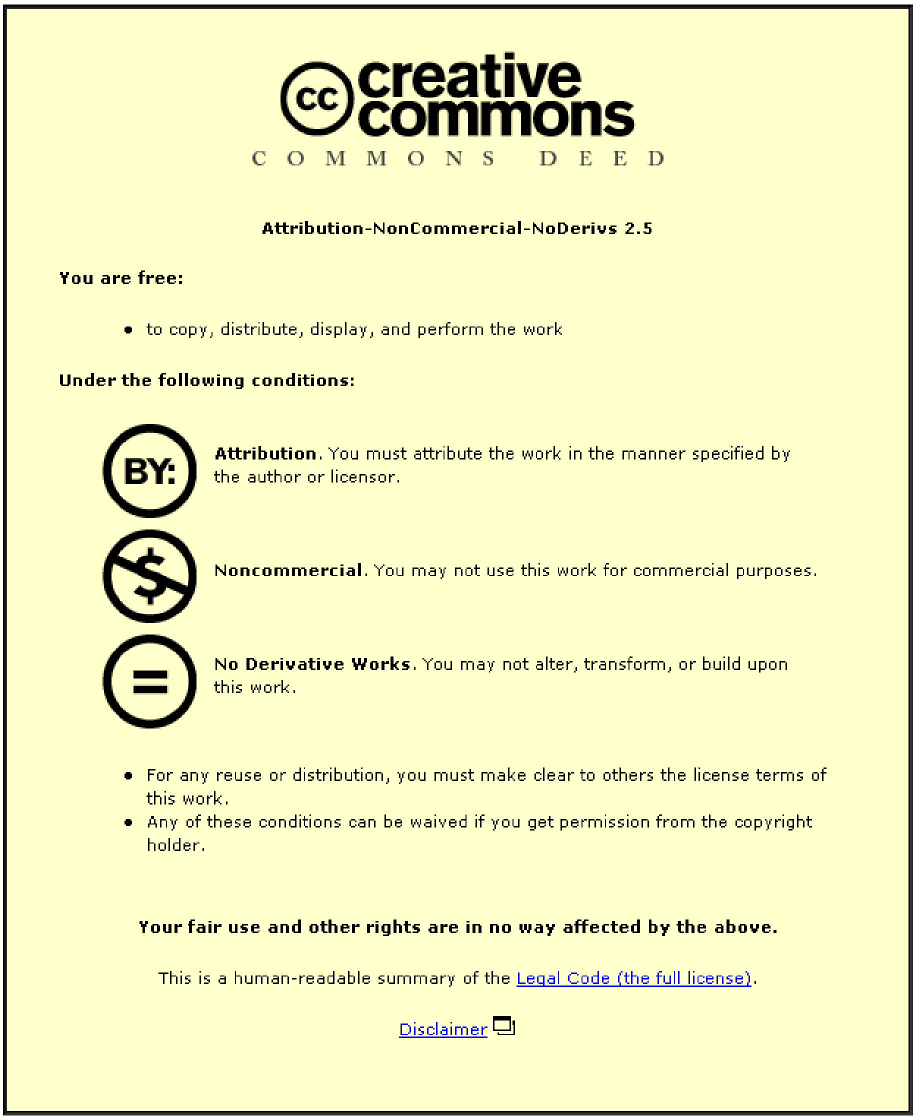


Development of a self-tuned drive-train damper for utility-scale variable-speed wind turbines

by

Athanasios Mathiopoulos

Doctoral Thesis submitted in partial fulfilment of the requirements
for the award of the degree Doctor of Philosophy (PhD) of Loughborough
University

December 2011

Abstract

This thesis describes the development of a procedure that tunes a wind turbine drive-train damper (DTD) automatically. This procedure, when integrated into the controller of any utility-scale variable-speed wind turbine, will allow the turbine to autonomously and automatically tune its DTD on site. In practice this means that the effectiveness of the damper becomes independent on the accuracy of the model or the simulations used by the control engineers in order to tune the damper. This research is motivated by the fact that drive-train failures are still one of the biggest problems that stigmatises the wind turbine industry. The development of an automatically tuned DTD that alleviates the drive-train fatigue loads and thus increases the reliability and lifetime of the drive-train is thus considered very beneficial for the wind turbine industry.

The procedure developed begins by running an experimental procedure to collect data that is then used to automatically system identify a linear model describing the drive-train. Based on this model a single band-pass filter acting as a DTD is automatically tuned. This procedure is run for a number of times, and the resulting DTDs are compared in order to select the optimal one.

The thesis demonstrates the effectiveness of the developed procedure and presents alternative procedures devised during research. Finally, insight into future work that could be performed is indicated in the last chapter of the thesis.

Acknowledgements

I would like to thank all the people who helped me during my research. First and foremost I would like to thank Prof. Simon Watson, my very supportive supervisor who dedicated his time and effort to helping me through this research. I am really glad and thankful for having had the opportunity to work with him, for his kindness and understanding, as well as his readiness to help me in my goal in a completely altruistic way.

I would also like to thank my former colleagues David Witcher and Ervin Bossanyi for the time they spent in training me in designing various industry standard controllers, and the support and guidance they later provided.

The Loughborough University and its staff members are gratefully acknowledged for the financial support during the first years of this research, but most importantly for the very welcoming and hearty attitude that encouraged me to continue with my research over its long eight-year period.

Finally, I would like to thank all my friends and family for bearing with me and my very busy schedule for the last eight years.

Contents

Chapter 1

An introduction to wind energy is presented in this chapter. A very brief historical overview of wind turbines is given, followed by a technology overview of utility-scale wind turbine types available today. Then, a slightly more detailed overview of the control strategy used in variable-speed wind turbines is presented. Having a basic understanding of the way a modern variable-speed wind turbine operates is crucial for the comprehension of this research. Finally, the components of a modern wind turbine drive-train are described, again in order for the reader to acquire the necessary understanding of this particular wind turbine subsystem and aid them in the comprehension of this research.

1.1 Introduction

Wind has been shaping the earth as we know it for millions of years. Our landscape, the plants around us and the animals living on earth would not be the same without it. Humans have been using wind energy for thousands of years, for transport, water pumping and grinding corn. Probably the earliest documented reference to a machine that uses wind as a power source is the wind-wheel, called 'anemurion' ("ανεμούριον"). The anemurion was designed to power a pipe-organ, and was discovered by the Greek engineer Heron of Alexandria, in the 1st century A.D. [1]. The first practical windmills, used for pumping water and grinding corn, were the vertical axle windmills invented in Persia in the 7th century A.D [2]. A lot of different windmill types were developed ever since, but it was after 1887, when James Blyth erected the first electricity producing vertical axis wind turbine[3], that we started to use wind energy in order to cover part of our ever increasing electrical energy needs.

During the last few decades a large number of wind turbine designs have emerged. A small number of these designs have been universally adopted by wind turbine manufacturers for the utility-scale wind turbine market. This is not to say that some of the other designs do not have substantial merits. However, as the wind industry is maturing, there is an evidently growing consent that there are a few design concepts which have a clear overall advantage over all other designs.

Most of these design concepts utilise a drive-train composed of a number of moving parts usually including gearboxes and shafts. Unfortunately though, the drive-train of a modern wind turbine is one of the most unreliable mechanical sub-assemblies on a modern utility-scale wind turbine. Spinato F. et al. in [4], through an analysis of more than 6,000 wind turbines, and an analysis period of 11 years, have shown that gearbox failures are not only relatively frequent, at approximately 0.1 failures per wind turbine per year, but also lead to the highest average repair times compared to all other wind turbine sub-assembly failures. The relatively high failure frequency and the very high cost of gearboxes, lead to prohibitive replacement expenses. It should be noted that according to the author's experience, every such replacement amounts to a cost of approximately 10% to 20% of the cost of installing a new wind turbine. For example, a MW class wind turbine, from a reputable European wind turbine manufacturer, installed in Europe, including transportation and installation, costs around 0.8 to 1.0 M EUR per MW (2010 prices), while the replacement of its gearbox with a new one costs around 0.12 to 0.16 M EUR per MW (2010 prices). By adding to this cost the associated revenue loss due to the long wind turbine downtime, it is clear that the service reliability of the drive-train is very important.

For variable-speed wind turbines, where the drive-train of the wind turbine is not sufficiently damped by the generator, a means of adding damping is necessary in order to reduce fatigue loading on the drive-train components. Reducing fatigue loading is one of the ways manufacturers are trying to tackle the reliability problem of the drive-train components. One technique used in order to provide additional damping is to control the torque of the generator in a way that increases the damping of the drive-train.

Usually, a Drive-Train Damper (DTD) is employed to perform the task of constantly changing the torque of the generator in a way that damps the drive-train. The DTD structure, which may either be a filter, or a routine incorporated into the controller of the wind turbine, is defined by an engineer. The engineer using a model of the wind turbine, can tune the DTD in order to achieve the added damping of the drive-train.

This thesis outlines the work undertaken by the author on the development of a procedure that tunes a DTD automatically. The purpose of this procedure is for it to be used by the controller of any utility-scale variable-speed wind turbine in order to allow the turbine to autonomously and automatically tune its DTD on site. Practically this means that the effectiveness of the damper will not be dependent on the accuracy of the model or the simulations used by the control engineers in order to tune the damper.

1.2 Technology Overview

A brief overview of the utility-scale wind turbine types available today is given in this section, so that the reader who is not familiar with such information can follow this thesis.

A wind turbine is a machine designed to convert the kinetic energy of air into electrical energy. The way in which this goal is achieved is by first converting the kinetic energy of the air into mechanical energy of the wind turbine’s rotating components, and then converting this mechanical energy into electricity via the wind turbine’s generator.

The most common way of converting the kinetic energy of the air into mechanical energy is by the use of a rotor comprised of one or more blades with an aerofoil shaped cross section. The most widespread rotor configuration, adopted by almost all large-scale commercial wind turbines today, is that of a three-bladed rotor turning on a horizontal axis. As the blades sweep through the rotor plane, the resulting wind velocity imparts a lift force on the aerofoil shaped blades, which increases the rotor torque and provides the required mechanical energy to the wind turbine. This research will focus only on the three bladed, horizontal axis rotor configuration, although in practice, the principles described here and the methodology developed are easily transferred to rotors of greater or fewer number of blades, either with a horizontal or vertical rotor axis configuration.

The mechanical energy produced by the rotor is converted via a generator into electricity. The generator is usually rated at a power level the wind turbine can achieve at 11 to 14 m/s wind speed. This wind speed is usually referred to as the rated wind speed. At lower than rated wind speeds, the wind turbine needs to extract as much energy from the wind as possible, and convert it to electrical energy. At higher than rated wind speeds the wind turbine needs to limit the energy it extracts from the wind to its designed limit.

The rated wind speed of modern utility-scale wind turbines is above the average wind speeds observed at most wind farm sites, but is at the same time frequently exceeded.

Figure 1 below shows a Rayleigh distribution, representing the wind distribution of a typical site.

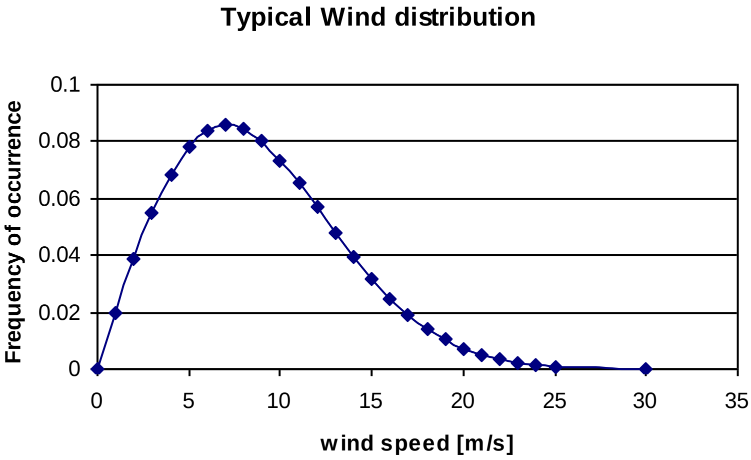


Figure 1: Typical wind distribution, represented by a Rayleigh distribution.

Although the average wind speed of this particular site is 8 m/s at hub height, the wind speed exceeds 12 m/s for approximately 1,500 hours per year. Thus, a wind turbine with a rated wind speed of 12 m/s would have to work above this limit for most of these 1,500 hours every year.

The reason for not rating the generator, and the other wind turbine components, for a power level that could be achieved at higher wind speeds is that this would increase the cost of the wind turbines without achieving an analogous benefit from the increased energy production. If the generator's, and thus wind turbine's, power rating were too high, this rated power would only be achieved very rarely. Imagine a wind turbine installed at a site with the wind distribution of Figure 1. If this wind turbine were for example rated at 20 m/s, this rated wind speed, would be reached and exceeded for only 65 hours during a whole year. Thus, the increased cost of the power-train and of most other wind turbine components associated with the very high power rating would not be justified by the relatively small extra energy yield in the wind turbine's operating lifetime.

On the other hand, the reason for not rating the generator and the other wind turbine components to low wind speeds, is that the only benefit of such rating would be to decrease the cost of only specific parts of the wind turbine, mainly the power-train of the wind turbine. The cost of the rest of the wind turbine components, and especially the blades, tower and foundations would then be excessive in relation to the total energy yield achieved. Again, imagine a wind turbine installed at a site with the wind distribution of Figure 1. If this wind turbine were for example rated at 5 m/s, the wind turbine would be limited to work at its low rated wind speed (and thus for a constant rotor geometry, low rated power), for more that 6,400 hours per year, wasting all the energy it could capture if the wind turbine were rated at a higher level.

Having a rated wind speed that is going to be frequently exceeded, means that a wind turbine needs to be able to operate up to its rated power by extracting as much energy from the wind as possible, and then, when its rated power (or wind speed) is reached, it needs to limit the power extracted from wind to the desirable level. The alternative of just shutting down the wind turbine at rated wind speed would lead to a significant overall energy loss and is thus avoided.

The two main factors differentiating the wind turbine types available today are their speed and power control characteristics. The speed control characteristics classify the wind turbines into fixed or variable-speed wind turbines, whilst the power control characteristics classify the wind turbines into stall-controlled or pitch-controlled wind turbines.

Fixed-speed Wind Turbines

The first widely used and commercially successful wind turbines were fixed-speed wind turbines. They were considered simple and reliable, and due to their very simple and low cost electrical components were initially preferred in comparison to variable-speed wind turbines, the technology of which matured much later. This type of wind turbine utilises a simple induction generator directly connected to the grid. Due to its slip characteristics, the induction generator was considered ideal for wind energy applications as it provided damping to the drive-train. More recently however, and with the evolution of power electronics, this type of wind turbine is now less often installed.

Variable-speed Wind Turbines

Variable-speed wind turbines (VSWTs) are currently the most common type of wind turbines available on the market. There exist limited variable-speed wind turbines, allowing only a small variability in the rotor speed, extended variable-speed wind turbines and full variable-speed wind turbines.

Limited variable-speed wind turbines are wind turbines whose rotor can change rotational speed by an amount typically of the order of 10% of the nominal rotational speed. A typical example of such wind turbines are wind turbines equipped with the early versions of the variable rotor resistance induction generator system patented by Danish Manufacturer Vestas, under the name ‘Optislip’. In these wind turbines, the resistance in the windings of the induction generator’s rotor is varied to achieve variable slip and thus variable generator speed [5,6]. The Vestas V-47 wind turbine for example, achieves generator speeds varying for 1500 to 1650 rpm, i.e. a total variation of 10% [7].

Extended variable-speed wind turbines are wind turbines whose rotor can change rotational speed substantially, typically ±30%, but are limited by the generator to certain variability around the nominal speed. The most common type of such wind turbines are the wind turbines equipped with a doubly-fed induction generator. Most of the wind turbine manufacturers, including Vestas, Siemens, GE, Nordex, Suzlon, Gamesa and Acciona, produce wind turbines that employ such generators. The method in which speed variability is achieved, is by adding or subtracting energy to or from the generator’s rotor windings. This is performed via slip rings connected to power electronics that transform grid frequency A/C voltage to the appropriate frequency A/C voltage of the rotor windings. By adding or subtracting energy from the rotor windings the slip of the induction generator is changed and thus the generator and thus rotor speed is changed [8]. Another example of extended variable-speed wind turbines are the wind turbines with the more technologically advanced ‘Optislip’ system, such as the Vestas V-52 that can achieve generator and thus rotor speed fluctuations of around ±30% of its rated speed [9].

Full variable-speed wind turbines are wind turbines, the rotor speed of which can be freely changed according to practical needs. The way this is achieved is by decoupling the generator from the grid, using power electronics rated at the full capacity of the generator. In this way, all the produced electricity is first transformed to DC power, and then to AC of the appropriate grid frequency. Some wind turbine manufacturers such as Enercon, GE and Lagerway are using such an approach.

The advantages of being able to operate at variable-speed are related to increased energy capture [10,11], reduction of some of the wind turbine loads [12] and reduced noise at low wind speeds. The increased energy capture is achieved by operating the wind turbine at a different rotational speed according to the wind speed. The reason why this increases energy capture is explained in the following section of this introduction. In addition, incoming wind variations are absorbed by the inertia of the rotor via changes in the rotor speed. This reduces the stress fluctuation on the wind turbine components, and smoothes power production. Finally, by being able to operate at low rotational speeds when the wind speed is low, variable-speed wind turbines can operate more quietly than fixed-speed wind turbines. The main source of noise in modern utility-scale wind turbines comes from the aerodynamic noise created by the blade itself and by the blade moving past the tower. Consequently, as the rotor speed decreases the aerodynamic noise also decreases.

The disadvantages compared to fixed-speed wind turbines are the increased unreliability related to the extra power electronic components used to interface the generator to the grid, and the cost related to these extra components. Spinato et. al. in [4] have demonstrated that the power electronics show the second highest failure rate amongst all wind turbine sub-assemblies.

The advantages however of the variable-speed wind turbine concept are now outweighing the disadvantages, as variable-speed wind turbines are achieving a higher market penetration reaching above 80% of the yearly installed market share in 2005 [13].

As previously mentioned, a wind turbine needs to be able to operate up to its rated power by extracting as much power from the wind as possible, and then when its rated power is reached, it needs to limit the power extracted to the desirable level. The two dominant power regulation methods are presented here.

Stall controlled wind turbines

The simplest form of power control employed on modern utility-scale wind turbines is that of stall control.

Figure 2 below demonstrates how lift, the aerodynamic force producing the required rotor torque for power extraction, is reduced when the blades become stalled:

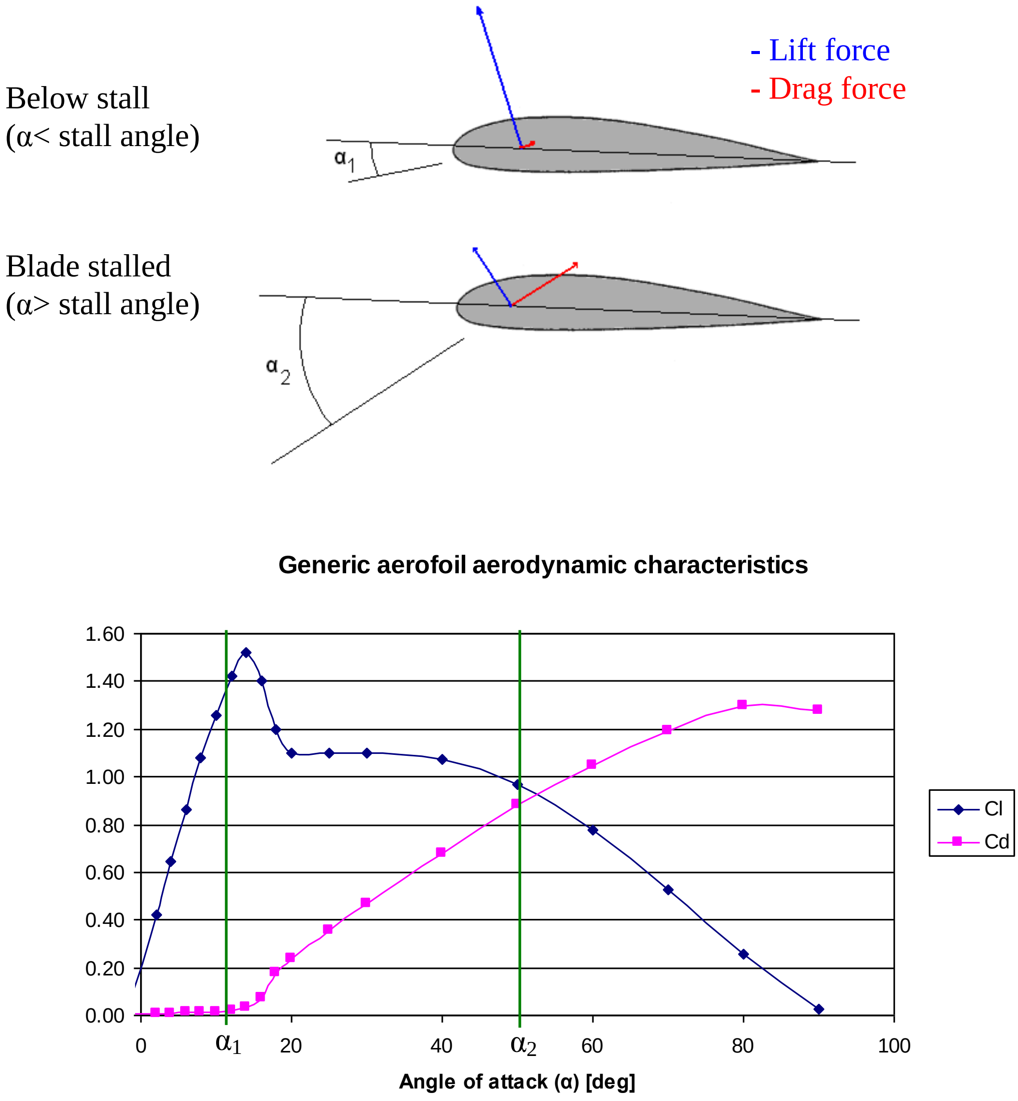


Figure 2: Lift and drag forces on an aerofoil, before and after stall.

With stall control, the blades have fixed geometry. As the wind speed increases the blades start stalling and thus less power is captured from the wind. This is achieved by designing the blades of the wind turbine to gradually stall as the wind speed increases.

Stall is designed to start just after the rated wind speed, but in a gradual manner, thus in effect limiting the power produced at the rated power level.

Figure 3 below shows how the increase in wind speed changes the angle of attack from an angle α_1 below stall angle to an angle α_2 above stall angle, and thus leads to stalling:

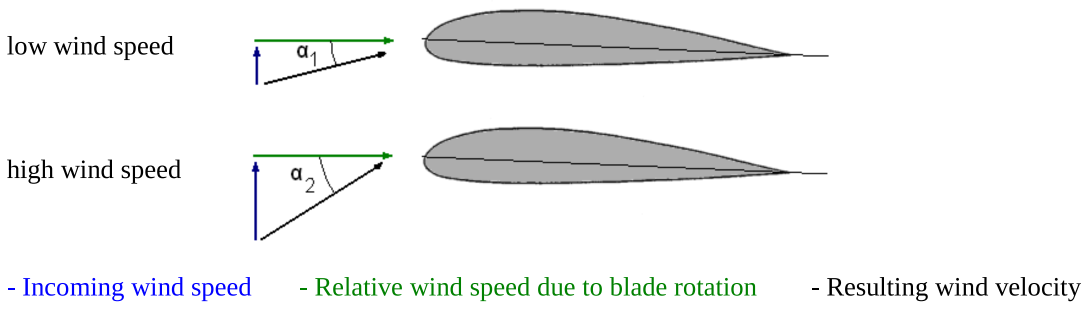


Figure 3: Wind velocity at low and high wind speeds.

Limiting the power extracted to the necessary level, is unfortunately not possible to achieve in practice with the desired precision. In practice, stall controlled wind turbines extract less power than their rated power at high wind speeds*.

Pitch controlled wind turbines

The more modern approach of power control employed on modern utility-scale wind turbines is that of pitching the blades.

By pitching the blades, the effective angle of attack of the blades changes, and thus the power extraction can be controlled. The pitching action can be used to lower the angle of attack of the blade, thus reducing the lift gradually (this is usually referred to as pitch to feather), or by increasing the angle of attack to partially stall the blade (this is usually referred to as pitch to stall, or assisted stall).

* Based on the published power curves of commercial stall controlled wind turbines, and the author's experience from monitoring the production of stall controlled utility scale wind turbines.

The main advantage of this power regulation method is that it allows more accurate power control. This power regulation method also makes it possible to control the wind turbine in a way that makes its power output much less susceptible to ambient conditions, blade soiling and icing[*]. Additionally the blade pitch mechanism limits the high mechanical stress experienced by high wind speed gusts more effectively than stall controlled wind turbines. By pitching the blades into the wind, or out of it, the wind turbines employing pitch control can also start-up at lower wind speeds, and perform faster emergency stops, with less structural loading to the wind turbine.

This research will focus on the horizontal-axis, three-bladed rotor wind turbine operating at extended variable-speed, and regulated by active pitch. This configuration was chosen because most utility-scale wind turbines available on the market today are of this configuration, and according to Hansen et. al. [[13]], this configuration dominates the market achieving more than 60% of the yearly installed market share in 2005.

[*] This much reduced susceptibility to ambient conditions, blade soiling and icing, is commonly quoted by both manufacturers and researchers, and has been verified by the author’s monitoring experience of more than 5 years of operational data of approximately 100 wind turbines employing either stall or pitch control.

1.3 Control Strategy

In order to be able to produce energy under constantly varying wind conditions, a wind turbine needs to be controlled by a control system. As mentioned in the previous section, at low wind speeds the wind turbine needs to extract as much energy from the wind as possible, and convert it to electrical energy. At high wind speed, it needs to limit the energy it extracts from the wind to its designed limit.

More specifically, the main roles and objectives of the wind turbine control system can be summarised by the following list [14,15]:

- Perform general control actions such as starting the wind turbine at the cut-in wind speed, stopping it at the cut-out wind speed or at any appropriate error signal, maintaining the rotor yawed towards the wind direction, etc.

- Maximize the extracted wind energy at low operating wind speeds.

- Control the aerodynamic power and the rotor speed at the rated level for high wind speeds.

- Control the electrical power quality in order to match the grid regulator requirements

- Minimize the loading on the mechanical and electrical components of the wind turbine, by decreasing loads, load fluctuations and control actions.

As it can be seen from the wind turbine control objectives, the wind turbine operates in two regions with somewhat different control objectives. The region below rated wind speed, henceforth referred to as the below rated region, and the region above rated wind speed, henceforth referred to as the above rated region. Figure 4 in the next page shows the power production characteristics of a pitch controlled variable-speed wind turbine vs. wind speed, highlighting the two regions:

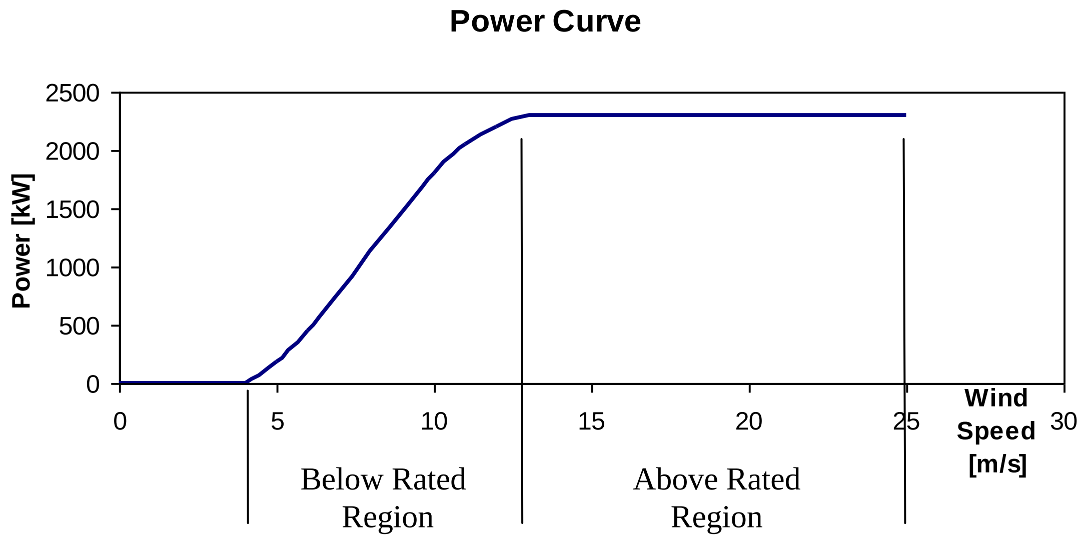


Figure 4: Power curve of a typical utility-scale variable pitch wind turbine.

In the below rated region the wind turbine operates at variable rotor speed in order to capture the maximum amount of energy available in the wind. The generator torque is used as a control output in order to vary the rotor speed. The blade pitch angle is usually held constant in this operating region. In the above rated region, the wind turbine operates with a constant torque demand, and is controlled via the blade pitch to a constant rotor speed set point, thus achieving constant power output.

The above described control methodology in these two regions is analysed in the following paragraphs.

Operation below rated power

In the region below rated power, the generator torque demand is varied in order to control the rotor speed of the turbine. The speed of the rotor is controlled in such a way as to achieve maximum aerodynamic efficiency for all wind speeds. Pitch angle (β), is the angle of the blade chord with respect to the direction of rotation. Zero or fine pitch is usually set at the angle of the blade chord where the angle of attack of the blade sections maximises power extraction in this operation region.

In general, wind turbine aerodynamic efficiency depends on the wind turbine design and ambient conditions. The aerodynamic efficiency of a wind turbine rotor can be represented by the power coefficient (C_P). The power coefficient (C_P) is the percentage of the available kinetic energy of the incident air mass that is converted into mechanical energy by the rotor, and it is expressed as follows:

$$C_P = \frac{Power\_Extracted}{Power\_Available} = \frac{Power\_Extracted}{\frac{1}{2}\rho U_\infty^3 A_D} \qquad \textbf{Eq. 01}$$

Where,

Power_Extracted is the aerodynamic power extracted by a rotor,

Power_Available is the available kinetic energy of the incident air mass, in the absence of the rotor,

ρ is the air density,

U_∞ is the free stream wind speed (the wind speed far upstream the rotor, assuming that the wind speed is constant in time, and the wind direction is perpendicular to the rotor). It can also be considered as the wind speed that would flow through the rotor plane in the absence of the rotor.

A_D is the area swept by the rotor.

For a specific rotor design and pitch angle, and under specific ambient conditions, the power coefficient is a function of the tip speed ratio (λ). The tip speed ratio is the ratio between the speed of the blade tips (the outer, in a radial sense, part of the blade) and the free stream wind speed. Figure 5 in the next page shows a typical C_P – λ curve:

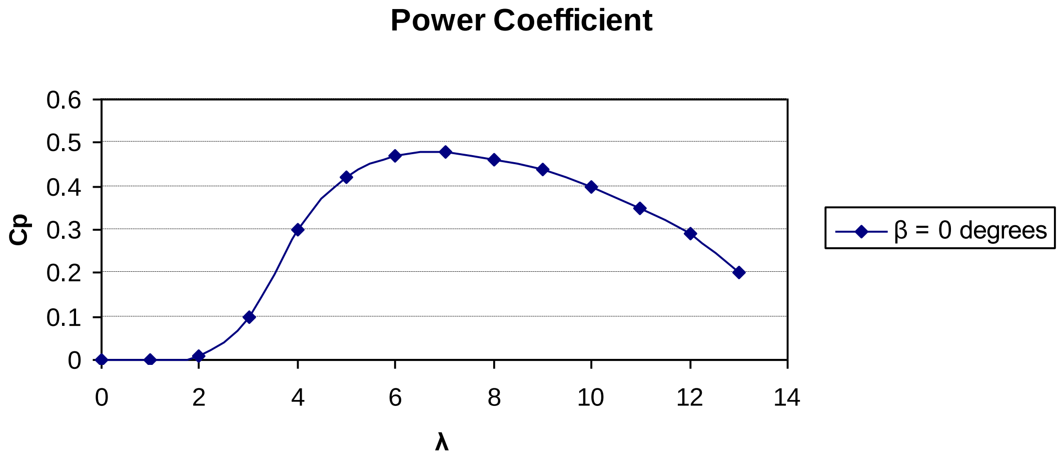


Figure 5: typical C_P – λ curve

As shown in Figure 5, there is a single tip speed ratio value, called the optimum tip speed ratio (λ_{opt}), where the maximum power coefficient (C_{Pmax}) is reached. In the specific example of Figure 5, λ_{opt} is equal to 6.7.

If the tip speed ratio is low, the efficiency is also low. This is mainly because a) a larger volume of air passes through the rotor unaffected by the blades (low tip speed ration means that the rotor blades travel slowly with respect to the wind speed, and thus the effective solidity of the rotor is low), and b) the angle of attack of the wind with respect to the blades (the angle between the blade chord line and the effective local wind velocity) is very large, thus the blades are partially (or totally) stalled, thus not being able to effectively extract energy from the wind.

If the tip speed ratio is high, the efficiency is again low. This is mainly because the angle of attack of the wind with respect to the blades is very small, thus the forces on the blades are relatively low, and become increasingly dominated by drag forces. Also, as the effective solidity of the rotor becomes higher at higher tip speeds, the air speed at the rotor disc is becoming lower in comparison to the free stream wind speed and this eventually leads in reduced efficiency. This follows the actuator disc theory that can be used to prove that:

$$C_P = 4a(1-a)^2 \quad \textbf{Eq. 02}$$

where a is the inflow factor, or axial flow induction factor, and is the factor by which the free stream wind speed is reduced up to the rotor disk.

Equation 02 maximises at a =1/3. So, as the effective solidity of the rotor increases at high tip speed ratios, the inflow factor becomes larger, and moves away from the optimum point, reducing the power coefficient.

The following graph shows a typical C_P – λ curve, this time with the blades set at different pitch angles:

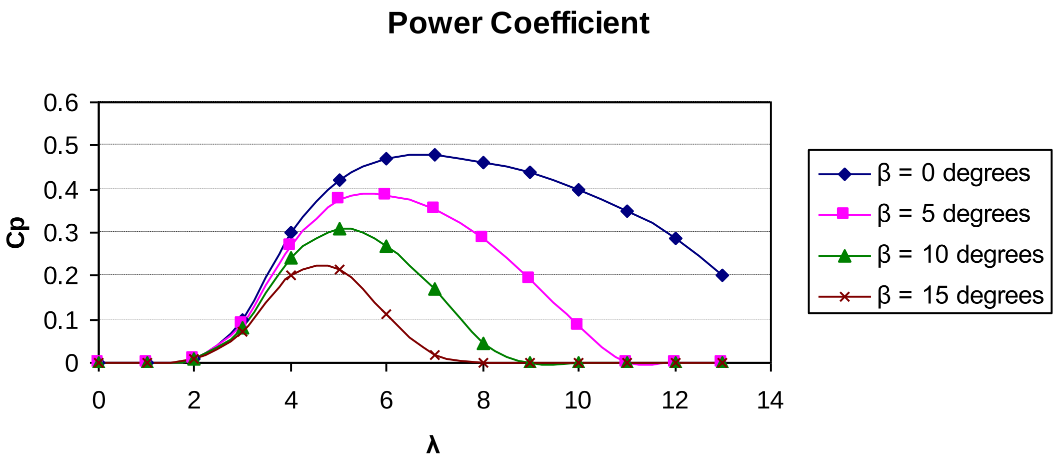


Figure 6: typical C_p – λ curve at various blade pitch angles.

It is evident from the above graph that as the blade pitch angle changes away from its optimum position (here assumed to be at zero degrees) the power coefficient drops.

As previously described the controller of a wind turbine at the below rated region is trying to achieve maximum aerodynamic efficiency. As it is evident from the above analysis of some of the C_P characteristics, this is accomplished by maintaining the optimum tip speed ratio (λ_{opt}) that produces the maximum C_P (C_{Pmax}). The pitch is maintained at its fine position (0 degrees) which is assumed to be the pitch angle that gives the maximum aerodynamic efficiency.

In order to stay at its optimum tip speed ratio, the controller of the variable-speed wind turbine tries to change the rotor speed in proportion to the wind speed. This is achieved by means of changing the generator torque demand.

One way of maintaining the optimum tip speed ratio, is by setting the generator torque Q_m to be proportional to the square of the rotor or generator speed:

$$Q_m = K \cdot \Omega^2 \ , \qquad \textbf{Eq. 03}$$

where Ω is the rotor speed and *K* is a constant that, in steady state, balances the rotor's mechanical torque with the aerodynamic torque:

$$K = \frac{1}{2}\rho\pi R^5 \frac{C_{P\max}}{\lambda_{opt}{}^3}$$

where *R* is the rotor radius.

Equation 03 can be further refined by the inclusion of a term describing the mechanical losses in the drive-train, thus correctly balancing the generator rotor demand with the mechanical and aerodynamic power.

Many researchers, such as Bossanyi [16,17], Leithead et. al. [18] and B. Boukhezzar et. al. [19], have proposed alternative control strategies to maximise the power extraction in the below rated region. According to all researchers, the control signal demanded by this simple approach is slow in reacting to the changes in wind. This happens because of the inertia of the rotor. As the wind speed changes, there is a delay in the rotor speed tracking this change. The larger the inertia of the rotor, the slower the rotor speed will respond to the change in wind speed. This means that in practice, and unlike the steady state case, the rotor will operate at a non-optimal C_P point, if the generator torque is directly set to a value proportional to the rotor speed. Bossanyi [17] has proposed introducing an additional term to Equation 03 that is proportional to the rotor acceleration. This term is shown to reduce the effective inertia of the rotor, thus allowing the rotor speed to respond more quickly to changes in the wind. Leithead et. al., and B. Boukhezzar et. al. have proposed estimating the wind speed and the aerodynamic torque respectively, in order to control the wind turbine rotor closer to the C_{Pmax}.

A number of other approaches are presented in bibliography, for example in [20] and [21]. All these approaches share the same basic goal, i.e. to maintain the wind turbine operating at the maximum aerodynamic efficiency (at C_{Pmax}), under real non steady state conditions. Examples found in the referenced bibliography include controllers based on the Maximum Power Point Tracking (MPPT) methodology, PI based control and others. Some approaches, such as Linear Quadratic Gaussian (LQG) approach further discussed in the literature review section of this thesis, apart from this single main goal, also try to alleviate fatigue loads.

Irrespective of the strategy used to control the wind turbine in the below rated power region, the wind turbine cannot be operated at its maximum aerodynamic efficiency in the whole region, not even under the assumption of steady state conditions.

In most wind turbines, close to the cut-in wind speed, the generator reaches its minimum operating speed and thus the tip speed ratio cannot be maintained at its optimum value. There, the controller tries to control the rotor torque in order to maintain the wind turbine’s operation at a minimum rotor speed set-point (set close to, but above, the minimum generator operating speed limit).

Close to the rated wind speed, tracking C_{Pmax} may produce unacceptably high thrust loads. Also, by maintaining the optimum tip speed ratio at high wind speeds, even before reaching the rated wind speed, the tips of the blades are usually moving at high speeds producing unacceptably high aerodynamic noise. Thus, the maximum allowable rotor speed is reached at a relatively low wind speed. As the wind speed increases further the controller increases the torque demand trying at the same time to maintain the maximum allowable rotor speed. This continues up to the rated power of the wind turbine.

Figure 7 below, shows the above described control strategy:

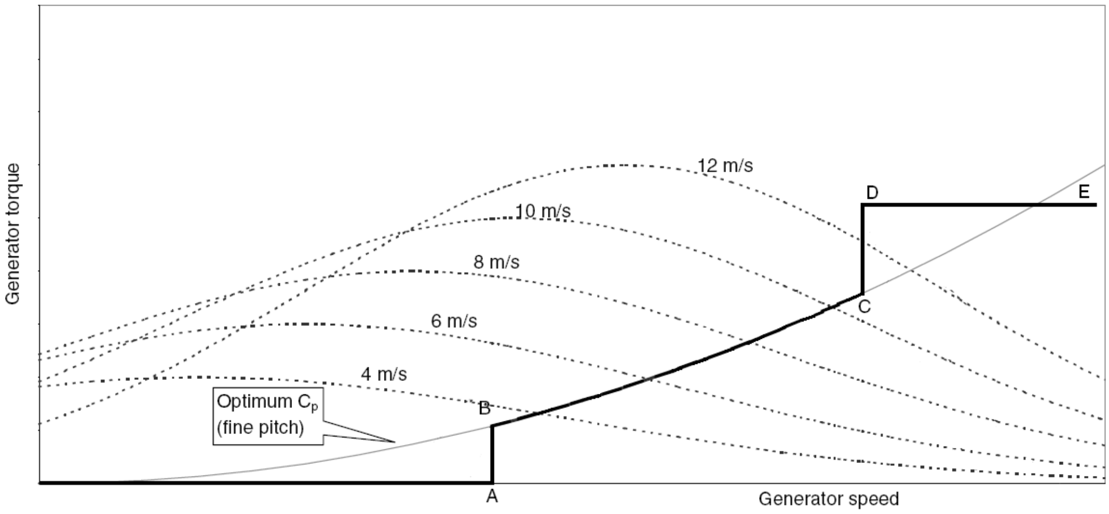


Figure 7: Generator Torque vs Speed curve for variable-speed operation. Reproduced from [22], with minor amendments.

The doted lines in the above figure show the generator torque versus generator speed characteristic of a typical wind turbine at specific wind speeds. The solid grey line shows the C_{Pmax} curve at optimum pitch, which if followed would lead to the maximum possible energy capture. It should be noted that as the above shown diagram is a torque versus speed diagram, the maximum power coefficient line does not cross the constant wind speed lines at their maxima (as it happens if plotted on a power versus speed diagram), but rather at a point further to the right of these maxima. As explained earlier, no torque is applied before point A which is the lowest generator speed control point. As the cut-in wind speed is reached and the wind turbine starts operating, the generator torque is allowed to vary up to point B. This is usually done via a PI controller. After point B and until point C, the maximum power coefficient is tracked, using one of the methodologies previously described. At point C, where the maximum allowable generator rotor speed is reached, the generator torque is varied at constant speed, again usually via a PI controller, up until point D. At point D, the wind turbine reaches the rated power, and the control algorithm switches to the above rated power operation algorithm.

Operation above rated power

In the above rated region, the wind turbine operates with a constant torque demand (line D-E in Figure 7), and is controlled via the blade pitch to a constant rotor speed set point, thus achieving constant power output.

