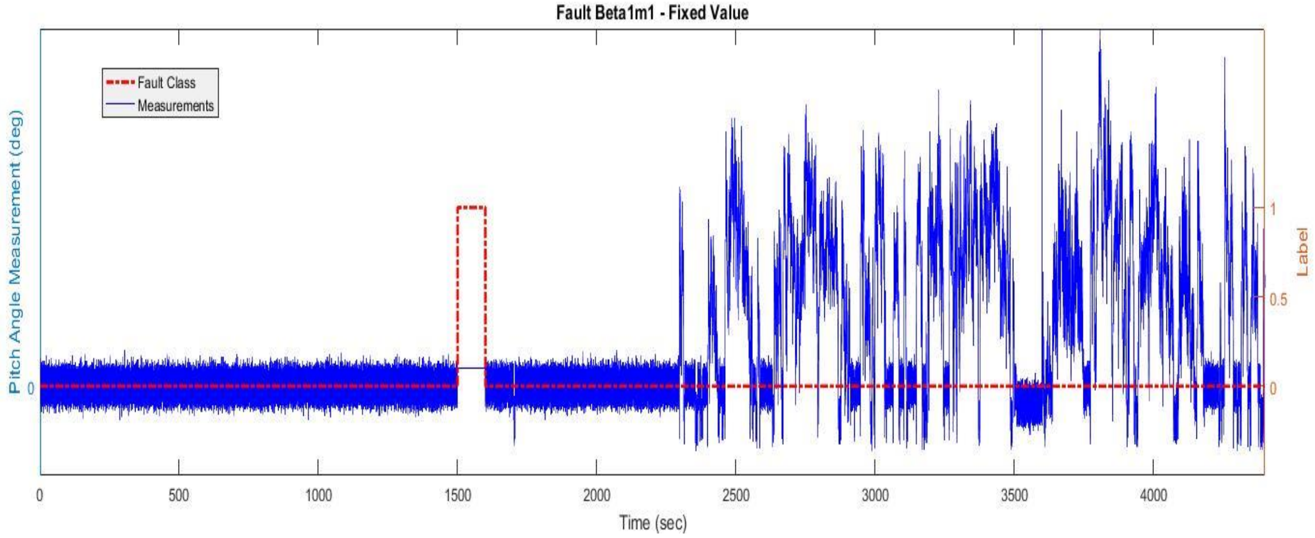


Figure 4.3: Fixed Value of the pitch sensor $\beta_{1,m1}$ at time interval [1500-1600] - (fault 1a)

As can be seen in figure 4.3, SVM classifier detects a malfunction in $\beta_{1,m1}$ during the time interval [1500-1600], where the blade pitch has to be set to zero (although having noisy measurements), but it deviates from the healthy measurements. Figure 4.4, shows how quick was the detection. To be exact, the fault is identified just after two samples i.e. 0.02 second and meets the FDI requirements. The same results are obtained in other fault scenarios.

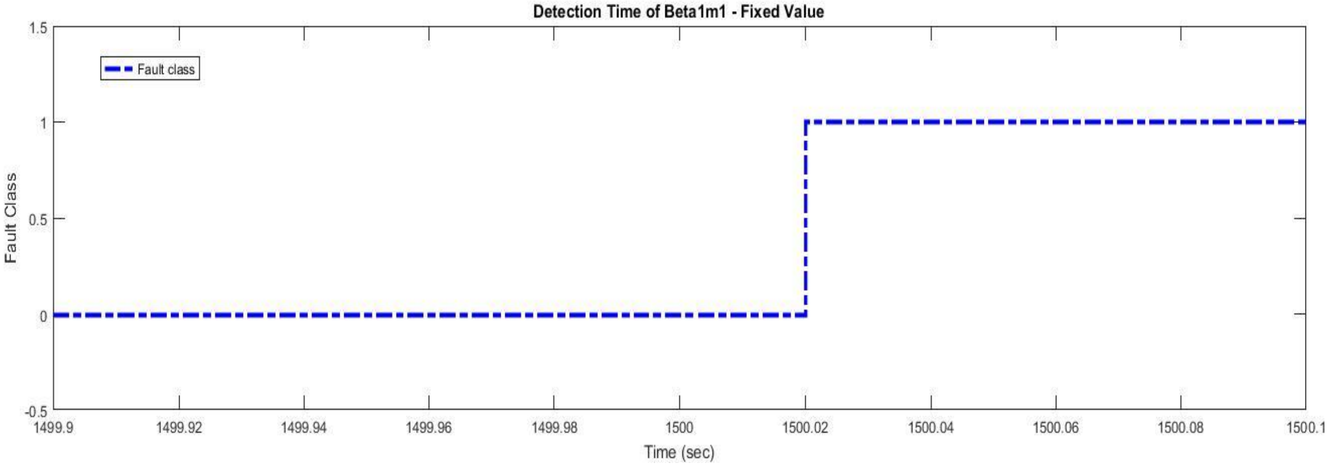


Figure 4.4: Detection Time for the fixed value fault of $\beta_{1,m1}$

Figure 4.5 shows another fixed value type that is imposed to the third pitch sensor in the time interval [4000-4100], where the controller is working in the power reference following zone. Same as above, it is well-detected after two samples and the FDI requirements are satisfied

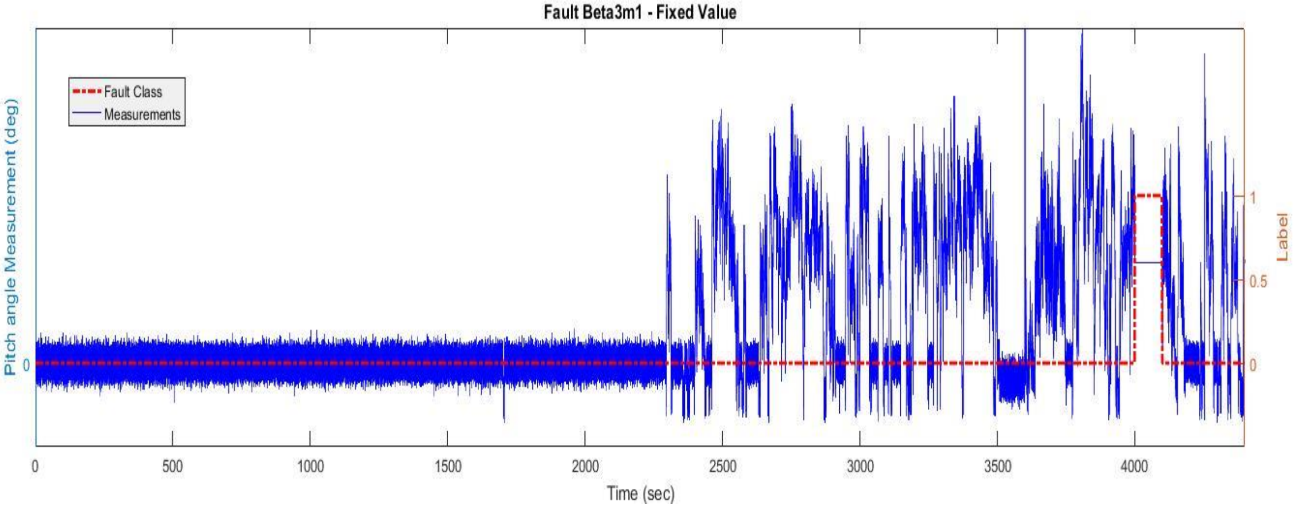


Figure 4.5: Fixed value fault of the third blade sensor $\beta_{3,m1}$at [4000-4100]

4.2.1.2 Gain Factor

In order to check the reaction of the SVM to a multiplicative fault, a gain factor k=2 is imposed to the redundant sensor of the second blade measurements during the time interval [4000-4100].

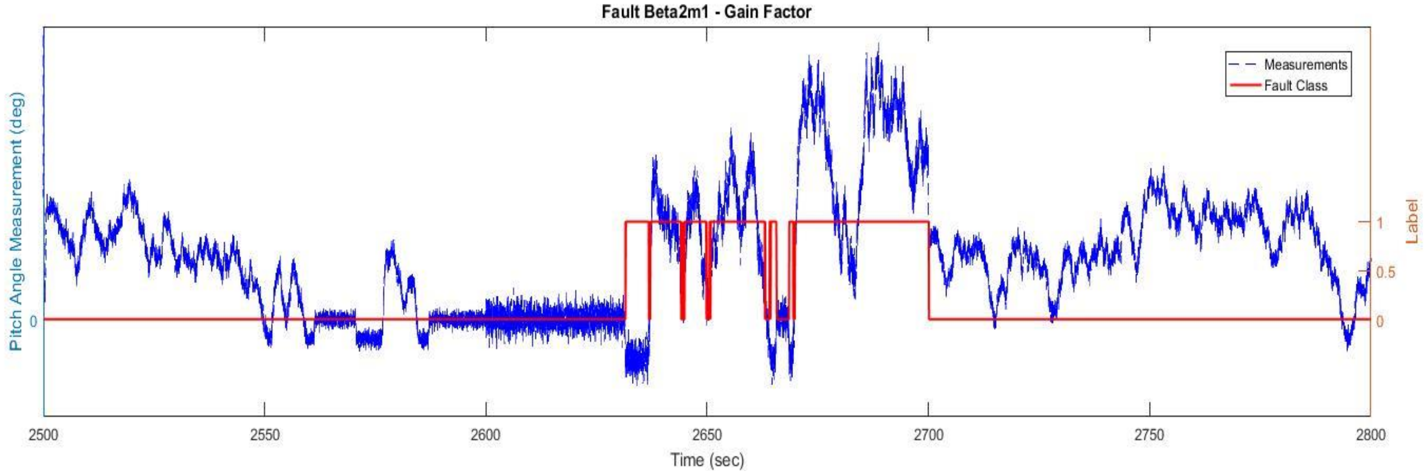


Figure 4.6: Gain factor fault of the second blade sensor $\beta_{2,m2}$ at [2600-2700] with gain k=2

Figure 4.6 indicates that a multiplicative fault (type 1b) can be detected with some oscillations that are proportional to the gain value. The appearance of such oscillations is due to the slower dynamic change with respect to the fixed-value case, where the dynamic changes suddenly. If the gain is reduced, the detection time increases and vice versa. For a gain factor of 2, the detection time is 31.5 seconds shown in figure 4.7 that fails to satisfy the FDI requirements according to Odgaard et al [1].

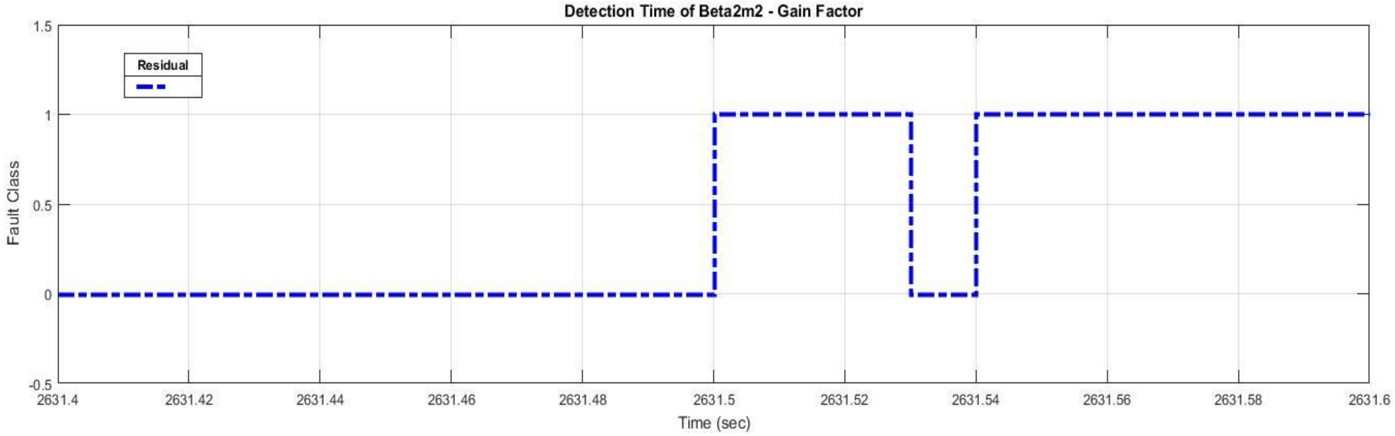


Figure 4.7: Detection Time for the Multiplicative fault $\beta_{2,m2}$ with gain k=2

For a multiplicative fault with gain 5, the detection time shows a significant decrease from 31.5 to 0.33 seconds. However, the oscillation nature of the residual still holds and it becomes even worse.

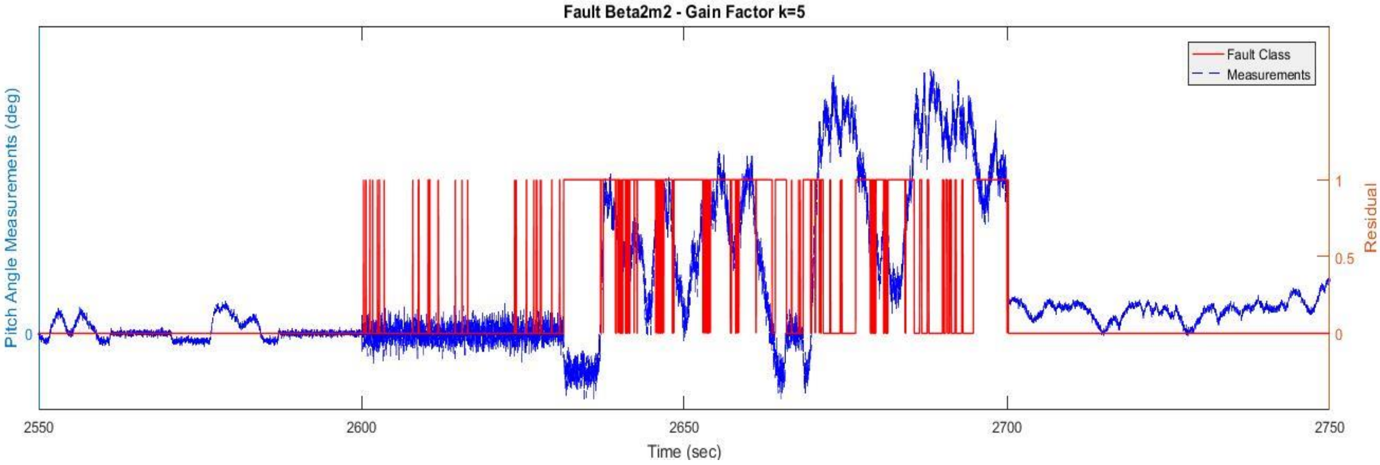


Figure 4.8: Gain factor fault of the second blade sensor $\beta_{2,m2}$ at [2600-2700] with gain k=2

Hence, there is an inverse trend between the detection time of a multiplicative fault and its magnitude. A rise in the gain is followed by faster detection, but with more oscillations.

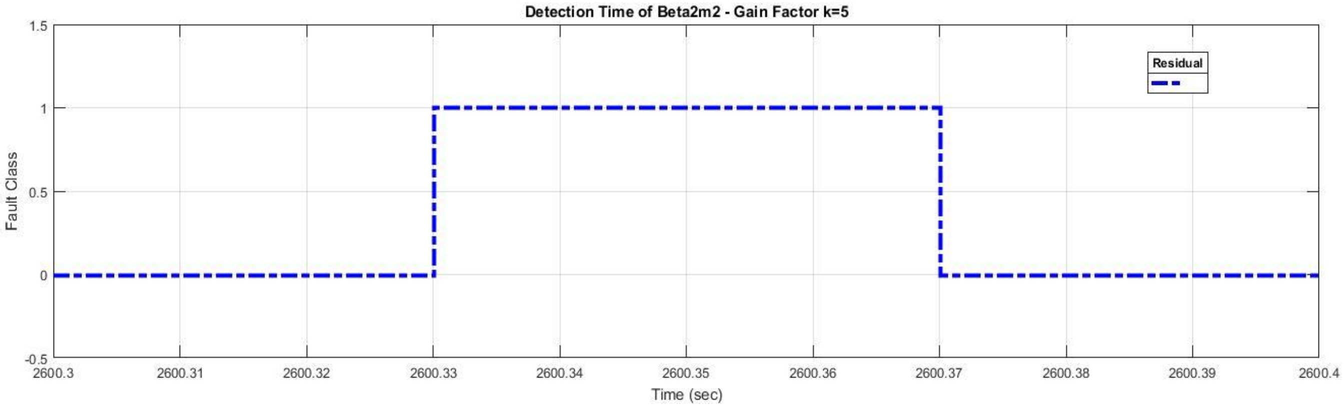


Figure 4.9: Detection Time for the Multiplicative fault $\beta_{2,m2}$ with gain k=5

4.2.2 Generator and Rotor Speed Sensor Faults

An almost similar scenario as above occurs for the generator and rotor speed sensors, which their dynamics and nature are fully described in section 2.4.3. Considering physical redundancy of the sensors, this chapter covers four sensors, being exposed to two types of faults. Consequently, eight potential malfunctions are discussed.