As shown in Figure 6 in the previous section, by increasing the pitch angle of the blades away from the optimum position, the power coefficient decreases. This happens because the angle of attack is lowered and both lift and drag forces are lowered reducing the aerodynamic torque. This behaviour is used by the wind turbine controller in order to regulate the rotor speed at the required level, thus producing constant power.

It is also possible to pitch the blades towards stall, i.e. pitch them in the opposite direction, in order to increase the angle of attack even more and reach stall. Being stalled, the blades exhibit lower lift forces thus reducing the aerodynamic torque on the rotor. This allows for the regulation of the rotor speed via pitch control towards stall. It must be noted that whilst stalled, the blades exhibit much larger drag forces thus increasing rotor thrust, and are less aerodynamically damped. On the positive side, the lift forces vary less with the change in the angle of attack in the stalled region. This results in reduced control action requirements and lower aerodynamic load fluctuations, with an associated positive effect on the reduction of fatigue damage to the wind turbine components. Finally, it should be noted that most wind turbine models sold today feature a pitch to feather control methodology, suggesting that for most manufacturers the lower thrust loads and higher aerodynamic damping offered by this solution outweigh the benefits of the pitch to stall control methodology.

As the wind speed changes, the rotor accelerates or decelerates depending on the wind change. The controller of the wind turbine detects the resulting change in rotor speed and commands the appropriate pitching action in order to bring the rotor back to the required rotor speed set point. This is usually performed by a PI controller, but other controller types, such as the ones discussed in the literature review section of this thesis, have also been developed and are used on commercial wind turbines to achieve this speed regulation via pitch control.

As the rotor inertia of the utility-scale wind turbines is large, the rapid wind velocity fluctuations that naturally occur are filtered by the large rotor inertia that prohibits the rotor speed from changing rapidly. This makes it possible to control the rotor speed via pitching the blades, without excessive control action.

In some cases, the torque is allowed to slightly change (typically up to 3%) in order to track constant power more effectively. In this case, line D-E in Figure 7 would be slightly inclined. As the rotor speed accelerates after a wind gust (the operating point moves towards point E), and before the pitch controller is able to bring the rotor speed back to the desired rotor speed, the generator torque is decreased in a proportional manner to the speed increase, thus maintaining the power constant. Similarly when the rotor speed decelerates (the operating point moves towards point D), the torque is increased in a proportional manner to the rotor speed decrease, and the power remains constant.

After this brief review of the wind turbine types available and the control strategy employed in a variable-speed wind turbine, a brief overview of the drive-train of a wind turbine is presented in the next section.

1.4 The Wind Turbine Drive-Train

One of the most important mechanical moving parts of a wind turbine is the drive-train. A conventional drive-train for an upwind wind turbine consists of a series of components, described below.

Figure 8 shows these components:

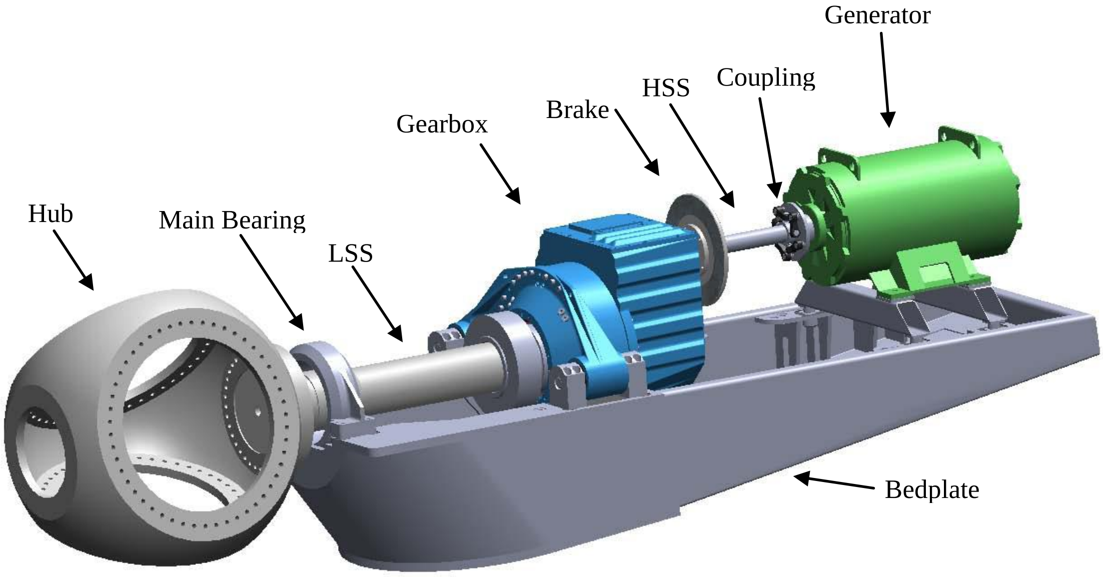


Figure 8: Conventional drive-train layout of a wind turbine, partly reproduced from [23].

Describing the components starting upwind we first encounter the low-speed shaft (LSS) that connects the rotor hub to the low-speed stage of the gearbox.

The LSS is supported by a double spherical roller bearing, called the main bearing, on the upwind side and the LSS gearbox bearing on the downwind side. The main bearing is a large and expensive component that acts as a locating bearing, holding the LSS in place, and counteracting the aerodynamic thrust load, transferring it from the rotor to the nacelle’s bedplate.

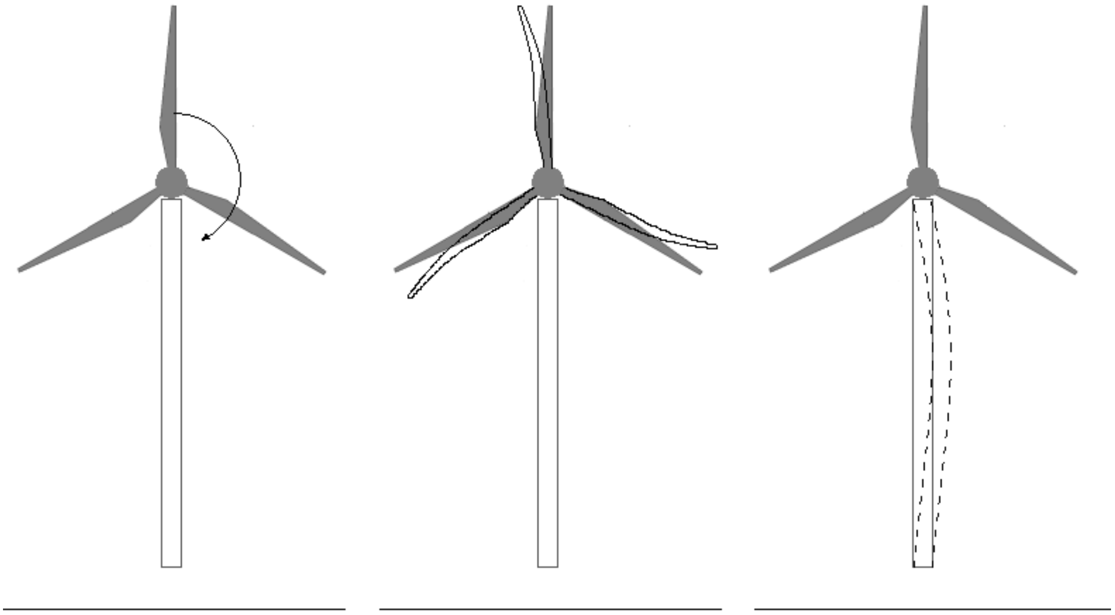
The next component is the gearbox. The gearbox transforms the high torque low speed mechanical power of the rotor, to high speed mechanical power so that a conventional generator can transform it to electricity. A number of elastic bushings connect the gearbox to the nacelle bedplate. The bushings are elastic in order to allow for small misalignment of the LSS, thermal expansion of the LSS, and to damp vibrations in the drive-train, thus reducing fatigue loads and noise.

At the downwind side of the gearbox, the high-speed shaft (HSS) transmits the available power from the gearbox to the generator. A mechanical brake is usually installed on the gearbox side of the HSS, and a flexible coupling is installed on the other side of the HSS. The flexible coupling is installed in order to allow for small misalignment between the generator and the gearbox, and to act as a mechanical 'fuse', i.e. a weak point, that will break after an abrupt mechanical failure of the generator or the gearbox. If such a flexible coupling did not exist, the abrupt mechanical failure of one of these two components would frequently lead to the mechanical failure of the other component as well.

The drive-train structure as described above is not isolated from its environment, but rather is an integral part of the wind turbine, interacting with the rotor, the generator, and the nacelle. The nacelle on the other hand interacts with the tower, which in turn interacts with the foundation. Thus, in order to describe the dynamics of the drive-train, the whole wind turbine dynamics must be taken into account.

The oscillations of the drive-train are governed by the drive-train torsional degree-of-freedom. As explained above, the drive-train structure is not isolated from its environment and so this torsional motion of the drive-train is coupled to other wind turbine component motions.

The oscillations of the drive-train in the torsional degree-of-freedom are predominantly affected by three oscillation modes: a) the oscillation of the uncoupled drive-train itself in the torsional degree-of-freedom, i.e. the drive-train torsional mode b) the oscillation of the rotor blades in the in-plane direction (also referred to as the edgewise direction), and more specifically the first rotor in-plane collective mode and c) to the tower's side-to-side motion, and more specifically the second tower side-to-side mode. This is described in bibliography [16,38], and was also confirmed by the simulations in this research. These modes are displayed graphically in Figure 9 presented in the next page:



(a) Drive-train torsion (b) 1st rotor in-plane collective (c) 2nd tower side-to-side

Figure 9: Wind turbine modes affecting the torsional drive-train oscillations

Representing the drive-train by a mass-spring model facilitates the understanding of the modes, and clearly shows how the previously introduced physical components contribute to the oscillatory modes of the drive-train. Considering solely the effect of the dynamics of a simplified drive-train, and assuming a rigid rotor and tower we can represent the wind turbine drive-train by a two-mass model:

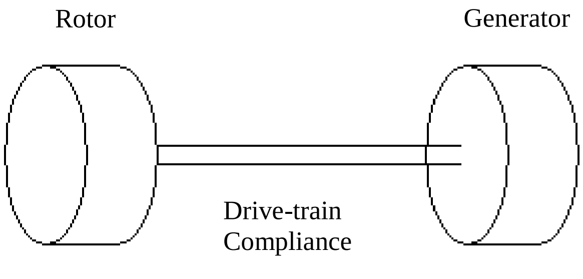


Figure 10: Wind turbine drive-train two mass model

The mass labelled ‘Rotor’ in Figure 10 includes the blades, the hub and the low speed shaft. The mass labelled ‘Generator’ includes the gearbox, the high speed shaft, the brake disk, the coupling and the generator rotor. The spring element ‘Drive-train Compliance’ represents the elasticity between the ‘Rotor’ and ‘Generator’ components. This simplified wind turbine drive-train system has two degrees of freedom and one oscillatory mode. It must be noted at this point that this is not the only way to structure a two mass model representation of the wind turbine drive-train. One could opt for incorporating different components of the drive-train to either of the two masses. For example, if the high speed shaft and coupling assembly had a high compliance, one could opt for including the gearbox, coupling and high speed shaft into the 'Rotor' mass. The most appropriate choice depends on the relative compliance of the various components in the drive-train. Also, it should be noted that in these simplified mass model representations presented in this section of the thesis, only the system dynamics in the rotational sense are considered.

Considering a more realistic flexible rotor, whereby the blades are considered as a single mass element separated from the hub by an elastic component, the two mass model is converted to a more accurate three mass model shown in Figure 11 below:

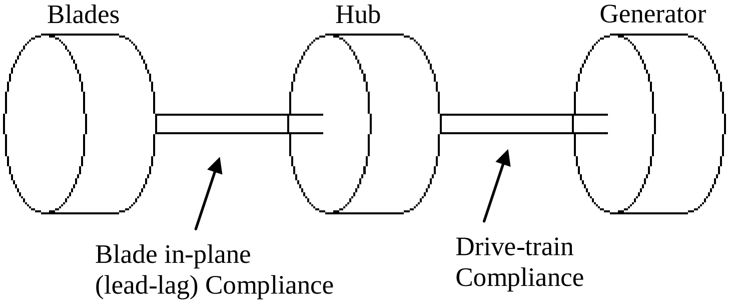


Figure 11: Wind turbine drive-train three mass model

This simplified three mass model representation of the wind turbine drive-train has three degrees of freedom and two oscillatory modes. The first mode, i.e. the low frequency mode, is created when both two adjacent masses, i.e. the 'Hub' and 'Generator' or the 'Hub' and 'Blades' masses, move together and in the opposite rotational sense to the movement of the third mass. The inertial and stiffness properties of the system define which of the adjacent masses will move together. For example, if the 'Blades' mass inertia is higher than the inertia of the 'Generator' mass – a realistic assumption for large scale variable speed wind turbines –, and assuming that the blade in-plane compliance is equal to the drive-train compliance, then the 'Hub' mass would be moving in the same sense as the 'Generator" mass. The second mode is created when both the ‘Blades’ and the ‘Generator’ masses move together in the opposite direction to the movement of the ‘Hub’.

For wind turbines with flexible towers and considerable tower top rotation, this motion also feeds into the drive-train torsional dynamics, mainly through the gearbox and generator mounting reaction torques. A four mass model representation can be constructed for such a wind turbine drive-train, an example of which is shown in Figure 12 below:

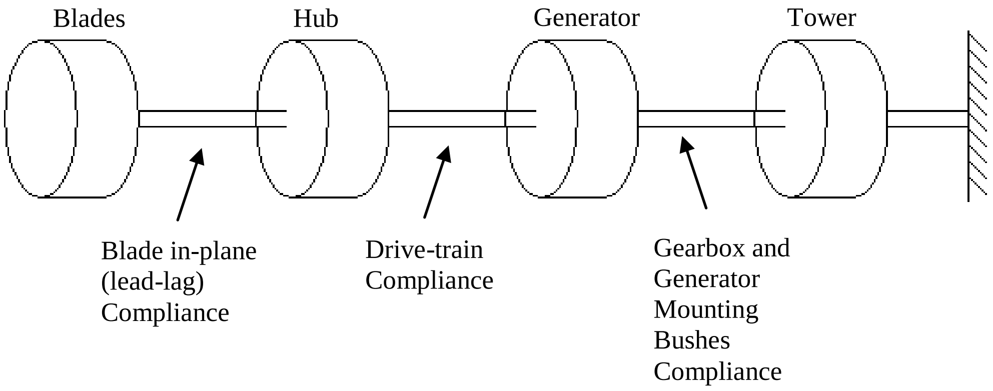

Blades
Hub
Generator
Tower
Blade in-plane (lead-lag) Compliance
Drive-train Compliance
Gearbox and Generator Mounting Bushes Compliance

Figure 12: Wind turbine drive-train four mass model

This simplified four mass model representation of the wind turbine drive-train has four degrees of freedom and three oscillatory modes. The first mode of the system is created when all masses move in the same direction. As for the two higher modes, inertial and stiffness properties of the drive-train components determine which of the masses will rotate in the same direction. For the interested reader, the methodology to calculate the modal properties of such a four degree of freedom system, and in fact any multi-degree of freedom system, can be found in [24]. This methodology is fairly easy to implement if one has access to an algebra analysis software package (for example the open-source wxMaxima software package available at http://andrejv.github.com/wxmaxima/).

Note that in order to keep all the above spring-mass model representations as simple as possible, the connection of the generator to the electrical grid is not shown. This is an acceptable simplification since for a variable speed wind turbine this connection has very high compliance [27]. Other simplified spring-mass model representations of a wind turbine drive-train can also be found in bibliography, for example in [25,26,33].

As previously described, in a VSWT, the generator torque demand is varied in order to control the rotor speed of the turbine in the region below rated. In the above rated region, the generator torque is maintained constant. In some cases, the torque is allowed to slightly change to maintain constant power. As the rotor speed slightly increases, the torque is decreased in a proportional manner, and thus the power remains constant. Similarly when the rotor speed slightly decreases, the torque is increased in a proportional manner, and the power remains constant again.

In a fixed-speed wind turbine that uses an induction generator, the generator torque vs. generator speed relationship is a steep curve, with the generator torque increasing substantially when the generator speed increases slightly. Figure 13 shown below shows the torque-speed characteristic of the induction generator:

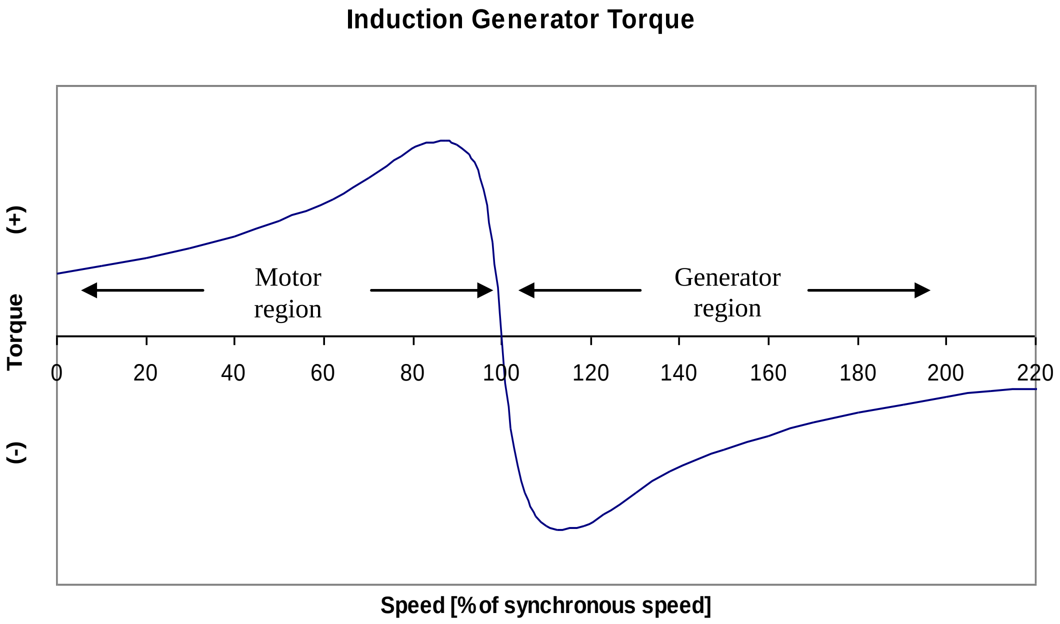


Figure 13: Induction generator torque vs. speed curve.

Note that in the region close to the synchronous speed, where the generator is operating under normal operation, the generator toque vs. generator speed relationship is characterised by a steep curve. This inherent behaviour of the induction generator not only maintains the rotor of the turbine at a constant speed, but provides considerable damping to the drive-train torsional modes. In a VSWT on the other hand, this beneficial inherent damping is absent.

As with the fixed-speed wind turbines described above, for a VSWT in the below rated region, the generator torque changes as the generator speed changes. However, the generator torque vs. speed relationship is far less steep. This means that for a small change in speed there is a relatively small change in torque. Thus the additional drive-train damping inherently provided by the generator torque-speed characteristic of a VSWT in the below rated region is far smaller compared to the damping provided by an induction generator based fixed-speed wind turbine.

In the region above rated, the generator torque of a VSWT provides no damping to the drive-train. This happens because the generator torque in this region is, as explained in the previous section, constant. Moreover, in the case where the VSWT controller slightly changes the generator torque in order to keep the power constant it is affecting the damping of the system negatively thus destabilising the system even further. Thus in a VSWT there is a necessity to damp the drive-train torsional modes in some way.

Mechanical components that increase the drive-train damping can be designed and constructed but add to the complexity and cost of the wind turbine. An alternative to such components is the Drive-Train Damper (DTD). The DTD is a filter applied to the generator torque demand that, in effect, adds a small ripple in the torque at the drive-train frequency and at such a phase that counteracts the effects of resonance [27]. Figure 14 shown below shows the effect of adding a DTD on a 600 kW VSWT:

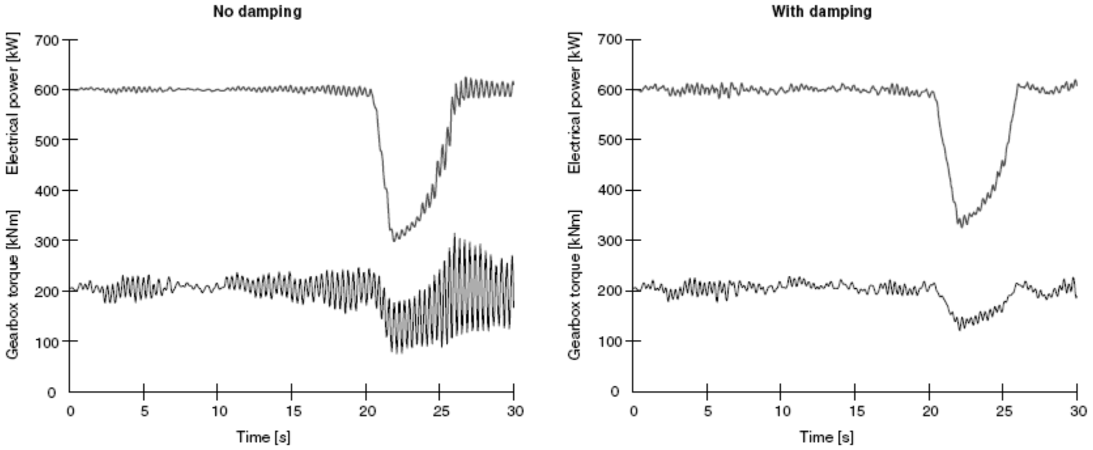


Figure 14: Effect of a drive-train damping filter reproduced from [22].

Figure 14 shows simulation results for a 600 kW VSWT operating without (left graph) and with (right graph) a DTD. On the left graph, a large drive-train resonance can be seen. Although the power is relatively smooth, the gearbox torque shows large fluctuations that would most probably produce excessive fatigue damage that would considerably lower the drive-train's expected lifetime. The effect of introducing a damping filter as described above is shown in the right graph. The result shows that the DTD effectively damps out the resonance without increasing the electrical power variations. This is because the torque ripple needed to damp the resonance is actually relatively small, ranging from very low values up to 8% of rated torque. Such a torque ripple, introduced by the DTD of a 750 kW wind turbine is shown in Figure 15 below:

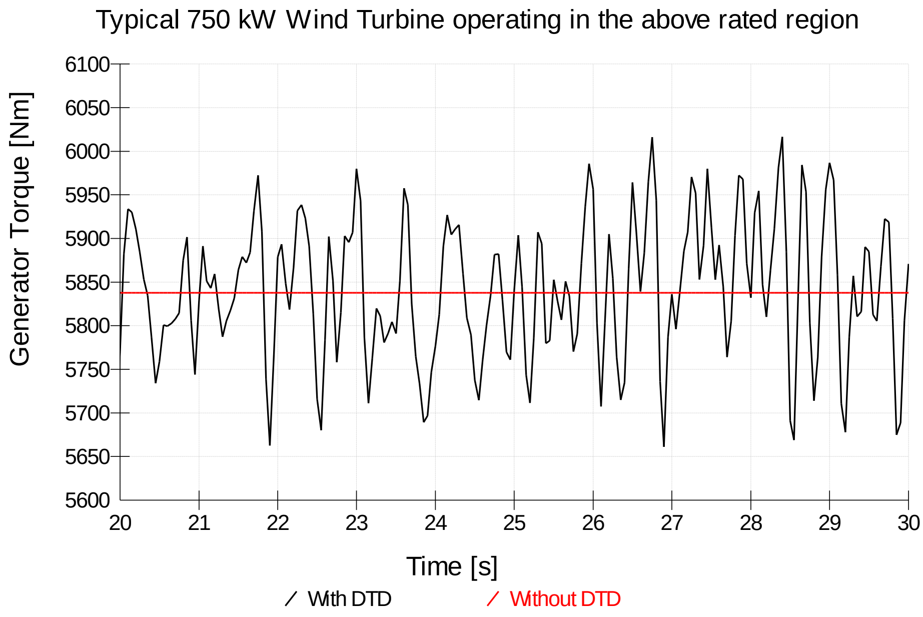


Figure 15: Drive-train damper's torque ripple

A more detailed analysis of the effects of a drive-train on the wind turbine loads and power quality is presented in the final results section of this thesis. Relevant work already undertaken by other researchers, along with the motivation for this research is presented in the next chapter. In chapter three, the theoretical basis on which all calculations were performed and the wind turbine models used are introduced. In chapter four, the methodology created by the author's research is presented and analysed. Some other methodologies also investigated are also introduced. In chapter five, the results obtained by the methodology created by the author’s research are analysed. Finally, in chapter six a summary of the results is submitted, underlining the conclusions of this thesis, along with a discussion of further work that could be performed by other researchers in this field.

Chapter 2

A literature review of relevant research in the control strategy used to damp the drive-train resonances is presented in this chapter. Some of the interesting findings and shortcomings of the existing research are highlighted. Finally, the motives that led to this research and the objectives of this research are also presented in this chapter.

2.1 Literature Review

Large wind turbines built in the 70s and 80s were fixed-speed with stiff drive-trains and high inertia rotors. According to Alan D. Wright [28], wind turbulence easily excited these machines' first drive-train torsional mode. As some of these machines used synchronous generators, the beneficial damping provided by the induction generator, as mentioned in the previous section of this report, was not realised. Thus control engineers of that era had to design pitch controllers that not only regulated power, but also damped the drive-train [27,29,30,31,32]. An example of such a control implementation was that of Wasynczuk et al. who investigated the application of such a control strategy on the experimental MOD-2 wind turbine in the 1980s [33]. However, controlling pitch in order to reduce drive-train load fluctuations was performed at the expense of power extraction efficiency. Thus, other control engineers, in their effort to maximize power extraction, designed pitch controllers to maximize power extraction, but added mechanical components to the gearbox in order to increase the drive-train damping. An example is the spring and dashpot mechanism attached to the gearbox as described in [34].

More recently and with VSWTs, new ways to damp drive-train torsion vibrations had to be devised. There is a rich literature showing that considerable effort has been expended in this field. An overview of recent efforts to damp drive-train torsion vibrations is presented in this chapter.

One of the most commonly used and referenced ways to damp drive-train torsion vibrations is achieved by using the torque demand signal [22, 28,35]. According to Ervin A. Bossanyi [22], this has been successfully implemented on many turbines in the following way:

In order to damp the drive-train, a filter is added to the generator torque demand control loop. This filter works by adding a small ripple in the torque demand at the drive-train frequency and at such a phase so as to counteract the effects of resonance. Bossanyi [16,22,36] suggests the use of a band-pass filter of the form:

$$H(s) = G\frac{2\zeta\omega s(1+s\tau)}{s^2+2\zeta\omega s+\omega^2} \qquad \textbf{Eq. 04}$$

Where: G = gain, ζ = damping, ω = frequency and τ = time constant.

Dixit and Suryanarayanan [37] have also used the band-pass filter proposed by Bossanyi, and explored the benefits of scheduling, i.e. changing, the filter parameters according to the blade pitch angle. As discussed in the previous chapter, the main contribution of one of the coupled natural frequencies of the drive-train is that of the motion related to the uncoupled first collective in-plane rotor mode. As the blade pitch angle increases, the frequency of the rotor in the in-plane direction is lowered. This happens because the shape of the blades is such that the blades are stiffer in the edgewise direction than in the flap-wise direction. Thus the higher the blade pitch angle is, the lower the stiffness of the rotor in the in-plane direction becomes. The approach proposed by Dixit and Suryanarayanan is simple to implement, and seems to lower the drive-train loads. One unexplored effect was the transition between the different band pass filters as the pitch angle changes, and the transient phenomena this generates. As this transition would occur very frequently during the wind turbine operation, it is important to validate these transient phenomena before arriving to the final implementation of this strategy.

Van Engelen, et al., [38] have shown that a slightly different control loop from the generator speed to the generator torque can be used to damp the drive-train. They propose a state-variable controller that uses a low order Kalman filter for state feedback, in conjunction with a band pass filter. The use of such type of control has been also proposed by other researchers for the control of wind turbines, most notably Bossanyi [39]. Here, instead of using directly the generator speed to produce the necessary control signals to damp the drive-train, the researchers use the vibrations in the modes affecting the drive-train. Since these vibrations cannot be measured with the appropriate accuracy and reliability on real wind turbines, the researchers estimate these signals (state estimation) based on a linear state-space model representation. Once the signals have been estimated, a band pass filter is used to produce the necessary generator torque signal that damps the drive-train in a similar concept to the band-pass filter described above. The results are very interesting as they show that the controller proposed by the researchers has a positive effect in alleviating the loads when compared to a single band-pass filter without using state estimation. Unfortunately, this research seems to be confined to the use of a single wind turbine model for validation purposes. Moreover, the results obtained by the researchers when they are using a single band pass filter DTD without state estimation, show that this filter is not capable of effectively damping more than one coupled drive-train frequency. In practice, however, a well-tuned single band-pass filter can usually damp effectively all the main coupled drive-train natural frequencies, as will be later verified by the results of this research. Thus, the positive effect of the proposed controller might have been exaggerated by the use of a particular wind turbine model, where the use of a single band-pass filter as a DTD cannot produce good results. Nevertheless, the results are very interesting and warrant further research and validation of this drive-train damping methodology.

The above described DTDs are based on adding a control loop in series to the main PI(D) control loop used for the wind turbine power and speed regulation. Other researchers in the past few years have designed more modern controllers that provide for good power and speed regulation and inherent drive-train damping capabilities. This research is based on wind turbines controlled by the classical PI controller. The reason behind this is that this controller is still the most widely used controller and it has been proven to be robust and reliable during its many years of use in the majority of the wind turbines operating to date. Moreover, at the beginning of this research, there was no publicly available information regarding the successful use of any other controller in commercial utility-scale wind turbines. However, as the years have progressed since the beginning of this research, more and more researchers have been focusing on the creation of different types of main wind turbine control algorithms, and some of these control algorithms have been successfully implemented on a number of commercial utility-scale wind turbines. As these control algorithms have shown good power and speed control capabilities, as well as load reductions to the gearbox, they are also presented in this section. The fact that this research is based on wind turbines controlled by a PI controller does not necessarily exclude the use of the DTD developed by this research in conjunction with other types of controllers. Of course, however, the interaction of the DTD developed by this research and the controller regulating pitch and torque demand must be thoroughly investigated.

Many researchers have used Linear Quadratic Regulator (LQR) controllers to control both fixed-speed and variable-speed generators. Some examples include Mattson [29], Liebst [40], Muhando, et al. [41], Munteanu et. al. [42], and others.

The Linear Quadratic Regulator (LQR), in it simplest form, is a controller that not only tries to minimize, or track, a certain system output, but at the same time tries to minimize the control signal itself. On the one hand, in order to minimize, or track, a certain control system output, a large control signal is required. On the other hand, a small control signal will lead to large deviations from the output target. Thus, a trade-off between these two goals is established by a cost function weighting the importance of the two goals. In a similar manner, a LQR can be designed with many control targets. Moreover, a LQR can be used for multiple-input/multiple-output processes for which classical designs such as the PI(D) are difficult to apply, and is thus perfectly suited for use as a wind turbine controller where both pitch angle and torque are controlled, and multiple control goals are requested.

In its most robust form, the LQR uses the whole state of the system as its control input. In order to do this, the whole state of the system should be measurable. Such a controller is called a state feedback LQR. In practice, measuring all the system states of a real system is usually not possible – at least not with the required accuracy and reliability. To overcome this, a single (or more) measured system output(s), for example the generator speed, can be used to estimate the states of the system. The most usual practise is to use a Kalman filter to perform such estimation. The basis of this method is to assume that both the disturbances and the noise on the measured signal are uncorrelated from each other and are zero-mean Gaussian noise processes. Combining a Kalman filter based estimation method for estimating the system states and a state feedback LQR thus seems to be a reasonable approach. The resulting controller is called a Linear Quadratic Gaussian (LQG) controller, owing its name to the Gaussian noise characteristics assumed for disturbances and noise on the measured signal. The schematic in Figure 16 on the next page shows the implementation of such a LQG controller for minimizing system output:

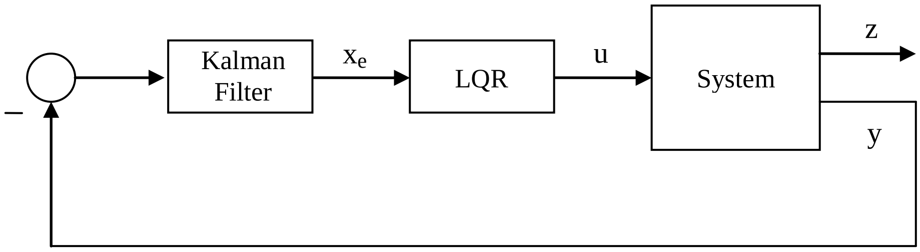


Figure 16: LQG output feedback controller

Where,

y is the system measurable control output,

z is the system output,

Kalman Filter, is the Kalman filter based state estimator,

x_e is the estimated system state array

LQR is a state feedback LQR, and

u is the control signal.

When the control goal is to maintain the system output to a desired set-point, instead of its minimization to zero, a very similar implementation is used. The difference is that the set point is added to the input of the LQG system estimator and the input to the system by using the appropriate gains.

The first attempts to use such controllers were made as early as the fixed-speed wind turbine era. As an example, Mattson in 1984 [29], used a state estimator in combination with a LQR in order to regulate power for a fixed-speed machine using blade pitch. The developed controller was not only designed to regulate power, but also reduce drive-train loading by adding damping to the first drive-train torsional mode.

Later, Knudsen et al. [43], used a H_∞ controller for the same purpose on a grid connected 400kW wind turbine. The LQG / LQR controllers are not necessarily robust. This means that they can be sensitive to errors in the turbine model, or to excessive noise in the control signals. To overcome this, some researchers, like Knudsen et al., have tested the use of a similar approach that is less sensitive to model errors. The H_∞ controller can take the uncertainties in the turbine and wind model explicitly into account. The H_∞ controller is a filter based controller that, unlike the Kalman filter is not based on the assumption of Gaussian noise disturbances and measured signal noise. The results of using this controller showed a reduction in pitch activity and reduced fatigue loads compared to a simple PI controller.

Another interesting implementation of the LQG controller is an earlier attempt of Muhando et al. [41] in 2002, who have shown the positive effects of including a neural network (NN) controller in parallel to an LQG based controller. The researchers have developed an LQG controller for controlling both pitch angle and generator torque of a wind turbine in the above rated wind speed region. The objective is to limit the power extracted by the wind and thus not exceed the operation limits of the wind turbine, but also try to do this in such a way as to maximise the energy capture with respect to a normal PI controller. Since the proposed LQG controller can track changes in wind more quickly, it works with positive energy capture results. For the same reason, however, i.e. due to the excessive controlling actions, the loads of the drive-train are higher. A neural network controller has been used in parallel with the LQG controller to smooth out the control actions of the latter, and thus alleviate the drive-train loads. Unfortunately, a very simple drive-train model is used in this research whereby the coupled drive-train modes created from the tower and rotor structures' interaction are disregarded. Finally, as there is no comparison of the drive-train loads achieved by the use of the proposed controller with those achieved by a conventional PI controller and DTD setup, one cannot assess the effectiveness of the proposed controller in reducing drive-train loads. Still, as in the case of the research by Wright et al., the results are promising and warrant more investigation into these controllers.

Lescher et al. [44,45], have developed a multivariable Linear Parameter Varying (LPV) dynamic output-feedback controller that changes the pitch angle of the blades and the generator torque in order to control the wind turbine throughout its operating range. Among its main control objectives, i.e. to maximise power extraction without exceeding maximum power and rotor speed limits, one of the control objectives of the proposed controller was to reduce the drive-train load fluctuations. This is done by penalizing high frequency torque fluctuations in the LPV controller during controller synthesis. In order to evaluate the proposed controller, Lescher et al. compare the results obtained by this controller with the results obtained by a PI controller and a LQG controller, on a simulation run at above rated wind speed. The results show that the LQG controller shows a reduction in the drive-train mechanical fatigue of 41% when compared to the PI controller. The LPV shows a reduction of 43% when compared to the PI controller. Moreover, the LPV controller shows similar control behaviour to the PI controller. They are both much smoother in their controlling actions when compared to the LQG controller that exhibits excessive blade pitch controlling actions. Although these results seem very promising, the PI controller implemented by the researchers does not use any kind of DTD. As was previously mentioned, it is a common practice to use a single band pass filter as a DTD when using a PI controller. Thus the results presented by the researchers, unfortunately give little insight as to whether the LQG controller and the proposed LPV controller reduce the drive-train mechanical fatigue in comparison to a PI controller with a DTD acting on the generator torque demand signal. Moreover, it is not possible to even draw firm conclusions on the comparison of the drive-train load reduction capability of the LPV and the LQG controllers. This is because the load reduction capabilities seem similar, and the results presented in the research paper are obtained by a single experiment, using a single wind turbine model with a simple drive-train model.

Finally, another research activity worth noting is that of Wright et al. [35], who in 2006 implemented a state-space controller using Disturbance Accommodation Control (DAC) theory [46,47] in the above rated region to control both pitch angle and torque. The goal of the DAC based torque controller was to minimize drive-train loading, whilst the goal of the pitch controller was to maintain the rotor speed within the predefined limits, thus regulating power. A state-space controller using Disturbance Tracking Control (DTC) theory [48,49] was used to control torque in the below rated region in order to maximize power extraction.

In their usual implementation, state-space controllers used in wind turbine control employ state estimation based on the generator rotational speed as an input in order to estimate unmeasured plant states. The state-space controller, based on the estimated states, produces the desired control signal. The basic principle of both DAC and DTC theory state-space controllers is the expansion of the state-estimator for the turbine with additional states used to estimate the wind-speed disturbances. In the DAC based controller, these states are used with the appropriate wind disturbance gain in order to minimize the effect of wind-speed disturbances in the above rated region. Similarly, in the DTC based controllers the estimated wind speed is again used, this time however, in order to maintain a constant tip-speed ratio.

The proposed controllers were implemented in the National Renewable Energy Laboratory's (NREL) Controls Advanced Research Turbine (CART). The CART is a 2-bladed 600kW Variable-speed Wind Turbine (VSWT), with variable pitch control and teetered hub. The results of implementing this controller were compared to those obtained when using a simple PI controller and the comparison showed a considerable reduction in the drive-train loads. In a specific 300 second subset of experimental data, where the wind turbine was operating mainly in the above rated region, the DAC based controller achieved a 51% reduction of fatigue loads on the low speed shaft (or more accurately a 51% reduction of damage equivalent load on the low speed shaft torque). Unfortunately, as with the case of the Lescher et al. controller tests, the comparison was made with a simple PI controller that did not use a DTD [[50]] and thus the results on their own remain inconclusive.

It is worth also noting that 2 years before, in 2004, Wright et al. [[51]] used a Disturbance Accommodation Control (DAC) theory based state-space controller to regulate pitch in the above rated operation region, whilst keeping the generator torque demand constant. This time the control objective was not only to maintain the rotor speed and thus producing power to the required set point, but also to alleviate the drive-train load fluctuations. Again, the results seem promising, as they showed a reduction in loads when compared to the simple PI pitch controller, while maintaining similar load following and pitch actuator duty characteristics. Based on what is presented separately in [[35]] and [[51]], it is worth noting that the drive-train fatigue load reduction of this methodology is similar to that achieved by the methodology developed by Wright et al. in 2006.

The very promising results of Wright et al., will hopefully lead other researchers to investigate this type of state-space controller implementation on wind turbines even more in the coming years, and try to prove its commercial feasibility by test implementation in commercial wind turbines and a comparison of the proposed controller with other types of controllers.

2.2 Motivation and Objectives for this Research

The research on a self-tuned drive-train damper was motivated by two major reasons. The first reason comes from the fact that drive-train failure is still one of the biggest problems that affects the wind turbine industry [52 53 54 55]. Replacing a gearbox is associated with prohibitive operational expenses (OPEX) and revenue loss due to wind turbine downtime. Not only is replacing a gearbox very troublesome and expensive, but also no single definitive cause or solution for gearbox failures in wind turbines has been identified (although a few have been explored [52]).

Creating a code that could be programmed into the existing controller of any large VSWT in order to reduce the fatigue loads on the gearbox is thus of great importance to wind turbine operators and the wind turbine industry in general.

The second reason derives from the fact that when a wind turbine is sold in large numbers, and for a succession of years, some components will inevitably have to be changed over the years of manufacture. If any major part of the turbine is changed, the whole wind turbine design is checked and the wind turbine controller is re-tuned (if needed). If, however, a component such as the generator or gearbox is changed using a similar part (say due to a change in part supplier), it sometimes does not make economic sense to re-tune the controller and have a multiple of controllers for the same turbine model depending on the parts each turbine uses. Also there are always components in a wind turbine that are in practice difficult to model in detail, and their interaction with the other components is not modelled with a sufficient accuracy in the existing structural models used for system design.

Following the above argument, the existing structural models cannot be used to perfectly model wind turbines at the design stage. This calls for a need of a controller (or part of a controller in this case) that can tune itself based on data collected through an experimental procedure run on the actual wind turbine and does not rely on the existing structural models used for system design.

The objective of the research undertaken by the author of this thesis is to create a DTD that can autonomously and automatically tune itself. This is a completely novel approach that has not yet been investigated by any other researcher to the author's knowledge. By tuning itself on the actual wind turbine, the effectiveness of the DTD will depend neither on how well the various components of the wind turbine were modelled, nor on the sophistication of the turbine simulator used. By tuning itself autonomously, the DTD's effectiveness will not depend on the skills of the wind turbine installer, nor will it need any adaptation for the specific turbine on which the code is being installed.

As will become apparent to the reader of this thesis, the development of the methodology that is used in order to create the self-tuned DTD, was result-oriented. The author's goal was to develop the best possible methodology in order to solve the problem at hand. This meant that the development was based on the use of available computational tools and not on new wind turbine or drive-train models created for the purpose of this research. The reasons behind this decision are analysed in more detail in the next chapter. Having said that, it must be stressed that good understanding of the basic physical principles involved was of utmost importance in the development of the self-tuned DTD and imperative for the analysis of the results.

Chapter 3

The simulation methodology and wind turbine models used for the development of the self-tuned drive-train damper are presented in this chapter. First, the theory based on which the simulations are run is demonstrated, analysing all the aerodynamic and structural considerations taken into account in these simulations. For the purposes of this research a number of wind turbine models, ranging in size and structural characteristics, were developed. The properties of these wind turbine models are presented and explored.