4.2.2.1 Fixed Value

Figure 4.10 indicates the occurrence of a fixed value (fault 2a) in rotor speed sensor $\omega_{r,m1}$ *during the time interval [2900-3000]. Like pitch sensor faults, it can be detected immediately, after two sample periods that is 0.02 seconds (figure 5.11).*

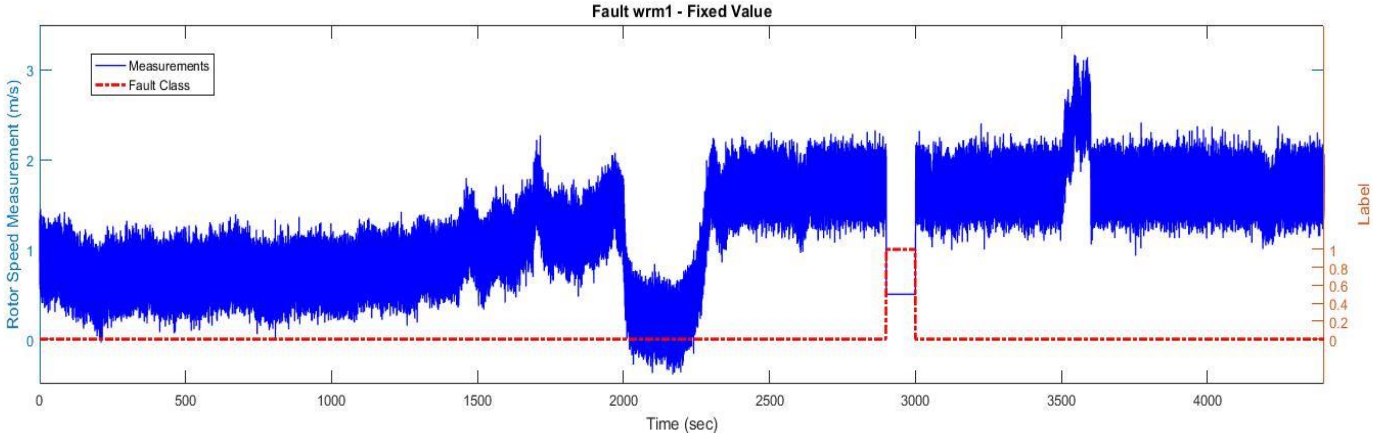


Figure 4.10: Fixed Value of the rotor speed $\omega_{r,m1}$ *at time interval [2900-3000] - (fault 2a)*

The same results are obtained upon varying the magnitude and exposure time of the so-called fault.

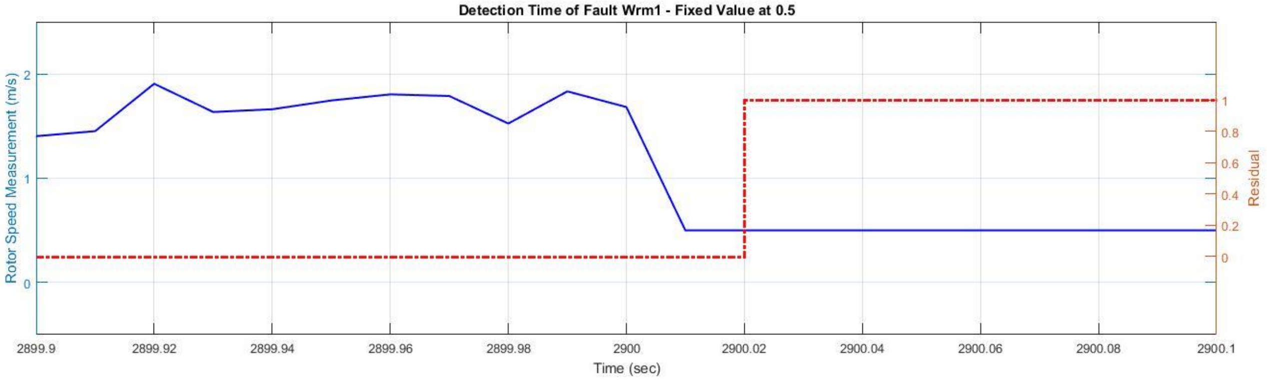


Figure 4.11: Detection Time for the Multiplicative fault $\omega_{r,m1} = 0.5$

As can be seen in figure 4.12, the generator speed is stuck into the value $\omega_{\mathrm{g},m1} = 150$ *(fault 3a) within the time interval [3500-3600] and the FDI system identified it within two sample periods (figure 4.13). To be noted, the requirements of all the other scenarios were satisfied in response to such kind of fault.*

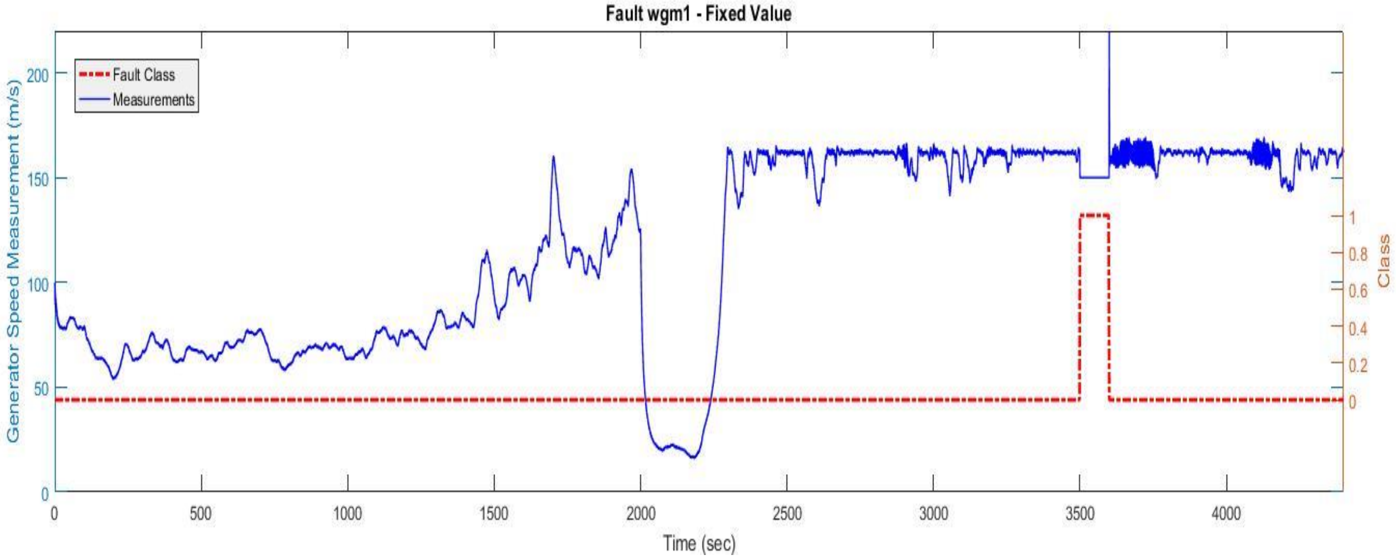


Figure 4.12: Fixed Value of the Generator speed $\omega_{g,m1}$ *at time interval [3500-3600] - (fault 3a)*

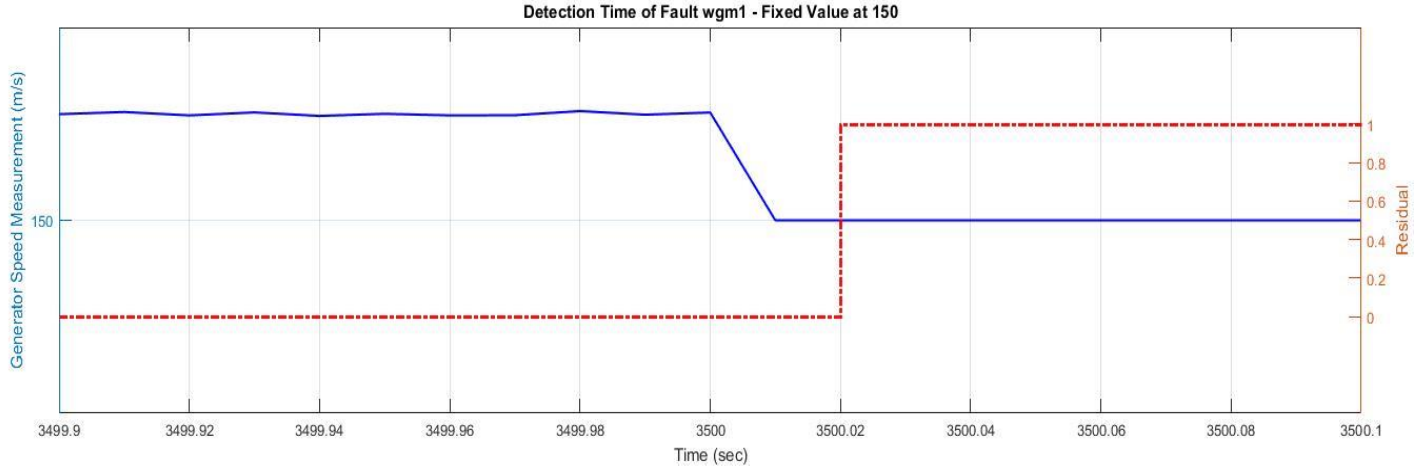


Figure 4.14: Detection Time for the Multiplicative fault $\omega_{g,m2} = 150$

4.2.2.2 Gain Factor

Unlike section 4.3.1.2, speed measurement sensors reacted properly to multiplicative faults in most cases, except where the gain is so close to one (Scenario 5).

The difference between a fixed value type and a multiplicative one is that the latter is gradual, while the former takes place instantly. As a result, the detection time of types (2b) and (3b) are slower. It slows down with a fall in the gain factor. As an example, T_D for the gain k=0.2 is 0.12 seconds, or equally 12 sample delays (figure 4.16), while for k=3, it is identified after 5 samples (figure 4.17).

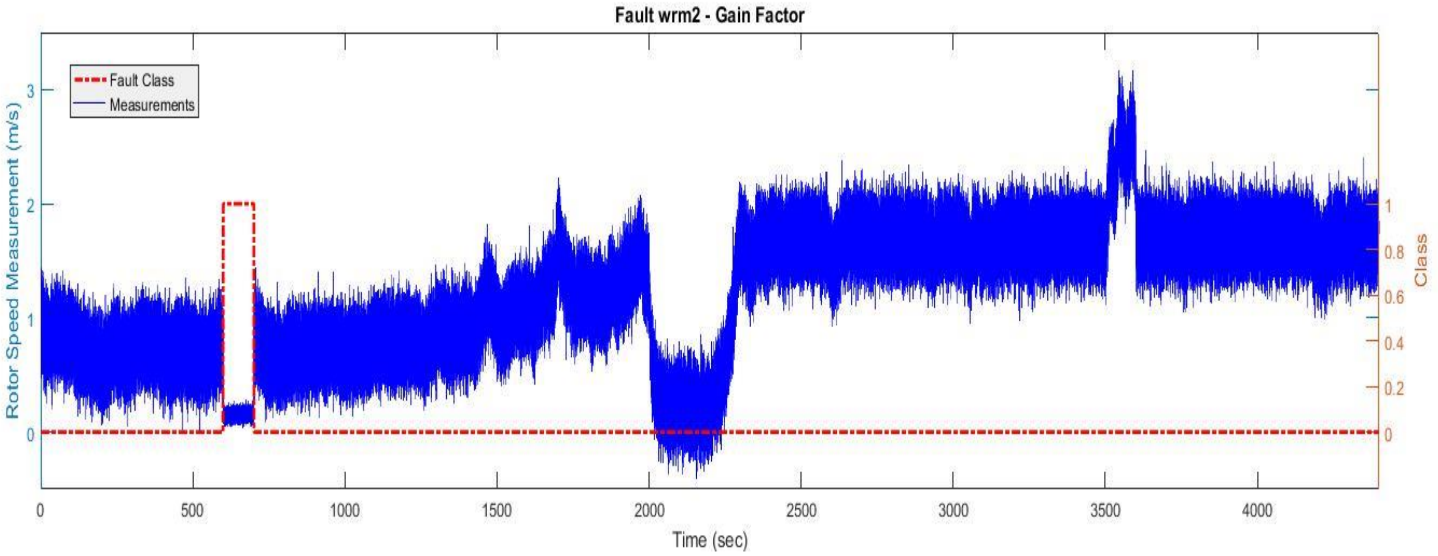


Figure 4.15: Gain factor fault of the rotor speed $\omega_{r,m2}$ at time interval [600-700] - (fault 2b) with gain k=0.2

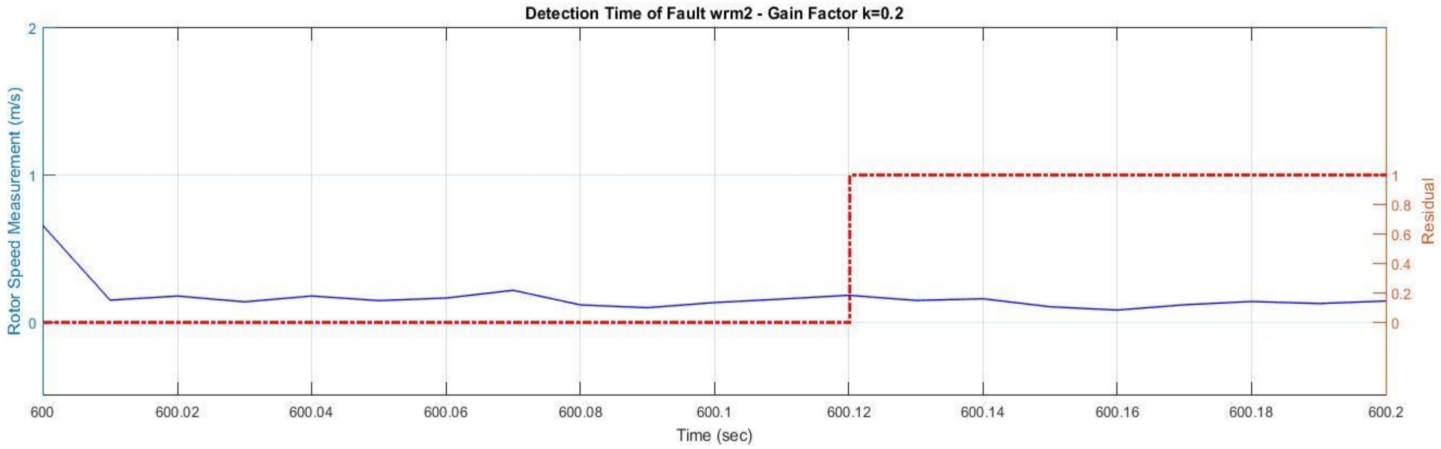


Figure 4.16: Detection Time for the Multiplicative fault $\omega_{r,m2}$ with gain k=0.2

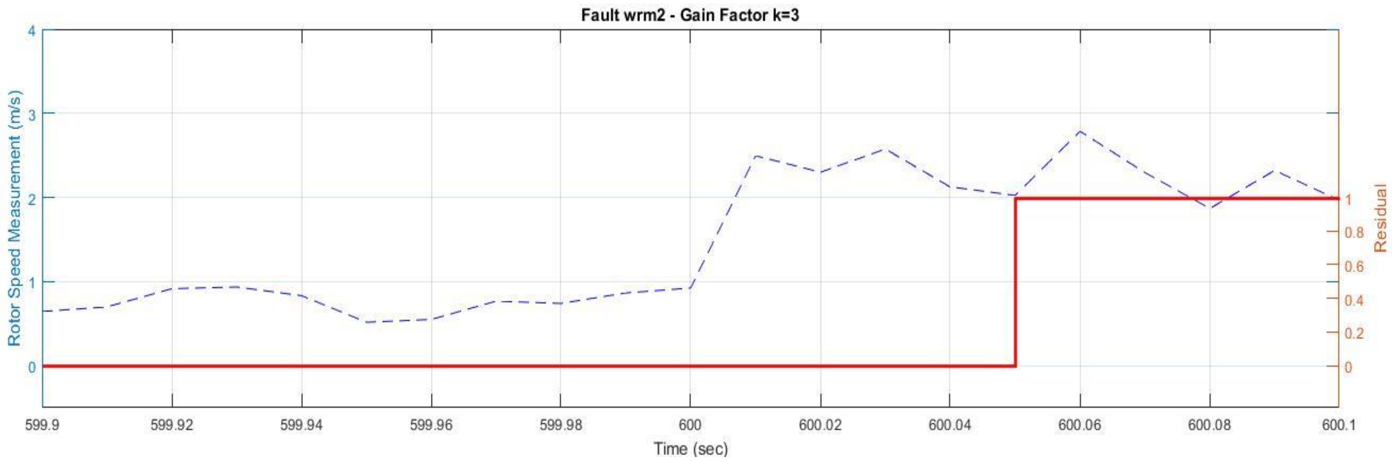


Figure 4.17: Detection Time for the Multiplicative fault $\omega_{r,m2}$ with gain k=3

Figure 4.18 shows the fault detection results of the generator speed sensor $\omega_{g,m2}$. Ranging the gain amplitude from 0.2 to 5 in different time intervals, the results were satisfactory. Although some oscillations are probable, due to the nature of this fault, extracted information from the vector x provided fully-relevant results.