3.1 Theoretical Background

In order to develop and test the self-tuning DTD procedure, a number of VSWT models were created. The models are implemented in, and used with, GL-Garrad Hassan's "Bladed" wind turbine simulation software [56]. GH Bladed is a full aeroelastic model and is thus able to perform dynamic load calculations. Along with FLEX[57], another full aeroelastic model based software package, these two software packages are the most widely used modelling packages in the wind turbine manufacturing industry. Other software packages used for wind turbine simulation and design include Alcyone, PHATAS, HAWC, Vidyn, FAST, ADAMS/WT and DUWECS. A comparison of the first four along with GH Bladed and FLEX is given in [58]. A comparison of FAST, ADAMS/WT and GH Bladed is given in [59]. More recently, a number of investigation and verification projects of aero-elastic codes for offshore wind turbines were performed under the "Offshore Code Comparison Collaboration" (OC3) project. Under this project GH Bladed, FAST, ADAMS/WT, and various versions of HAWC and FLEX codes, were investigated and the results have been presented in various papers [60,61].

GH Bladed is a modelling suite that combines the simulation of aerodynamic loading and the structural loading of a wind turbine. The calculation of the aerodynamic loading on the blades, which is the most demanding of the aerodynamic loading calculations for a wind turbine model simulation, is based on blade element - momentum (BEM) theory.

Blade element momentum theory is based on the application of the axial and angular momentum theories on blade sections. By breaking down the rotor into a number of very small annular sections, the axial component of the aerodynamic force δF_{axial}, and the torque δQ on annular sections of N blades can be calculated from the local aerofoil's known drag and lift characteristics, as follows:

$$\delta F_{axial} = (\delta L \cos\phi + \delta D \sin\phi) N \qquad \textbf{Eq. 05}$$

and

$$\delta Q = \left[(\delta L \sin\phi + \delta D \cos\phi) r \right] N \qquad \textbf{Eq. 06}$$

Where,

δF_{axial} is the axial aerodynamic force on the blade elements of span-wise length *δr*, acting in the undisturbed wind direction (for the purposes of the BEM theory it is considered to be normal to the rotor plane),

δQ is the torque developed by the blade elements of span-wise length *δr*,

δL is the lift force on the blade element,

δD is the drag force of the blade element,

φ is the flow angle of the air with respect to the moving blade

r is the radial distance of the *δr* blade element from the centre of the rotor,

and *N* is the number of blades

From axial and angular momentum theories [16] it can be shown that:

a) The change of axial momentum in the air passing through a swept annulus with a width *δr* imparts an axial force on the *N* blades sweeping this annulus, and can be calculated as follows:

$$\delta F_{axialFromAxialMomentum} = \text{rate of change of axial momentum} = 4\pi\rho U_\infty{}^2 a(1-a) r\delta r \qquad \textbf{Eq. 07}$$

Where,

ρ is the air density,

U_∞ is the undisturbed (upwind) wind velocity,

and *a* is the axial flow induction factor (a measure of the change of axial wind velocity by the presence of the rotor, *a* = (1-U_d / U_∞) where U_d is the axial wind velocity at the rotor plane),

b) The angular moment imparted to the wake increases the kinetic energy in the wake. This kinetic energy increase has to be balanced by an additional drop in static pressure. Again, this drop of pressure imparts an additional axial force on the blades. The axial force on the *N* blades sweeping an annulus with a width *δr* can be calculated as follows:

$\delta F_{axialFromAngularMomentum} =$ pressure difference across annulus **x** annular ring area

$$\delta F_{axialFromAngularMomentum} = \frac{1}{2}\rho(2a'\Omega r)^2 2\pi r\delta r$$ **Eq. 08**

where,
a' is the tangential flow induction factor (a factor used in the calculation of the tangential wind velocity),
and *Ω* is the angular velocity of the rotor

Note that $2a'\Omega r$ is the tangential wind velocity immediately downstream of the rotor.

c) The change of angular momentum in the air passing through a swept annulus with a width *δr* imparts tangential forces on the N blades sweeping this annulus that in turn create the rotor torque *δQ*. This can be calculated as follows:

$\delta Q =$ rate of change of angular momentum $\Leftrightarrow$
$\delta Q =$ mass flow rate **x** change in tangential velocity **x** radius $\Leftrightarrow$

$$\delta Q = 2\pi r\delta r\rho U_\infty(1-a) \quad \mathbf{x} \quad 2a'\Omega r \quad \mathbf{x} \quad r$$ **Eq. 09**

Now equating Eq. 05 with Eq. 07 and Eq. 08 and equating Eq. 06 with Eq. 09, and performing some simplifications, two equations with only two unknowns (*a* and a') are finally found:

$$(1-a)^2 \frac{\Omega^2 r^2 (1+a')^2}{U_\infty{}^2} N\frac{c}{R}\left(C_l\cos\phi + C_d\sin\phi\right) = 8\pi(a(1-a)+(a'\lambda r/R)^2)r/R$$ **Eq. 10**

and

$$(1-a)^2 \frac{\Omega^2 r^2 (1+a')^2}{U_\infty{}^2} N\frac{c}{R}\left(C_l\sin\phi + C_d\cos\phi\right) = 8\pi\lambda(r/R)^2 a'(1-a)$$ **Eq. 11**

where,
c is the chord length of the blade element

R is the rotor radius

C_l and C_d are the lift and drag coefficients

φ, as before, is the flow angle of the air with respect to the moving blade, and can be shown that it is equal to $\arctan(\frac{U_\infty(1-a)}{\Omega r(1+a')})$,

λ, is the tip speed ratio, and is equal to $\Omega R / U_\infty$

The detailed derivations of these equations is not shown here, as they do not aid the comprehension of this research, nor give any insight into the way GH Bladed performs the necessary aerodynamic calculations. For the interested reader, the detailed derivation can be found in [[16], pp. 61-63]. These equations can be solved (by an iterative process) and all blade forces can thus be calculated.

As previously mentioned, GH Bladed uses BEM theory as the basis of the rotor aerodynamic calculations it performs. GH Bladed uses BEM theory but also applies a number of correction factors to it:

a) For heavily loaded wind turbines, where the blades rotate with high tip speed ratios, the axial flow induction factor *a* is high. Under such a situation, the wake becomes turbulent, and by its turbulent nature forces air from outside the wake to enter in the wake region, thus re-energizing the wake. This obviously leads to a breakdown of the BEM theory. Thus, GH Bladed applies an empirical correction factor related to axial forces when the axial flow induction factor becomes larger than 0.3539, which once again is a limit that was empirically set.

b) BEM theory is based on the approximation of a permeable ‘solid disk’ representation of the rotor, whereby the axial induction factor is azimuthally uniform. In reality there are distinct blades, and the axial induction factor as experienced by a single particle (or at a single azimuth) depends on the proximity of this particle to the blade, having a higher value close to the blade, and a lower value further away from the blade it passes. These different axial induction factors that the particles experience are more apparent near the outer parts of the blade, and lead to the azimuthally average induction factor to change radially. In order to take this into account, a so called ‘tip-loss’ factor is applied to the value of both the axial induction factor and the tangential flow induction factor. In GH Bladed, the Prandtl tip loss factor approximation is used [62].

c) As with any aerofoil, there is a net circulation around the aerofoil. This in practice means that the air particles on the top of the aerofoil move quicker than the air particles at the bottom of the aerofoil. This is achieved by the overall shape of the aerofoil, and especially by its sharp trailing edge. At the root of the blade, at the point where the hub starts, or at the point where the blade’s cross section changes from an aerofoil shape to a cylinder, the circulation drops to zero. In doing so, vorticity is shed into the wake from the trailing edge. This introduces a loss in the extracted energy, and a deviation from the BEM theory, again taken into account in GH Bladed.

For non-uniform wind, and with rotors at an angle to the flow field, the wake structure under non-steady conditions must be taken into consideration. In a time domain calculation, the blade loading changes as the incident wind velocity changes. The incident wind velocity changes, either because the free-stream wind velocity changes, or because the wind flow over the rotor area is non-uniform and as the blades sweep around the rotor area, they experience this non-uniform flow field as a change to the incident wind velocity. The non-uniform flow field, apart from the turbulent nature of the wind, is caused mainly by the wind shear profile of wind and tower shadow.

As the blade loading and the wind velocity change in the time domain, the wake structure behind the rotor and the flow field in front of the rotor do not instantaneously change. In order to explain this in simple words: an air particle 10 metres behind the rotor will not be instantaneously affected by a wind speed change at the rotor plane. BEM theory, as was previously shown, is based on some assumptions that are only valid if the flow field is steady. Thus taking into account the structure of the flow field is crucial for the correct implementation of the BEM theory in time domain simulations. GH Bladed has various models that can be used to take this into account (or disregard it if the user chooses to do so). The model used for the purposes of this research is the ‘Dynamic wake’ model, which is based on Pitt and Peters [63] theory. Although the Pitt and Peters model was developed for an actuator disk, in GH Bladed the model is applied at actuator annuli level.

The second major assumption made by GH Bladed to correct BEM theory for time domain calculations is the introduction of dynamic stall effects. In the same way that the flow field in front and behind the rotor does not instantly change with changes in wind velocity and blade loading, the flow field around the blade does not change instantly with changes of angle of attack. Thus, although we are used to thinking that the aerofoil section used in the blade's various blade segments dictates the lift and drag expected under a specific wind velocity, this is only true under steady state conditions. Under a changing angle of attack scenario, the flow field does not instantly change all over the blade vicinity. This effect is small at lower angle of attacks, where the flow field is attached to the blade, and is thus disregarded in GH Bladed.

However, when the angle of attack becomes larger, then flow separation is experienced. This flow separation is called stall, and leads to a substantial drop in lift and a steep increase in drag. Thus, the flow structure just before a specific angle of attack, called the stall angle, and after the stall angle is quite different. When the angle of attack changes from a low angle (below the static stall angle) to a large value (larger than the stall angle), a vortex is created on the upper surface of the aerofoil (low pressure side). For as long as this vortex stays on the upper surface of the aerofoil, the lift coefficient continues to rise. As soon as this vortex detaches from the trailing edge of the upper surface, the flow on the upper surface becomes turbulent, and the lift coefficient rapidly drops.

A few semi-empirical methods for predicting the dynamic stall characteristics of aerofoil sections have been developed. GH Bladed uses a model largely based on the Beddoes model [[64]], a well-known semi-empirical method used for predicting dynamic stall effects.

The structural loading on the wind turbine model is calculated by applying the aerodynamic loads found using the above-described theory, with the inertial and gravitational loadings of the various wind turbine components, into a modal model.

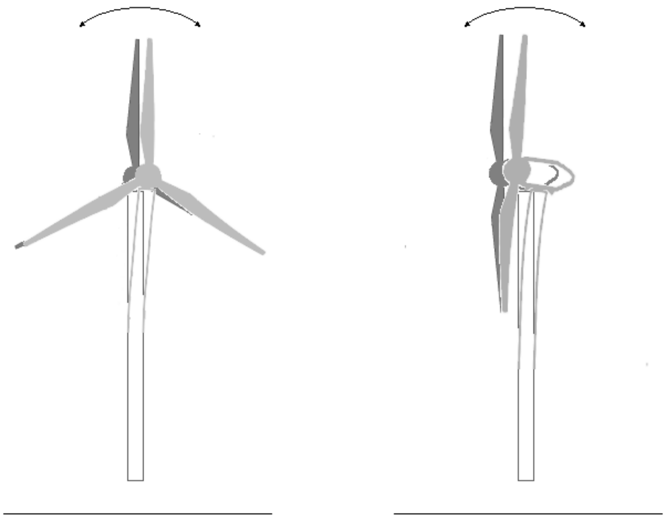
Because of the rotation of the blades of a wind turbine relative to the nacelle and tower structures, the equation of motion that describes the dynamics of the wind turbine contains terms with periodic coefficients. This means that as the blades change position, the structural dynamics of the wind turbine change. This periodicity makes the computation of the modal properties of the wind turbine as a complete structural entity not possible using a standard eigen-analysis.

GH Bladed uses a "component mode synthesis" methodology to solve this problem. Under this approach, the modal properties of the large rotating components (i.e. the rotor) and the non-rotating components (i.e. the tower) are computed independently. These modes are then coupled by an appropriate formulation of the equation of motion of the wind turbines in a forced response analysis.

The modal properties of the rotor are found by calculating all orthogonal, uncoupled 'normal' modes of the structure. The frequencies and model shapes of the rotor modes are computed from the eigen-values and the eigen-vectors of a finite element representation of the rotor. The finite element model of the rotor is based on the use of two dimensional beam elements to describe the mass and stiffness properties of the rotor blades. These can be calculated by defining the number of blades, mass distribution, bending stiffness along the blades, and twist distribution along the blades, pitch angle and rotor speed. The modal damping coefficients are a user-defined input. As the rotor speed and pitch angle of the blades vary during the operation of a wind turbine, and thus during the simulations run in GH Bladed, the above described analysis is run at various rotor speeds and pitch angles, and the most appropriate modal representation is selected during the simulation.

The modal properties of the non-rotating structural components of the wind turbine (i.e. the tower) are calculated based on the modal degrees of freedom in the fore-aft and side-to-side directions of motion. Fore-aft is a common terminology referring to the movement of the tower towards and away from where the nacelle is pointing, i.e. perpendicular to the plane of rotation of the rotor. Side-to-side refers to the tower movement perpendicular to the fore-aft movement. Therefore, if we imagine looking at a wind turbine that is facing us, the side-to-side movement would be when the tower is bending towards the left or the right.

Figure 17 below shows a visual representation of the fore-aft and side-to-side movements of a wind turbine tower (1st modes):



(a) 1st tower side-to-side mode (b) 1st tower fore-aft mode

Figure 17: 1st side-to-side and fore-aft tower modes

As with the rotor, the frequencies and modal shapes of the tower modes are computed from the eigen-values and the eigen-vectors of a finite element representation of the tower. The finite element model of the tower is also based on the use of two dimensional beam elements to describe the mass and stiffness properties of the tower. These can be calculated by defining the mass distribution and bending stiffness along the tower, the mass inertia and stiffness properties of the foundation, and finally the mass and inertia of the nacelle and rotor. As with the rotor, the modal damping coefficients are a user defined input.

The actual coupling of the modal degrees of freedom of these rotating and non-rotating components is complex, and thus needs a complicated algebraic manipulation in order to derive the equation of motion of the structural dynamics of a wind turbine.

According to GH Bladed's theory manual, GH Bladed carries out this derivation of the equation of motion by using 'energy principles and Lagrange equations by means of a computer algebra package'[65].

A power train model is also used in GH Bladed in order to allow useful simulations to be carried out. This model consists of the generator model and the drive-train model.

The generator model allows the simulation of the various types of generators installed on wind turbines, and also allows the implementation of a control strategy by means of regulating the generator speed or the generator torque, as described in the introduction to this thesis.

The drive-train model represents the low-speed shaft, the gearbox and the high-speed shaft. The drive-train is modelled in various ways according to the user selection. For this research, the drive-train is modelled as the low speed and the high speed shafts – each having its own damping and stiffness, and a torque/speed reducing step between them.

A mechanical equivalent of the drive-train model, as used in this research is shown in Figure 18 below:

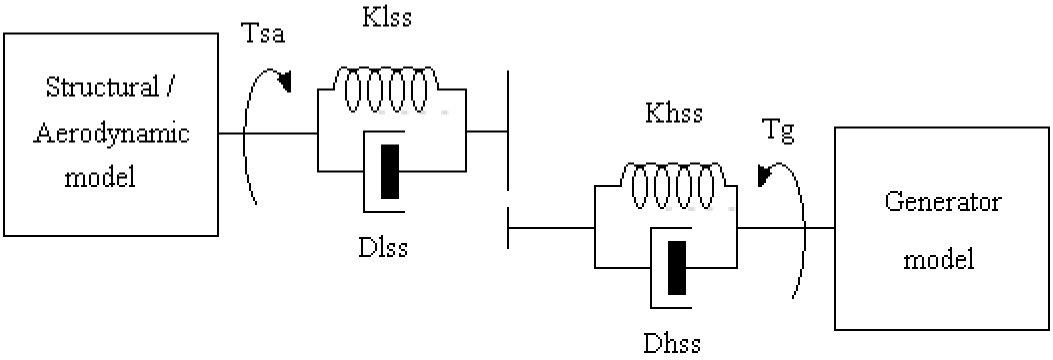


Figure 18: mechanical equivalent of the drive-train model

where:

The *Structural / Aerodynamic model* is the part of the GH Bladed model that performs all the calculations, as described in the previous paragraphs, in order to determine *Tsa*, the torque applied by the rotor’s hub to the Low Speed Shaft (LSS). The mass of the LSS is not shown in the above diagram as it is incorporated into the rotor mass.

Klss is the spring element that represents the stiffness of the LSS.

Dlss is the viscous damper that represents the damping properties of the LSS, main bearing (the bearing holding the rotor in place) and the gearbox bearings on the low speed gears.

Khss is the spring element that represents the stiffness of the High Speed Shaft (HSS)

Dhss is the viscous damper that represents the damping properties of the HSS, the gearbox bearings on the high-speed gears and the generator bearings.
The *Generator model* is the GH Bladed model that performs all the necessary calculations in order to determine *Tg*, the torque applied by the generator.

The equations of motion of the above drive-train model are quite simple:

Analysing the loads on the LSS:

$$Tsa = Klss(\theta_{lssr} - \theta_{lssgbx}) + Dlss(\dot{\theta}_{lssr} - \dot{\theta}_{lssgbx}) \qquad \textbf{Eq. 12}$$

Analysing the loads on the HSS:

$$Tg = Khss(\theta_{hssgen} - \theta_{hssgbx}) + Dhss(\dot{\theta}_{hssgen} - \dot{\theta}_{hssgbx}) \qquad \textbf{Eq. 13}$$

where:

θ_{lssr} and $\dot{\theta}_{lssr}$ are the angle and angular speed of the LSS at the rotor end,

θ_{lssgbx} and $\dot{\theta}_{lssgbx}$ are the angle and angular speed of the LSS at the gearbox end,

θ_{hssgen} and $\dot{\theta}_{hssgen}$ are the angle and angular speed of the HSS at the generator end,

θ_{hssgbx} and $\dot{\theta}_{hssgbx}$ are the angle and angular speed of the HSS at the gearbox end,

The necessity, however, to perform complex calculations in order to obtain a detailed equation of motion of the structural dynamics of the whole wind turbine, add the aerodynamic effects, the power-train model, and the control strategies used to operate a wind turbine, makes using a model like GH Bladed or FLEX, the most appropriate option for undertaking a research on the development of an automatically tuned drive-train damper. If such a model were not utilized, then a number of approximations would have to be used in order to construct a simple wind turbine model, thus substantially degrading the accuracy of results obtained.

Stol et al. [66], while developing a Disturbance Accommodating Control (DAC) based controller, showed just how important it is to test the controller behaviour with a good wind turbine simulation code / model. They showed that a controller they created was adequately controlling a turbine as modelled in SymDyn simulation code with just the rotor's rotation degree of freedom enabled. When, however, more degrees of freedom were enabled in the simulation, this system became unstable signifying that this controller was in reality not suited for its purpose.

Finally, the models and theoretical methods used in GH Bladed, have been extensively validated against monitored data from a number of wind turbines (28 are quoted in [65]), varying in both size and configuration. Such an extensive validation of a simulation code built specifically for this research would not be feasible within the timeframe and scope of this research.

For all the above-described reasons, the author decided to use GH Bladed in order to develop the procedure that automatically tunes the DTD.

3.2 Wind Turbine models

A total of 10 wind turbine models were designed and used both for development and validation. They are based on 3 generic wind turbines, created specifically for the purposes of this research. The models' main characteristics are shown in table 1 below:

Model name	750k-a	750k-b	2M-a	2M-b	2M-c	2M-d	3M-a	3M-b	3M-c	3M-d
Rated Power	750kW		2MW				3MW			
Rotor diameter (m)	50		75				96			
1st out-of-plane rotor freq. (Hz) *	1.65		1.06				0.83			
Tower Top Mass (tonne)	39		98				160			
Hub Height (m)	55	60	65	80	80	100	68	80	80	100
1st FA tower freq. (Hz)	0.64	0.7	0.66	0.51	0.44	0.42	0.48	0.41	0.35	0.34
Tower Mass (tonne)	75	96	184	255	196	367	192	255	196	367
Drive-train Stiffness (Nm/rad)	4.35E+07		1.70E+08				4.50E+08			
Drive-train Damping (Nms/rad)	3.82E+04		2.12E+05				4.87E+05			

* Uncoupled modal frequency at rated rotor speed

Table 1. The main characteristics of the models used in this research

Further information on the mechanical properties of the wind turbine models used is included in Appendix I.

Care has been taken, so that the turbine models devised are as realistic as possible. The models have been created in such a way as to have realistic component dimensions and mass along with realistic system frequencies.

Tower top mass, tower mass, hub height, rotor mass and rotor radius of these models were selected to resemble those of real turbines. All these parameters are quoted in the basic manufacturers' brochures available in the public domain. The term tower top mass is an industry specific term used to represent the mass of the blades, hub and nacelle. This is an often-quoted figure, as this mass has a profound effect on the loads on the tower and foundations of the wind turbine, and also its cost.

The uncoupled mechanical damping of the drive-train was set to 1% of critical. This is a figure commonly used for wind turbine drive-trains [67], and falls between standard damping for metals and standard damping for metallic structures with joints, i.e.:

- Metals (in elastic range) < 1%
- Metal structures with joints 3% - 7% [68]

The first tower fore-aft modal frequency lies between 1P and 2P, the first rotor out-of-plane modal frequency lies between 3.5P and 4.5P. The drive-train frequency, with the generator at constant speed, i.e. approximating a fixed speed wind turbine, lies between 2P and 3P for all configurations. These values were chosen as they are representative for a number of real turbines, of various sizes [69].

For the reader who is not familiar with the terms 1P, 2P, etc., 1P refers to the rotor frequency of rotation. So, for a rotor turning at 12 rpm, 1P is 12 rpm, usually expressed in Hz or rad/s, i.e. 0.2 Hz or 1.3 rad/s. Following this terminology, 2P is twice the rotor's frequency of rotation, 3P is three times this frequency, and so on. The use of this terminology is widespread in the wind turbine industry as these frequencies are the frequencies where some of the most important dynamic loads occur. The most important of these loads for a three-bladed wind turbine come from loading occurring at 3P. The main causes of this loading are briefly explained below:

a) The blades passing in front of, or behind, the tower. The wind speeds in front and behind the tower are lower than the wind speeds further away from the tower. This is because of the obstruction of the tower to the wind. As a turbine blade passes in front of or behind the tower, it experiences a reduction in the relative wind speed, and thus lift and drag forces are lower. As the blade moves away from this point, the relative wind speed it is subjected to starts quickly rising, and thus the forces on the blade start rising again. This relatively sudden unloading and loading of the blade is one of the most important fatigue loads in a wind turbine from a wind turbine design perspective.
b) The uneven wind speeds across the rotor disk. As the blades sweep across the rotor plane they are subjected to varying wind velocities. The rotor plane is crossed by wind with uneven wind velocities because of:

i. wind turbulence: the wind velocity at any point in space is never constant, and is never exactly the same as the wind velocity at a different point in space

ii. vertical wind shear: there is always a vertical wind shear profile (positive, negative, or mixed) that is caused by the interaction of the wind with the ground. In simple words, in flat terrain, the higher you are from the ground the stronger the wind blows

iii. nacelle misalignment to the prevailing wind direction, also known as yaw error. Assuming an even and constant wind blowing over the rotor, a blade would experience the same wind speeds as it crosses the rotor plane, but as the nacelle is misaligned, it will experience different wind directions, at different relative wind speeds, i.e. different wind velocities and thus different loading.

Although the loading caused by wind turbulence is stochastic in nature, and has a distributed frequency, the loading caused by the blades passing in front of or behind the tower, the vertical wind shear and the nacelle misalignment is always happening at 3P for three-bladed wind turbines. This is true for all wind turbine components, apart from the blades that individually experience the loads with a 1P frequency. Although the loads occur at 3P, they can have a profound effect on the wind turbine sub-structures that have natural frequencies not only close to 3P, but also close to the 3P harmonics at 6P, 9P, 12P etc.

Based on the rotor diameter and the turbine rating, the aerodynamic properties of the blades were synthesised. Based on the required out-of-plane rotor frequency, the mass distribution of the blades and their total mass were adjusted. Similarly, the tower modal properties were set in accordance with the desired mass, height and 1st tower natural frequencies.

All simulations run for this research were as realistic as possible, including advanced aerodynamic calculations, fully flexible turbine components and realistic wind regimes. The various parameters used for the simulations, are presented Appendix I. Note that all the parameters of the simulations, including wind regimes, are set based on the IEC61400-1 (2nd edition)[70] standard’s requirements for power production runs for wind turbine certification of Class IA wind turbines.

Chapter 4

The development of the self-tuned drive-train damper methodology is presented. Several variations of this methodology, tested during the development phase, are also discussed. Finally, other methodologies used for alleviating gearbox loads are examined. Some are completely novel and others rely on the automation of existing methodologies.

4.1 Methodology

In this section, the procedure for self-tuning the DTD is outlined and explained. The purpose of this procedure is for it to be used by the controller of any utility-scale variable-speed wind turbine in order to allow the turbine to autonomously and automatically tune its DTD on site.

The automatically tuned drive-train damper methodology can be divided in 4 stages. These stages are shown visually in Figure 19:

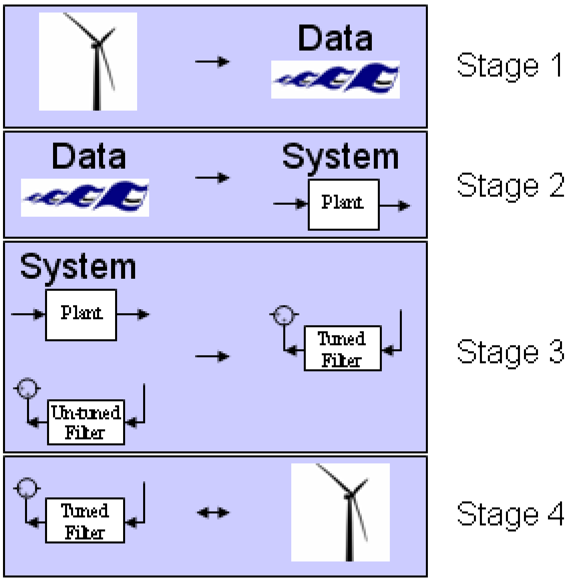


Figure 19: The stages of the automatically tuned drive-train damper methodology.

The first step in tuning the DTD is to create a linear model of the drive-train and more specifically the linear model representing the relationship between the generator torque and the generator speed (this is shown as stages 1 and 2 in the above figure). Usually the wind turbine designer builds a mathematical model of the whole wind turbine and simulates its behaviour. Based on the simulation results, a linear model of the drive-train is created. However, using the methodology developed in this present research, the model is created by system identification. The reason behind this choice is that the methodology relies on the actual system dynamics and not the design specifications, being also universally applicable on any variable-speed wind turbine. In order to create a model by means of system identification, an experiment must be run using the wind turbine (or the simulated wind turbine in the case of this research) in order to collect the necessary data. This is shown as Stage 1 in Figure 19. Then the collected data are used to construct a linearized model defining the generator torque – generator speed relationship shown as Stage 2 in Figure 19.

The second step is to use this model in order to automatically tune the DTD. This is shown as Stage 3 in Figure 19. As previously explained, the DTD is a filter applied to the generator torque demand that, in effect, adds a small ripple to the torque demand at the drive-train frequency and at such a phase that counteracts the effects of resonance. Tuning the DTD is normally done by first building a system comprised of a) a linear model of the wind turbine's drive-train (usually referred to as the 'plant' in control engineering terminology) and b) a filter (usually referred to as the compensator) to this system. By changing the properties of the compensator, i.e. by tuning the filter, the properties of the resulting closed loop system are changed. When this procedure is normally performed by a control engineer manually, his goal is to manually set the properties of the filter correctly. Setting the properties of the filter correctly minimizes the system's response in the regions around the drive-train's natural frequencies, which means that the system damping near the natural frequencies is damped more, and resonance and excessive loads are avoided. The methodology described here automatically replicates this procedure.

By performing the steps described above, a single DTD is tuned. As the quality of the tuned damper is dependent on the quality of the experimentally derived results, it is evident that the DTD will not always be well tuned. Thus the above steps are run a number of times, so that a well performing DTD can be tuned. Thus a final stage, Stage 4 in Figure 19, is needed to identify the optimal DTD among the DTDs tuned.

The stages of this methodology, briefly described in the preceding paragraphs, are presented analytically in the following sections.

4.1.1 Stage 1: The Experimental Procedure

Two different experimental procedures have been successfully devised. These experimental procedures have been run for all the wind turbine models, under 99 random wind realizations.

As explained in the introduction to this thesis, when a wind turbine is normally operating, depending on the way it is being controlled, it has a mechanism of maintaining the rotor operating within a predefined speed range. However, during the experimental procedure, and in order to be able to collect the necessary experimental data without any additional disturbances to the system identification procedure, the normal operation control mechanism was disabled.

Both experimental procedures ensure that the wind turbine is operated within preset rotor speed boundaries. When the turbine is operating within these boundaries, the wind turbine controller demands the experimental torque signal and halts any other rotor speed control mechanism the wind turbine controller would normally employ. When the rotor speed goes out of the preset boundaries, the wind turbine controller halts the experiment, brings the rotor speed back to the preset boundaries, and re-initialises the experiment. The reason behind this control strategy is that all variable-speed wind turbines can only safely operate within a defined speed limit. The lowest speed limit is determined by the capability of the variable-speed generator to produce energy at the grid frequency. The upper speed limit needs to be kept for two reasons. The first one is again related to the power output frequency which cannot exceed the grid frequency and is thus limited by the capability of the variable-speed generator. The second reason is related to the increase of the inertial and aerodynamic loads of the rotor, on the rotor itself and subsequently on the rest of the wind turbine components.

The optimal 10-minute mean wind speed at hub height for conducting the experimental procedures was found to be close to 6 m/s. The mean wind speed selected is high enough to maintain the wind turbine working within its lowest speed limit, and low enough not to make the rotor frequently over-speed. This allows the controller to demand the experimental torque signal for a longer period of time, before having to halt the experiment and bring the rotor speed back to the predefined limits. Consequently, the wind realisations used for conducting the simulated experimental procedures had a 6m/s 10-minute mean wind speed and IEC Class A turbulence, as explained in the next paragraph. In an experimental procedure run by an actual wind turbine, and not a simulated one as in the case of this research, this procedure should initiate when the 5 or 10 minute mean wind speed is close to 6m/s. In case the wind speed picks up or drops substantially during the experiment, the experimental procedure would halt, and re-initialise when the wind conditions were favourable.

The IEC turbulence classification is a definition of the level of turbulence in wind according to the EN61400-1 standard [70], and classifies turbulence into either Class A or Class B. This standard specifies turbulence by defining a characteristic turbulence intensity (I_{15}) and the way to calculate turbulence intensity at all wind speeds. The turbulence intensity (I) is a measure of turbulence which is equal to the standard deviation of the wind speed around the mean wind speed in a ten minute period over the mean of the wind speed in the same period. The characteristic turbulence intensity (I_{15}) is a value of this turbulence intensity at 15 m/s wind speeds. According to the standard, for high turbulence sites, i.e. class A sites, the value of I_{15} should be taken equal to 0.18. The way in which the standard proposes that the turbulence intensity varies with speed is the following:

$$I = I_{15} \cdot \left(\alpha + \frac{15}{\overline{U}} \right) \Big/ (\alpha + 1) \qquad \textbf{Eq. 14}$$

where I_{15} is the characteristic turbulence intensity, α is a constant set to 2 for Class A turbulence and $\overline{U}$ is the mean wind speed.

The EN61400-1 standard, now in its 3rd edition, is one of the most widely used design standards (if not *the* most widely used) in the industry, and most industrial sized wind turbines have been designed and certified according to this standard. Class A turbulence wind according to this standard was selected as it adequately defines the turbulence a wind turbine is expected to experience in a medium to high turbulence site. Such a site is where one expects the most structural failure problems to occur, and is thus more applicable to this research.

The difference between the two experimental procedures is the shape of the experimental torque demand signal. The first experimental procedure uses a torque demand based on the shape of a chirp signal, and thus will be called the “chirp experimental procedure” from here onwards. The second experimental procedure uses a torque demand based on a Pseudo Random Binary Signal (PRBS), and thus will be called the “PRBS experimental procedure” from here onwards.

The chirp experimental procedure

In the case of the chirp experimental procedure, the controller of the wind turbine demands a torque from the generator which changes with time according to Equation 15:

$$Q_d = Q_m + A \cdot \sin(2 \cdot \pi \cdot F_c \cdot t \cdot (t / t_c)) \qquad \textbf{Eq. 15}$$

where,
Q_d is the torque demand (Nm),
t is the time (s),
Q_m is the mean torque demand (Nm)
A is the amplitude of torque fluctuations (Nm)
F_c is a frequency constant (Hz), which is equal to the frequency of the chirp signal at time t_c,
and t_c = a time constant (s), where the frequency of the chirp signal is equal to F_c.

If t/t_c on the right hand side of Equation 15 were omitted, the equation would define a sinusoidal torque demand with a frequency = F_c Hz and an amplitude = A Nm. By multiplying the standard sine-wave equation with the t/t_c term, a modified sine-wave of constantly changing frequency is obtained.

Constant $\mathrm{Q_m}$ was set to 90% of the mean torque (in Nm) of a normal power production run at 6m/s mean wind speed. This value was chosen by a trial and error approach; ensuring that the rotor does not over-speed or slow down substantially. Constant A was set to 1% of rated torque. Again, this value was chosen by a trial and error approach: it needs to be large enough to excite the drive-train, but small enough not to excessively excite it and thus produce high loads in the drive-train of the wind turbine. Constants F_c and t_c were set to 10 Hz and 90 s respectively in order to ensure that in 70 seconds a full set of experimental data are collected. We consider that a full set of experimental data have been collected when the drive-train of the wind turbine has been subjected to a torque with a frequency content that contains all the excitation frequencies of the drive-train. Although these frequencies are not exactly known before or during the experimental procedure is run, it has been found in the course of this research, that for industrial size wind turbines (600kW and above) these frequencies would lie between 1.5 and 7.5 Hz.

This signal is reset, i.e. t starts counting again from 0, and starts the sweep through the frequency range 0 to 8Hz as soon as the signal goes over 8Hz. If the signal is not reset, and as the controller communication interval is 0.05s, the signal aliases after 10Hz and the frequency excitation is then in essence random (in the frequency range 0 to 10Hz) after the first sweep. This is not problematic by itself, but it has been found that resetting the signal gives more consistent results in most cases. This will be further discussed in the results analysis section.

Figure 20 on the next page shows the torque demand resulting from Equation 15 for a particular experiment:

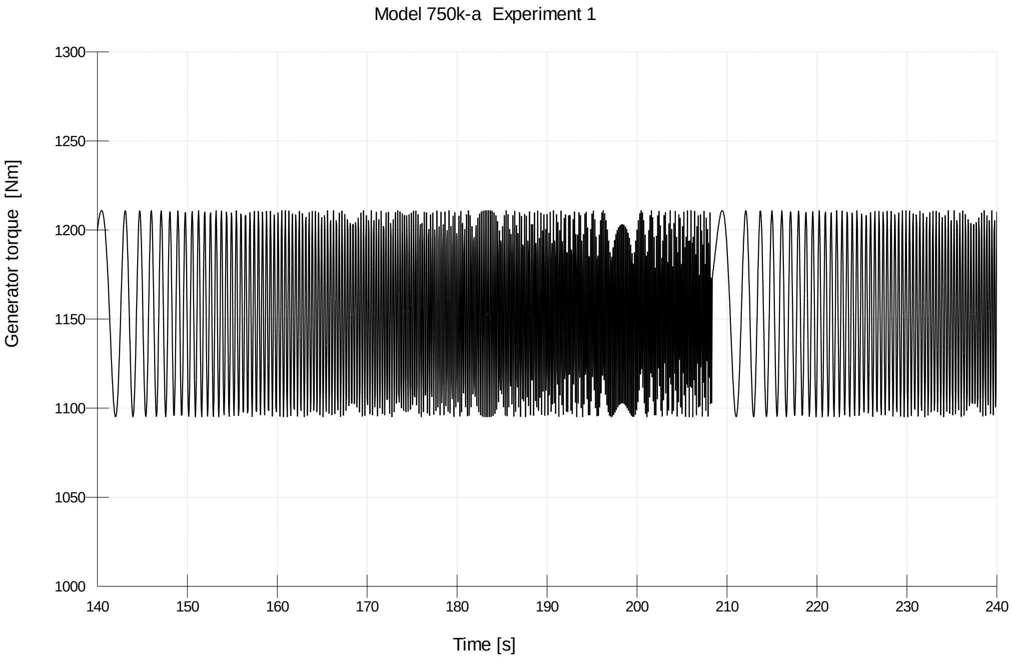


Figure 20: An example of torque demand while the controller is demanding a torque based on Equation 15.

In Figure 20 one can see the signal changing from low frequency to high frequency and then re-initializing (at approximately 209 s into the experiment).

The power spectrum of the generator torque confirms the quality of the generated signal. It is a signal exciting all the possible drive-train natural frequencies, i.e. from 1.5Hz to 7.5 Hz, more or less by the same amount. Figure 21 shows the power spectrum of the generator torque demand signal for a particular experiment:

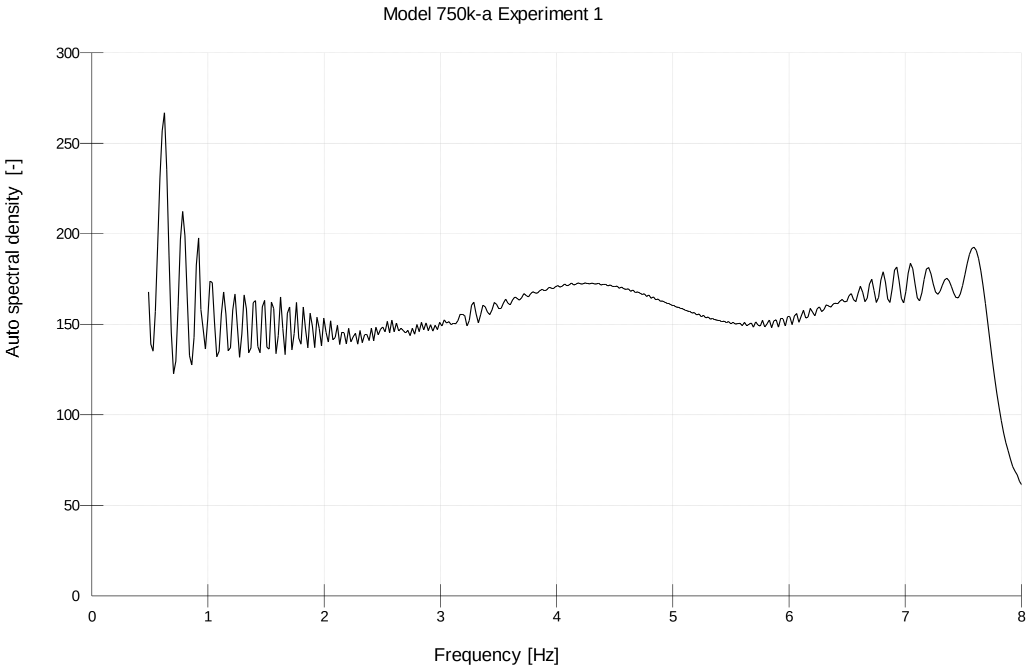


Figure 21: An example of the power spectrum of the torque demand while the controller is demanding a torque based on Equation 15.

Keeping torque at a constant mean torque value (equal to Q_m) makes the rotor change speed substantially during the experiment. If rotor speed goes outside of some predefined bounds, the controller tries to bring the rotor speed back within these bounds and then re-initializes the experiment. The lower limit is 0.9 of the minimum demanded generator speed (in power production)[*]. The higher limit is set at 1.1 of the demanded generator speed above rated.

When the rotor over-speeds, the torque controller switches to the turbine’s normal torque controller algorithm, i.e. the control algorithm the wind turbine uses under normal operating conditions. Since the rotor speed at this point is high, the torque increases and thus effectively brakes the rotor. At the same time, the blade pitch angle is also ramped to 20% of the maximum pitch angle (at 6deg/s) to stall the blades and slow down the rotor.

Similarly, when the speed of the rotor drops below the predefined lower limit, the torque controller switches to the turbine’s normal torque controller algorithm. Since

[*] Note that the minimum demanded generator speed for a VSWT is not the lower limit of the generator, but just the lowest control speed point for the torque controller in the below rated region.

the rotor speed at this point is low, the torque demand is brought to its lower limit and thus the rotor can regain its speed more easily.

Once the rotor speed is well within the predefined experimental limits (10% higher than the minimum demanded generator speed and 20% lower than the demanded generator speed above rated) and following an additional 5 second hysteresis, the experiment is re-initialized.

Figure 22 presented in the next page shows a typical torque demand in an experimental procedure. Note that in this 10 minute experiment, there is an experimental period of at least 70 seconds where a full set of experimental data has been collected (i.e. the torque demand has completed one full sweep between 0 and 7.5 Hz) :

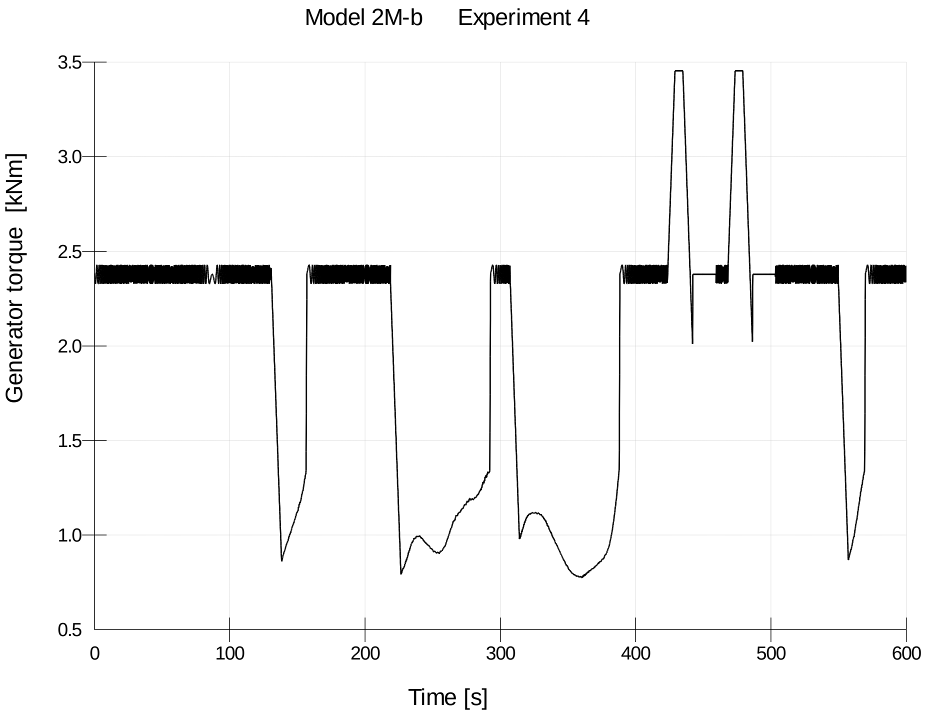


Figure 22: Torque demand during an experiment.