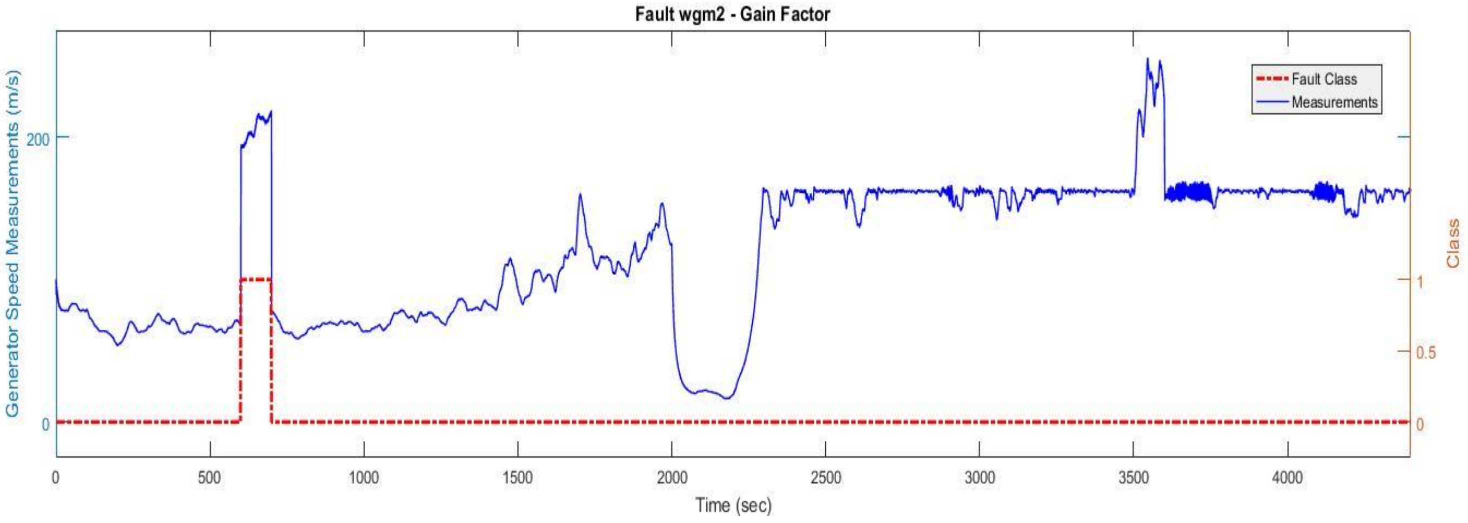


Figure 4.18: Gain factor fault of the generator speed $\omega_{g,m2}$ at time interval [600-700] - (fault 3b) with gain k=2.8

To be noted, modifying the gain, the detection time shows no particular change. Two examples are shown in figure 4.19 and 4.20.

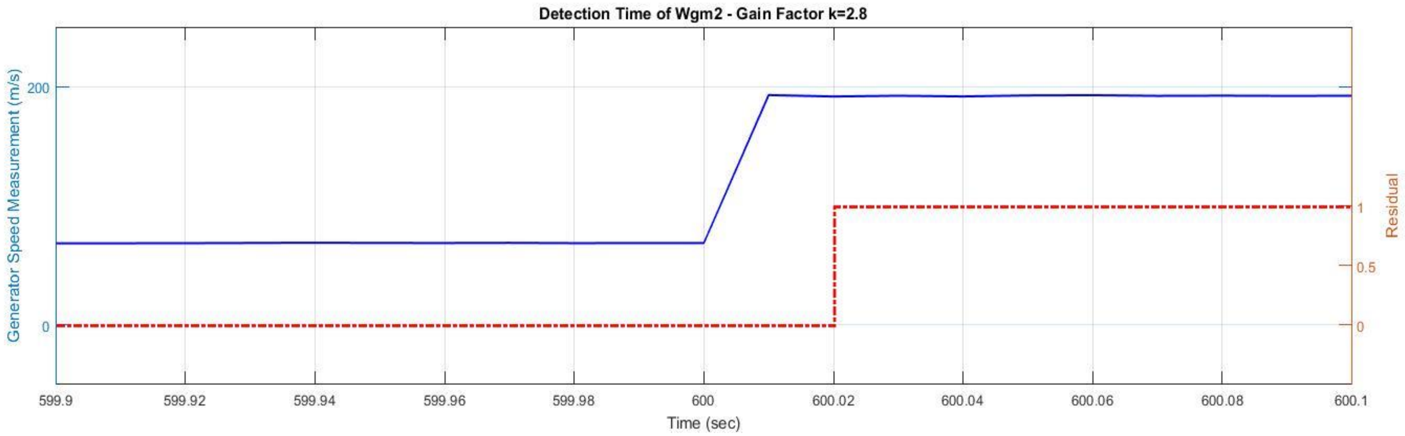


Figure 4.19: Detection Time for the Multiplicative fault $\omega_{g,m2}$ with gain k=2.8

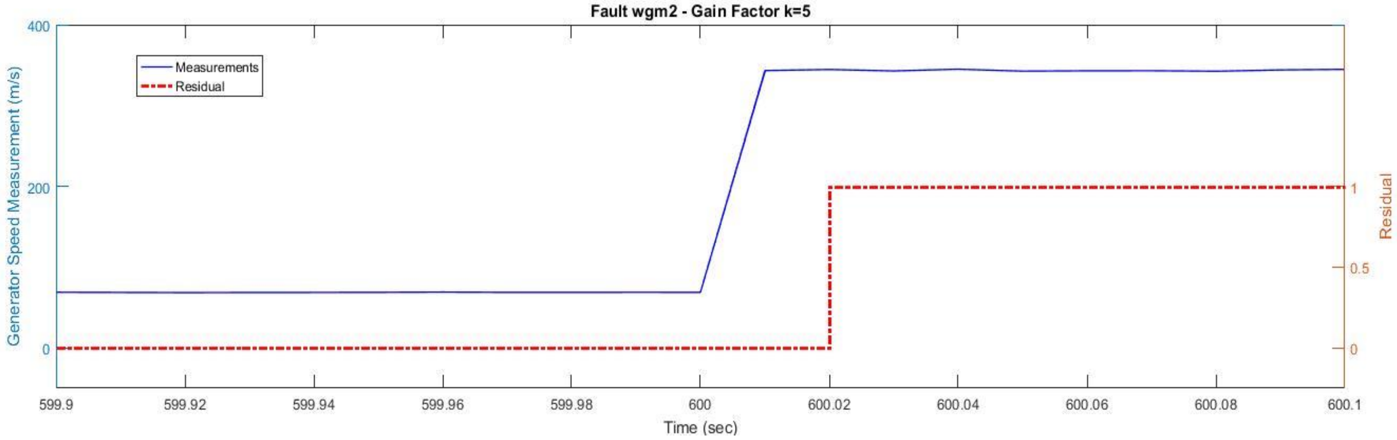


Figure 4.20: Detection Time for the Multiplicative fault $\omega_{g,m2}$ with gain k=5

4.2.3 Converter Torque Actuator Fault

Looking back to section 4.1, generator torque malfunction (fault 4) is of high severity that should be handled as quickly as possible in order to remain in the operational region. To test the functionality of the FDI system, offset values within 10 different intervals with different magnitudes have been deployed. Except for the case where this offset was small and introduced some fluctuations, others were detected as required in terms of fault level and rapidity. Note that x uses the desired torque value that is compared to the measured one with 2 sample periods delay. This fault could be detected in both of the controlled zones. Figure 4.21 is an instance, showing the reaction of the FDI system to the sixth fault scenario.