Care has been taken so that all torque changes happen smoothly and no sudden torque ramps arise. The experimental procedure code is written in C++ and the code is provided in Appendix II.

Through appropriately developing this controller code, the loads on the turbine are kept within acceptable levels. As this experimental procedure is meant to excite the drive-train, it is crucial to ensure that gearbox torque fluctuations and all other loads are kept within acceptable levels.

Figure 23 presented in the next page shows an example of the gearbox torque loads (on the low speed shaft side) during a 10 minute experimental procedure, and a 10 minute simulation run at a normal power production state for the same turbine.

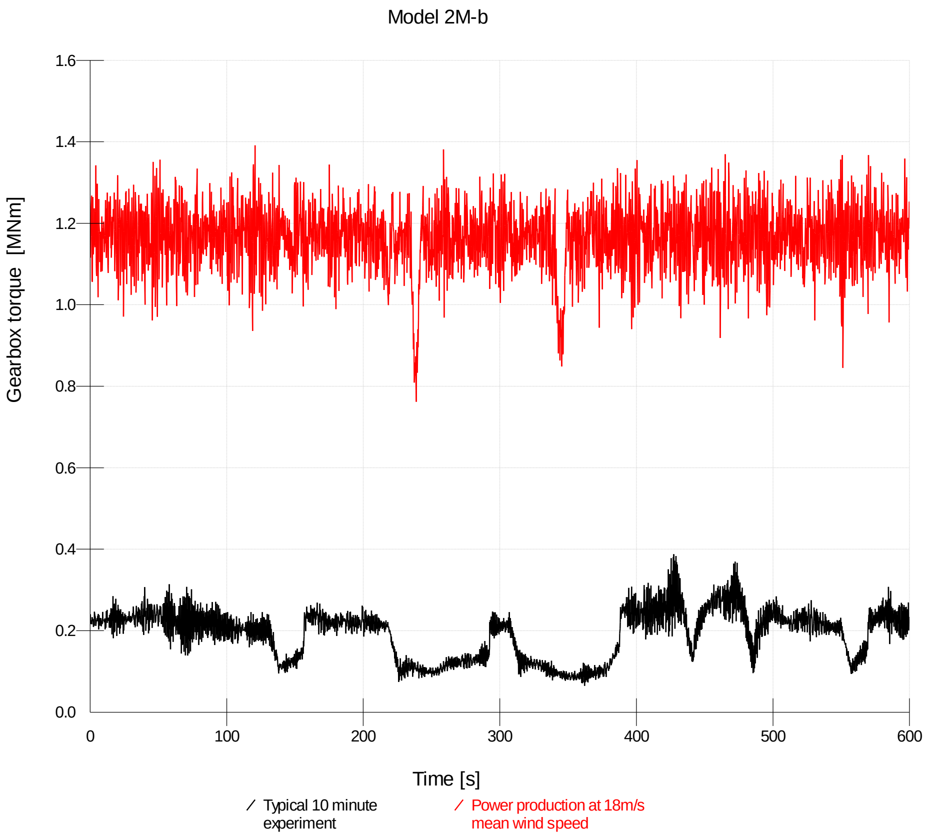


Figure 23: Gearbox torque comparison.

Note that both the gearbox torque loads and the load cycle amplitudes produced during the experiment are much lower than those created during a normal power production simulation.

The PRBS experimental procedure

In the case of the PRBS experimental procedure, the controller of the wind turbine demands the same torque from the generator as it would in a normal power production run below rated wind speed, adding a ripple on this demand based on a predefined PRBS signal.

Below rated wind speed, a variable-speed turbine usually tries to stay at its optimum tip speed ratio by changing the rotor speed in proportion to the wind speed. This maximises the power coefficient and hence the aerodynamic power available.

As was shown in the introduction to this thesis, this can be achieved by setting the generator torque Q_m to be proportional to the square of the rotor or generator speed:

$$Q_m = K \cdot \Omega^2 \ , \qquad \textbf{Eq. 16}$$

where Ω is the rotor speed and K is a constant that balances the rotor mechanical torque with the aerodynamic torque (in steady state).

Demanded torque (Q_d) during the PRBS experimental procedure changes with time according to Equation 17:

$$Q_d = Q_m \cdot (1 + A \cdot B_t) \qquad \textbf{Eq. 17}$$

where,

B_t is a PRBS based series, and takes the values of -1 and 1,

and A is a constant.

B_t PRBS based series has been created so that when it is sampled at the controller time step, all frequencies between 0 and 10Hz are excited equally (as far as this is possible). This was achieved by using the *idinput* built-in MATLAB command that uses an eighth-order Butterworth non-causal filter in order to create the time series. The *A* constant was set to 0.05. As in the chirp experimental procedure, the value of constant *A* was chosen by a trial and error approach: it needs to be large enough to excite the drive-train, but small enough not to excessively excite it and thus produce high loads in the drive-train of the wind turbine.

A power spectrum of the B_t series sampled at the controller communication interval is shown in Figure 24:

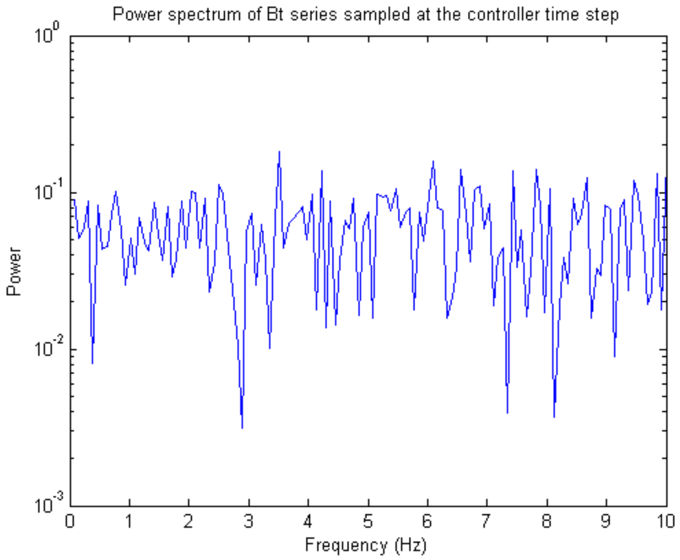


Figure 24: the power spectrum of the B_t series sampled at the controller communication interval (50ms).

Figure 25 shows the torque demand resulting from Equation 17 for a particular experiment run during this research:

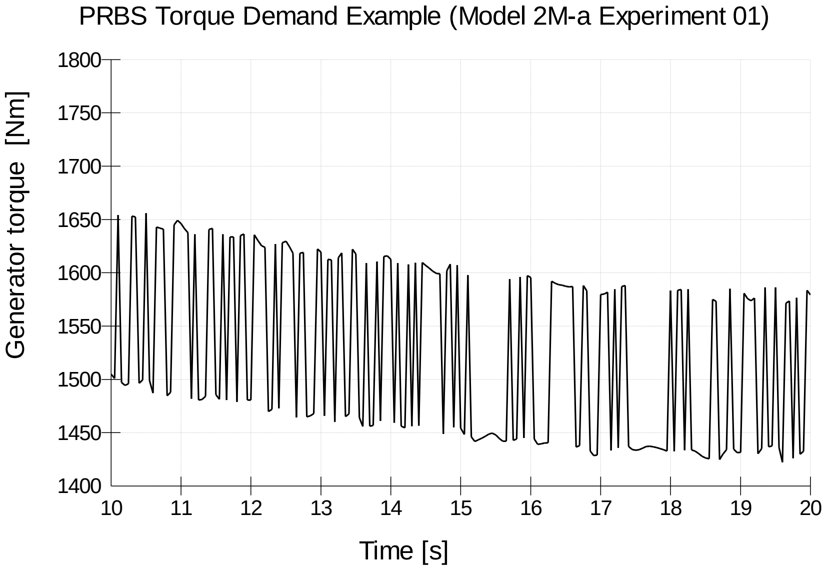


Figure 25: An example of torque demand while the controller is demanding a torque based on Equation 17.

As with the chirp experimental procedure, in this procedure the controller tries to limit the rotor speed variations in order to keep the turbine operating in a safe regime. However, by using the PRBS experimental procedure, the turbine controller self-regulates the rotor speed at all times during the experimental period and not only when the rotor speed exceeds the predefined experimental limits. In a normal power production run, the wind turbine controller tries to balance the mechanical torque applied to the rotor (from the generator) with the aerodynamic torque applied to the blades. As a result, the power extraction is maximized and the rotor accelerations are minimized. In the PRBS experimental procedure, the mean demanded generator torque signal is set in the same way as in a normal power production run, thus minimising rotor accelerations and maintaining rotor speed within the predefined experimental limits. This results in longer valid experiments and thus more data per experiment.

It is important to note that in the PRBS experimental procedure the normal torque demand is based on a closed loop control strategy: the torque changes in response to the generator speed. Normally, this would present a problem for the identification of the relationship between these two properties. This, however, is not a problem for the identification of the drive-train dynamics as the drive-train's resonant frequencies are much higher than the rotor speed variation frequencies due to the aerodynamic loading on the rotor.

An added benefit in using the PRBS experimental procedure is the fact that wind turbine operators would feel more comfortable running such an experiment on their machines as:

a) The loads on the machine during the experiment are very similar to those under normal operation and

b) A PRBS based ripple has been added on several occasions in experiments on real turbines [71 72].

The results obtained by the automatic DTD procedure developed by this research when using the chirp signal for its first stage are more consistent in comparison to the results obtained when using the PRBS signal. This will be further discussed in the results analysis section. However, as the results obtained by the self tuning procedure using both experimental procedures are similar, and the PRBS experimental procedure has the merits mentioned in the previous paragraphs, both these procedures and their results are analysed in this thesis.

The PRBS experimental procedure code is also written in C++ and the code is provided in Appendix II.

4.1.2 Stage 2: The Linearization Procedure

In this stage, a state-space linearized model describing the "Generator torque" - "Generator speed" relationship is created using the experimental data collected from stage 1. In order to create this model, MATLAB’s system identification toolkit is used.

System identification is the process of building a mathematical model for a dynamic system based on observed data. This process can be broken down into three steps. The first step is to select and collect the necessary data. The second is to select an appropriate model structure and define its variables. The third and final step is to assess the model quality, something that is usually based on how well the model can reproduce the measured data.

Therefore, the first step in the linearization procedure is to create an experimental procedure and then select the correct input and output signals one needs to monitor. In order to model the dynamic behaviour of a system, at least one output and one or more inputs need to be defined. The output at any given moment depends on the dynamic behaviour of the system (which is what we want to model), the inputs to the system (that is the external signals that can be manipulated by the observer), the output at previous time steps and any other external stimuli that we call disturbances.

For the system we want to model, the output is the generator speed. The input is the generator torque. The most important disturbance is the aerodynamic torque. Unfortunately, this is a disturbance we cannot easily measure with the required accuracy on a real turbine and thus this disturbance will not be measured or used in any way. So the collection of the necessary data, that is the generator speed and the generator torque, has already been accomplished in stage 1 of the self-tuning DTD procedure.

For the second step of the system identification procedure, an appropriate model structure must be selected. With the advance of computer hardware and the modern numerical software widely available, selecting the most appropriate structure is done in practice by selecting a number of model structures, defining the appropriate variables and comparing the quality of the models produced. For this research, a number of model structures were tested. These included an ARX model and various elaborations of the basic ARX model (namely the ARMAX, output-error (OE), and Box-Jenkins (BJ) models), a frequency-response model, and a state-space model[*].

The ARX model is a simple, discrete-time model used to describe the relationship between inputs and the output of a system.

In the ARX model the relationship between the input **u**(t) and the output **y**(t) is described as follows:

[*] A detailed explanation of these models can be found in [84]. General information on these models, and the associated theory, can be found in [73] and any other textbook that deals with system identification.

$$\mathbf{y}(t) + a_1\mathbf{y}(t-1) + ... + a_n\mathbf{y}(t-n) = b_1\mathbf{u}(t-1) + ... + b_m\mathbf{u}(t-m) \qquad \textbf{Eq. 18}$$

where a and b are weighting vectors, and t is time. The size of vectors a and b, i.e. n and m, are equal to the number of past output weightings and past input weightings respectively. The size of vector a, i.e. n, defines the number of poles of the model, whilst the size of vector b plus one, is equal to the number of zeros.

The vector containing the past values of inputs is called the regression vector in statistics. Models, such as Equation 18, that are based on regression of a time variable with itself at different time instants, are partly auto-regressions. For this reason, the model structure described by this equation is called Auto-Regression with eXogenus inputs (ARX).

Using a set of inputs and outputs, the above model can be trained, i.e. parameters a_{1-n} and b_{1-n} can be set, in order to minimize the error between the outputs predicted and the outputs observed. This can be achieved easily by modern numerical software such as MATLAB. The only input needed is the number of past outputs/inputs to be considered. This selection is based on the actual physics of the system to be modelled, and can be selected by an initial insight based on the physics of the model and then based on a trial and error approach.

The basic disadvantage with the above-described ARX model is that it does not take into account any disturbance to the system. If we add a disturbance term **e**(t) to the right hand side of Equation 18, a more complete ARX model is created. Still however, the lack of adequate freedom in describing the properties of the disturbance term has created the need for more complex models based on the ARX model: the Auto-Regressive Moving Average with eXogenous input (ARMAX), the output-error (OE) model and the Box-Jenkins (BJ) models. For example, the difference between the ARMAX and the simple ARX model is the moving average (MA) disturbance term:

$$\begin{aligned}&\mathbf{y}(t) + a_1\mathbf{y}(t-1) + ... + a_n\mathbf{y}(t-n) = \\ &b_1\mathbf{u}(t-1) + ... + b_m\mathbf{u}(t-m) + \mathbf{e}(t) + c_1\mathbf{e}(t-1) + ... + c_k\mathbf{e}(t-k) \qquad \textbf{Eq. 19}\end{aligned}$$

where **e**(t) is the disturbance and c is the weighting vector for the disturbance, with size k.

A more detailed explanation of the ARMAX model and other ARX based models can be found in Ljung [73], chapter 4.2.

The frequency-response model is based on a function that describes the steady-state response of a system to sinusoidal inputs. For a linear system, like our model of the drive-train, a sinusoidal input of a specific frequency results in an output that is also a sinusoid with the same frequency, but with a different amplitude and phase.

Such a liner system can be described by Equation 20 [74]:

$$\mathbf{y}(t) = G(\sigma)\mathbf{u}(t) + \mathbf{v}(t) \qquad \textbf{Eq. 20}$$

Where,

y is the output vector,
G is the frequency-response function that needs to be defined in order to construct the frequency-response model and describes the amplitude change and phase shift as a function of frequency.
σ is the differentiation operator p in continuous time and the shift operator q in discrete time,
u is the input vector,
and **v** is the additive noise.

A more detailed explanation of the frequency-response model can be found in Ljung [73], and other books such as [75] and [76].

The State-space models are models that represent a physical system as n first order coupled differential equations. The number of first order differential equations needed in order to adequately represent the actual system being modelled is equal to the number of poles of the system and defines the order of the model. In their most simple form, these differential equations relate input and output using the state variables in the following way:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \qquad \textbf{Eq. 21}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u} \qquad \textbf{Eq. 22}$$

where:

u is the input vector,
x is the state vector,
y is the output vector,
A is the state matrix,
B is the input matrix,
C is the output matrix,
D is the feedforward matrix

State vectors contain variables describing the state of the system. In simple terms, and if we compare the state-space representation to a physical representation of a system, the state variables would be the equivalent of the state of the system, i.e. the system's velocity, position, temperature etc. Matrices A, B, C and D contain elements with physical significance—for example, physical coefficients or material constants– that relate the system state, inputs or past outputs with the current system output.

For the purposes of this research all the above models were initially used. The parameters of these models were set using a number of estimation methods in order to find the best-suited model to describe the wind turbine drive-train model.

For the ARX model, the Least Squares (LS) and the Instrumental Variable (IV) methodologies were used in order to estimate the a and b vectors shown in Equation 18. The LS methodology works by minimizing the sum of squares of the RHS minus the LHS of the Equation 18, with respect to a and b. The IV methodology works by determining a and b so that the error (RHS-LHS in Equation 18) becomes uncorrelated with certain linear combinations of the inputs.

For the ARMAX, output-error (OE), and Box-Jenkins (BJ) models only one estimation method was used. This was the MATLAB System Identification toolbox method that is based on a quadratic prediction error criterion that is minimized using an iterative search algorithm. The initial parameter values for the iterative search are constructed by a LS-IV combining algorithm. The exact details regarding this method are not disclosed by MATLAB, but there is reference to the iterative search methods presented in Ljung [73], Chapter 10.2. In this reference, a number of iterative search methods are analysed, including Newton and quasi-Newton algorithms.

As mentioned earlier, the frequency-response model is based on a function that describes the steady-state response of a system to sinusoidal inputs. Using this approach, a linear time-invariant system can be described by Equation 20. Spectral analysis methods for determining frequency-response function G(σ) have been developed from statistical methods for spectral estimation [77,78]. For this research, the ETFE (Empirical Transfer Function Estimate), the SPA (Blackman-Tukey), and the SPAFDR (SPectral Analysis with Frequency Dependent Resolution) estimate methods were used.

The ETFE model is effectively the ratio of the Fourier transform of the output to the Fourier transform of the input. In this most basic model, the frequency-response function, is approximated by the EFTE, as shown in Equation 23:

$$\hat{\hat{G}}_N(e^{i\omega}) = \frac{Y_N(\omega)}{U_N(\omega)} \qquad \textbf{Eq. 23}$$

where,

$Y_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} y(t) e^{-i\omega t}$, with N being the length of the input – output data set,

and $U_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} u(t) e^{-i\omega t}$

denote the discrete Fourier transforms of $y(t)$ and $u(t)$ respectively [79].

By analysing the EFTE, it can be shown that this method produces a good quality estimate of the system at the specific frequencies that are present in the input, in case of a periodic input signal [73]. This follows the fact that the variance of the EFTE, at the specific frequencies that are present in the input, decays with N. So, assuming a sufficiently large size of data, a good model can be estimated for the specific frequencies. However, if the input is not periodic, the quality of the method deteriorates, with the error variance produced by this method not decaying with N, but rather remaining equal to the noise-to-signal ratio at the corresponding frequency.

In order to avoid this model's shortcoming, a number of methods have been developed whereby the system's behaviour at one frequency is assumed to be related to that at another frequency.

One of these methods is the SPA, or Blackman-Tukey method, where windowed versions of the covariance functions are Fourier transformed [80]. The transfer function estimate in the SPA model is given by Equation 24:

$$\hat{G}_N(e^{i\omega_0}) = \frac{\hat{\Phi}_{yu}^{N}(\omega_0)}{\hat{\Phi}_{u}^{N}(\omega_0)} \qquad \textbf{Eq. 24}$$

Where $\hat{\Phi}_{yu}^{N}(\omega_0)$ is the output-input cross-spectrum estimate and $\hat{\Phi}_{u}^{N}(\omega_0)$ is the input spectrum estimate. These are both standard estimates for spectra and cross spectra as smoothed periodograms. For a derivation of this equation, and a detailed explanation of the used spectra see [77,78,80].

Another estimation method used, is the SPAFDR. Under the SPAFDR method, Fourier transforms of the outputs and inputs are first formed. Products of the inputs and outputs with the conjugate input transform are smoothed over local frequency regions, whose width may depend on the frequency. In reality, this method is a variant of the SPA method with frequency-dependent resolution. In MATLAB, the SPAFDR transfer function estimate is computed in the following way: "First, the algorithm computes Fourier transforms of the inputs and outputs. Next, the products of the transformed inputs and outputs with the conjugate input transform are smoothed over local frequency regions." "The ratio of these averages computes the frequency-response estimate" [81].

For the state-space model, the N4SID and the PEM estimation methods were used. The N4SID method is a particular implementation of a subspace method. The subspace method is a type of linear system identification method developed in the beginning of the 1990's. In this method a projection (estimation) of the so-called extended observability matrix and/or the states of the unknown system is first made. Then based on this extended observability matrix or the projected system states, the state-space system model matrices are retrieved. A number of subspace method versions that apply the above-described procedure in different ways have been developed, such as the N4SID, the MOESP, the CVA and others [[82]]. For this research the N4SID method developed by Van Overschee and De Moor [[83]] is used. The N4SID method is one of the subspace methods that use the estimated system states in order to retrieve the system matrix. The exact way in which this method works is out of the scope of this thesis, and as it is quite cumbersome to reproduce, the presentation of this method has intentionally been omitted. Although knowing how this method works is not essential to the reader for the comprehension this thesis, the interested reader can find more information in the above referenced bibliography.

The PEM (prediction–error identification method) is a method very similar to the method used for estimating the ARMAX, OE and BJ models. The main difference is that the iterations are started from parameter values that are computed from N4SID. The parameterization of the matrices A, B, C, D, in Equations 20 and 21 and an additional matrix K related to the disturbances, (not shown in Equation 22 for simplicity) is free and adjusted to be numerically well conditioned.

Apart from the selection of the estimation method, the MATLAB system identification toolbox gives the possibility to 'focus' the estimation method, in a specific frequency region. The first option, called *'Prediction'*, is to let the estimation method find a solution that minimizes the total prediction error. Typically, this favours a good fit at high frequencies. The second option, called *'Simulation'*, is to perform a frequency weighting of the transfer function fit by the input spectrum. Frequency ranges where the input has considerable power will thus be better described by the model. In other words, the model approximation is such that the model will produce as good simulations as possible, when applied to inputs that have the same spectra as the input used for the estimation. The third option, called *'Filter'*, is to use a custom frequency range weighting, whereby the user defines the frequencies in which the user wants the model to predict better the system's actual behaviour. The fourth and last option, called *'Stability'*, is similar to the first option with the difference that the model is forced to be stable.

There are a number of model validation methods that can be used in order to assess the quality of a system identified model. According to Ljung [73], a particularly useful method is the residual analysis technique. Under this technique, the residuals, i.e. the difference between the predicted output of the model and the actual output, are tested in a number of ways in order to gain insight into the quality of the identified model. The problem is that these methods try to help us understand if the identified model agrees with observed data. Recognizing if the model agrees with observed data is not always the same as identifying if the model is a good representation of the true system. This would happen only if the following criteria were met: a) there were no excessive unmeasured disturbances affecting the output of the system during the experiment and b) the system identification experiment was run for long periods and thus collected a large number of 'observed data'. As neither (a) nor (b) can be sufficiently met for our purposes, the conformity of the identified model with the observed data serves a limited purpose. This point is further addressed in the Results and Discussion section of this thesis.

Instrumental at this stage of designing the self-tuning DTD procedure is the frequency response of the models drawn on Bode plots. The frequency response of the models constructed were compared to the frequency response of the linearized "Generator torque" - "Generator speed" relationship, as this is provided by the linearization module in GH Bladed[*]. According to Ljung [73], this method of comparing the input and output properties of a system identified model with the properties of models created with quite different underlying assumptions, is a good approach for understanding how well the identified model represents the true system.

[*] As GH Bladed is a certified and widely accepted simulation code, the resulting linearized models it produces will be considered as being very close to the "true" linear representations of the system. Of course as the wind turbine is not a linear system, a 'true' linear representation does not exist. What is sought after is the best possible linear approximation.

The system identified models, the frequency response of which most resembles that of the GH Bladed linearized models, are those based on the state-space model. A comparison of the system identified state-space models, produced using the N4SID methodology, with the models created by other system identification methodologies, is presented in the Results and Discussion section of this thesis. This comparison validates the choice of the N4SID methodology for the self tuning DTD procedure developed by this research.

Figures 26 and 27 show examples of the Bode plots of system identified models and GH Bladed linearized models for wind turbine models 750k-a and 2M-a respectively:

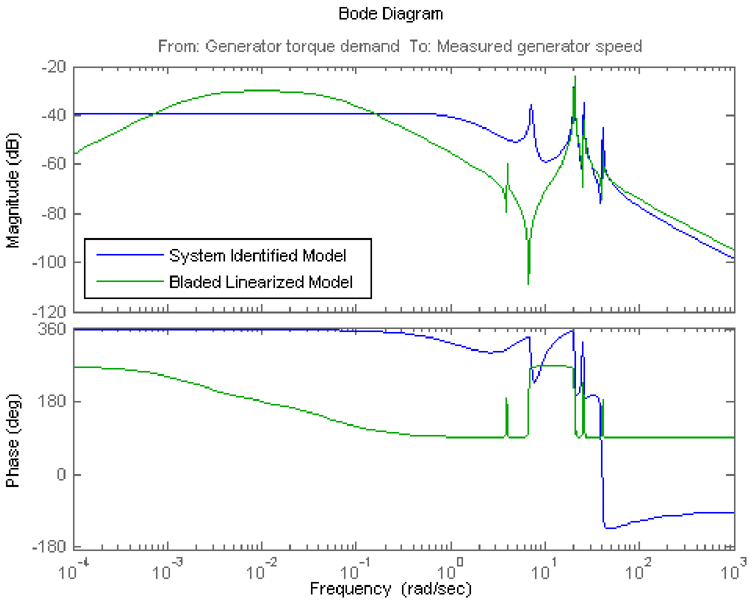


Figure 26: System Identified state-space model (wind turbine model 750k-a, experiment 39) and the GH Bladed linearized model (wind turbine model 750k-a).

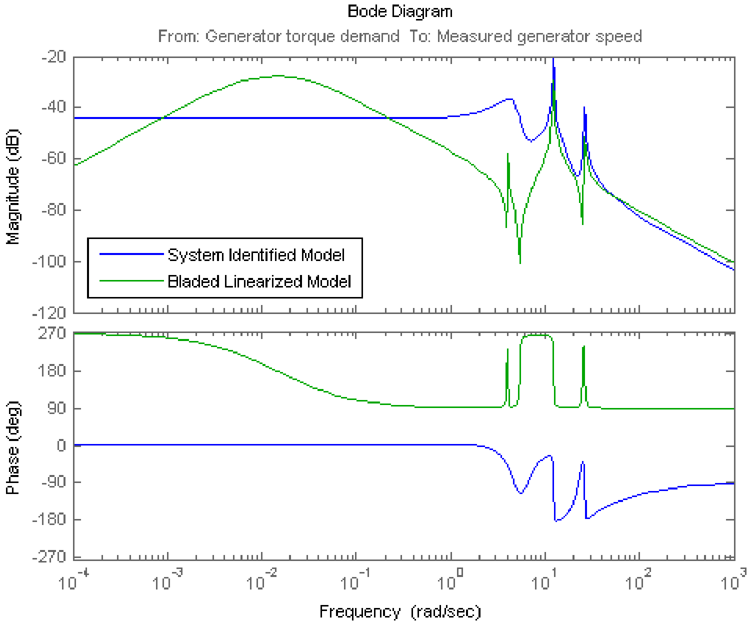


Figure 27: System Identified state-space model (wind turbine model 2M-a, experiment 5) and the GH GH Bladed linearized model (wind turbine model 2M-a).

For the reader who is unfamiliar with the Bode plot representation, a Bode plot is a combination of a Bode magnitude plot and a Bode phase plot.

The Bode magnitude plot is a plot where the magnitude response of a transfer function (measured in dB, which is inherently a logarithmic measure) of a linear system is plotted versus frequency on a logarithmic x-axis. Very simplistically, in a single input single output (SISO) linear system, where the input to the system is a constant frequency input signal of known amplitude, the magnitude value of a Bode magnitude plot at that specific frequency shows what the system output gain will be. Multiplying this gain with the input signal amplitude, the amplitude (magnitude) of the output signal is found. It is useful to note that the peaks on a Bode magnitude plot are created by the closed system poles of the system, while the troughs are created by the closed system zeros.

The Bode phase plot is a plot where the phase response of a transfer function of a linear system is plotted versus frequency on a logarithmic x-axis. Very simplistically, in a single input single output (SISO) linear system, where the input to the system is a constant frequency input signal at a specific frequency, the phase response value of a

Bode phase plot at that specific frequency shows the phase difference between the input and output signals at that frequency.

Comparing magnitude plots of the two models in both Figures 26 and 27 one can see that, with the exception of the first peak of the magnitude plots, all the peaks (the poles of the system) are located at the correct frequencies and correct magnitudes.

The low frequency system pole in the GH Bladed Linearized Model should be disregarded, according to GH Bladed's developers, as it is a pole created in the Linearization procedure GH Bladed uses, and does not represent an actual system pole. The low frequency system pole in the System Identified Model is a pole created by the tower interference on the flow field the blade experiences when passing in front of the tower. Such a pole does not exist in the GH Bladed Linearized Model, since this model is a model of the system's own dynamics and no wind interference is introduced.

In the phase plots, although the plots on first inspection seem quite different, the phase changes happen at the same frequencies, and with comparable magnitudes, thus showing good agreement.

Based on the above findings, the state-space model was considered the correct structure to use for system identification. Thus, a short MATLAB code was written in order to automatically create a state-space model for any given set of data created in stage 1 of the self-tuning DTD procedure. A step-by-step explanation of the code is given below:

i) "Generator Torque" (the input) and "Generator Speed" (the output) data series, called *the data* from this point on, are imported (after Stage 1 has run).

ii) The means and the trends of *the data* are removed. The means are removed since the mean magnitude of the input or output is not important since the mean magnitude is directly related to the wind speed and not the dynamic characteristics of the drive-train. The trends are also removed, since during the experiment the wind speed may decrease or increase, making the rotor slow down or accelerate. This again is something that should be disregarded as it is the result of the change in wind speed which is considered a disturbance to the system we are trying to linearize. Removing the means and trends from the data when these are not important is a standard practice suggested for the system identification process [73]. An example of this is shown in the following graphs.

The original output series for the 25th experiment of the 2M-a model is shown in Figure 28.

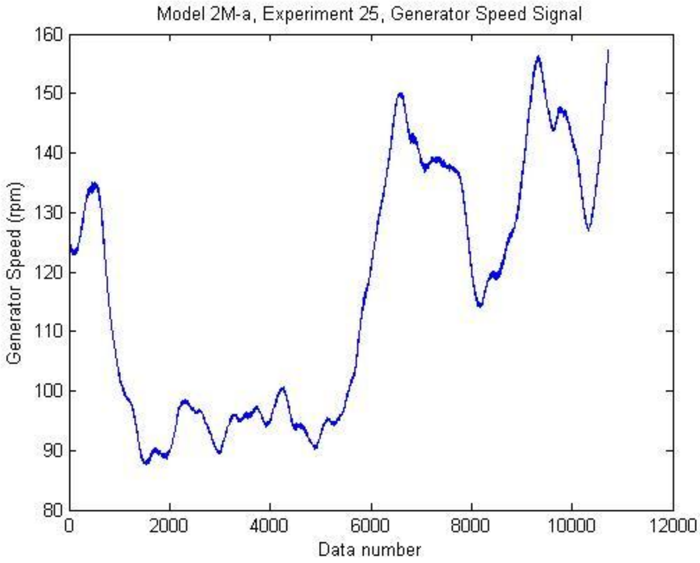


Figure 28: The Generator Speed signal (output data) of the 2M-a wind turbine model as it is recorded during the 25th Experiment of the PRBS procedure.

After removal of the mean and trend, the time series becomes as shown in Figure 29.

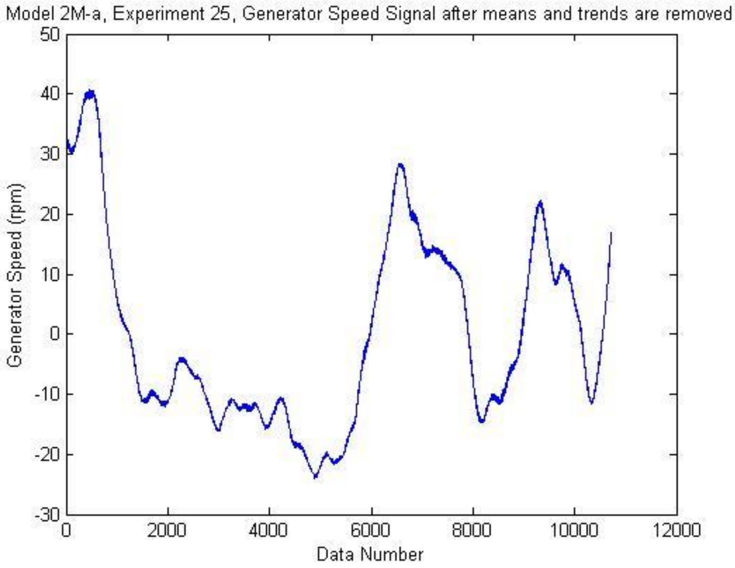


Figure 29: The Generator Speed signal of the 2M-a wind turbine model in the 25th PRBS experiment after the means and trends are removed.

iii) *The data* are then filtered to further isolate the effects of the turbulent wind. The filter is set to filter out data below 1.5 rad/s. Filtering the data at this frequency ensures that no significant dynamic characteristic of the drive-train will be masked out and that a substantial part of the turbulent wind effects will be filtered out. As this filtering distorts the first few seconds of data, these first few seconds are removed from the data set. The exact number of data removed is dynamically set depending on the distortion introduced by filtering by an algorithm that was developed for this research. This was a simple algorithm, which worked in the following way: By using a large set of experiments, it was found that filtering never affected the data after the first 200 to 300 data points. So the data set after the first 400 data points was considered to be always unaffected by the filtering process. Removing the first 400 points would, however, lead to the discarding of many useful data in the cases where the filtering only distorted the first few data points.

The first step was to check if the data were distorted. First, the minimum and maximum data points in the unaffected portion of the already filtered, de-trended, and normalized output data series were found. These are henceforth referred to as the Unaffected Minimum and Unaffected Maximum. Then the minimum and maximum of the first 400 data points were calculated. These points are now referred to as the Affected Minimum and the Affected Maximum. If the Affected Minimum was less than 80% of the Unaffected Minimum, or the Affected Maximum was more than 120% of the Unaffected Maximum, then the algorithm assumed the data are distorted and continued to remove the distorted data.

In order to make sure that only the distorted data were discarded, the following procedure was used. The first 400 data points were averaged in groups of five data points. Each one of these groups was tested and if its average value was found to be more than the Unaffected Maximum or less than the Unaffected Minimum, then the group of data was considered to be distorted. All data from the first data point up to the last data point group found to be distorted were deleted, along with an additional 80 data points discarded as a ‘safety’ margin. An example of this is shown in the following graphs.

The output series for the 25th experiment of the 2M-a model, after filtering is shown in Figure 30.

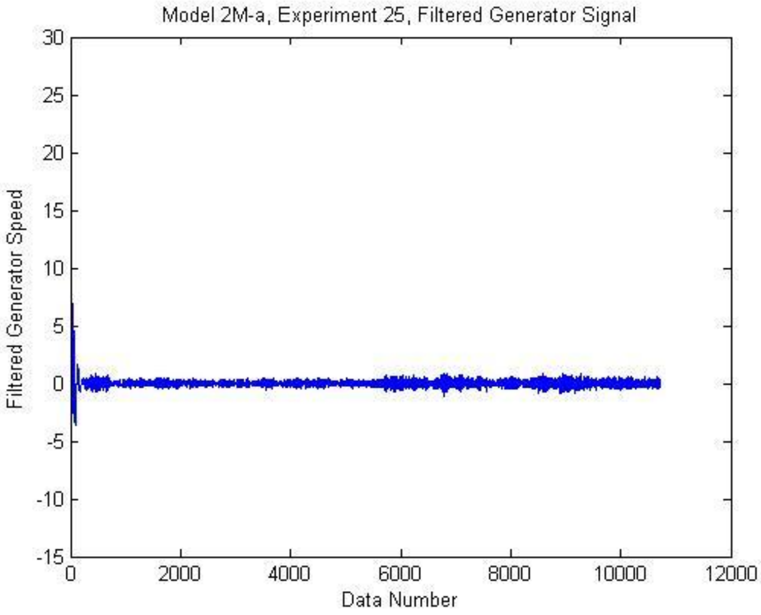


Figure 30: The Generator Speed signal of the 2M-a wind turbine model in the 25th PRBS experiment after the means and trends are removed and the signal is filtered.

The first 1000 data points are plotted in Figure 31 below, showing the distortion produced by the filter.

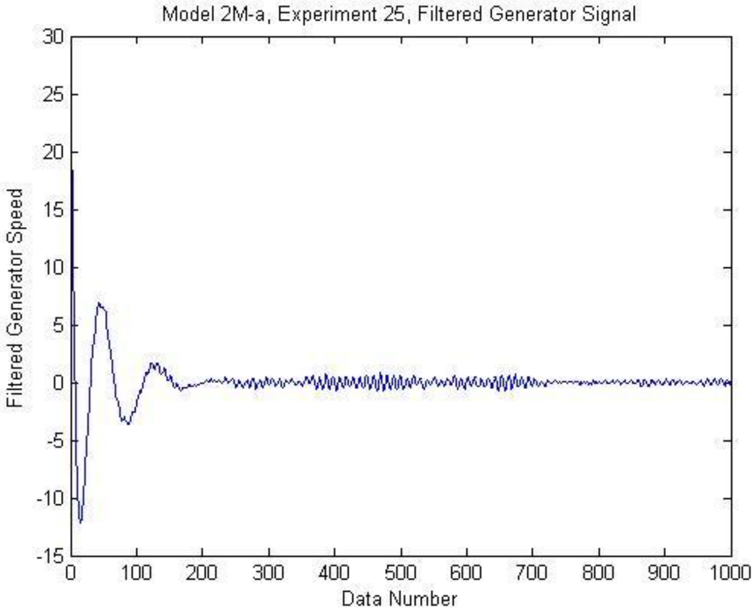


Figure 31: The first 1000 points of the Generator Speed signal of the 2M-a wind turbine model in the 25th PRBS experiment after the means and trends are removed and the signal is filtered.

By applying the above-described simple algorithm, the first 221 data points are removed, and the first 1000 data points of the remaining data set are shown in Figure 32.

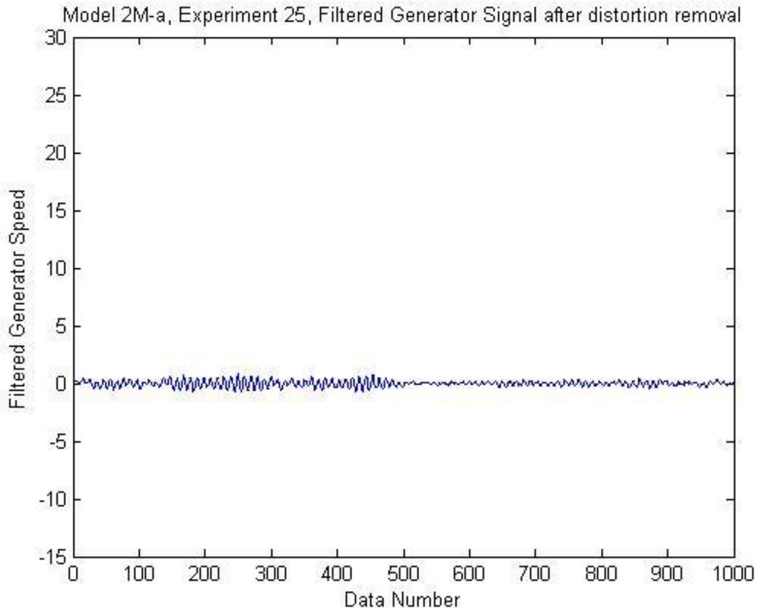


Figure 32: The first 1000 points of the Generator Speed signal of the 2M-a wind turbine model in the 25th PRBS experiment after the means and trends are removed, the signal is filtered and the initial points affected by the filtering distortion have been discarded.

iv) The State-Space model representing the linearized system is then calculated using the N4SID subspace method. For the implementation of this method MATLAB's System Identification toolbox "n4sid" command [[84]] is used. Automatic order selection and focus on stability options are selected for this command, as this results in models that are consistently closer to the GH Bladed linearized models. Although setting the order to a predefined value might seem to be more prudent, this is not suited to the purposes of this research. This is so because this procedure is designed to be used - without any changes - for a wide range of wind turbines that may have a different number of system poles. As MATLAB's automatic order selection has been tested with all the available models and it was shown that it correctly selects the optimal order number for every model turbine, MATLAB's automatic order selection feature was used in order to select the model order.

v) Finally, the discrete state-space model is transformed into a continuous time single input single output (SISO) transfer function. This is done in order to prepare the model for the next stage. Note that the transfer function does not need to be continuous. Working with a system and a compensator (the DTD) either in continuous or discreet time is a preference of the control engineer [85]. The final DTD transfer function is usually used in its continuous form, with the discretisation carried out at run-time, because that allows for a different discrete controller time step to be used from the one that was assumed during tuning. This results in a slightly more robust approach. Thus, and as a preference of the author of this thesis, all the controller and DTD tuning processes were carried out in continuous time.

Two other alternative procedures were also created in order to create a linearized model for the "Generator torque" - "Generator speed" relationship; namely the 'multi-experiment' and the 'Blade Pass Correction' system identification procedure. In effect, these procedures are the same as the procedure described above, with the following differences:

'Multi-experiment' system identification procedure

In the 'Multi-experiment' system identification procedure, instead of using *the data* from a single experiment, these data are merged from a number of experiments and are used for steps *ii* to *v* in the same way as in the original procedure. This approach was not found to produce good quality linearized models, as the Bode plots of these models were not a good match to the Bode plots of the linearized systems produced by GH Bladed. Thus, this procedure was not pursued further.