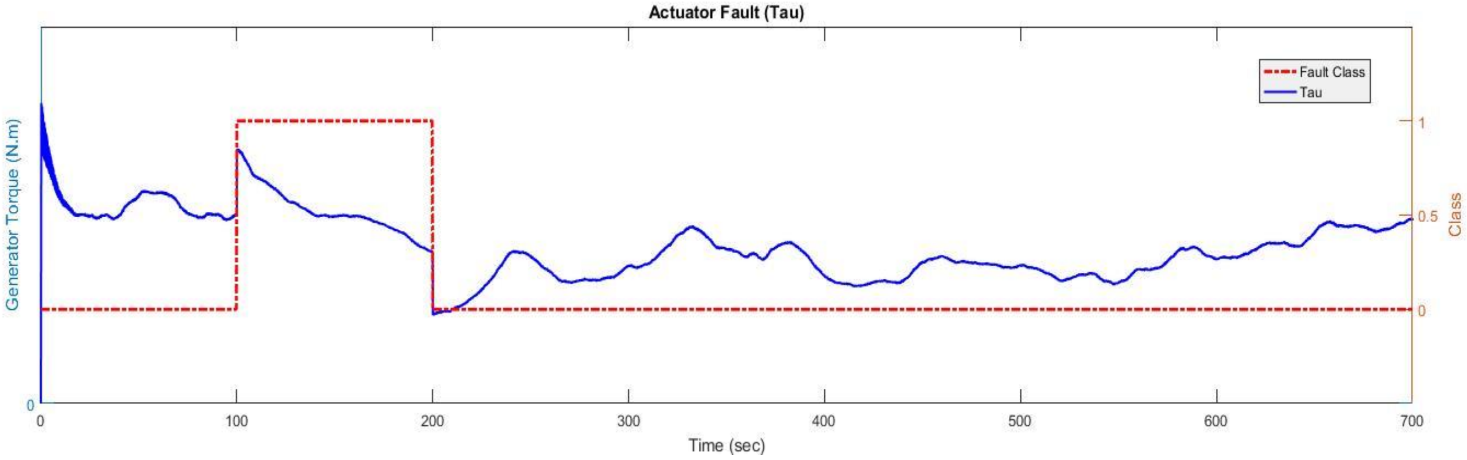


Figure 4.21: Fault in torque control action τ_g within time interval [100-200] - (fault 4) with Offset=2500

The detection time, as shown in figure 4.22 is one sample delayed and thus, the requirements mentioned in table 4.1 are satisfied.

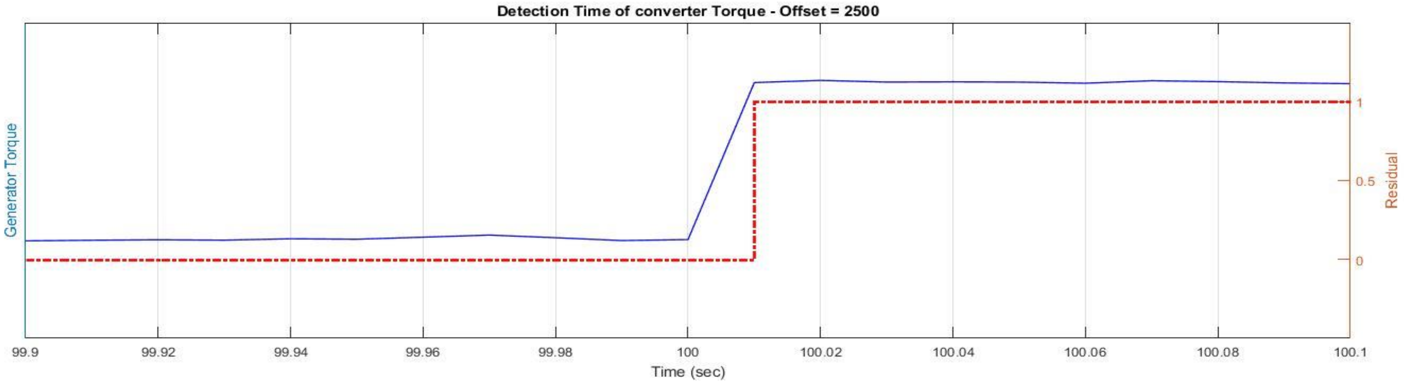


Figure 4.22: Detection time for the offset fault in torque control action τ_g - Offset=2500

4.2.4 Pitch System Actuator Faults

There are two other sources of actuator faults with the medium and high severity level, as explained in section 2.4. They appear in the form of pitch system dynamic change (section 1.4.2) and directly affect the pitch angles $\beta_{k,mi}$*. Equation 4.6 provides the information related to vector x in the learning phase. These data are correlated with pitch angle sensors fault. Hence, the outputs of this system have to be compared with those of section 4.3.1 to reach out the pitch system ageing faults.*

4.2.4.1 Abrupt Change

The second pitch system was imposed to an abrupt dynamic change (fault 5a), caused by the hydraulic pressure drop, in order to test the FDI functionality. None of the fault scenarios shows a reaction to this abnormal behaviour. It means the vector x is not well trained to detect such faults. Any suggestion to provide a better investigation to extract the hidden information in the learning phase would be welcomed as a future task.

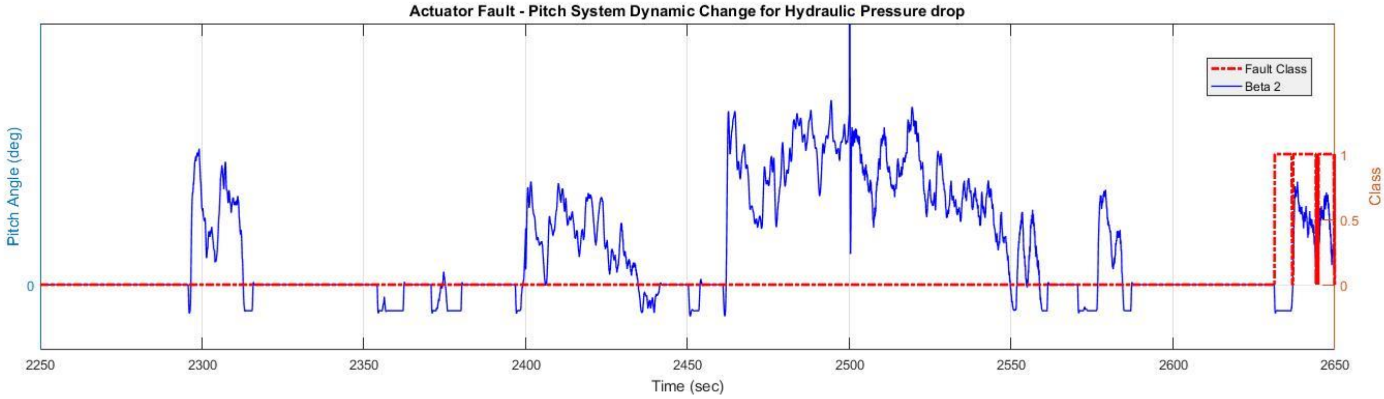


Figure 4.23: Undetected dynamic change fault in the second pitch system β_2 *within time interval [2400-2500] - (fault 5a)*

To be noted, the reaction of the SVM in the right-hand side of the figure 4.23 is due to the multiplicative sensor fault and no reaction is detected within the time interval [2400-2500], where the dynamic change is imposed.

4.2.4.2 Slow Change

The slow dynamic change (fault 5b) is due to the air blubs in the oil that might lead to the control loss over time. The third pitch system is considered as a victim of this sort of faults. A gradual (Trapezoidal Pulse) fault has been considered in this case. However, the results were unsatisfactory, like the abrupt dynamic change, in all fault scenarios and the fault remained hidden.

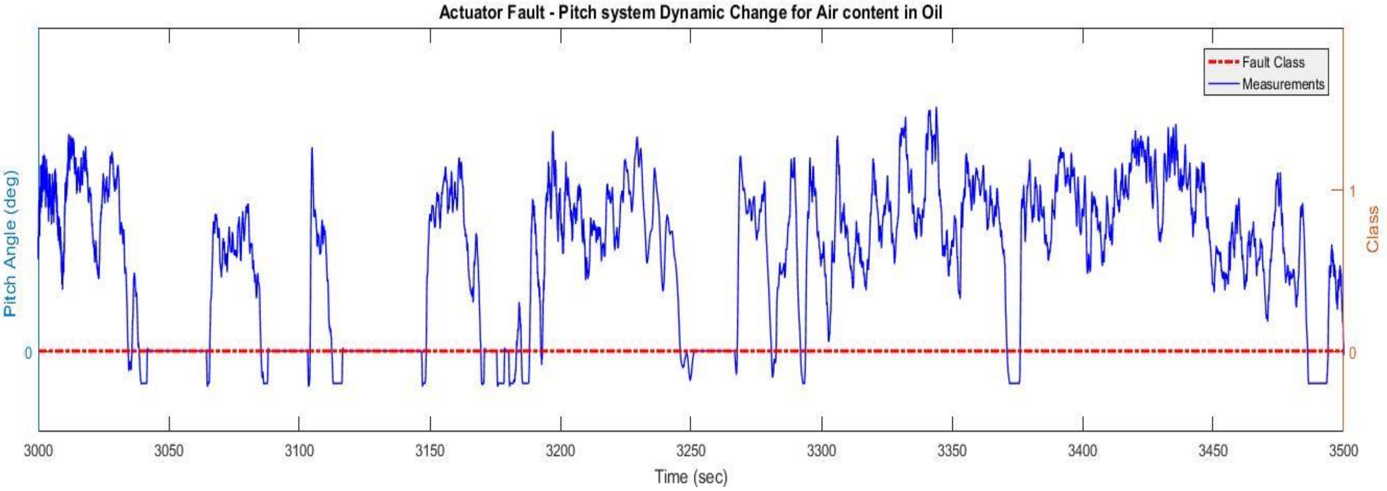


Figure 4.24: Undetected dynamic change fault in the third pitch system β_3 within time interval [3200-3300] - (fault 5b)

4.2.5 Drive Train system Fault

Finally, the slowest dynamic change that is taken into consideration is related to the drive train system wear-out that appears in the form of efficiency change and affects the generator speed. This error was modelled by varying the values of the model parameter η_{dt}. It is detectable only if a massive difference with its nominal value is observed. For example, while the efficiency decreases up to 50%, some oscillations in the FDI residual is visible. In the sixth fault scenario, the so-called amount reduces up to 10% within the time interval [2000-2200]. Figure 4.24 depicted a slow, fluctuating response to such a fault. However, this very slow trend does not lead to a catastrophic event quickly and no constraint over its detection time has been considered. In *order to obtain more meaningful results, further investigation over training vector x is needed, since it was not detected in most fault scenarios.*

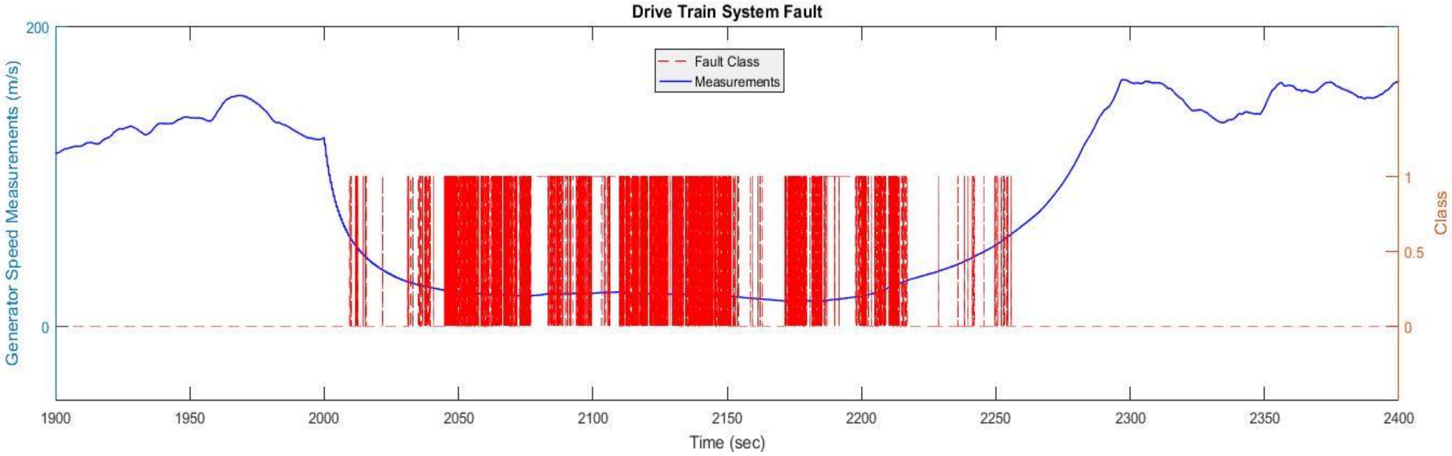


Figure 4.25: Drive train friction fault η_{dt} within time interval [2000-2200] - (fault 6)

Figure 4.26 implies that the first reaction of the FDI to this fault appears after 9.51 seconds (951 samples)

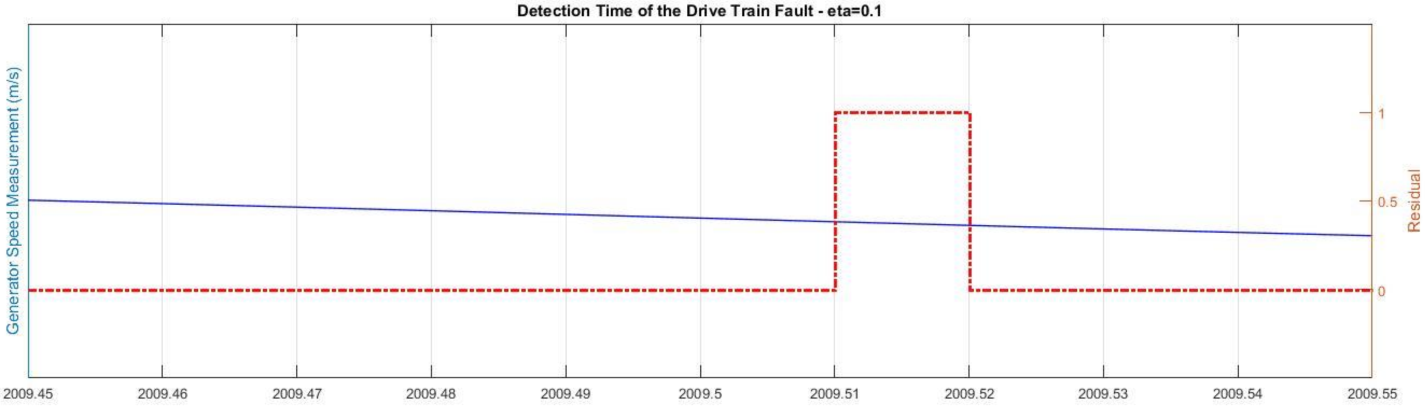


Figure 4.26: Detection time for the drive train fault η_{dt} after 9.51 seconds

4.3 Conclusions

Wind turbines as modern sources of green energy production, like any other novel technology, face many challenges. One of the most challenging tasks in the wind turbine problem is accurate detection and isolation of the faults within the rigorous time constraints. A promising architecture, named support vector machines, has been presented to detect them in wind turbines, particularly the horizontal axis variable-speed benchmark.

SVMs are state-of-the-art classifiers that are widely used in real problems because of their performance, including good generalization and structural risk minimization.

A model comprising of the most supportive, pertinent information extracted from the dynamics of the system under observation was learned to detect all the sensors, actuators and system faults. Gaussian kernels for the sake of dealing with nonlinear datasets have been applied. Then, nine different fault scenarios with various magnitudes, covering both operational control zones have been tested.

The results obtained in this study have provided evidence that most of the FDI requirements, based on Odgaard et al. have been met. Fixed value faults are well-detected. For multiplicative faults, some oscillations have been observed. However, some improvements are obtainable by re-tuning the kernel or low pass filters. The same analysis holds for generator torque and drive-train fault. On the contrary, the efforts to detect pitch actuator faults were unsuccessful.

Future investigations can focus on (i) finding a better pre-processing block (vector x) that leads to the improvement of the SVM performance (ii) the application of this methodology to other potential types of wind-turbine or any other technological system faults.

References

[1] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines–A benchmark model," in Proc. 7th IFAC Symp. Fault Detection, Supervis., Safety Tech. Process., Barcelona, Spain, Jun./Jul. 2009, pp. 155–160.

[2] N. Laouti, N. Sheibat-Othman, and S. Othman, "Support Vector Machines for Fault Detection in Wind Turbines" in Proc. 18th IFAC world congress, Volume 44, Issue 1, Pages 7067-7072, January 2011.

[3] *https://it.mathworks.com/matlabcentral/fileexchange/35130-award-winning-fdi-solution-in-wind-turbines*

Appendix

In this part, the required tool to solve a "Quadratic Programming" problem with Lagrange multipliers approach is described. Constrained optimization by Lagrangian relaxation and its dual form is the technique that is used to solve our SVM problem.