'Blade Pass Correction' system identification procedure

In the 'Blade Pass Correction' system identification procedure, a signal based on the azimuth of the rotor is introduced in the system identification as a known disturbance. The reason for adding this input signal is to take out, or at least reduce, the effect of the interference caused by the abrupt aerodynamic load change taking place when the blade of the wind turbine passes in front of the tower. This interference is evident at the passing frequency of the blades (usually referred to as 3P, for a three-bladed wind turbine, as previously mentioned) and its harmonics.

A detailed investigation of the results obtained with the original linearization procedure showed that there was substantial 3P interference in the identified models.

As an example, the Bode diagram of an identified system (750k-a model, system identified based on experiment 2) is shown in Figure 33 below.

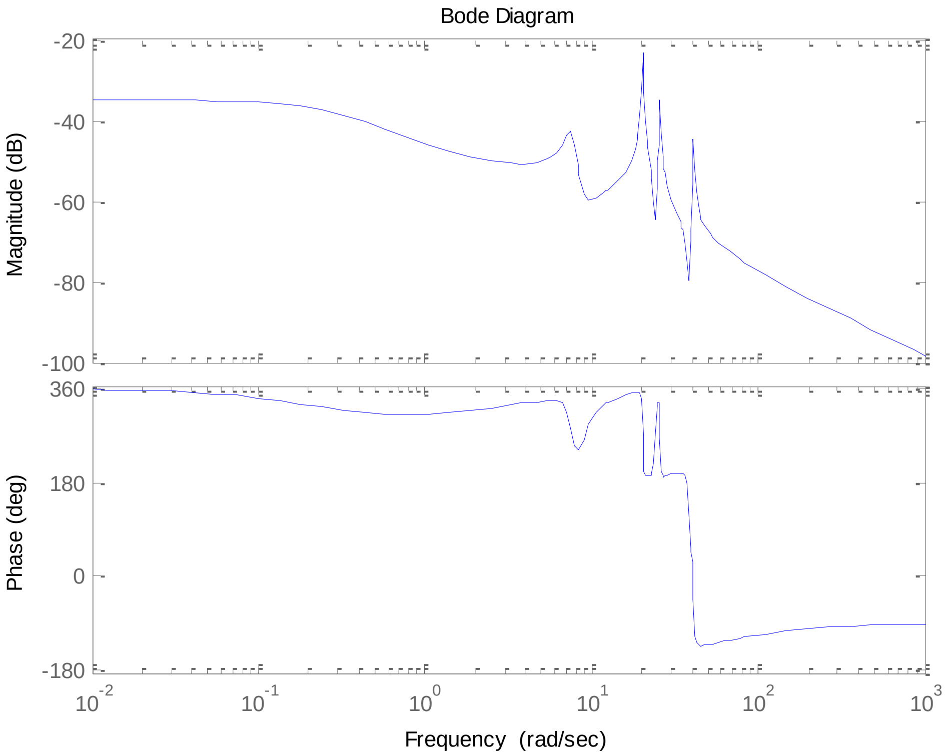


Figure 33: Example of an identified system showing 3P interference.

In this example, the mean 3P frequency during the experiment used for system identification was approximately 7 rad/s. The first peak on the Bode magnitude plot is at 7.4 rad/s and is created by the 3P excitation as there is no natural frequency of the system nor any other significant excitation close to this frequency.

The above shown interference, caused by the experimental 3P excitation of the identified system, is - to some extent - common in all experiments. The results obtained when using linearized systems, modified substantially by 3P excitation, were found to be inferior to the results obtained by using linearized systems that were only very lightly modified by the 3P excitation. This is because the filters developed based on linearized systems that were substantially modified by 3P excitation were strongly biased toward the low 3P frequencies. This caused to the filters to try to damp the 3P loading and not the loading near the drive-train frequency. Thus, a number of ways to reduce the effect of the 3P excitation to the identified system were investigated.

First, a procedure to directly delete the pole associated with the 3P interference was examined. This proved unsuccessful, as deleting the pole distorted the system. Then another procedure that filtered the experimental data around the 3P frequency was employed. This was done using both a band stop filter and a high pass filter. Using a band stop filter resulted in the creation of two smaller peaks in the frequency response of the system on either side of the original 3P peak. Using a high pass filter resulted in a diminished magnitude peak at a higher frequency than the original 3P peak. Unfortunately, all these methodologies proved problematic as they eventually led to badly tuned drive-train dampers, i.e. dampers that lead to greater drive-train damage.

To overcome the large 3P interference, a multiple input system identification method was finally used in order for the system identification routine to be able to distinguish between the abrupt aerodynamic load changes taking place when the blade of the wind turbine passes in front of the tower and the rest of the excitation. In this method, instead of using one input signal, namely the “Generator Torque”, two input signals were used, as explained below.

A signal based on the rotor azimuth was fed into the system identification routine as a second input with the first input being the Generator torque as in the original procedure. This signal, from now on called “Azimuth Load Indicator”, was created by the following conversion: azimuths from 0 to 180 degrees were converted linearly to values ranging from 0 (for an azimuth of 0 degrees) up to 1 (for an azimuth of 180 degrees). Azimuths from 180 to 360 degrees were converted linearly to values ranging from 1 (for an azimuth of 180 degrees) down to 0 (for an azimuth of 360 degrees).

Including the “Azimuth Load Indicator” as a second input to the system identification procedure resulted in a considerably more consistent DTD performance. These findings will be further discussed in the results analysis section of this thesis. Thus, in its final implementation the linearization procedure is based on two input signals, the generator torque, and the “Azimuth Load Indicator”.

The final implementation of the system identification procedure code is provided in Appendix II.

4.1.3 Stage 3: The DTD Tuning Procedure

In ‘stage 2’ previously discussed, a model for the generator torque - generator speed relationship of the wind turbine was created. In ‘stage 3’, the model created in ‘stage 2’ is used in order to find the appropriate constants for the DTD. As previously mentioned, the DTD used in this methodology is a band-pass filter of the form presented in Equation 04 and reproduced here for the reader’s convenience:

$$H(s) = K\frac{2\zeta\omega s(1+s\tau)}{s^2+2\zeta\omega s+\omega^2} \qquad \textbf{Eq. 04}$$

Where: K = gain, ζ = damping, ω = frequency and τ = time constant.

As explained earlier, for the purpose of this research, the filter design is being performed in continuous time. We also assume no time-lag and thus set τ to 0. Note that the real wind turbine system will have a time-lag, but since we are using a system identification routine to create the model of the wind turbine drive-train, time lags are implicitly taken into account. This means that Equation 04 becomes:

$$H(s) = K\frac{2\zeta\omega s}{s^2+2\zeta\omega s+\omega^2} \qquad \textbf{Eq. 25}$$

The single band-pass filter described by Equation 25 is introduced as a positive feedback loop to the generator torque demand control loop as shown in Figure 34:

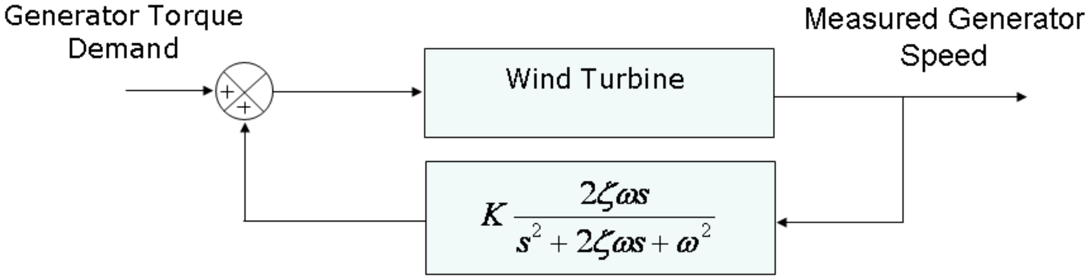


Figure 34: DTD block diagram.

In order to 'tune' the DTD, so that it performs optimally, i.e. reduces the fatigue damage on the drive-train, one must select the most appropriate values for the filter's gain, damping and frequency constants. The usual way for control engineers to achieve this in practice, is to use graphical computer tools that allow them to graph the closed system root-locus on a pole–zero plot and continually modify all the parameters of the filter dynamically. By evaluating the shape of the root-locus plot, the position of the closed system poles and the response of the closed system via Bode plots, the control engineer can modify the filter parameters to achieve the desired goal.

How this is done in practice is explained with the aid of the following example.

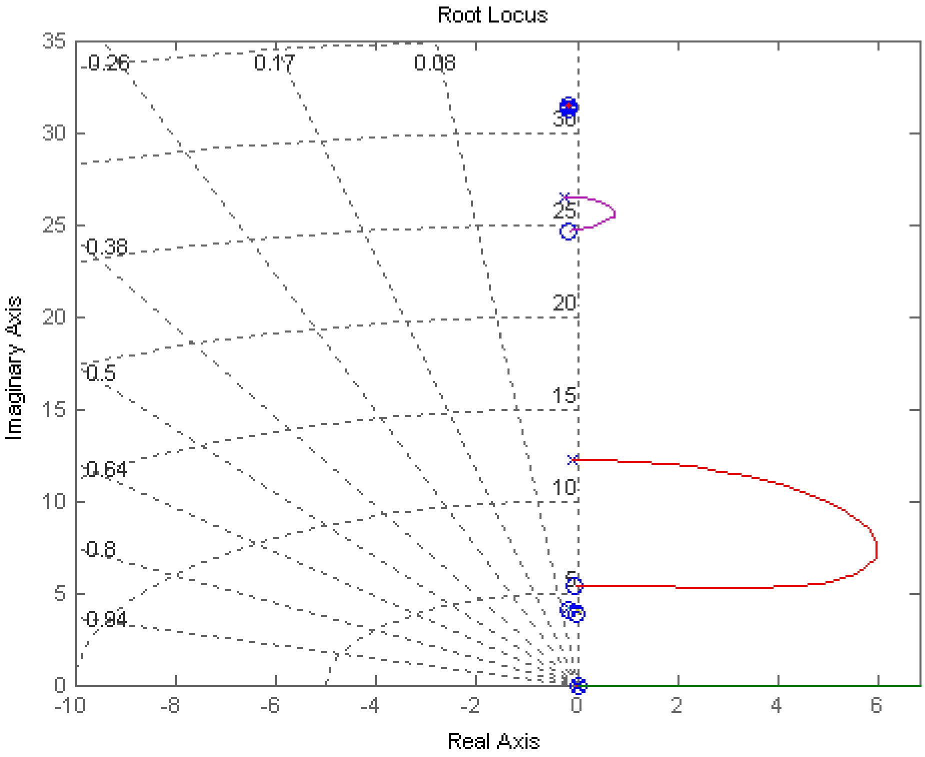


Figure 35: Example system root-locus plot.

In Figure 35 above, the zero and pole locations of the linear model of a wind turbine drive-train are shown. These are indicated as blue circles and blue crosses respectively. It should be noted that a number of zero and poles, created during the GH Bladed linearization process but were cancelling each other out without affecting the system, have been erased from the Bode plot in order to aid to the clarity of this figure. The response of the linear model shown in Figure 35 above is displayed in the following Bode plot in Figure 36.

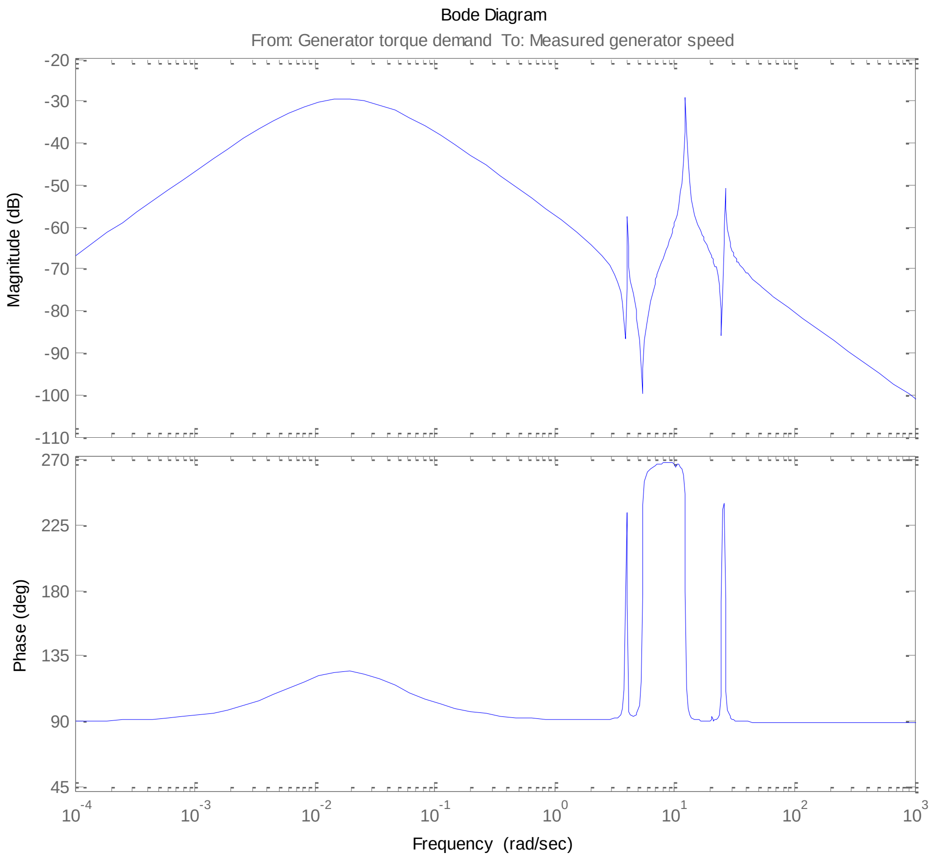


Figure 36: Example system Bode plot.

The engineer trying to create a single band pass filter based DTD would start by introducing a single band pass filter as a compensator to the system. Then he would use a graphical computer tool to plot the closed system root-locus on a pole–zero plot and dynamically modify all the parameters of the filter. The goal of the engineer would be to maximize the damping of the system poles, in such a way that minimizes the system's amplification near the resonant drive-train frequencies. An example of such a tool is MATLAB's *SISO design tool* that allows the user to interactively modify the compensator parameters by changing the compensator's gain and its pole location on the root locus plot. An example screen shot of the root locus plot used within this tool is shown in Figure 37:

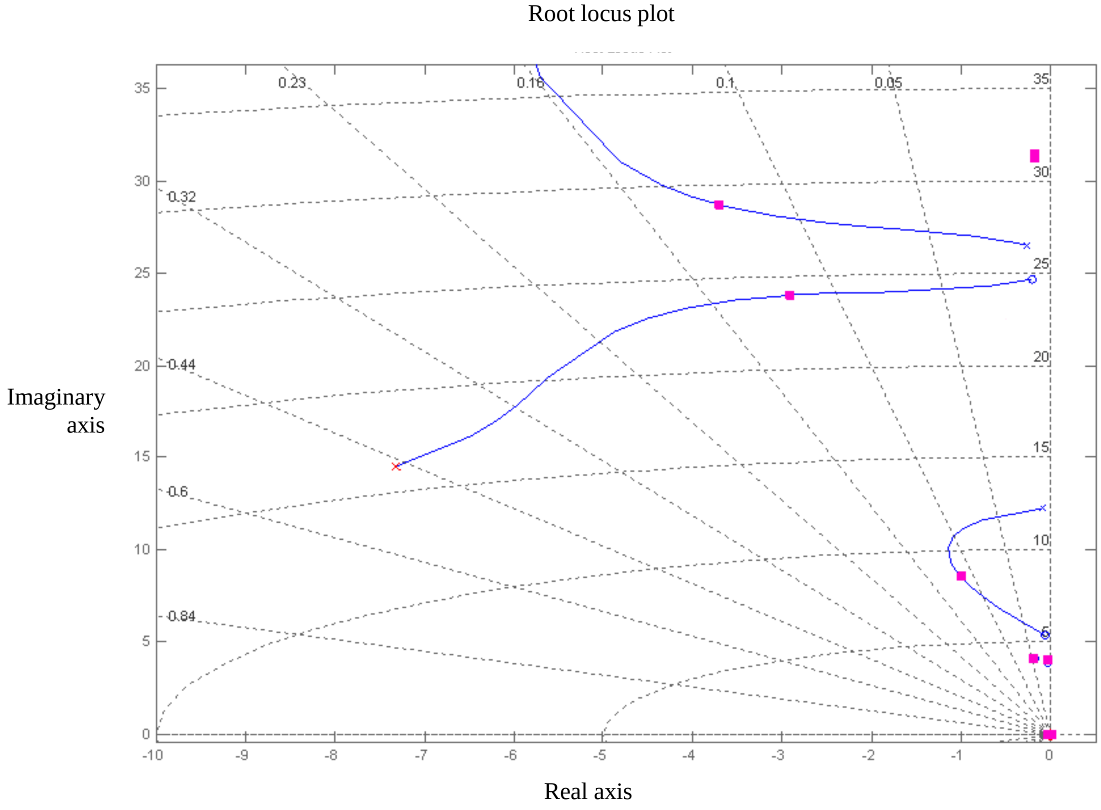


Figure 37: Root locus plot showing closed system zeros and poles.

A single band-pass filter has been introduced as a compensator to the example system. The pole of this compensator is shown as a red cross on the left side of the plot (at -7.3 + 14.5i). The purple squares indicate the closed system poles. The engineer’s task is to move the compensator’s pole, in order to find a position where the closed system poles have the largest damping. At the same time the pole’s location is changed, the compensator’s gain should also be varied in order to optimize both pole position and gain. Note that the dotted straight lines are lines of constant damping.

By inserting the compensator shown in Figure 37 above, the closed system response is modified, and exhibits magnitude attenuation around the system’s natural frequencies. This is shown in Figure 38:

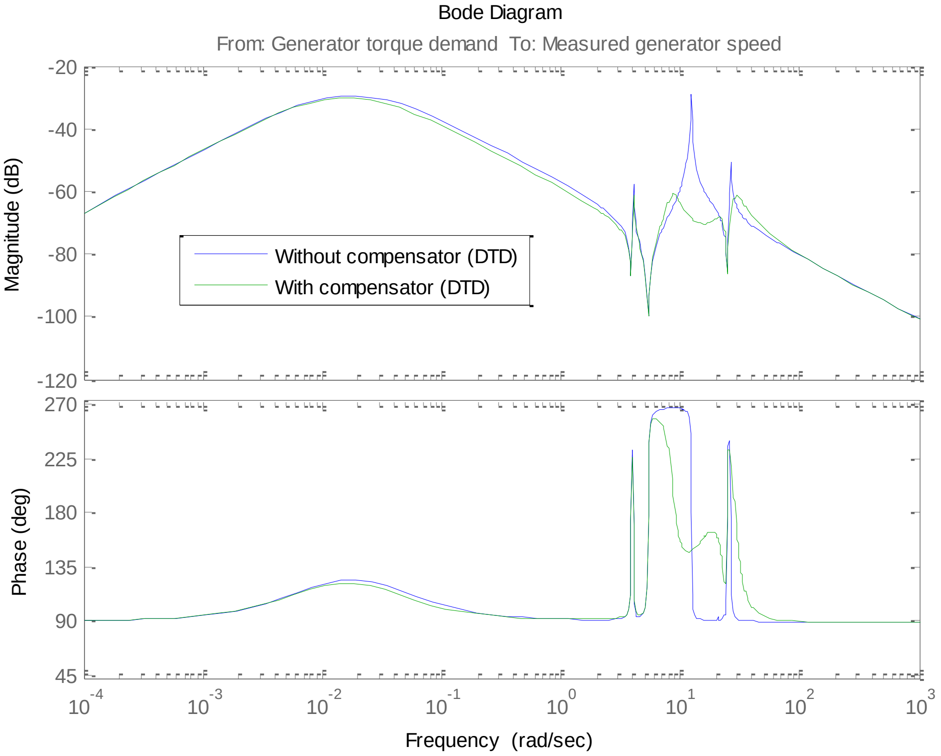


Figure 38: Example system Bode plot, with and without compensator.

In order to find the appropriate constants for the filter automatically, a MATLAB script was created in order to emulate this procedure. This script creates a closed loop system, with the plant being the linearized model from stage 2 and the compensator being a filter of the form presented in Equation 25 each time using a different set of filter parameters. The resulting closed systems are then evaluated and the four best are selected based on four "cost functions".

The following sections describe how the parameters of the filters are set, and how the cost functions are used in order to evaluate the filter's effectiveness.

Filter parameters

The filter's damping constant (ζ) is allowed to either take a value of 0.29 or 0.45.

Low values of the damping constant never produce good enough dampers, as their frequency response is very narrow. They produce a filter that only damps a very small range of frequencies, that are not sufficient to damp the resonances produced in the drive-train of a wind turbine. The developed algorithm allows the damping constant of the filter to take the low value of 0.29, only as an error checking provision. If one of the filters tuned has a damping constant value of 0.29 then the filter is automatically discarded. The usual cause leading the self-tuning DTD procedure to tune such a filter is a badly identified linear model. Note that the 0.29 value was arbitrarily set to be a low value. It could be set to 0.25 or 0.35 or any other such low value.

Higher values of the damping constant usually produce better results. But it is found in practice that damping constant values greater than 0.45 may introduce unwanted low frequency torque variations that adversely affect power quality [85]. As the damping constant value of the filter increases, the frequency bandwidth of the filter is broadened, and the low frequency content of the speed signal produces torque signal variations through the DTD filter. Thus for the purposes of this research, the filter's damping constant is set to a value of 0.45. Of course, after automatically tuning the first acceptable DTD, and using it in the controller, running the wind turbine in normal power production mode, the power fluctuations can be assessed (preferably near rated wind speed, where the induced low frequency torque fluctuations are most evident), and the value of the upper boundary for the damping can be incrementally increased, assessing each time the low frequency power fluctuations, until the optimum value is reached.

The boundaries of the frequency variable (ω) are dynamically set using the identified system parameters as follows: the frequency (ω) is allowed to vary between the lowest significant natural frequency of the drive-train and the highest significant natural frequency of the drive-train. The lowest and highest significant natural frequencies are automatically detected by a simple script finding the lower and higher maxima of the magnitude Bode plot. Care has been taken so that the algorithm developed distinguishes between the 3P excitation and the lowest significant coupled natural frequency of the drive-train, in order to avoid setting the lowest frequency boundary based on the 3P excitation. It must be noted that it has been found during this research, that a well-tuned DTD always has a frequency variable (ω) that is within these boundaries, and this is why these boundaries were selected. Consequently, if the frequency were allowed to take values outside this boundary, the tuning procedure would be slower, with no performance benefit.

Finally, the gain (G) is allowed to vary between 0.5 and 7000.5, as it was found that in practice, the gain of a well-tuned single band pass filter acting as a DTD would always lie within this range.

The variables are incremented in such a way so that they cover the whole allowable range with the required resolution. This resolution should not be very high, as the time it takes for the procedure to run is proportional to the number of increments selected. As an example, if the resolution of the gain is 100 (i.e. the DTD will be tested with 71 different gain values), the time it will take for the automatic DTD tuning procedure to run will be double compared to the time it will take for the procedure to run if the resolution of the gain were 50. At the opposite end, the resolution should not be very small, as this would lead to sub-optimal setting of the variables.

The resolution was set so that on a standard PC the whole automatic DTD tuning procedure would not take more than 10 minutes per identified system.

In order to optimize this procedure, by both increasing the resolution of the domain of the variables used and decreasing computing time, the following procedure was employed: First, a set of 11,500 combinations of gain, damping, and frequency are evaluated. When the best combination is selected, a second ‘round’ of evaluation starts, this time checking 55 combinations around the best combination selected in the first ‘round’ of evaluation, this time with a higher resolution.

As an example, for a wind turbine with two distinct drive-train modes, the first drive-train mode at 6 rad/s and the second at 12 rad/s, the first round parameter values would be:

Gain (G): 0.5 to 7000.5 with 71 increments,

Damping (ζ): 0.29 and 0.45

Frequency (ω): from 6 to 12 with 81 increments.

If the best combination was identified to be Gain = 700.5, Damping = 0.45 and frequency = 11.25, the second round parameter values would be:

Gain (G): 630.5 to 770.5 with 11 increments, Damping (ζ): 0.45,

Frequency (ω): from 11.13 to 11.37 with 5 increments.

The reason behind incrementing gain with a higher resolution than frequency in the second round is that the closed system dynamic behaviour is affected more by small increments of gain rather than frequency near the optimal combination of parameters. This is due to the shape of the root locus of the closed system.

The relatively simple optimization methodology described above was used instead of more complex algorithms, for example genetic algorithms, because of the robustness of this simple methodology. More complex algorithms would lead to a less computationally intensive procedure, but inherently could sometimes lead to non-optimal solutions. This is because the closed loop system damping has local maxima that can lead such algorithms to end up at those false local maxima locations, instead of the global maximum damping location.

In order to evaluate how well a particular combination of filter parameters is performing, a set of cost functions was devised. All the cost functions penalize the performance score of the DTD according to the system's closed loop poles' damping constant values. In general, the more damped the closed system poles are, the more damped the system is. The goal of the DTD is to damp the system poles created at the system's natural frequencies as much as possible in order to minimize the load amplification naturally occurring near the system's natural frequencies. Thus a DTD that damps the system well (i.e. damps the system's poles more) gets a smaller penalty than a DTD that damps the system lightly.

In order to calculate the first two cost functions, the closed system poles are divided into five categories based on their damping ratio values: very low (ζ <0.04), low ($0.04\leq \zeta < 0.07$), medium ($0.07\leq \zeta < 0.1$), high ($0.1\leq \zeta < 0.15$) and very high ($0.15\leq \zeta$). Then the number of closed system pole pairs in each category is multiplied by a "penalty" value specified for each category and cost function. If two DTDs result in systems whose closed loop poles are binned in the same categories (say three closed loop poles at the medium damping and two closed loop poles at the high damping categories) then the sum of the closed loop poles' damping values is used to determine which DTD is performing better.

Equations 26 and 27 presented below show how the cost functions are calculated:

$$\text{Costfunction1} = 0.1\cdot\text{vhighZ} + 1\cdot\text{highZ} + 10\cdot\text{medZ} + 100\cdot\text{lowZ} + 1000\cdot\text{vlowZ} - 0.001\cdot\text{sumZ} \qquad \textbf{Eq. 26}$$

$$\text{Costfunction2} = 1\cdot\text{vhighZ} + 1\cdot\text{highZ} + 10\cdot\text{medZ} + 100\cdot\text{lowZ} + 1000\cdot\text{vlowZ} - 0.001\cdot\text{sumZ} \qquad \textbf{Eq. 27}$$

where: vlowZ, lowZ, medZ, highZ, vhighZ represent the number of poles in the very low, low, medium, high and very high damping categories respectively. SumZ is the sum of the closed loop poles' damping constant values.

The third and fourth cost functions are calculated based on an exponential "penalty" function.

Equations 28 and 29 presented below show how the third and fourth cost functions are calculated:

$$\text{Costfunction3}= \sum_{x=\text{Npoles}}^{x=1} 3200\cdot\exp(-58.13\cdot Z(x)) \qquad \textbf{Eq. 28}$$

$$\text{Costfunction4}= \sum_{x=\text{Npoles}}^{x=1} 3000\cdot\exp(-2\cdot Z(x)) \qquad \textbf{Eq. 29}$$

where: *Z(x)* is equal to the damping constant value of the *x*'th closed system pole if this value is lower than 0.2, or 0.2 if the damping of the *x*'th closed system pole is larger than 0.2,

Npoles is the number of closed system poles

At the beginning of this research, a number of additional cost functions similar to the ones presented above were used but they became obsolete, as they were always out-performed by the four above mentioned cost functions. Therefore, they were discarded in order to reduce the computational time. This point is further analysed in Results and Discussion section. Also, a number of cost functions giving weight to both the damping and the frequency values were devised, but as they did not give rise to good DTDs they were also discarded.

After evaluating all pre-defined variable combinations, based on the four cost functions presented above, four variable combinations, i.e. filters, have been selected. A next step is required in order to select the best filter out of the four. This is done by comparing the closed loop systems’ Bode plot peaks’ magnitudes. The system showing the lowest peaks is selected as the self-tuned DTD. It should be noted here that a frequency weighting is applied to the peaks’ magnitude in order to weight more the low frequency reductions in peak magnitudes. The energy content in the lowest modes of vibration, i.e. lowest frequency peaks in the magnitude Bode plot, is higher that that of the higher frequency modes, as there is more excitation at these lower frequencies during the wind turbine normal operation. The exact weighting algorithm can be seen in Appendix II.

So the selection of the best filter variable combination is performed with a two step approach. The first step uses the cost functions to arrive at a selection of four filter variable combinations, and the second step uses the closed loop systems’ Bode plot peaks’ magnitudes to select the best filter variable combination out of the four. It was found during research that if only one of the four cost functions was used, a suboptimal DTD would be tuned for at least one of the wind turbine models. On the other hand, if only the magnitude Bode plot methodology was used in order to select the best filter variable combination, the selected variables would always lead to a badly tuned DTD. Thus, the two-step approach was pursued.

4.1.4 Stage 4: DTD Selection

After stages one to three have run several times, a number of DTDs have been tuned. Selecting the best one is done in stage 4. In this stage, the DTDs are introduced for short periods of time on the controlling loop of the wind turbine when the wind turbine is operating at an above rated wind speed.

The straightforward way of evaluating the performance of the tuned drive-train dampers is by measuring the gearbox loads. Unfortunately however, a gearbox load sensor is almost always not readily available on commercial wind turbines. Thus, a way to assess the quality of each DTD using commonly available signals was investigated.

Two signals that are always available to the controller of a variable-speed wind turbine, the generator speed and torque signals, are used in order to produce an indication of the best DTD. The wind turbine is allowed to run normally at the above rated region for approximately 5 minutes and the generator speed and torque data are collected. This is done once per tuned DTD. These data were manipulated in many ways in order to find the best way to use them in order to predict gearbox damage.

The two most promising ways of identifying the best DTD are presented here. The first is based on the derivation of the generator acceleration from the generator speed. After this has been performed, the generator acceleration is filtered at 1.5 Hz in order to better correlate the data with the system dynamics and less with the wind turbulence. The standard deviation of the filtered generator acceleration can then be used as an "indicator" for the best DTD. Figure 39 shows a plot of this "indicator" and the gearbox damage for model 2M-a using a selection of 34 DTDs, sorted in ascending order with respect to gearbox damage. The gearbox damage is a measure computed by a methodology explained in chapter 5 of this thesis. Note that both the indicator and the gearbox damage have been normalized to facilitate the data representation.

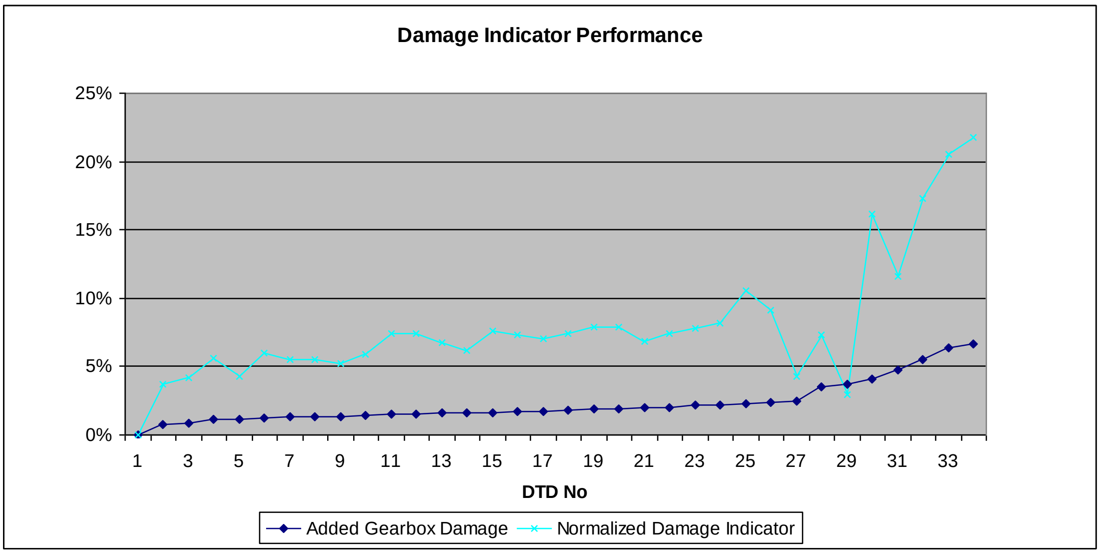


Figure 39: Damage Indicator performance.

From the example presented in Figure 39 it is evident that the indicator value is to some extent correlated with the gearbox damage. It is however also evident that the indicator is not reliable enough to be used for the selection of the best DTD. This becomes obvious when examining DTD 29 in Figure 39. Based on the indicator this is the second best DTD. In reality however, this DTD is ranked 29th based on the added damage to the gearbox. Added to this, the use of the derivative of a measured signal (the generator acceleration is not measured directly) might be problematic if there is substantial noise on the measured data.

Another "indicator" was also developed that does not rely on the derivation of a measured signal. It uses a combination of a filtered generator speed Damage Equivalent Load (DEL) and a generator torque DEL. A detailed explanation of the term Damage Equivalent Load, and its calculation is given in chapter 5 of this thesis. At this point, it is sufficient to say that the DEL, in the context of loads, is a constant frequency and amplitude load that would impart the same damage to a component as the actual frequency and amplitude changing load being measured. Although the DEL of a speed signal has no physical significance, it is a means to represent the speed fluctuations by a single number. There are several ways to do this of course, for example by using the standard deviation of the signal, but the DEL was found to be the best suited method for the derivation of this indicator. The generator speed signal is first filtered at 1.5 Hz with a high pass filter in order to make the data less affected by wind turbulence and more correlated to the system dynamics. The filtered generator speed DEL is then multiplied by the generator torque DEL producing a number used as an indicator for the gearbox damage comparison. In most cases, this "indicator" identifies the best DTD or a DTD which is very close to the best (i.e. a DTD that produces at most 1% more damage in the gearbox in comparison to the damage produced by the best DTD). Unfortunately though, in some cases this "indicator" identifies a DTD that is much worse than the best DTD (in the author's investigation a DTD producing up to 13% more damage to the gearbox was identified as the best DTD using this methodology).

The shortcomings of these indicators, which are the most promising of all indicators evaluated, signify that the use of the commonly available signals is probably insufficient.

Since it is very important to use the best available DTD in the final implementation of stage 4, a gearbox torque signal needs to be used for the comparison of the effectiveness of the DTDs. Since only the relative magnitude of the gearbox torque needs to be known and the evaluation can be done in a relatively short period of time, the gearbox load sensors do not have to be calibrated. Thus, the use of a simple un-calibrated strain gauge on the gearbox should be sufficient for the purposes of selecting the best DTD. The procedure of selecting the best DTD would then be straightforward: the various self-tuned DTDs would be tested for short periods of time under similar wind speeds, and the best performing one would be selected.

Normally the effect of the DTD is identifiable in the above rated region where torque is maintained constant, or nearly constant. Thus in order to compare the effectiveness of the various DTDs tuned, the wind turbine should be operating in the above rated region. However, high wind speeds are not so frequent and the loads on the drive-train are much larger at high wind speeds. Therefore, it is recommended to operate the wind turbine in the above rated region operation mode, at a lower than normal wind speed and rated power. This will ensure that the drive-train loads will not become excessive even when testing badly tuned dampers and that the self-tuning procedure can be concluded more quickly as the lower wind speeds needed will be more frequently available. Operating the wind turbine at a lower operating point, will not increase any of the loads of the wind turbine, and thus should not pose any problem to the wind turbine operation. As the wind turbine will be operating at its nominal rotor speed, the drive-train dynamics at this low wind speed will be almost identical to the drive-train dynamics the wind turbine would have at the higher wind speeds. The only difference would be that of slightly different aerodynamic damping. As the aerodynamic loads are lower at these lower wind speeds, the aerodynamic damping of both blades and tower will be lower [86]. This however should not affect the natural frequencies of the system and thus the comparison of the DTDs. The controller code needed for comparing the drive-trains is straightforward. Care should be taken to stop the experiment in case the amplitudes of the measured loads become larger than those found under operation with the original DTD, thus ensuring that the badly tuned dampers do not damage the wind turbine.

It should be noted at this point that an alternative to measuring the gearbox torque would be to use the blade root bending load signals. These signals will most probably be available in state of the art wind turbines with independent pitch control.

Through chapter 4.1, the methodology followed to produce a self-tuned DTD has been explained in detail. First, an experiment is being run, where the wind turbine works with pre-defined settings in order to collect required data. These data are then used in a system identification procedure, where a linearized model of the wind turbine drive-train is created. This model describes the relationship between the generator torque and the generator speed. Using this model, a band-pass filter is tuned in order to operate as the drive-train damper. The effectiveness of this methodology to produce good quality DTDs is analysed in the results section.

4.2 Other self-tuned DTDs

Other self-tuning DTDs were investigated as part of this research. These included: a double band-pass filter tuned by a procedure similar to the one finally adopted and presented through this thesis, a PI based controller, a neural network based controller, and a controller based on load prediction.

As the self-tuned DTDs obtained by the above-mentioned methodologies were sub-optimal on their own, the development of these methodologies was abandoned at some point during the research. However, as it might be informative to the reader, these dampers and the methodology for creating them will be briefly outlined in this section.

4.2.1 Double Band Pass Based Filter

Considerable effort was placed in trying to develop a self-tuning procedure that adjusts a double band-pass filter, i.e. a cascade of two single band pass filters. A number of tuning procedures were developed including a number of procedures that tuned one single band-pass filter and then tuned again a second band-pass filter, and a number of procedures that tuned the double band pass filter in one stage. Unfortunately, all failed to produce consistent results. Moreover, adjusting the gain and the position of two poles and one zero at the same time (as in the procedures that tune the double band-pass filter in one stage, which were found to be the best performing tuning procedures for this filter type) produces a very large number of possible filter realizations that need to be checked. This results in a very slow procedure. In practice, this means that a PC connected to an actual wind turbine would need approximately a week to self-tune a double band-pass filter.

In order to make the self-tuning procedures quicker to run, a study was carried out to identify the most prominent filter zero pole locations but the results were mostly inconclusive. For instance, the best results for one model were obtained when both poles of the filter were located in the high damping regions of the root locus plot. This, however, did not hold true for other models tested. The trend was that the poles needed to be spaced apart, one being in the high frequency region and the other in the low frequency region. N.B. Low frequency region is the region that is close to the lowest coupled drive-train resonant frequency of the system, and high frequency region is the one being close to the highest coupled drive-train resonant frequency.

As the computational time was quite extensive, and the results obtained were not consistent, the use of a double band-pass filter acting as the self-tuned DTD was abandoned. It must be noted however that in some cases, a manually tuned double band-pass filter sometimes gives better results than a single band pass filter [[85]]. In these cases, a manually tuned double band-pass filter is chosen instead of a single band pass filter, and thus automatically tuning a double band pass filter would be very beneficial for some wind turbines.

4.2.2 PI / Neural Network Controller

The basic principle behind the PI controller based DTD and the Neural Network controller based DTD is similar. The generator acceleration would be measured and a controller would try to adjust the generator torque to limit its acceleration. It was assumed that this could reduce the gearbox torque fluctuations.

Both these controllers were tested using MATLAB's Simulink and FAST's S-function interface. FAST [87] is an Aeroelastic Design Code for Horizontal Axis Wind Turbines, developed by the Oregon State University for NREL. Although this code is not as complicated as GH Bladed, it is still quite advanced and has been verified by Germanischer Lloyd for the calculation of loads on wind turbines. As it allows the interface with MATLAB's Simulink, it has been used to test various innovative damping methodologies including the PI and Neural Network controller based dampers presented here.

In order to create a self-tuned DTD that effectively damps the drive-train, these controllers would need to be tuned with an automated procedure, based on the system identified model of the drive-train.

In the first implementation, a PI controller was used. In this case the derivative of the generator speed, i.e. the generator acceleration, was fed into a high pass filter set to filter out frequencies below 1.5 Hz and then into the PI controller. Finally, the signal from the PI controller was added to the normal torque demanded by the wind turbine controller. The generator acceleration was fed into the high pass filter as the controller should not regulate the generator at zero acceleration (the wind turbine must be left to vary its speed) but it should try to minimize only high frequency accelerations (due to structural frequency excitations) thus minimizing the gearbox loading. An experiment was run and the data obtained were used in order to system identify the generator torque – filtered generator acceleration relationship. The identified system was then used in order to tune the PI controller. A specific implementation of this PI based system is shown graphically in Figure 40:

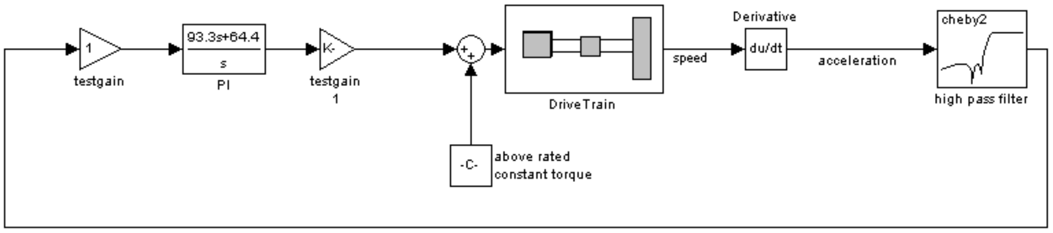


Figure 40: PI controller based system.

Note that the block labelled “Drive Train” in Figure 40 is not a simple drive-train model but a FAST S-function running a simulation of the whole wind turbine under turbulent wind.

In the second implementation, a neural network controller was used. Except for the actual controller, the rest of the system is the same as with the PI based system. The neural network controller used is a feedback linearization control or NARMA-L2 controller [88]. This controller was selected as it can be trained off-line and in contrast to other neural network controllers like the Model Predictive Controller, requires very modest computational power when working. The first step in using such a controller is to train its neural network to represent the forward dynamics of the system. After doing this, the NARMA-L2 controller should regulate its output in a way that makes the system follow the reference signal. The reference signal is set to zero so that the NARMA-L2 controller will try to reduce all high frequency accelerations of the system. A specific implementation of this NARMA-L2 controller based system is shown graphically in Figure 41 below:

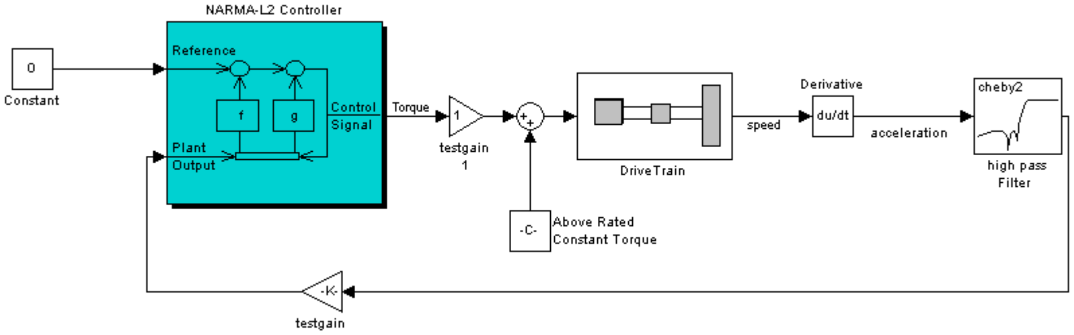


Figure 41: NARMA-L2 controller based system.

The results obtained by both these systems were discouraging. The controllers did not seem to be able to control the system and produce any reduction in gearbox loading. In fact, the controllers increased gearbox torque fluctuations and could not be configured in a way that would produce even the most modest load reductions. A comparison of the gearbox torque with and without the PI controller in a section of a typical experiment is presented in Figure 42:

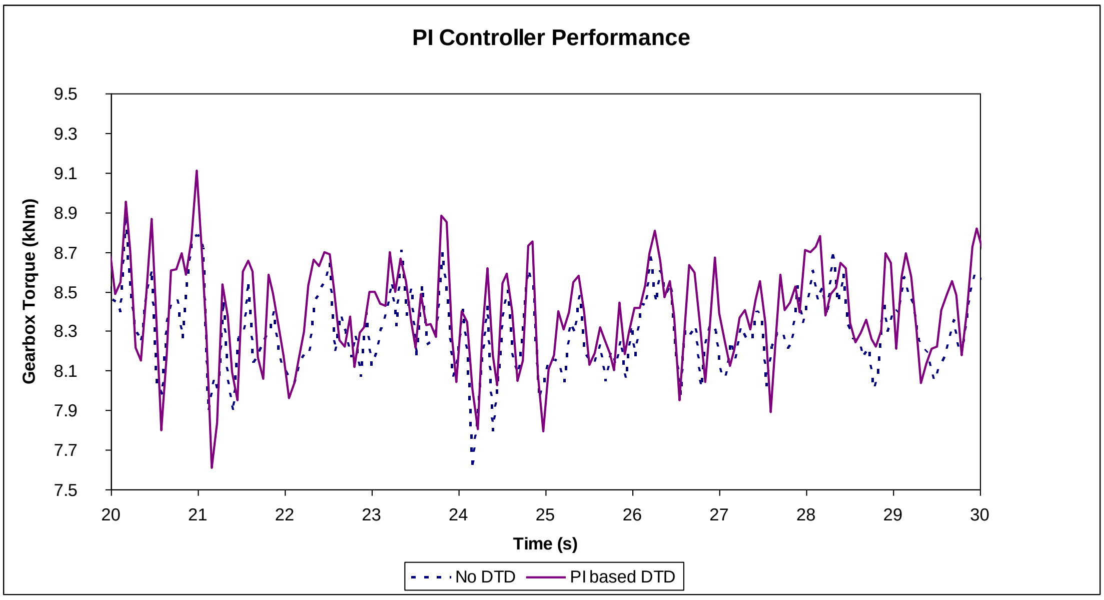


Figure 42: Typical performance of the PI controller based DTD.

A similar graph is presented for the neural network based controller in Figure 43:

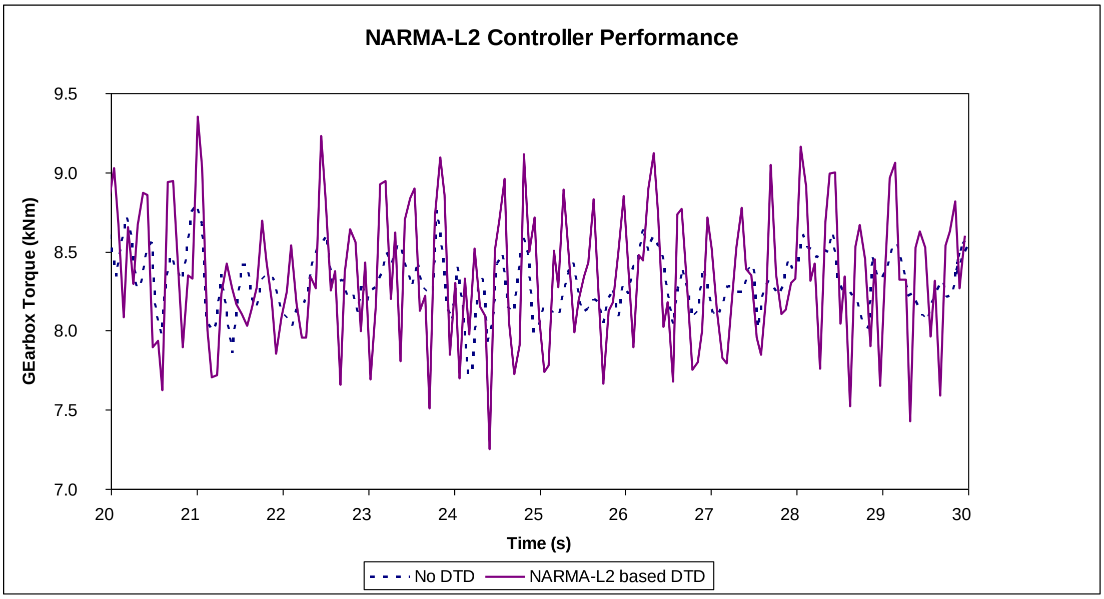


Figure 43: Typical performance of the NARMA-L2 controller based DTD.

As the above examples' results clearly indicate, these systems were not successful in reducing the gearbox loads. The NARMA –L2 controller implemented here shows more exaggerated oscillatory control signals, something that seems to be in line with what is observed by other researchers [89] using this type of network controller.

Moreover, using the derivative of any measured signal (in this case the acceleration) for control purposes is not advisable, as the derivative of a signal amplifies noise. So even if more development of these techniques could lead to small gearbox load reductions in theory, in practice, noise on the generator speed signal would probably hinder the performance of the actual damper, cancelling any benefits predicted by the simulated experiments. Thus, any further investigation into these types of drive-train damping methods was abandoned.

4.2.3 Load Prediction Technique

Other, completely novel, drive-train damping techniques were also investigated. The most effective of these was inspired by the noise cancellation procedure used in audio applications. The basic philosophy behind this methodology is that one predicts the gearbox torque of the next time step and demands a generator torque that counteracts this gearbox torque in the next time step. This is done by using two neural networks.

The first neural network is a linear network used to correlate generator speed to the gearbox torque. This neural network is adjusted once and requires two time series: one of the generator speed and one of the gearbox torque. Thus, either a power production simulation of the wind turbine (using any simulation software) or measured data from the actual turbine is needed. This means that this damping technique will require either a good model of the turbine to be available or would require a gearbox torque sensor on the turbine (at least for enough time to record an adequate period of normal turbine operation). This neural network was realised by a single linear layer neural network with 16 inputs, also called delays as they are the same signal sampled at the current plus last 15 time-steps, and a single neuron. Once this neural network is adjusted, the only data required are a constant stream of the generator speed from the wind turbine controller, allowing an estimation of the gearbox torque.

The second neural network is a predictive filter. This neural network predicts the gearbox torque at the next time step based on the estimated gearbox torque of the current time step and that of the previous 13 time steps. The number of time steps was set by a trial and error approach. Including more time steps in the neural network did not improve the results, so there was no need to increase the number of time steps beyond 13. Reducing the time steps, however, decreased the accuracy of the neural network. Thus, 13 time steps were used in the neural network. This neural network is a linear neural network, with one layer and one neuron, but unlike the first neural network used, its weights and biases are adjusted at each time step (each time it receives a new input). More on this neural network structure can be found in related bibliography [[90]]. The predicted gearbox torque signal is then multiplied by a constant and added to the normal torque. The value of this constant was set by manually trying a number of constant amplitudes and selecting the optimal value for the specific wind turbine model.

Although, initially, this load prediction technique was designed so that it would act as the drive-train damper, it was found that it was not sufficient on its own. If, however, this load prediction based damper is used in conjunction with the main drive-train damping filter it can further reduce gearbox loads even if the main drive-train damping filter is well tuned.

Figure 44 shows the gearbox torque of a normal run and a run employing the load cancellation technique described above.

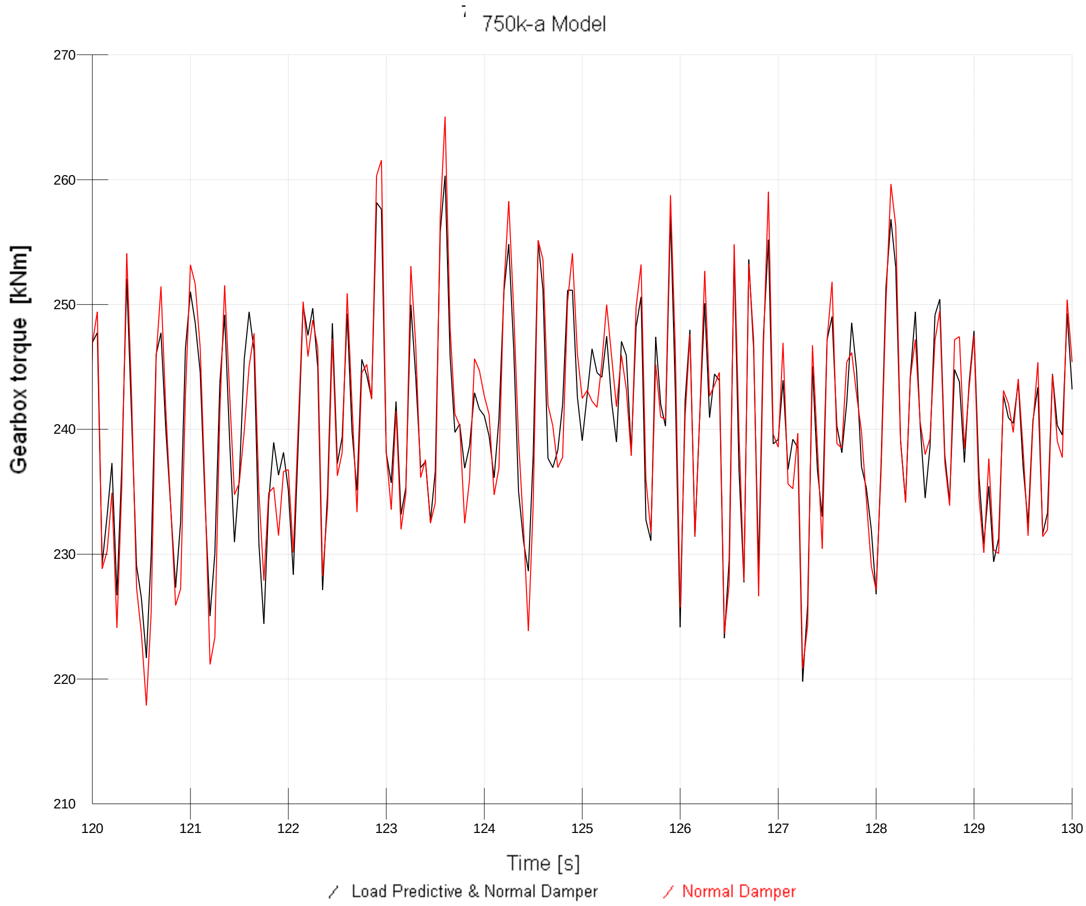


Figure 44: Time series of gearbox torque with and without the load cancellation algorithm.

As one can see from the above figure, the gearbox torque shows an identifiable reduction in fluctuations which is enough to give a small reduction in gearbox damage. Note that this damage reduction was approximately 5% in a 18m/s wind speed simulation. Therefore, in its life-time, the gearbox should be expected to attain a 1-2% damage reduction, as the drive-train damper is not affecting the loads as much at lower wind speeds, and only marginally at the below rated region. Figures 45 to 47 show the behaviour of the generator torque demand, the generator speed and the pitch angle of the blades:

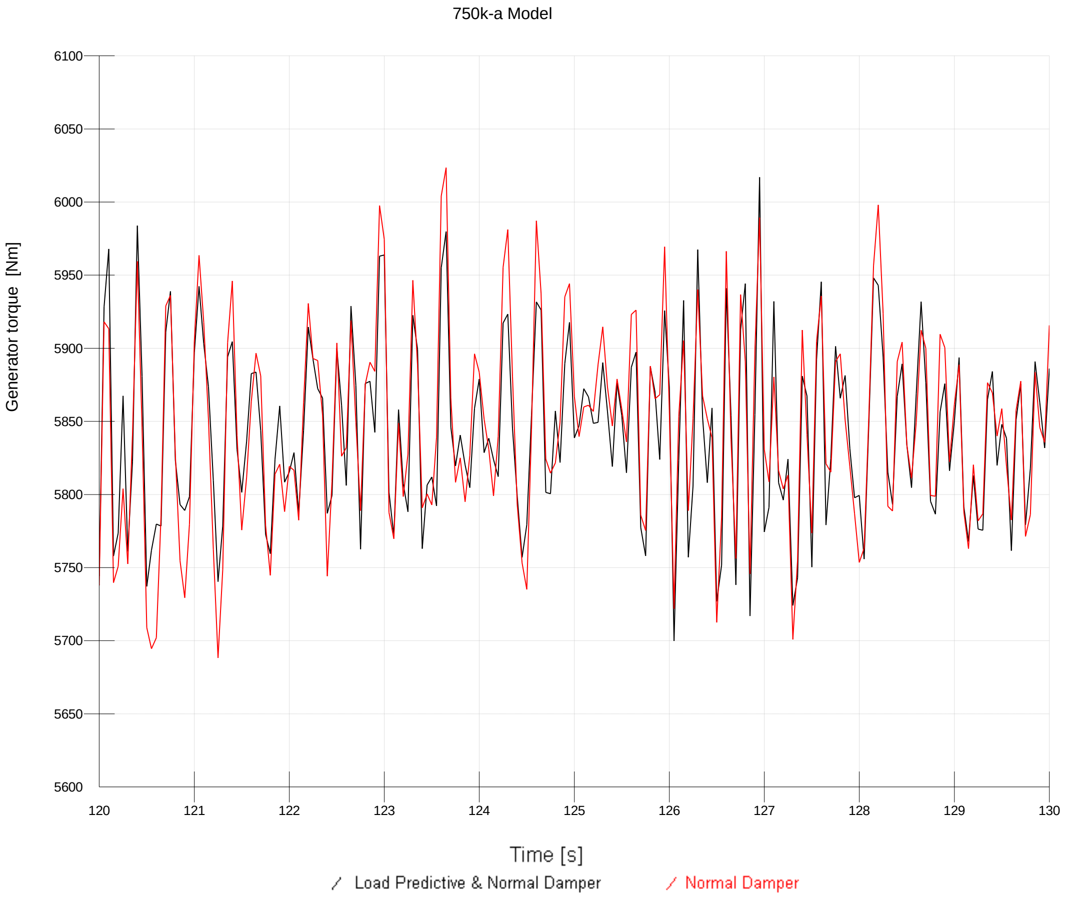


Figure 45: Time series of generator torque with and without the load cancellation algorithm.

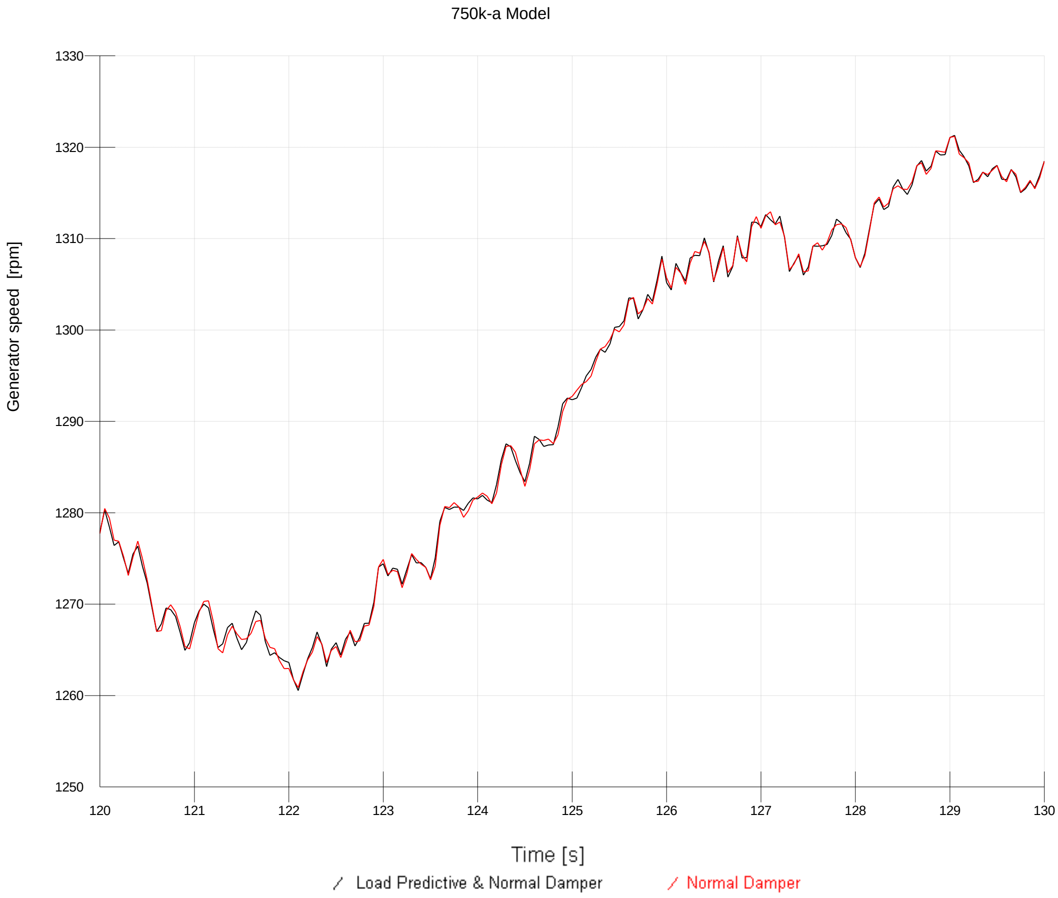


Figure 46: Time series of generator speed with and without the load cancellation algorithm.

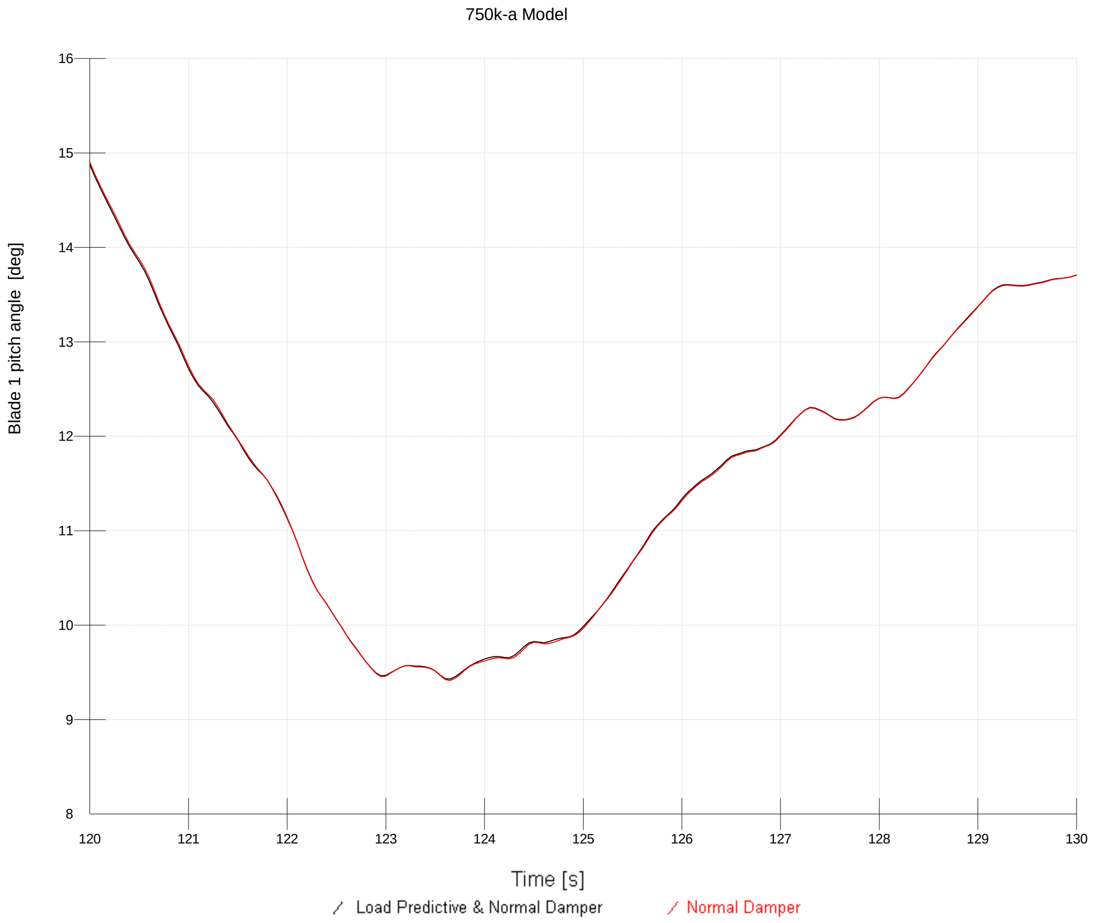


Figure 47: Time series of the blade pitch angle with and without the load cancellation algorithm.

From the above figures, one can see that the gearbox load reduction has been achieved by effectively reducing the generator torque fluctuations. This has been accomplished with no adverse effect on either the speed of the generator (the speed fluctuations are kept to the same levels) or the blade pitch actuation (no excessive pitching is observed).

As this predictive torque cancellation technique cannot be used on its own, further investigation has been deemed unnecessary, since the scope of this research is to develop a self-tuned drive-train damper that can be used on its own. Thus, only one test run using one wind turbine was used to validate its effectiveness. This technique did seem to work and gave a reduction of 5% in the gearbox DEL during the simulated test run.

One important detail to note is that the predictive neural network is being adjusted at every time step (0.05 second time steps are used for simulations). This means that only a dedicated processor can handle the task of adjusting the neural network and predicting the next time step in less than 50 milliseconds.

Chapter 5

In this chapter, the results obtained using different variations of the main methodology are compared, in order to show why a specific variation was chosen at each step. Finally, the results obtained using the self-tuned drive-train damper are analysed.

5.1 Results and Discussion

The self-tuning DTD procedure was run for all ten wind turbine models. As described in the previous chapter, the parameters and thus efficiency of the self-tuned DTD depends on the system linearization performed on experimental data (real or simulated in the case of this research). The usefulness of the experimental data collected depends on the specific ambient conditions prevailing during the experiment, and the state of the wind turbine at the beginning of the experiment. Thus, the efficiency of the self-tuned DTD depends on the random wind regime that the wind turbine was subjected to before and during the experimental procedure. In order to compare the results of the various alternatives of the self-tuning DTD procedure that were examined during the development of the procedure, a large number of experiments needed to be run for each self-tuning DTD procedure alternative. Of course, running a very large number of experiments would give the best basis for comparison, but this is practically unrealistic, as the whole procedure is quite computationally intensive and consequently time consuming. For each wind turbine model, the procedure was run 99 times. It should be noted that the exact number was not set to this level for any particular reason; it could be set to 95 or 105 without any notable effect on the results obtained by this research. The resulting DTDs were imported into the models and normal power production simulations using the same 18m/s mean wind speed for each wind turbine model were run. Running the procedure 99 times, for ten wind turbine models, and then using the resulting DTDs in order to validate their effectiveness was fully automated, and took approximately five to six days each time it was run.

The main objective of having a well-performing DTD is to reduce the loads on the drive-train as much as possible, and more specifically to increase the life-time of the gearbox by reducing the gearbox torque fluctuations. In order to quantify the lifetime increase of the gearbox, the Damage Equivalent Load (DEL) of the gearbox torque is used. For the purposes of this research, the DEL of the gearbox torque over a 400 second power production experiment at an 18m/s mean wind speed regime is used. The DEL is a sinusoidal load of the specified frequency that would create the same damage to the component under question as the complex load pattern experienced in reality. Thus, the lower the magnitude of the DEL, the lower the fatigue damage to components and thus the components have longer operational lifetimes. The DEL load depends on the number of actual load cycles and the load cycle amplitude experienced by the component under investigation. Likewise, the DEL load depends on the S-N slope* of the component and the specified frequency of the equivalent load (a detailed explanation is given in [65]). In this research, an inverse S-N slope of 3 was used and the equivalent load's sinusoidal frequency was set to 0.4 Hz. The frequency was chosen so that it is close to 1P for all turbine models. The inverse S-N slope of 3 was selected as it is the suggested value for bearings in a gearbox [91]. Note that a variety of S-N slopes need to be considered for a gearbox, depending on the specific component one needs to design since the S-N slope depends on the material, heat treatment, stress concentration and loading mode [91,92]. Note also that the DEL approach is too simplistic for the actual detailed design of a gearbox, where it might be more sensible to use three-dimensional Markov matrices, which are then fed into Finite Element Method (FEM) analysis software. On the other hand, the DEL representation is very compact and thus facilitates a meaningful presentation of the damage on the gearbox.

The self-tuning DTD procedure was run 99 times per wind turbine model, each time with a different wind realisation. However, some of the experiments (stage 1 of the procedure) did not produce sufficiently large numbers of *data*, i.e. the valid experiment period was less than eighty seconds, and thus they were automatically screened out. This was more apparent in the chirp based rather than the PRBS based experiments.

During the development of the self-tuning DTD procedure, a number of alternatives were tried out and the most appropriate one was chosen through the analysis of the results. A number of such alternatives have been discussed in the previous chapter and will be presented here again, along with the effect these alternative approaches have on the results. This will, hopefully, give more insight into the reasons the final methodology was chosen.

[*] S-N slope is the slope of the S-N curve of a component. The S-N curve is a curve made up from points where N is the number of load reversals which will cause structural failure at a specific stress amplitude S for the specific component. As many loading cycles of different stress amplitudes must be converted into a single 'equivalent' fixed amplitude loading cycle, the slope of the S-N curve (i.e. the sensitivity of the component to stress fluctuation amplitude) needs to be taken into account.

Resetting the Chirp Signal

A chirp signal was used as the control signal for the generator torque demand during the experiment in order to excite the drive-train natural frequencies. This signal was initially created as a continuously increasing frequency signal. This allowed the signal to cover the whole relevant frequency region once, and then by aliasing after 10Hz (the controller communication interval is 0.05s) the frequency of the signal became in essence random. If this signal is reset and starts the sweep through the interesting frequency range as soon as the signal goes over 7.5Hz, one would expect a more concise coverage of the frequencies that affect the drive-train and thus a better system identification result. In fact, as was mentioned in the methodology, it has been found that resetting the signal gives better results in most cases.

The direct results of the system identification are difficult to compare because 1) hundreds of identified models are assessed simultaneously and 2) standard assessment methodologies are not effective due to the large impact of unmeasured disturbances on the resulting system. Thus, instead of trying to assess the system identification results, the quality of the tuned DTDs is directly assessed. As previously explained, the efficiency of the drive-train damper is determined on the basis of the magnitude of the DEL on the gearbox.

Figures 48 to 50 show a comparison between the results obtained by a continuously increasing chirp signal, and resetting the chirp signal for three representative models, namely 750k-a, 2M-a and 3M-a.

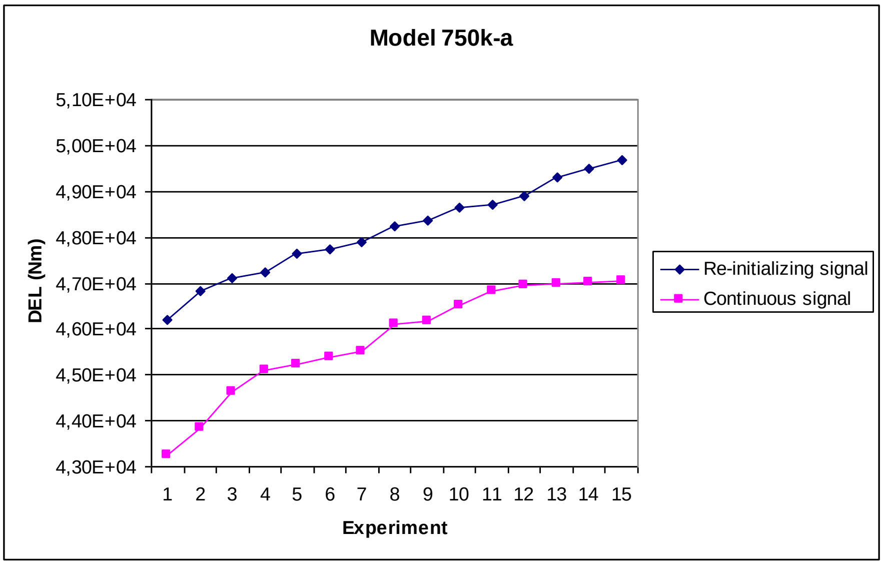


Figure 48: Performance of the top 15 DTDs for the Model 750k-a wind turbine, as they were tuned using experiments run with the re-initialized chirp signal and experiments run with the continuous signal.

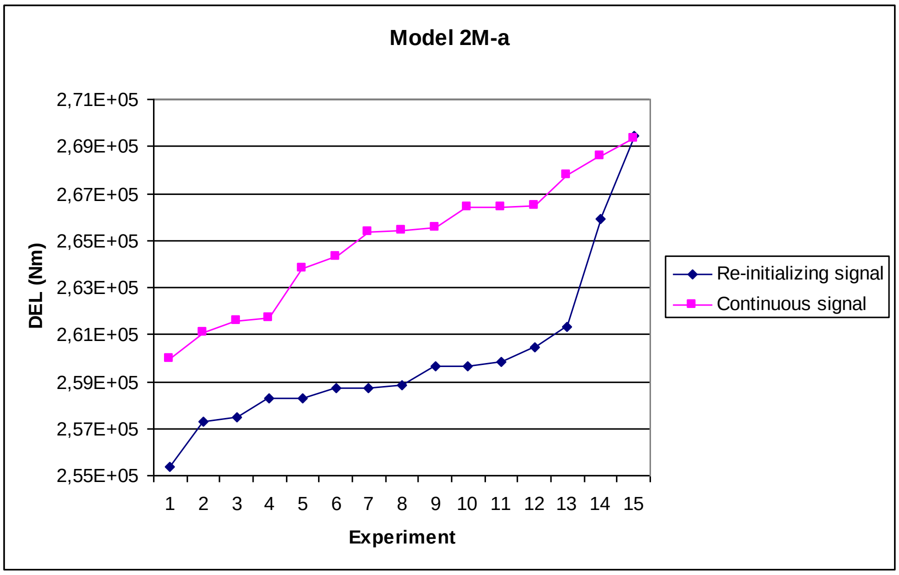


Figure 49: Performance of the top 15 DTDs for the Model 2M-a wind turbine, as they were tuned using experiments run with the re-initialized chirp signal and experiments run with the continuous signal.

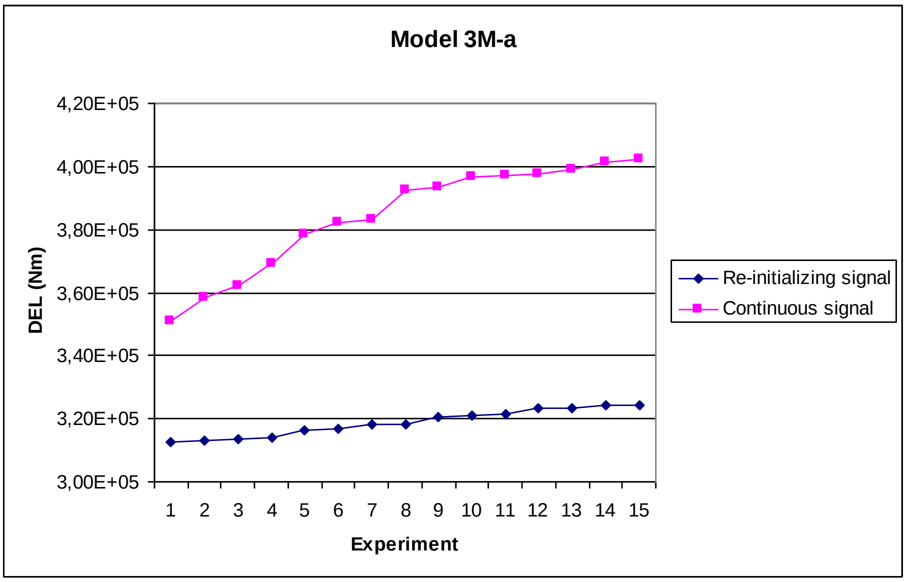


Figure 50: Performance of the top 15 DTDs for the Model 3M-a wind turbine, as they were tuned using experiments run with the re-initialized chirp signal and experiments run with the continuous signal.

As it is evident from the above graphs, the performance of the tuning methodology is better when using the continuously changing chirp signal for the 750k-a model, and the re-initialized chirp signal for the 2M-a and 3M-a models.

As explained in the previous paragraphs, a more concise coverage of the frequencies that affect the drive-train should in theory give a better system identification result. Analysing the results for all eight 2M and 3M models confirms that using the re-initialized chirp signal provides better results than using the aliasing chirp signal as expected.

Both 750k models, however, show a similar behaviour to the 750k-a model: that is, better results are obtained when using the aliasing chirp signal. The reason behind this behaviour can be attributed to the fact that the 750k models have higher natural frequencies that need to be excited to provide useful inputs for the identification procedure. Note that in the duration of a re-initialized chirp signal based experiment, the relevant natural frequencies for the 750k models are usually excited once, whilst for the 2M and 3M models, these frequencies are lower and thus there is enough time during the experiment to pass twice through these frequencies. Therefore, a reinitializing signal passes through the frequencies that excite the drive-train less often than the aliasing signal in the case of the 750k models.

This leads to the conclusion that the excitation frequency range used in the experiment plays an important role in the correct tuning of the DTD. A combination of a re-initialized chirp signal, and a restriction of the frequency excitation range to only the frequency range around the drive-train modes is thus recommended. This allows the drive-train natural frequencies to be excited more than once, and thus produces more useful results.

Choosing a multiple input System Identification procedure

As already discussed in the methodology section, in the ‘Blade Pass Correction’ system identification procedure, a signal based on the azimuth of the rotor is introduced in the system identification as a known disturbance. The reason for adding this input signal is to eliminate, or at least reduce, the effect of the blade pass interference in the "Generator torque" - "Generator speed" relationship.

First, the rotor azimuth was fed into the system identification routine as a second input (the first input being the Generator torque as in the original procedure). This did not seem to affect the identified system. According to bibliography [73, page 10], good practice in model linearization includes converting the input signals in a way that will promote the simplest correlation between the input and the output. Thus, the azimuth signal was modified in two ways in order to achieve the best correlation between the input and the output. The first azimuth based signal is called the ‘1P Load Indicator’ and the second one is called the ‘3P Load Indicator’. The conversion of the first signal is based on the rotor position, whilst the second signal conversion is based on the blade position:

For the ‘1P Load Indicator’, azimuths from 0 to 180 degrees are converted linearly to values going from 0 (for an azimuth of 0 degrees) up to 1 (for an azimuth of 180 degrees). Azimuths from 180 to 360 degrees are converted linearly to values going from 1 (for an azimuth of 180 degrees) down to 0 (for an azimuth of 360 degrees).

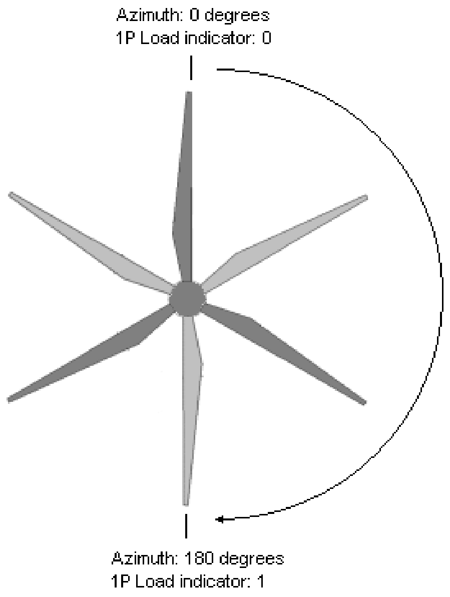


Figure 51: 1P load indicator.

For the ‘3P Load Indicator’, azimuths from 0 to 60 degrees are converted linearly to values going from 0 (for an azimuth of 0 degrees) up to 1 (for an azimuth of 60 degrees), azimuths from 60 to 120 degrees are converted linearly to values going from 1 (for an azimuth of 60 degrees) down to 0 (for an azimuth of 120 degrees), and so on for the full 360 degrees.

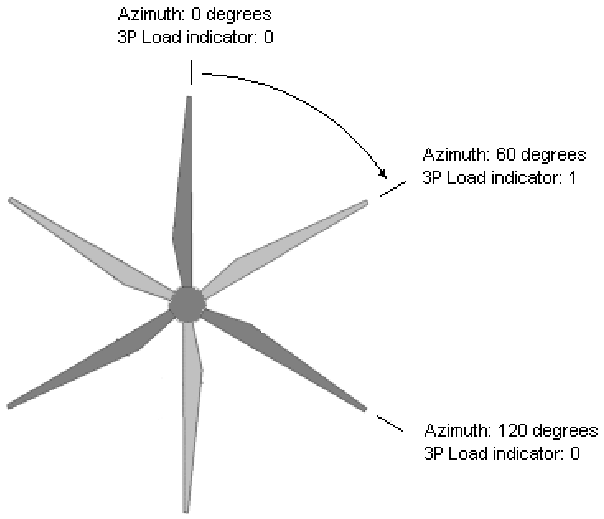


Figure 52: 3P load indicator.

Including either of the “Azimuth Indicators”, resulted in considerably more consistent DTD performance. The figures presented below show the DTD performance in reducing the loading on the gearbox obtained by using the various linearization procedures: the single input procedure, the multi-input procedure using ‘1P Load Indicator’ and the multi-input procedure using ‘3P Load Indicator’.

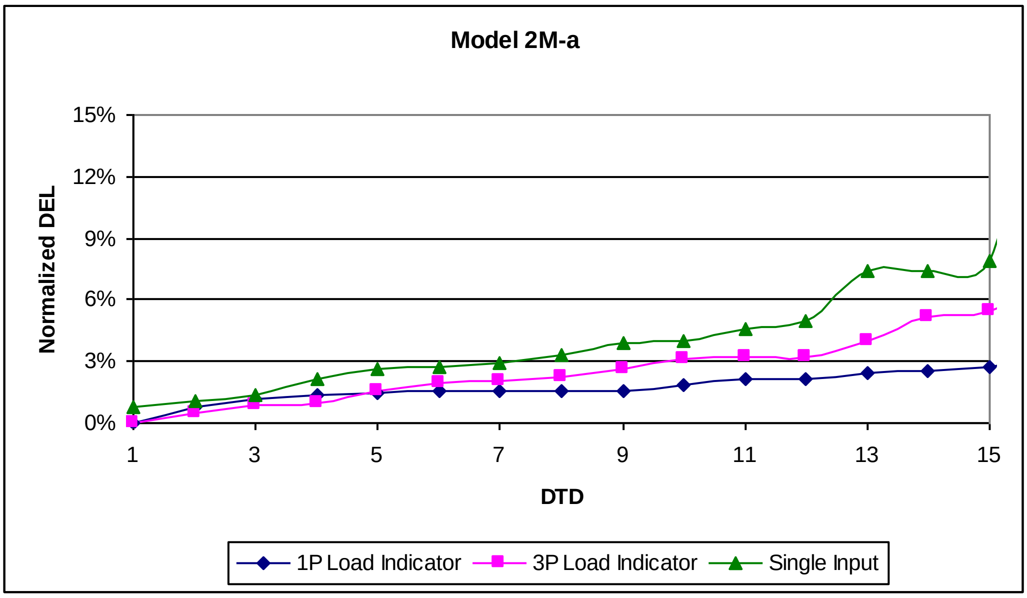


Figure 53: The performance of the DTD tuned based on various system identification methods (2M-a wind turbine Model).

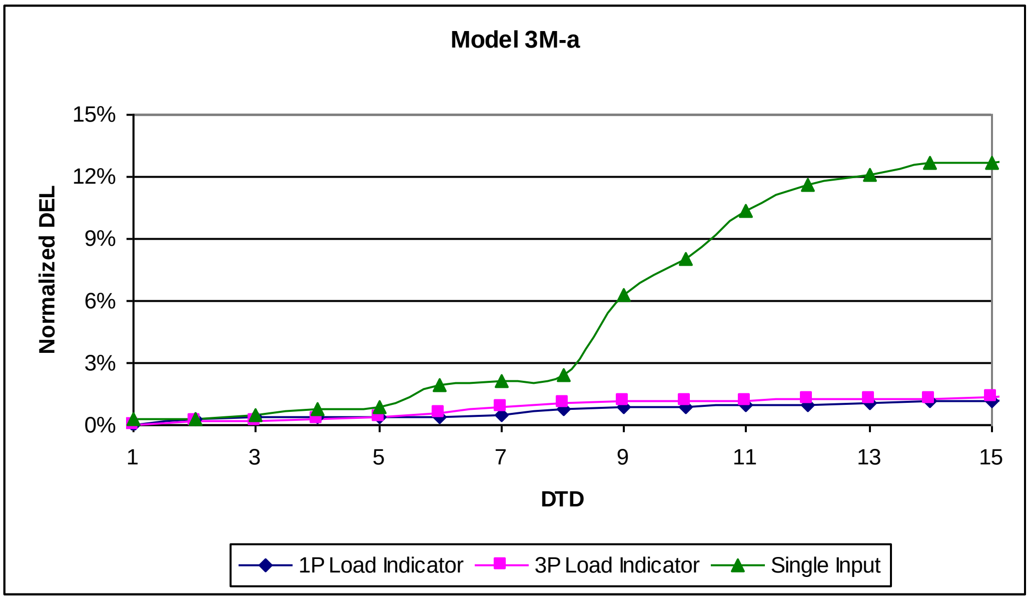


Figure 54: The performance of the DTD tuned based on various system identification methods (3M-a wind turbine Model).