App.1 Quadratic optimization for finding the Optimal Hyperplane

Our goal is to develop a computationally efficient procedure for using the training sample $\{(x_i, y_i)\}\ i = 1, \cdots, N$ *to find the optimal hyperplane that minimizes* $||w||^2$, *subject to some constraints, shown in equation (3.9). It is equivalent to solve a quadratic programming problem with the following format:*

$$Find \quad \arg max_u \quad c + d^T u + \frac{u^T R u}{2} \quad \leftarrow \quad Quadratic\ Criterion \qquad (\text{App.}1)$$

$$subject\ to \begin{cases} a_{11}u_1 + a_{12}u_2 + \cdots + a_{1m}u_m \le b_1 \\ a_{21}u_1 + a_{22}u_2 + \cdots + a_{2m}u_m \le b_2 \\ \vdots \\ a_{n1}u_1 + a_{n2}u_2 + \cdots + a_{nm}u_m \le b_n \end{cases} \quad \begin{pmatrix} n\ additional \\ linear \\ inequality \\ constraints \end{pmatrix} \qquad (\text{App.}2)$$

$$And\ subject\ to \begin{cases} a_{(n+1)1}u_1 + a_{(n+1)2}u_2 + \cdots + a_{(n+1)m}u_m = b_{(n+1)} \\ a_{(n+2)1}u_1 + a_{(n+2)2}u_2 + \cdots + a_{(n+2)m}u_m = b_{(n+2)} \\ \vdots \\ a_{(n+e)1}u_1 + a_{(n+e)2}u_2 + \cdots + a_{(n+e)m}u_m = b_{(n+e)} \end{cases} \quad \begin{pmatrix} e\ additional \\ linear \\ equality \\ constraints \end{pmatrix} \qquad (\text{App.}3)$$

App.1.1 Constrained Optimization by Lagrangian Relaxation

Suppose we want to minimize $f(x)$ *subject to* $g(x) = 0$. *A necessary condition for* x_0 *to be a solution is*

$$\begin{cases} L(x, \alpha) = \frac{\partial}{\partial x}\big(f(x) + \alpha g(x)\big)\Big|_{x=x_0} = 0 \\ g(x) = 0 \end{cases} \qquad (\text{App.}4)$$

where α is the Lagrange multiplier that incorporate the effect of constraints into the objective function. In case of having multiple constraints, $g_i(x) = 0$, $\mathrm{i} = 1, \dots, \mathrm{m}$, *we need a Lagrange multiplier α_i for each of them.*

$$\begin{cases} L(x, \alpha_i) = \dfrac{\partial}{\partial x}\Big(f(x) + \sum_{i=1}^{n} \alpha_i g_i(x)\Big)\Big|_{x=x_0} = 0 \\ \qquad g(x) = 0 \qquad\qquad for \quad i = 1, \cdots, m \end{cases} \qquad (App.5)$$

The case for inequality constraints $g_i(x) \leq 0$ is similar, except that the Lagrange multipliers α_i should be positive and the KKT conditions to be satisfied. This means that only active constraints can contribute in the optimality of the Lagrangian function.

If x_0 is a solution to the constrained optimization problem

$$\begin{cases} \min\limits_{x} f(x) \\ subject\ to \quad g_i(x) \leq 0 \quad for \quad i = 1, \cdots, m \end{cases} \qquad (App.6)$$

There must exist $\alpha_i \geq 0 \quad for \quad i = 1, \cdots, m$ such that x_0 satisfies:

$$\begin{cases} \dfrac{\partial}{\partial x}\Big(f(x) + \sum_{i=1}^{n} \alpha_i g_i(x)\Big)\Big|_{x=x_0} = 0 \\ g(x) \leq 0 \qquad for \quad i = 1, \cdots, m \qquad primal\ feasibility \\ \alpha_i g_i(x) = 0 \qquad for \quad i = 1, \cdots, m \qquad complementary\ slackness \end{cases} \qquad (\mathrm{App.}7)$$

In the SVM case, the solution to the constrained optimization problem is determined by the saddle point of the Lagrangian function $L(w, b, \alpha)$, which has to be minimized with respect to w and b. It also has to be maximized with respect to α. Thus, differentiating $L(w, b, \alpha)$ with respect to w and b and setting the results equal to zero, we get the following two conditions of optimality:

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \qquad (App.8)$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \qquad (App.9)$$

To solve the above problem, first we have to construct and minimize the Lagrangian:

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \big[y_i(\mathrm{w}x_i \; + \; b) - 1\big] \qquad (App.10)$$

Wrt. Constraint $\alpha_i \geq 0 \;\;,\;\; i = 1, \cdots, N$

Then, take derivatives with respect to w and b and equate them to 0.

$$\begin{cases} \dfrac{\partial L(w,b,\alpha)}{\partial w} = w - \displaystyle\sum_{i=1}^{N} \alpha_i y_i x_i = 0 \\ \dfrac{\partial L(w,b,\alpha)}{\partial b} = \displaystyle\sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \rightarrow \begin{pmatrix} \text{parameters are expressed} \\ \text{as a linear combination of} \\ \text{training points} \end{pmatrix} \qquad (App.11)$$

To relax the KKT condition, including the complementary slackness, the following equation holds.

$$\alpha_i[y_i(wx_i \; + \; b) - 1] = 0 \qquad (App.12)$$

That means only those constraints who get nonzero multipliers remain active and the rest are inactive. Those active constraints are related to the support vectors.

The Lagrange multipliers α_i *are called 'dual variables' and each training point has an associated dual variable.*

App.1.2 Lagrangian Dual

The weak duality theorem says that for the general problem, the optimal value of the Lagrange dual problem (d) and the optimal value of the primal minimization problem (p*) are related by:*

$$d^* \leq p^* \qquad (App.13)$$

This means that the dual problem provides the lower bound for the primal problem. The opposite holds true for a primal maximization problem. This theorem always holds. The difference between the two optimal values is called the optimal duality gap.

The strong duality theorem says that for convex problems that satisfy certain conditions, the optimal duality gap is zero, meaning that the optimal values of the primal and dual problems are the same. Assuming that strong duality holds, x^* is the optimal solution of the primal problem, and α^* is the optimal value of the dual problem, then:

$$\alpha_i f_i(x^*) = 0 \qquad for\ i = 1, \cdots, m \qquad (App.14)$$

In particular, we may state the following duality theorem:

1. *If the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are equal.*
2. *In order for w_0 to be an optimal primal solution and α_0 to be an optimal dual solution, it is necessary and sufficient that w_0 is feasible for the primal problem and*

$$L(w_0, b_0, \alpha_0) = \min_w L(w, b_0, \alpha_0) \qquad (App.15)$$

In the SVM problem, given the training sample $\{(x_i, y_i)\}$ for i=1,...,N; by substituting $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ in equation (App.10), we obtain:

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N} \alpha_i y_i x_i^T \sum_{j=1}^{N} \alpha_j y_j x_j + \sum_{i=1}^{N} \alpha_i \left(1 - y_i \left(\sum_{j=1}^{N} \alpha_j y_j x_j^T x_i + b \right) \right)$$

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i y_i \sum_{j=1}^{N} \alpha_j y_j x_j^T x_i - b \sum_{i=1}^{N} \alpha_i y_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \alpha_i \qquad (App.16)$$

Where α_i multipliers are non-negative and $\sum_{i=1}^{N} \alpha_i y_i = 0$ is zero by virtue of the optimality condition according to the equation (App.11).

The function $L(\alpha)$ to be maximized depends only on the input patterns in the form of a set of dot products $\left\{x_j^T x_i\right\}$ for i=1… N.

Then, the objective function of the dual problem that needs to be maximized is

$$\max\ w(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \qquad (App.17)$$

$$subject\ to\ \ \alpha_i \geq 0\ \ , \qquad \sum_{i=1}^{N} \alpha_i y_i = 0\ , i = 1, \cdots, N \qquad (App.18)$$

This is a quadratic programming (QP) problem. Solving this problem, a global maximum of α_i can always be found. w can be recovered by:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i = \sum_{i \epsilon SV} \alpha_i y_i x_i \qquad (App.19)$$

When set (α^*, b^*) is obtained by solving the so-called QP, we are able to apply the SVM to classify the new sets of data. If x is a new set of data, its classifier is as follows:

$$sign[f(x, \alpha^*, b^*)] \qquad (App.20)$$

where:

$$f(x, \alpha^*, b^*) = w^* x + b^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i\, x + b^* = \sum_{i \epsilon SV} \alpha_i^* y_i x_i\, x + b^* \qquad (App.21)$$