Examining the results one can identify that the consistency of the DTD performance improves when using two inputs to perform the system identification. This was expected, as the second input, an input based on the rotor azimuth, has been introduced in order to correlate the output variations, that are caused by the interference due to the blade passing in front of the tower, with the position of the blades. By making this correlation, the generator torque – generator speed correlation is modelled with greater accuracy. As the drive-train damper is tuned based on this model, constructing the model more accurately facilitates the tuning of a better performing damper.

Based on the results obtained, using either “1P Load Indicator” or “3P Load Indicator”, comparable results are achieved. Nevertheless, using “1P Load Indicator” gives slightly more consistently performing DTDs. Thus, in its final implementation, the linearization procedure is based on two input signals, the generator torque, and the “1P Load Indicator”.

The large deviation in performance of the DTD shown in model 3M-a (Figure 54), between a single input and a multiple input identification procedure, demonstrates how great the effect of the blade pass interference can be in the construction of a good model. In this case, the increase in the quality of the models identified (and thus the

DTD tuned) is not caused by the reduction of the 3P interference, but it is caused by the reduction of the interference due to the 9P excitation of the structural natural frequency of the drive-train. Figure 55 shows model 3M-a identified by the single input system identification procedure and the 'Blade Pass Correction' system identification procedure:

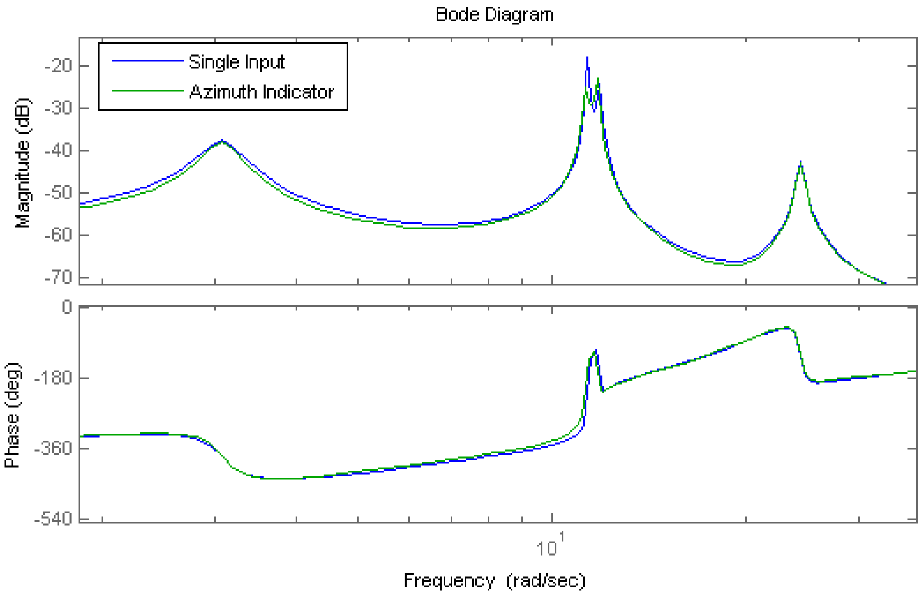


Figure 55: Example comparison between the single input and multi input system identified model (3M-a wind turbine Model).

From Figure 55 it is evident that the use of the azimuth indicator signal has not affected the large 3P interference shown at 3.1 rad/s. Figure 56 shows the region around the most significant structural natural frequency:

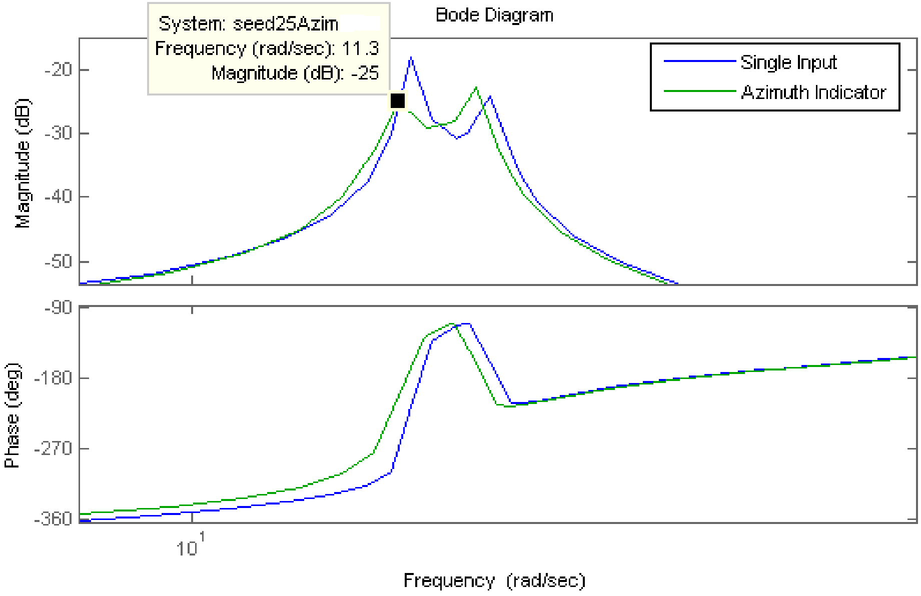


Figure 56: The effect of including the azimuth indicator input signal (3M-a wind turbine Model).

From Figure 56 it is evident that the use of the azimuth indicator signal has affected the model response near the first natural frequency of the system. During this experiment, the 9P loading frequency ranged from 8.7 to 14.1 rad/s. It seems that during the experiment, the 9P interference affected the first natural frequency of the system. By introducing the azimuth indicator input, the system identification procedure was able to distinguish between the 9P-induced excitation and the generator torque induced excitation, thus better identifying the model. In the specific example (model 3M-a, experiment 25), the DTD tuned using the ‘Blade Pass Correction’ system identification procedure shows a reduction of 4% in DEL with respect to the DEL achieved by the DTD tuned using the single input system identification procedure.

System Identification algorithm

As described in chapter 4.1.2 the N4SID system identification method was employed for the linearization procedure of this research. This method achieves the creation of a linearized model that most closely resembles the linearized model as predicted by the well referenced GH Bladed Software, and also allows the self tuning methodology to produce the best tuned DTDs.

Figure 57 below shows a Bode plot comparing the amplitude and phase responses of the various frequency models (also called nonparametric models) to the GH Bladed linearized model of the 2M-a wind turbine model.

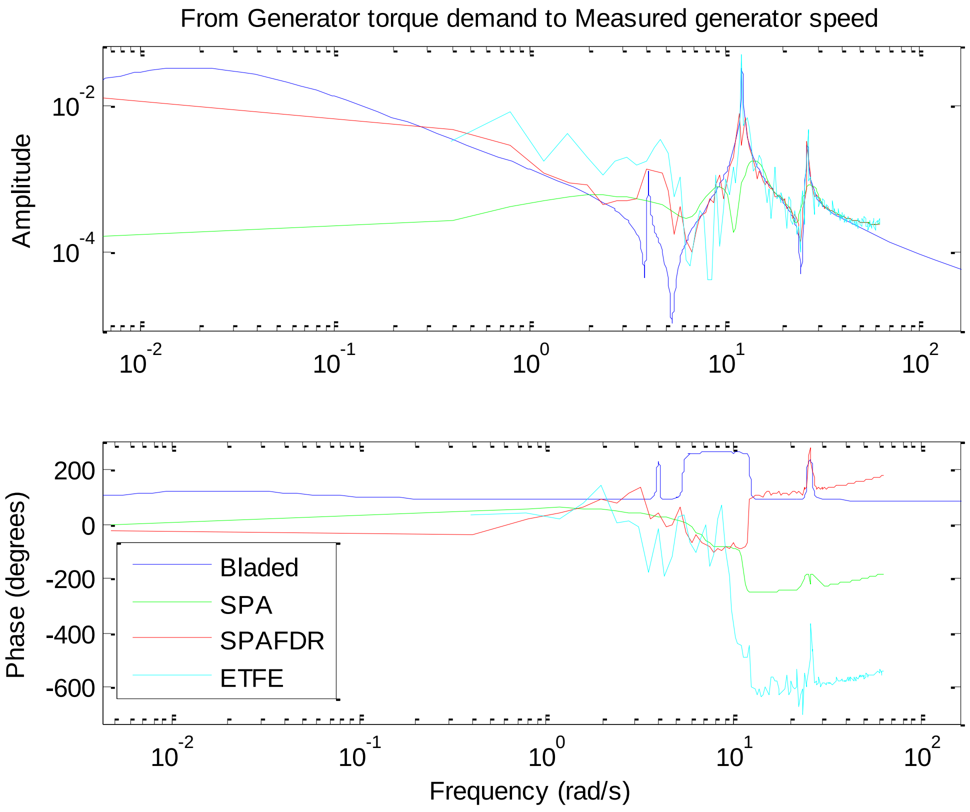


Figure 57: Amplitude and phase Bode plots of frequency response models compared to the GH Bladed linear model (Model 2M-a).

The responses labelled SPA, SPAFDR and ETFE in the Bode plots above are produced by frequency response models created by the SPA, SPAFDR and ETFE methodologies respectively, as described in chapter 4.1.2 of this thesis. It is evident from the above graphs that only the ETFE and SPAFDR models describe the generator torque demand to generator speed relationship with the required accuracy. Between these two models, the ETFE model seems to better represent the system dynamics in the region of the first resonant frequency (12.2 rad/s), as it shows a distinct peak of high magnitude at this frequency, closely resembling the response described by the GH Bladed linear model. The following figure shows the magnitude peaks created at the natural frequencies of the drive-train, as predicted by the various models:

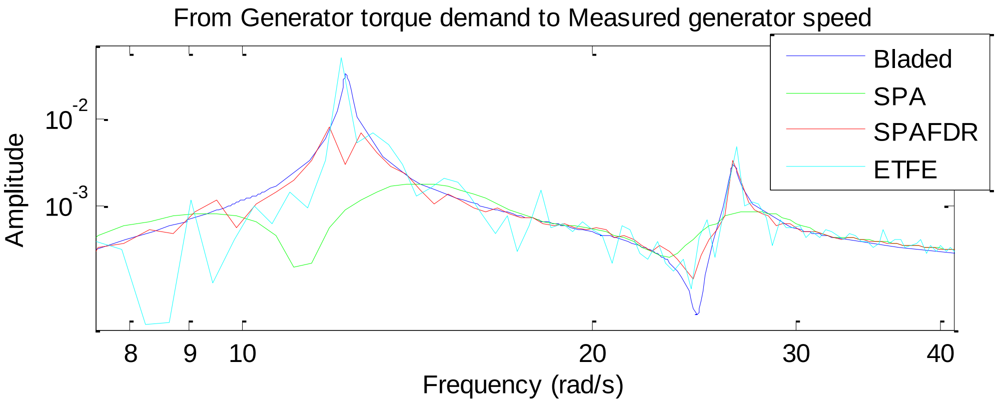


Figure 58: Amplitude Bode plot of frequency response models and GH Bladed linear model near the system's natural frequencies (Model 2M-a).

Note that for the above comparison, and all other similar comparisons following in this subsection, the system identified models - i.e. all models presented apart from the GH Bladed linear models - are based on experimental data collected by the 'experimental procedure' as it was described in section 4.1.2 of this thesis. Thus for the same wind turbine model, a different system identified model of the drive-train is created depending on the experiment realisation selected. To make the comparison between the various system identified models and the GH Bladed linear models more appropriate, a particular experimental realisation is used. This realisation has shown to give good results when used with the self tuning DTD methodology developed by this research. In the case of the 2M-a model a PRBS experiment with the fifth wind realisation is used.

Next, a Bode plot comparing the amplitude and phase responses of the various ARX based models with the GH Bladed linearized model is shown in figure 59:

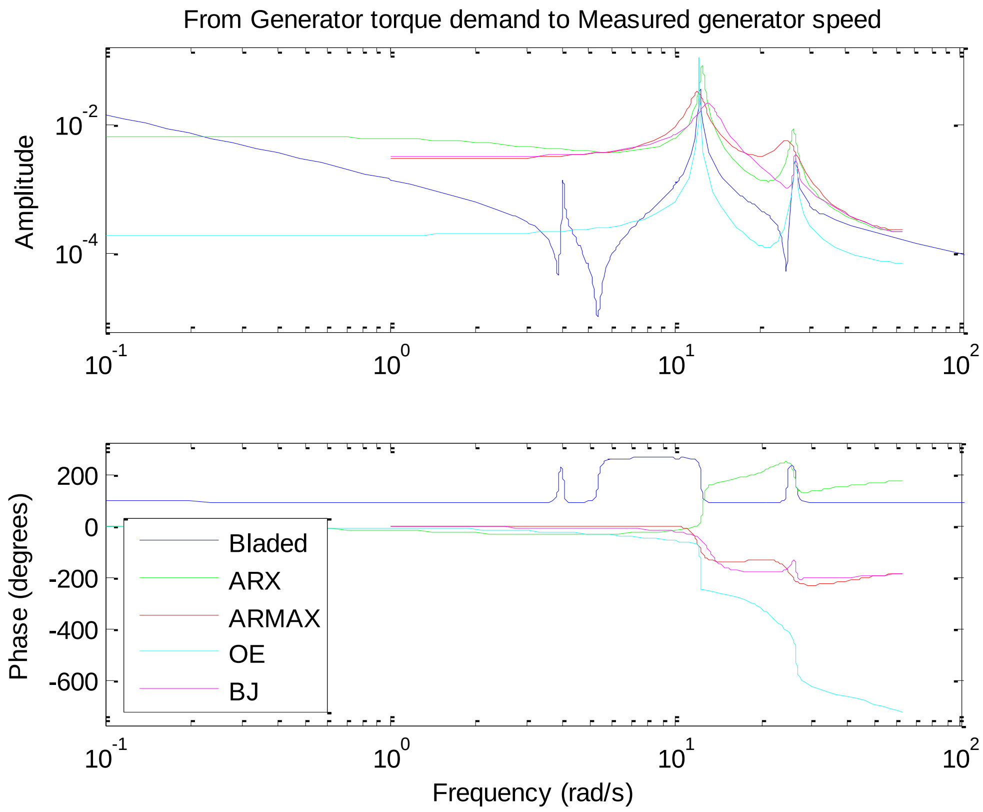


Figure 59: Amplitude and phase Bode plots of ARX based models compared to the GH Bladed linear model (Model 2M-a).

The responses labelled ARX, ARMAX, OE and BJ in the Bode plots above are produced by ARX based models created by the ARX, ARMAX, OE and BJ methodologies respectively, as described in chapter 4.1.2 of this thesis. Figure 60 shows the magnitude peaks created by the natural frequencies of the drive-train, as predicted by the various models:

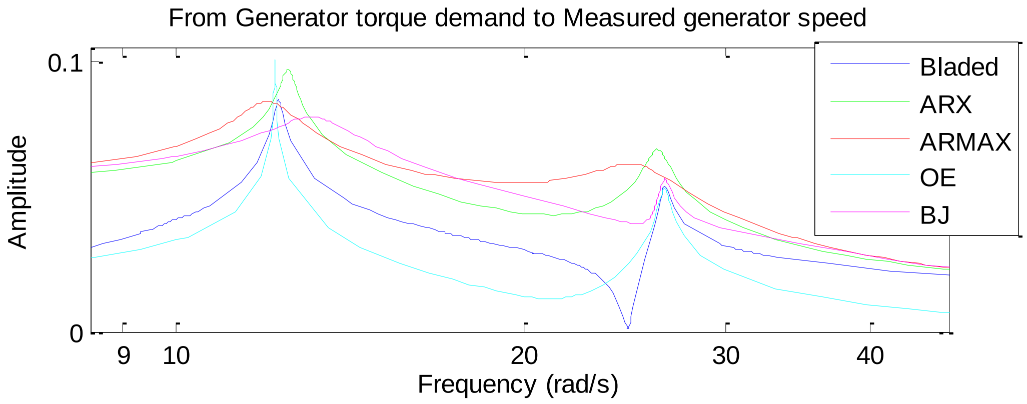


Figure 60: Amplitude Bode plot of ARX based and GH Bladed linearized models near the system's natural frequencies (Model 2M-a).

All ARX based models show comparable behaviour, but some distinctive features can be observed. The Box-Jenkins (BJ) model predicts a slightly higher frequency of the first mode, i.e. the first peak shown in the amplitude Bode plot, in comparison to the other models. The ARMAX model predicts slightly lower frequencies for the first two modes. Both the ARMAX and the BJ models show smoother amplitude peaks, and smaller phase shifts at the modal frequencies, when compared to the other three models.

It has to be noted that the above displayed frequency response of the various models is specific to the order of the appropriate polynomials in the models and the number of delays from input to output used. In the above examples, these parameters were set by a trial and error approach in order to force each model's frequency response to best match the frequency response of the GH Bladed linearized model.

Finally, a Bode plot comparing the magnitude and phase responses of the N4SID and PEM state-space models with the GH Bladed linearized model is shown in figure 61 below:

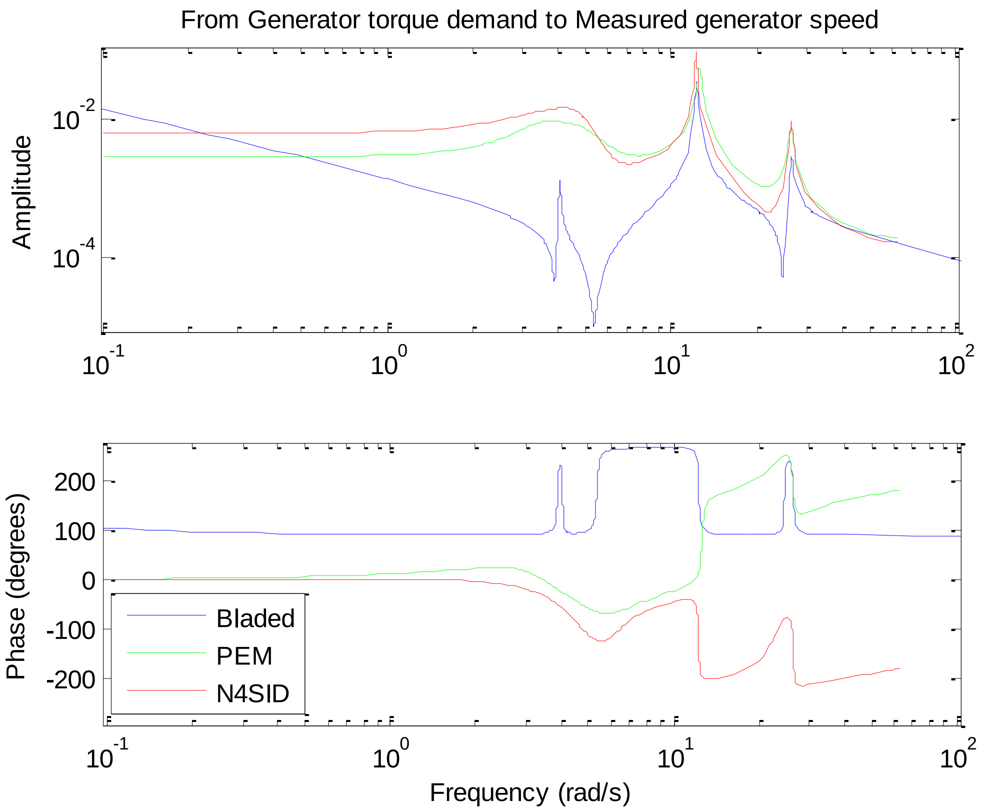


Figure 61: Amplitude and phase Bode plots of state space models compared to the GH Bladed linearized model (Model 2M-a).

The PEM and N4SID models presented in the Bode plots above are state-space models created by the PEM and N4SID methodologies respectively, as described in chapter 4.1.2 of this thesis. Figure 62 shows the magnitude peaks created by the natural frequencies of the drive-train, as predicted by the various models:

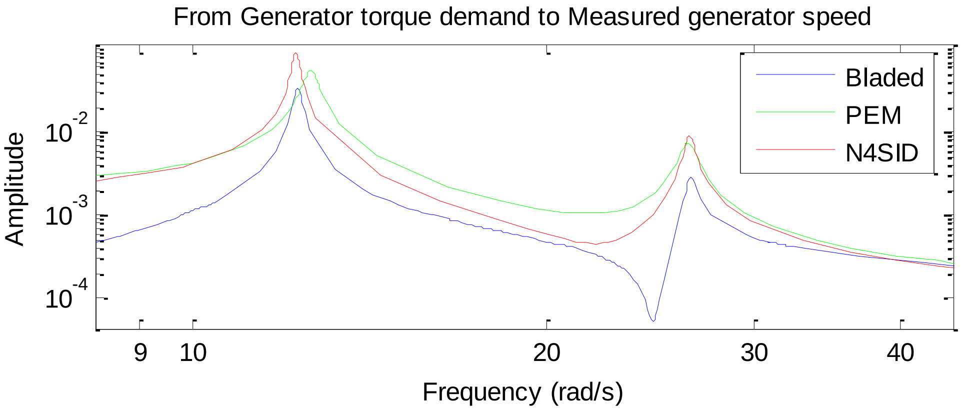


Figure 62: Amplitude Bode plot of state-space and GH Bladed linearized models near the system's natural frequencies (Model 2M-a).

Both the PEM and the N4SID models show comparable frequency responses, with only minor difference identified at the amplitudes predicted near the low frequency mode. In terms of the frequencies at which the system modes are identified at, the first mode identified by the N4SID model shows better correlation with the first mode indicated by the GH Bladed linearized model.

Again, it has to be noted at this point that the above displayed frequency responses of the various models is specific to their order. In the above examples the order of the models were set by a trial and error approach in order to force each model's frequency response to best match the one of the GH Bladed linearized model.

Apart for the ARMAX, SPA and BJ models, the remaining models showed similar amplitude and phase responses with respect to the GH Bladed Linearized model at the frequency range where the drive-train modes reside. Thus for the selection of the most appropriate model, a further comparison needs to be made. The self-tuning methodology created by this research was used to tune a DTD based on all the remaining system identified models. The DTDs tuned are compared against the DEL they allow on the 2M-a wind turbine gearbox during a 400s, 18m/s average wind speed with class A IEC turbulence, power production simulation in Table 2:

System Identification Method	Un-weighted DEL [Nm]	Delta from GH Bladed Linearized Model
GH Bladed Linearized Model	2.54E+05	0%
ARX	3.44E+05	36%
ETFE	2.66E+05	5%
SPAFDR	2.56E+05	1%
N4SID	2.53E+05	0%
OE	2.64E+05	4%
PEM	3.53E+05	39%

Table 2: DTD performance, based on different system identification methodologies (Model 2M-a).

A similar analysis run for the 3M-a and 750k-a models shows the following results in Tables 3 and 4, respectively:

System Identification Method	Un-weighted DEL [Nm]	Delta from GH Bladed Linearized Model
GH Bladed Linearized Model	3.12E+05	0%
ARX	6.54E+05	110%
ETFE	3.18E+05	2%
SPAFDR	3.16E+05	1%
N4SID	3.13E+05	0%
OE	3.15E+05	1%
PEM	3.18E+05	2%

Table 3: DTD performance, based on different system identification methodologies (Model 3M-a).

System Identification Method	Un-weighted DEL [Nm]	Delta from GH Bladed Linearized Model
GH Bladed Linearized Model	4.82E+04	0%
ARX	4.90E+04	2%
ETFE	4.74E+04	-2%
SPAFDR	7.26E+04	51%
N4SID	4.51E+04	-6%
OE	5.25E+04	9%
PEM	N/A	N/A

Table 4: DTD performance, based on different system identification methodologies (Model 750k-a).

As shown by the above results, the N4SID system identification methodology leads to the best tuned DTDs for all models, whilst other methods such as the ETFE and OE methods also show overall good results. No result is shown for the DTD tuned using the PEM based identified model for the 750k-a wind turbine model. This is because the DTD tuned was of poor quality and was discarded by the error checking mechanisms included in the automatic tuning procedure.

The above results constituted the main reason for selecting the N4SID system identification methodology for the self tuning-procedure. Another advantage of the N4SID method that led to its selection over other well performing methodologies, such as the ETFE and OE methods, was its suitability for use in an automatic procedure with no user input. For the N4SID method, only the model order needs to be set, whilst for the ETFE and OE methods, a number of parameter combinations need to be properly set in order to achieve the creation of a well-suited model that finally leads to the tuning of a well-tuned damper. In addition, MATLAB gives the flexibility to automatically select the order of the N4SID state-space model based on the decay rate of the singular values of the Hankel matrices of the impulse response. A knee in the decay rate of the singular value is identified, where a further increase in model order does not reduce the singular value appreciably. The order at this point is selected as the order of the state-space system to be identified. The exact way in which this knee in decay is selected, i.e. what further reduction in the Hankel matrices singular values is considered the minimum for identifying the knee in the decay, and thus the model order, is not disclosed. However, the results obtained by using MATLAB's automatic selection has proven to be adequate for all wind turbine models used in this study, so this methodology of automatic order selection was kept for the final version of the self-tuning procedure.

It must be noted here that there is no singularly optimal model order, as the number of distinct drive-train modes that need to be modelled correctly, is not the same for all the models. For example, model 750k-a has three identifiable drive-train modes that need to be correctly modelled by the system identified state-space model (see Figure 26), whilst the 2M-a model only has two (see Figure 27).

To the knowledgeable reader, it might seem unusual that the above comparison between different system identification methodologies is based only on a) the similarity between the frequency responses of the GH Bladed identified model and the system identified models and b) the final results of the DTD procedure. One might for example expect a comparison based on a more conventional residual analysis. In fact, the developed methodology does not make use of any residual based analysis at all, not even for the selection of the most appropriate DTD amongst the ones created by different experiments. The reason behind this has been briefly discussed in section

4.1.2. The problem is that residual analysis tries to help us understand if the identified model agrees with observed data. Recognizing if the model agrees with observed data is not always the same as identifying if the model is a good representation of the true system. This would happen only if the following criteria were met: a) there were no excessive unmeasured disturbances affecting the output of the system during the experiment and b) the system identification experiment was run for long periods and thus collected a large number of 'observed data'. As neither (a) nor (b) can be sufficiently met for our purposes, the conformity of the identified model with the observed data serves a limited purpose.

The above argument can be highlighted by the use of a simple example. In the 2M-a models constructed based on the experiment using the 5th wind realisation, it was shown that the N4SID based model allowed for the tuning of a much better DTD than the ARX and PEM based models (see table 2). The root-mean-square (RMS) value of the residuals, i.e. the RMS error between the predicted and actual system output, for these models is 0.419, 0.418 and 0.400 for the N4SID, ARX and PEM based models respectively. If the selection of the most appropriate system identification methodology were based on the RMS of the residuals then the PEM methodology would have been selected, leading to substantially worse performing DTD to be tuned, as shown in table 2.

Cost functions (and the usefulness thereof)

As part of the DTD tuning algorithm presented in section 4.1.3, a number of cost functions used to indicate the best combination of DTD filter parameters were created. All these cost functions penalize the performance score of a combination of DTD filter parameters according to the system's closed loop poles' damping constant values.

As it was mentioned in section 4.1.3, as a general rule, the more damped the closed system poles are, the more damped the system is. The goal of the DTD is to damp the system poles created at the system's natural frequencies as much as possible in order to minimize the load amplification naturally occurring near the system's natural frequencies. Thus the cost functions are defined in a way that penalize the performance score of a combination of DTD filter parameters more when the system's closed loop poles' damping constant values are lower and less when they are higher. The exact penalty assigned to each damping constant value or range, is what differentiates one cost function from another.

It must be noted that apart from cost functions 1 to 4 already presented in section 4.1.3, another 15 cost functions were also used during the development of the tuning procedure. These do not show up in the results, as they did not lead to a well-tuned DTD. These 15 cost functions are presented below:

Costfunction 5 $= 0.6\cdot\text{vhighZ} + 1.1\cdot\text{highZ} + 2.1\cdot\text{medZ} + 4.1\cdot\text{lowZ} + 8.1\cdot\text{vlowZ} - 0.001\cdot\text{sumZ}$ **Eq. 30**

Costfunction 6 $= 0.1\cdot\text{vhighZ} + 5\cdot\text{highZ} + 20\cdot\text{medZ} + 200\cdot\text{lowZ} + 2000\cdot\text{vlowZ} - 0.001\cdot\text{sumZ}$ **Eq. 31**

Costfunction 7 $= 0.1\cdot\text{vhighZ} + 1\cdot\text{highZ} + 10\cdot\text{medZ} + 600\cdot\text{lowZ} + 1000\cdot\text{vlowZ} - 0.001\cdot\text{sumZ}$ **Eq. 32**

Costfunction 8 $= 0.1\cdot\text{vhighZ} + 1\cdot\text{highZ} + 50\cdot\text{medZ} + 200\cdot\text{lowZ} + 400\cdot\text{vlowZ} - 0.001\cdot\text{sumZ}$ **Eq. 33**

Costfunction 9 $= 0.1\cdot\text{vhighZ} + 0.5\cdot\text{highZ} + 3\cdot\text{medZ} + 30\cdot\text{lowZ} + 800\cdot\text{vlowZ} - 0.001\cdot\text{sumZ}$ **Eq. 34**

Costfunction 10$= 1\cdot\text{vhighZ} + 1\cdot\text{highZ} + 10\cdot\text{medZ} + 100\cdot\text{lowZ} + 101\cdot\text{vlowZ} - 0.001\cdot\text{sumZ}$ **Eq. 35**

Costfunction 11$= \sum_{x=\text{Npoles}}^{x=1} 3100\cdot\exp((-58.13\cdot Z(x))$ **Eq. 36**

Costfunction 12$= \sum_{x=\text{Npoles}}^{x=1} 3300\cdot\exp((-58.13\cdot Z(x))$ **Eq. 37**

Costfunction 13$= \sum_{x=\text{Npoles}}^{x=1} 3000\cdot\exp(-10\cdot Z(x))$ **Eq. 38**

Costfunction 14$= \sum_{x=\text{Npoles}}^{x=1} 2900\cdot\exp(-10\cdot Z(x))$ **Eq. 39**

Costfunction 15$= \sum_{x=\text{Npoles}}^{x=1} 3100\cdot\exp(-10\cdot Z(x))$ **Eq. 40**

$$\text{Costfunction 16=} \quad \sum_{x=\text{Npoles}}^{x=1} 3000 \cdot \exp(-11 \cdot Z(x)) \qquad \textbf{Eq. 41}$$

$$\text{Costfunction 17=} \quad \sum_{x=\text{Npoles}}^{x=1} 3000 \cdot \exp(-9 \cdot Z(x)) \qquad \textbf{Eq. 42}$$

$$\text{Costfunction 18=} \quad \sum_{x=\text{Npoles}}^{x=1} 3200 \cdot \exp(-60 \cdot Z(x)) \qquad \textbf{Eq. 43}$$

$$\text{Costfunction 19=} \quad \sum_{x=\text{Npoles}}^{x=1} 3200 \cdot \exp(-56 \cdot Z(x)) \qquad \textbf{Eq. 44}$$

The cost of each closed system pole pair is plotted against the damping ratio of the pole pair for all cost functions used in the following figure:

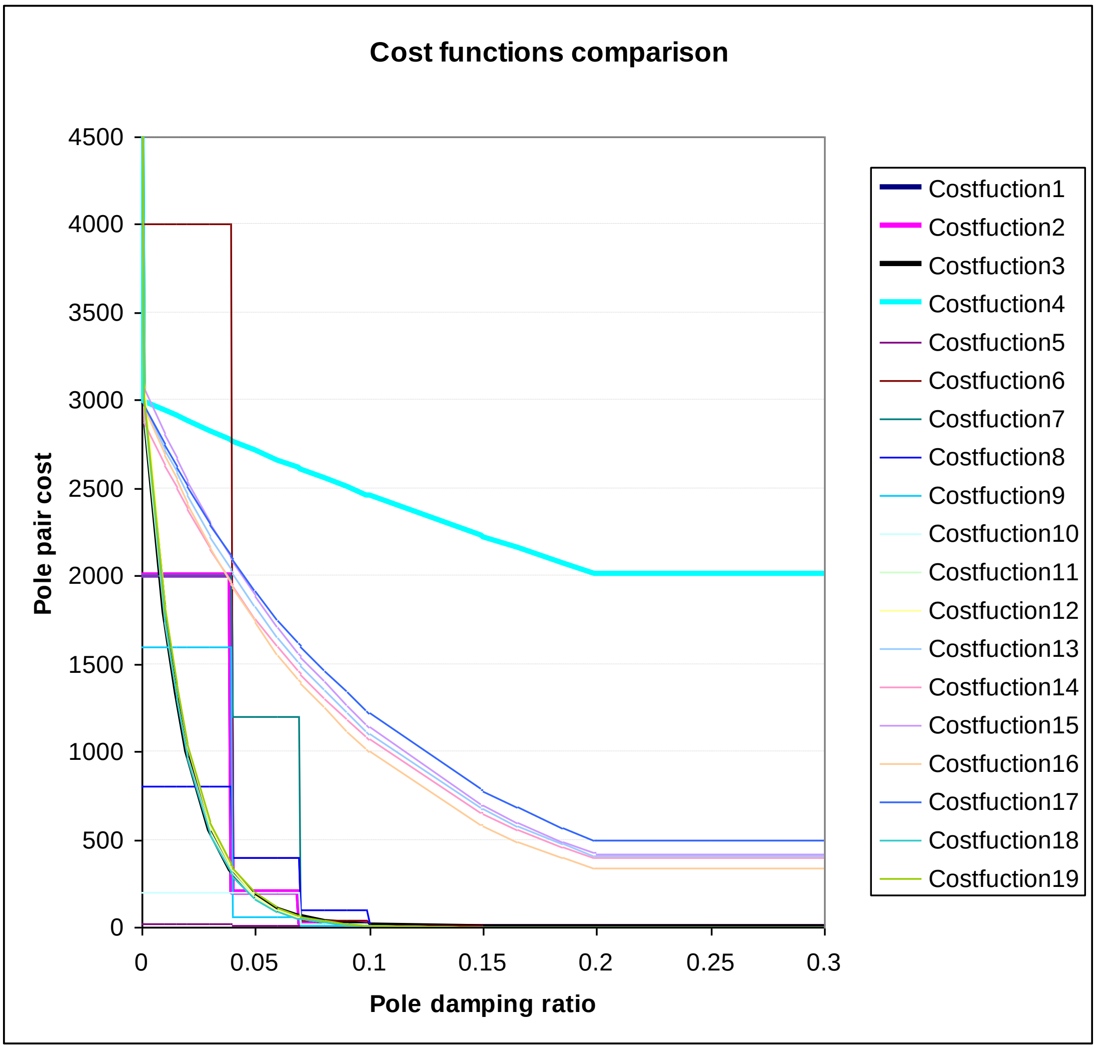


Figure 63: Cost function penalties.

One could argue that a total of 19 cost functions are not enough to evaluate whether the selected cost functions are appropriate or not. However, finding the best cost function by exploring a much larger set of possible combinations of penalties is not practically feasible by an iterative procedure. This is because it is hard to identify the best DTD based on a simplified and thus computationally inexpensive, test based on the linearized model of the drive-train only. In order to assess the quality of a DTD with some reasonable accuracy one needs to run a dynamic production simulation of the wind turbine.

An example proving the above statement follows:

Using model 2M-a, the performance of three DTDs are compared. DTD A is a well-tuned DTD. DTD B is a slightly worse DTD that would lead to 3% more DEL on the gearbox torque during a particular 400s, 18m/s average wind speed production run. Finally, DTD C is the worst of the three DTDs that would lead to 9% more DEL on the gearbox torque during the same production run when compared to DTD A.

Figures 64 and 65 below show the Bode diagrams for the closed loop systems using the 3 DTDs described in the previous paragraph:

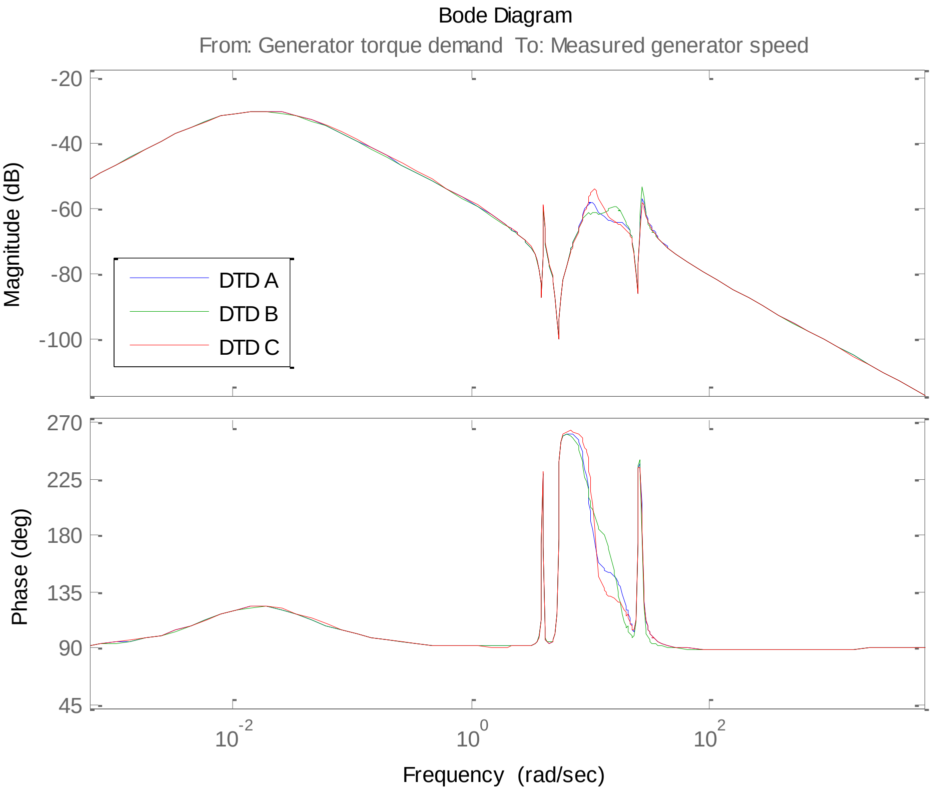


Figure 64: Bode plots of the GH Bladed linearized model of the drive-train of model 2M-a wind turbine, using three DTDs.

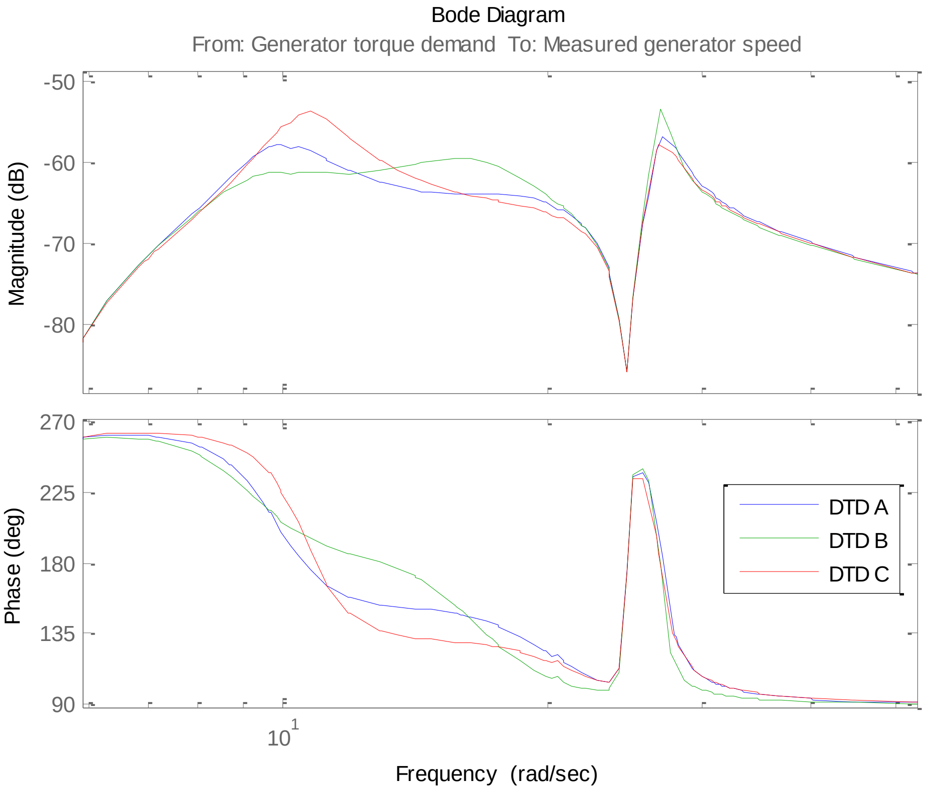


Figure 65: Bode plots of the GH Bladed linearized model of the drive-train of model 2M-a wind turbine, using three DTDs (only showing resonant frequencies).

The closed loop system utilising DTD A has a better-damped high frequency magnitude peak compared to the closed loop system utilising DTD B. On the other hand, the maximum of the first peak is slightly lower for the system utilising DTD B. The system utilising DTD C has the lowest high frequency peak, but also the highest low frequency peak. Figure 66 shows the impulse response of the systems:

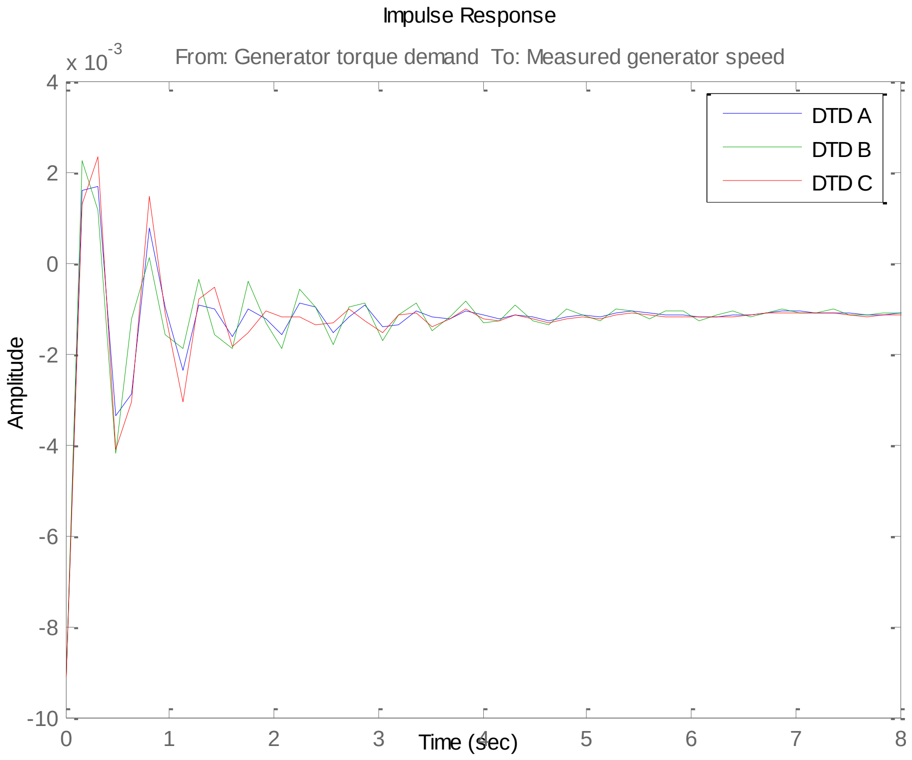


Figure 66: impulse response of the GH Bladed linearized model of the drive-train of Model 2M-a wind turbine, using three DTDs.

Judging from the impulse response, both in terms of percent overshoot and the settling time of the response, the system utilising DTD A seems to be the best damped system, which is what we expected. However, comparing the systems utilising DTD B and DTD C, it is not clear which of the two systems is better damped. Comparing the impulse response of these two systems, one could conclude that DTD C would act as a better drive train damper; a result which is false.

The above example shows that a direct comparison of the effect of the various dampers on the linearized systems, using a simple impulse response, is not suitable for evaluating the performance of the DTDs and as a consequence not suitable for evaluating the performance of the cost functions. This is mainly because the energy content of the exiting load is not uniformly distributed over all frequencies, and thus the impulse response of the system fails to provide us with the necessary information needed to identify the best tuned DTD.

On the other hand, running a dynamic production simulation for a wind turbine is quite a computationally expensive procedure that would need to be run for a very large number of combinations of penalties for ten wind turbine models. Thus, only 19 combinations of penalties, i.e. 19 cost functions, were tested for the development of this methodology.

Ideally only one cost function would be used in this methodology. However, finding a universally preferred cost function was not possible.

To demonstrate this, table 5 is presented below:

Wind Turbine Model	Wind realisation No	Best Performing Cost Functions	Well Performing Cost Functions (up to 0.5% more damage allowed)
750k-a	24	1, 2, 5, 6, 7, 8, 9, 10	
	48	1, 5, 6, 7, 8, 9, 10	
	61	1, 5, 6, 7, 8, 9, 10	
	39	1, 5, 6, 7, 8, 9, 10	
	16	1, 5, 6, 7, 8, 9, 10	2
750k-b	88	2	
	54	2	
	79	2	
	87	2	
	86	2	
2M-a	1	4, 17	
	88	1, 2, 5, 6, 8, 9, 13, 14, 15, 16, 17	
	7	3, 11, 12, 18, 19	1, 6, 9
	74	13, 14, 15, 16, 17	1, 6, 9
	28	1, 2, 6, 9, 13, 14, 15, 16, 17	
2M-b	78	2	1, 3, 5, 6, 8, 9, 11, 12, 18, 19
	55	2	
	98	3, 11, 12, 19	
	33	All Cost Functions	
	47	1, 5, 6, 7, 8, 9,10	
2M-c	15	1, 4, 5, 6, 8, 9	
	98	4, 13, 14, 15, 16, 17	3, 11, 12, 18, 19
	82	13, 14, 15, 16, 17	
	35	4	
	58	4	17
2M-d	82	2	1, 5, 6, 7, 8, 9, 10
	54	2, 7	
	86	2	
	7	1, 5, 6, 7, 8, 9, 10	2
	25	2	
3M-a	38	1, 5, 6, 7, 8, 9, 10	
	27	1, 5, 6, 7, 8, 9, 10	
	31	1, 5, 6, 7, 8, 9, 10	18
	6	3, 4, 11, 12, 13, 14, 15, 16, 17,	

		18, 19	
	57	3, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19	
3M-b	14	3, 11, 12, 18, 19	
	70	3, 11, 12, 18, 19	
	79	3, 11, 12, 18	4, 13, 14, 15, 16, 17, 19
	75	3, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19	1, 2, 5, 6, 7, 8, 9, 10
	77	All Cost Functions	
3M-c	14	All Cost Functions	
	79	3, 11, 12, 18, 19	1, 4, 5, 6, 7, 8, 9,10, 13, 14, 15, 16, 17
	40	1, 5, 6, 7, 8, 9, 10	2, 3, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19
	90	3, 11, 12, 18	
	99	3, 11, 12, 18, 19	4, 13, 14, 15, 16, 17
3M-d	6	3, 11, 12, 18, 19	4, 13, 14, 15, 16, 17
	48	3, 4, 11, 12, 14, 15, 16, 17, 18, 19	
	4	3, 4, 11, 12, 14, 15, 16, 17, 18, 19	1, 5, 6, 7, 8, 9, 10
	75	3, 4, 11, 12, 14, 15, 16, 17, 18, 19	
	26	3, 4, 11, 12, 14, 15, 16, 17, 18, 19	1, 2, 5, 6, 7, 8, 9, 10

Table 5: Cost function performance.

The table shows cost functions leading to the best-performing and “well-performing” DTDs, based only on the “5 best experimental wind realisations” per model. A DTD is considered “well-performing”, if the DTD it identifies leads to a maximum increase of 0.5% on the DEL on the gearbox torque (in a 18m/s wind speed realisation) when compared to the best DTD identified by the other 18 cost functions. The “5 best experimental wind realisations” per model, are the experimental wind realisations based on which the self-tuning methodology created by this research tuned the 5 best performing DTDs per model.

Only 5 experimental wind realisations per model are used for this comparison because the experiments that lead to the tuning of suboptimal DTDs should not be allowed to affect the selection of the most appropriate cost functions. To make this clear, let us suppose that 10 experimental wind realisations - and thus experiments - that lead to badly identified systems (and thus badly tuned DTDs) were included in this table. Let us then suppose that cost function 9 was the best performing cost function when the self-tuning procedure was run based on these 10 experimental wind realisations, whilst cost function 1 was the best performing cost function when the self-tuning procedure was run based on the 5 best experimental wind realisations. Then, by just comparing these results, one might conclude that both cost functions 1 and 9 are beneficial as both have been shown to produce best performing DTDs. In reality however, based on this hypothetic example, only cost function 1 is of value to the methodology developed.

Going back to the actual results obtained, as they are presented in table 5, one can identify that the use of cost functions 2 and 4 is unavoidable, since if either were not used, a well-tuned DTD would not be identified for a number of wind turbine models. As an example, if cost function 2 were not used, a well-performing DTD for the 750k-b wind turbine model would not be tuned. Also, if cost function 4 were not used, only 3 out of the best 5 DTDs for the 2M-c wind turbine model would be tuned, thus increasing the chances of not finally being able to tune a well-performing DTD.

Selecting only cost functions 2 and 4 would lead the methodology developed to only be able to tune a well-performing DTD in 36 out of the 50 cases presented in table 5 (see table entries coloured in green and pale blue). Including cost functions 1 and 3, however, allows the developed methodology to be able to tune a well-performing DTD in 49 out of the 50 cases presented. Since adding yet another cost function only marginally improves the effectiveness of the algorithm, and at the same time increases the computational time of the algorithm, the number of cost functions is maintained at 4.

One last point that should be highlighted is the usefulness of the sumZ term in cost functions 1 and 2. SumZ is a comparatively small number, as it is calculated by adding the damping ratio of the poles of the closed loop system. Multiplying this small number by 0.001 makes this number even smaller. A typical cost function 1 has a value of 2500, while the sumZ term is for example equal to 1.

Nevertheless, the importance of this term is crucial. This will be explained with the aid of a simple example. Consider two closed loop systems, created using the same drive-train model employing two different DTDs. Let us assume that each closed loop system has three pole pairs each. For system A, one pole pair has a damping ratio of 0.16, and two pole pairs a damping ratio of 0.06. System B has one pole pair with a damping ratio of 0.16, one with a damping ratio of 0.06 and the last one with a damping ratio of 0.05. Now if we consider cost functions 1 or 2 without the sumZ terms, the cost functions for the two systems would be identical, whilst it is clear that system B is more damped than system A. By including the sumZ term, the cost functions for system B become lower, thus correctly identifying the system that is damped more.

The reason why the sumZ term is maintained very low (by multiplying by 0.001) is to make sure that the cost functions are not biased towards DTDs that create systems that have only a few very well damped pole pairs leaving the remaining poles relatively un-damped.

Again, this is explained with the aid of a simple example. Consider as before, two closed loop systems, created using the same drive-train model and two different DTDs, with three pole pairs each. System A has one pole pair with a damping ratio of 0.15 and two pole pairs with a damping ratio of 0.30. System B has one pole pair with a damping ratio of 0.1 and two pole pairs with a damping ratio of 0.45. Both systems have 2 pairs of highly damped poles and one less damped pole. From a damping ratio value and above, additional damping makes little difference to the response of the system. On the other hand, small changes on the damping ratio of lightly damped poles have a substantial impact in the behaviour of the system. According to this logic, and assuming that poles are very well damped when their damping value is at a value of 0.3, system A indicates a better tuned DTD. This is because both DTDs are able to damp two pole pairs well enough, but the DTD used in system B will not damp the lightly damped pole as well as the DTD used in system A. If the sumZ term were multiplied by a large number, say 10, then System A (with sumZ equal to 1.5) would be penalised considerably more that System B (with sumZ equal to 2) thus leading the cost function to indicate the DTD of System B as the better tuned damper. As an example, assuming cost function 1 used a sumZ term multiplied by 10 instead of 0.001, the cost of System A would be -14.7, whilst the cost of System B would be -18.8, thus wrongly indentifying the DTD of system B as the best DTD of the two.

Assessing the number of experiments that need to be run

In order to obtain a well tuned DTD, a number of experiments need to be run. If too few experiments are run then the best DTD tuned will probably not be a very good one. On the other hand, running a very large set of experiments can interfere with normal turbine operation and might not be ideal since the turbine owner and the turbine manufacturer will probably want the DTD tuned and tested within the first few days of commissioning, and certainly before the end of commissioning, which usually takes a couple of weeks per wind turbine.

Since the developed self-tuning DTD procedure is intended for use on real turbines, one does not know beforehand by how much a good DTD can reduce the fatigue loading on the drive-train. One can obviously not run the experiments for an indefinite number of times in order to guarantee that the best DTD has been tuned.

Thus, there is the need to define a suggested number of experiments of a specific type, that must be run in order to ensure, to some extent, that a suitable DTD has been tuned. Obviously, a statistical approach needs to be taken in order to define this suggested number.

Framing the question in a more formal and quantitative way: How many successful experiments need to be run in order for us to be 99% certain that at least one of the DTDs tuned causes at most 1% more "damage" to the drive-train than the "ideal" DTD?

The DEL of the gearbox torque signal is used in order to compare the DTDs with respect to the damage they allow to the gearbox, in the same way as it has been done throughout this research. The "ideal" DTD is the DTD that would reduce fatigue damage by as much as possible. In this investigation, we approximate the "ideal" DTD with the best DTD found by 99 random experiments.

For simplicity, a DTD that causes at most 1% more "damage" to the drive-train than the "ideal" DTD will be called an "acceptable DTD". It would take a very long time to complete all the test runs needed to assess the lifetime weighted DEL on the gearbox* for each of the 99 experiments for each of the models. Therefore, the

[*] The lifetime weighted DEL on the gearbox is the equivalent load that would be experienced by the Gearbox in its lifetime, assuming that the wind turbine will be operating in Class IA climatic conditions for a period of 20 years.

following approximation has been made: it has been assumed that a 3% increase in DEL of the gearbox torque at a 10 minute-long 18m/s wind speed production run results in a 1% increase in the lifetime weighted DEL of the gearbox torque. This assumption was taken based on the findings of the following exercise: A number of 2M-a models were selected, each one created by using the base 2M-a model and a different DTD, and a full 20 year lifetime set of IEC Class IA loading Design Load Cases was run. By doing this, the lifetime weighted DEL on the gearbox was calculated for all these wind turbine model alternatives. It was found that the turbine models experiencing approximately 1% more DEL in their lifetime were using DTDs that showed approximately 3% difference in DEL under a 10 minute-long 18m/s wind speed production run.

So in practice, an *acceptable* DTD is the DTD which, when used in a 10 minute-long 18m/s wind speed production run, will cause a maximum increase of 3% DEL in comparison to the DEL caused when an ideal DTD is used. This is an approximation based on the above described analysis of a small set of results but has been accepted for this study as it is consistent with what is expected: one would expect that the overall DEL increase caused by a non optimal DTD would be much less that the DEL increase caused by this DTD at an 18m/s wind speed production run, where the loads are higher and the drive-train resonance is high.

The probability of finding an *acceptable* DTD by undertaking one random successful experiment (indicated here on by p_A) is equal to the number of experiments producing an *acceptable* DTD over the number of successful experiments. Table 6 below, shows p_A for all models.

Model p_A	750k-a	750k-b	2M-a	2M-b	2M-c	2M-d	3M-a	3M-b	3M-c	3M-d
PRBS experiments	25%	0%	18%	2%	5%	39%	13%	12%	17%	20%
Chirp experiments	1%	1%	17%	11%	12%	47%	21%	13%	26%	38%
Best	25%	1%	18%	11%	12%	47%	21%	13%	26%	38%

Table 6. p_A for all wind turbine models.

The probability of not finding an *acceptable* DTD by undertaking one successful experiment is thus equal to 1- p_A. By applying the Multinomial Theorem [93], the probability of not finding an acceptable DTD by undertaking *n* successful experiments is found to be equal to (1- p_A)^n. Thus the probability of finding at least 1 acceptable DTD by undertaking n successful experiments is equal to 1-(1- p_A)^n. Now, since this probability has already been set to 99%, Equation 45 just needs to be solved for *n*:

$$0.99 = 1-(1-p_A)^n \Leftrightarrow \quad \textbf{Eq. 45}$$

$$\Leftrightarrow 0.01 = (1-p_A)^n \Leftrightarrow$$

$$\Leftrightarrow \ln(0.01) = n \cdot \ln(1-p_A) \Leftrightarrow$$

$$\Leftrightarrow n = \frac{\ln(0.01)}{\ln(1-p_A)} \quad \textbf{Eq. 46}$$

From table 6, for the self-tuning procedure using the Chirp experimental procedure, the most conservative p_A is equal to 1% and if presented with a greater accuracy runs to 1.4%. Thus, Equation 46 becomes:

$$\Leftrightarrow n = \frac{\ln(0.01)}{\ln(1-0.014)} = 326.6 \approx 327$$

So in order to be 99% certain that at least one of the acceptable DTDs has been tuned, the self-tuning DTD procedure needs to be run using the Chirp experimental procedure 327 times, each time running a different experiment and tuning a new filter. In order to collect a valid set of experimental data, approximately 5 minutes of experiment time are needed. Thus the turbine would need to operate in the data collection mode (i.e. running the pre-defined experiment) for approximately 27 hours. As these 27 hours of experiments need to be run at wind speeds close to 6 m/s, this in practice means that the experimental procedure could take a few days to be completed.

It should be noted however that this is the worst case scenario. If the same calculation is performed for the average and the highest p_A for the Chirp experiment based self-tuning DTD procedure, equal to 19% and 47% respectively, the results indicate that the self-tuning procedure needs to run 22 times and 8 times respectively. These results are considerably lower than the results obtained in the worst case scenario. This signifies that for some wind turbines, the developed procedure can be used to find a well tuned DTD much quicker than in the worst case scenario.

The results obtained, using the self-tuning DTD procedure employing the PRBS experiment, show that this procedure fails to tune a well performing DTD in the case of the 750k-b wind turbine model. They also show that this procedure in general obtains lower p_A in most cases. However, it should be remembered that the PRBS experiment has some merits over the Chirp based experiment, as discussed in section 4.1.1, and that this procedure shows a much higher p_A in one of the models tested (750k-a).

It is interesting to note that relatively low values of p_A were obtained for some of the wind turbine models. This indicates a high number of unsuccessfully tuned dampers. Since the DTD tuning procedure was always accurately tuning the DTD when a GH Bladed drive-train model was used, it is safe to conclude that the cause of the problem was the drive-train model created by the system identification process. Thus, the low values of p_A obtained for some models indicate that some of the model wind turbines used, were more susceptible to unmeasured disturbances and 3P loading during the experiment than others, thus leading to badly identified drive-train models. Although this is something one could reasonably expect, to the authors knowledge, it has not been investigated by other researchers.

5.2 Final Results

Before discussing the final results obtained by the developed methodology, a brief overview of the benefits of a well tuned DTD is provided, so that the reader understands better why tuning a well performing DTD is very important.

Figure 67 on the next page compares the gearbox torque loading of a wind turbine (model 2M-a) with a well-tuned DTD to the gearbox torque loading of the same wind turbine without a DTD, for a 50s period during a 400s normal operation simulation.

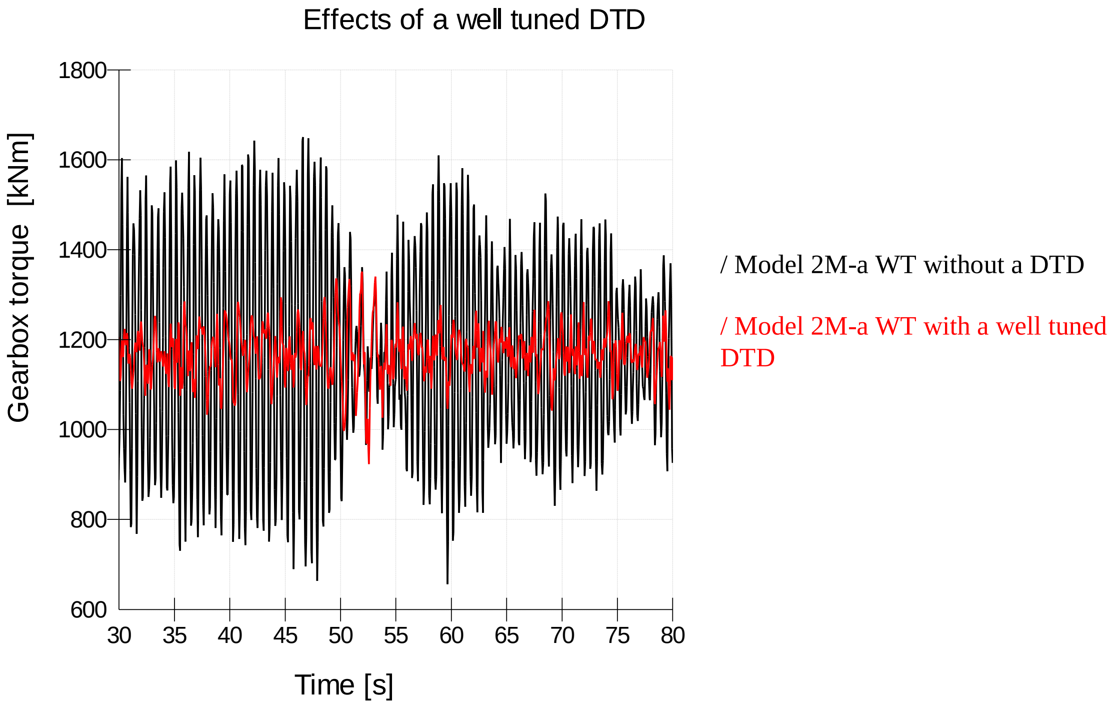


Figure 67: Gearbox torque produced with and without a DTD.

The red line in the above graph shows gearbox torque fluctuations that are maintained within normal operating limits for the wind turbine using a well-tuned DTD. However, if a DTD is not used (black line) the drive-train resonates, producing high load fluctuations that are not well damped by the variable speed wind turbine. Obviously, such behaviour would have catastrophic effects on a real gearbox.

In order to understand the cause of this large difference, one must examine the frequency content of the loading forces. Figure 68 shows the spectral density of the gearbox torque loading during the whole 400s experiment.

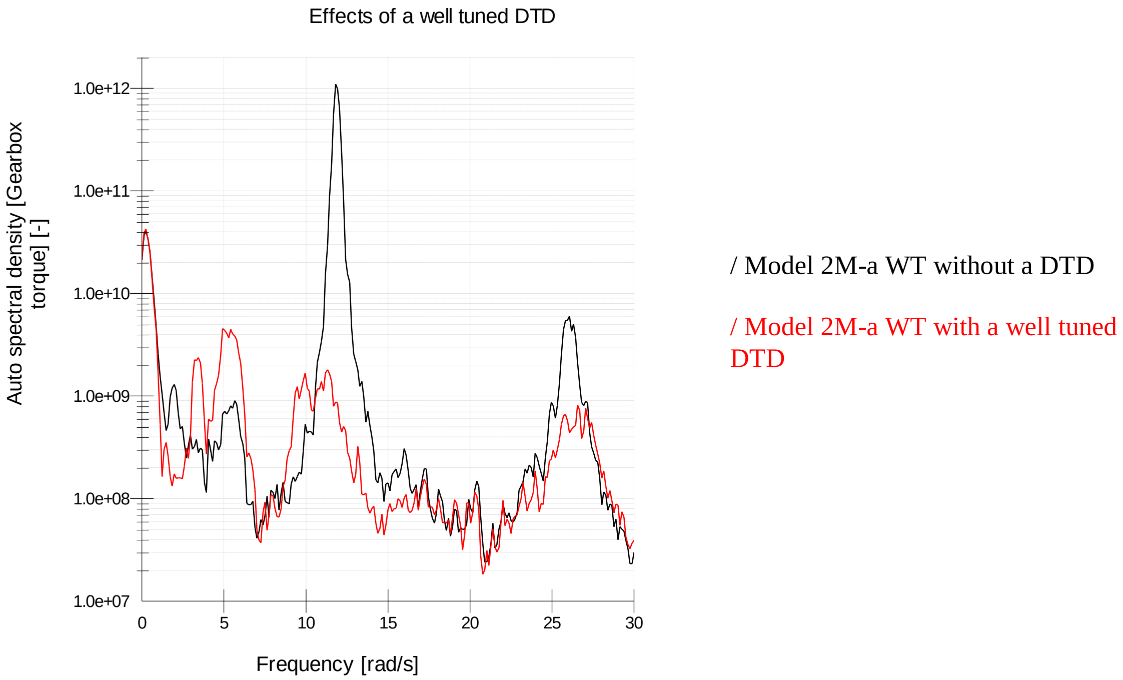


Figure 68: Gearbox torque spectral density with and without a DTD.

The red line in this graph shows the frequency content of the loading experienced by the gearbox when using the well-tuned DTD, whilst the black line shows the frequency content of the loading experienced by the gearbox when no DTD is used.

It is evident from the graph that the DTD effectively damps the resonances at the first and second natural frequencies of the drive-train. It should be pointed out that the y-axis of the graph is plotted on a logarithmic scale, so the highest peak of the gearbox torque spectrum for the WT model without a DTD shown at 12 rad/s is substantially larger than the peak of the WT with the DTD at this frequency region.

A comparison between a well-tuned and a sub-optimally tuned DTD is shown next. Figure 69 shows the same 50s of power production simulation, this time comparing the gearbox torque fluctuations using the well-tuned DTD presented before, and a second badly tuned DTD. In fact, both DTDs were tuned by the self-tuned DTD methodology using the PRBS experimental procedure. What differentiated the performance of the two DTDs, was the specific wind realisation used for each experiment that led to different experimental data being collected, and thus different DTDs tuned.

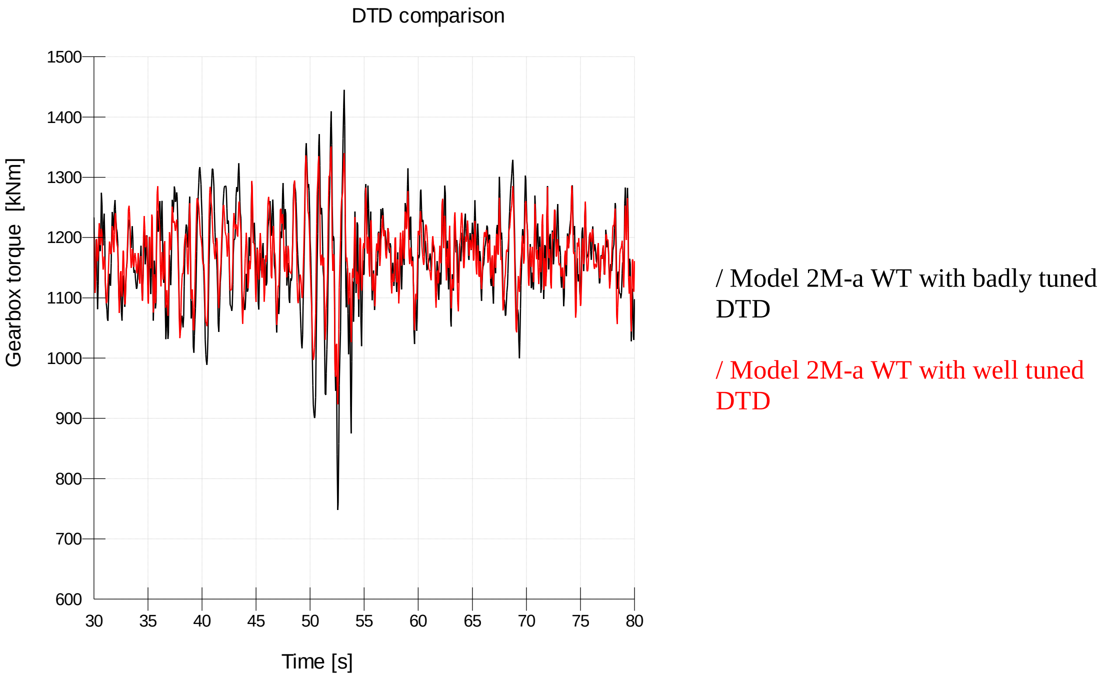


Figure 69: Gearbox torque produced with a well tuned DTD and a badly tuned DTD.

Again, the spectral density of the gearbox torque loading during the whole 400s experiment is shown in the figure below:

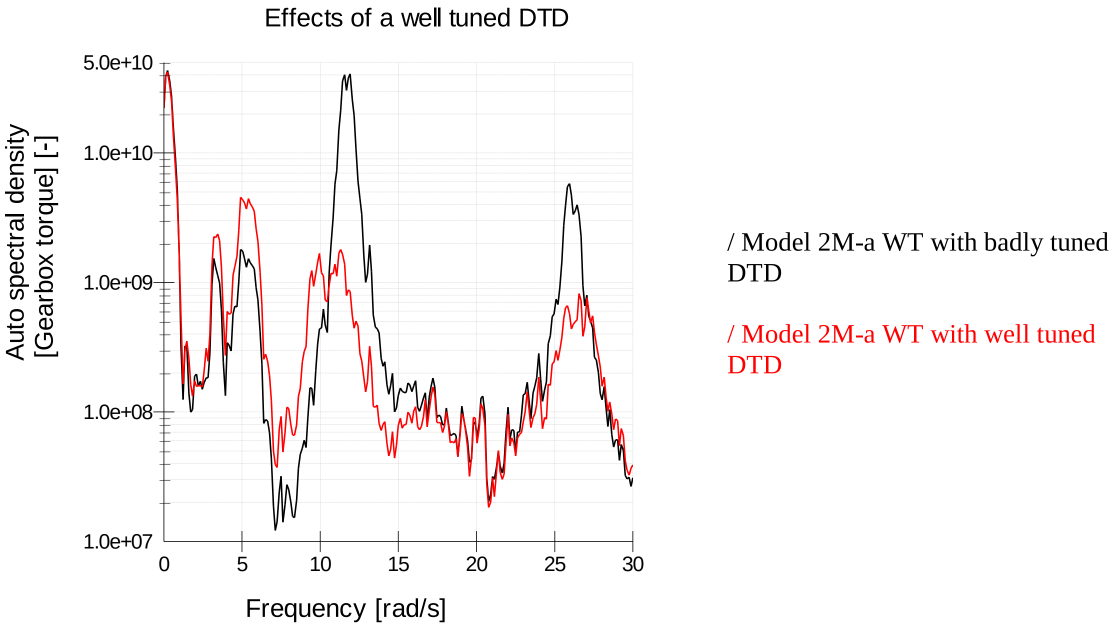


Figure 70: Gearbox torque spectral density with a well-tuned DTD and a badly-tuned DTD.

A useful graph for understanding the energy content at each frequency is the cumulative variance plot, which is presented in Figure 71 below:

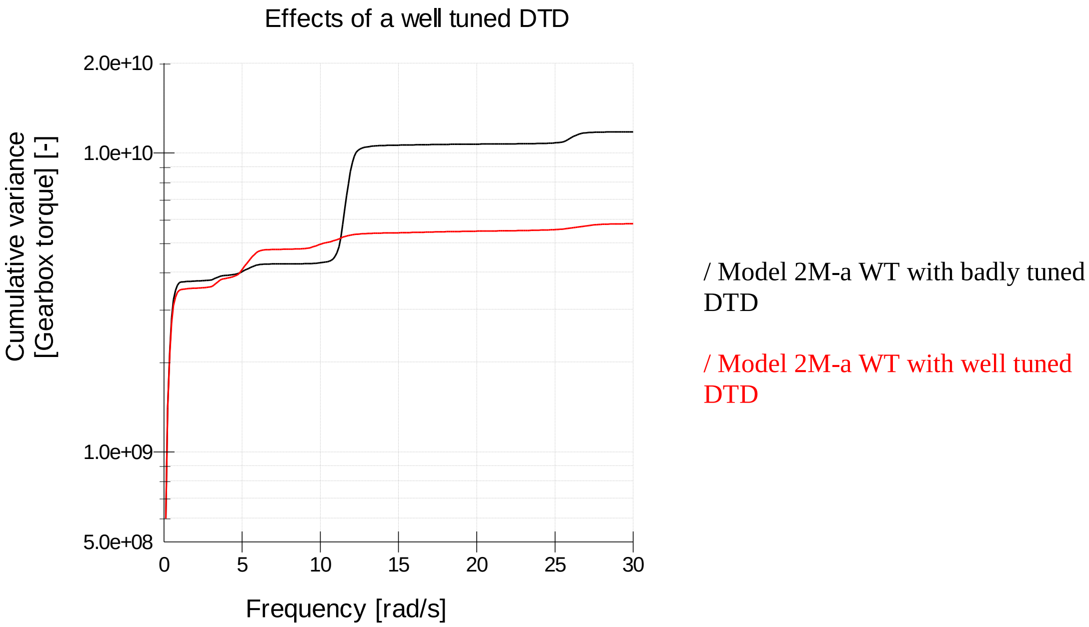


Figure 71: Gearbox torque cumulative variance with a well-tuned DTD and a badly-tuned DTD.

The above graphs indicate that the well-tuned DTD damps better both the resonance in the first and the second natural frequencies of the wind turbine drive-train, shown at 12 and 26 rad/s. The main difference evident on the graph is in the damping of the resonance at the first natural frequency.

The following table summarises the most important loads on the wind turbine during the 400s experiment, with the WT utilising the well-tuned DTD and the sub-optimally tuned DTD.

	Channel statistics	Badly tuned DTD	Well tuned DTD	Delta
Drive train	Generator speed Std.Dev. [rad/s]	1.90	1.65	-13%
	Rotor speed Std.Dev. [rad/s]	0.021	0.019	-6%
	Generator torque Std.Dev. [kNm]	0.718	0.740	3%
	Electrical power Std.Dev. [kW]	114	116	1%
	Gearbox Torque Std.Dev. [MNm]	0.109	0.076	-31%
	Max Gearbox Torque [MNm]	1.48	1.35	-8%
Blades	Tip to tower closest approach [m]	2.3	2.6	16%
	Blade 1 pitch angle Std.Dev. [deg]	19.4	19.2	-1%
	Blade 1 pitch rate Absolute Max [rad/s]	0.244	0.140	-43%
	Blade 1 flap-wise Max bending moment [MNm]	2.33	2.23	-4%
	Blade 1 edge-wise Max bending moment [MNm]	1.60	1.62	1%
	Blade 1 flap-wise bending moment Std.Dev. [MNm]	0.394	0.387	-2%
	Blade 1 edge-wise Max bending moment Std.Dev. [MNm]	0.471	0.471	0%
Tower	Tower base Max fore-aft bending moment [MNm]	27.0	25.6	-5%
	Tower base Max side to side bending moment [kNm]	6.35	5.88	-7%
	Tower base Std.Dev. fore-aft bending moment [MNm]	4.62	4.50	-3%
	Tower base Std.Dev. side to side bending moment [kNm]	1.37	1.21	-12%

Table 7: DTD performance comparison.

As expected, the well-tuned DTD reduces both maximum gearbox torque and gearbox torque fluctuations. Apart from this positive effect it also leads to a smoother wind turbine operation by lowering the pitch control action, as is evident by the lower maximum pitch rate and the lower pitch angle standard deviation. This is achieved by substantially lowering the rotor and generator speed fluctuations. This also has a positive effect on flap-wise blade moments and tower loads. A marginally negative effect can be observed in the blade edge-wise bending moment.

The main cost for the control action is the increase in electrical power fluctuations, that are inherent to this type of DTD. The reduction in the drive-train fatigue loads is achieved by introducing a small torque ripple at the frequency of the resonant mode that needs to be damped, at a phase at which the torque damps the drive-train oscillations. An example of this torque ripple is shown in Figure 72:

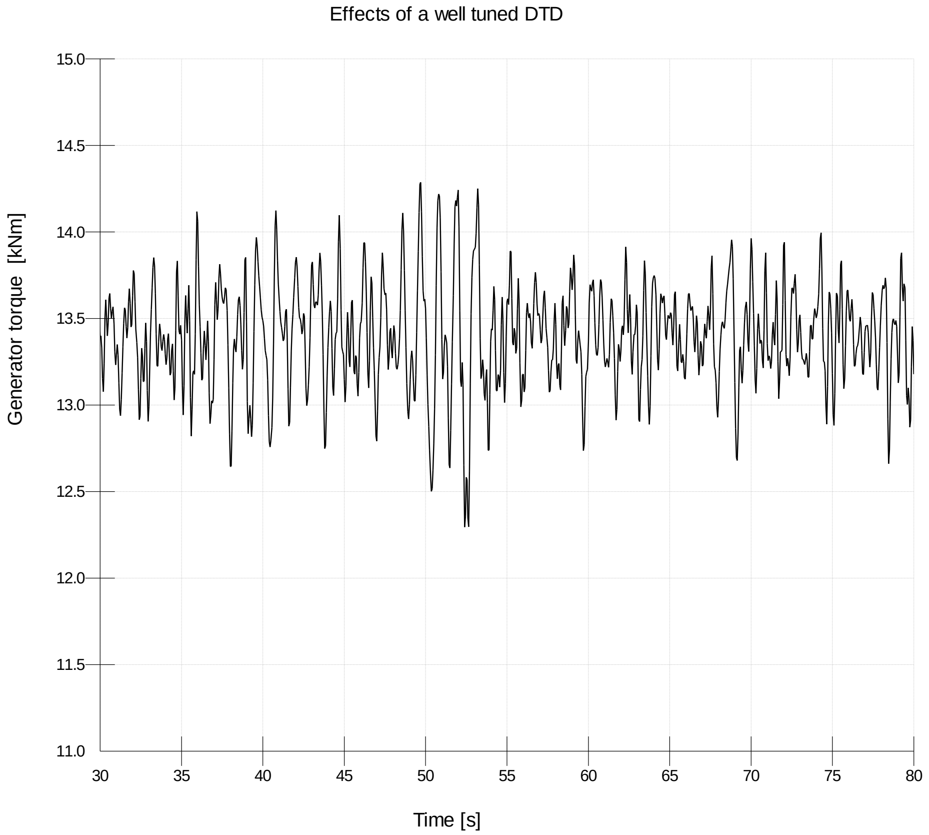


Figure 72: Example generator torque ripple introduced by the DTD (model 2M-a).

These generator torque fluctuations introduced in order to damp the drive-train, result in power fluctuations degrading the power quality of the produced electricity. This is a well-known issue for this type of DTD [[16]]. Its effect is limited by selecting a low damping constant (ζ) in the DTD equation, as was explained in section 4.1.3. In the example above, a 1% increase in electrical power standard deviation has been introduced. The following graph shows the electrical power output of the generator during the same 50s of the 400 second experiment introduced above, for the 2M-a wind turbine model without a DTD, with the sub-optimally tuned DTD and with the well-tuned DTD:

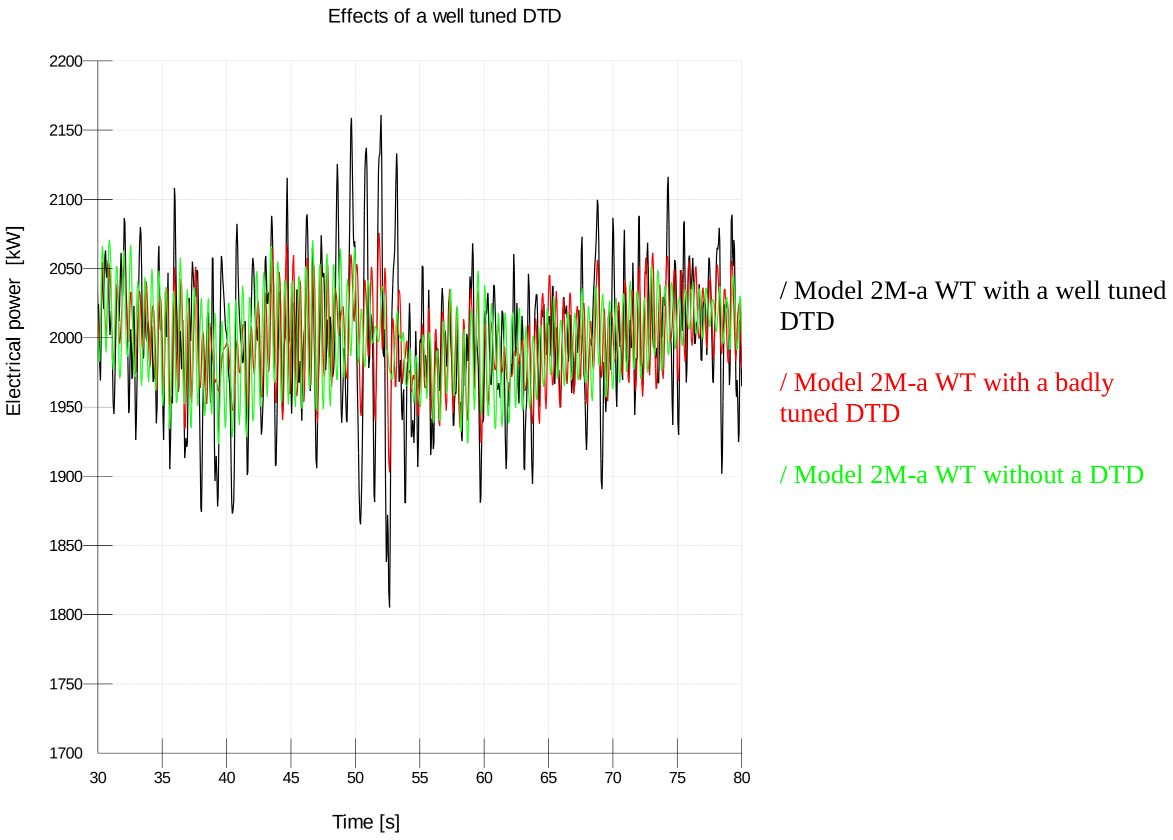


Figure 73: Example electrical power output without the introduction of a DTD, with a badly-tuned DTD and a well-tuned DTD introduced in the control algorithm of the wind turbine (model 2M-a).

The slight deterioration in electrical power quality is however counterbalanced by the substantial reductions in the structural loading and pitch actuator duty cycle, thus making a DTD a critical part of the wind turbine controller.

As previously discussed, the DTD obtained by the recommended procedure will result in an acceptable DTD. This means that the wind turbine will be using a DTD that will reduce loads on the gearbox as much as possible, and in the worst case, it will allow 1% more “damage” to the drive-train than the “ideal” DTD.

At a first glance this might seem less than an ideal result, as a well performed manual tuning of the damper seems to be a more effective approach than using the self-tuned DTD proposed by this research.

At a closer reading of the results however, the methodology has proven itself in two ways.

The main reason behind developing an automatically tuned DTD was to create a damper that would be suitable for a wind turbine regardless of (a) how well the model of the turbine in the design phase resembles the real turbine manufactured, and (b) the quality and accuracy of the software used to simulate the turbine.

It is correct to say that by using a self-tuned DTD as per the methodology developed by this research, the drive-train of the wind turbine will in the worst case be experiencing 1% more damage in its 20-year lifetime, compared to the drive-train of a similar wind turbine equipped with the “ideal” DTD (as this was previously defined). However, one should not forget that an ideal DTD can only be manually tuned in the case where both the model used in the design stage and the real machine behave in the exact same way, and the control engineer spends considerable effort in making sure that he has tuned the best possible DTD.

The model of the turbine used in the design stage will probably not have the exact same dynamic characteristics as the real machine for a number of reasons.

Firstly, the mechanical characteristics of the various drive-train components might not be the same. It is common practice to use more than one gearbox supplier for the same wind turbine type, providing equivalent but not identical gearboxes. For example the NM 52 / 900kW wind turbine uses a number of different gearboxes from at least three gearbox manufacturers, and each gearbox manufacturer has produced a number of gearbox designs (versions) that have been installed on the NM 52 wind turbine. These gearboxes might have similar loading characteristics but at the same time might have quite different inertias, leading to different natural frequencies. The same, but certainly to a lesser extent, applies to the rest of the drive-train components including hub, low speed shaft, high-speed shaft, brakes, coupling and generator. Other wind turbine components also affect the dynamic characteristics of the drive-train, such as the tower and the blades. Again, the actual characteristics of these components are not exactly the same as the ones predicted at the modelling stage. Finally, but to a much lesser extent, even the foundation and soil characteristics feed into the drive-train dynamic response by altering the side-to-side tower bending mode, thus aiding to the deviation of the modelled drive-train dynamic response from the actual drive-train dynamic response.

Secondly, the same software used to model the wind turbine is used by the engineers manually tuning the DTD at the design stage to a) model the coupled dynamic characteristics of the drive-train (and thus tune the DTD), and b) run some simulations to validate the effectiveness of a few DTDs and select the best one. These models however, are not always 100% accurate, as they are based on a number of simplifications and approximations used to describe the actual physical processes taking place. All these simplifications and approximations have an effect on the dynamic characteristics of the turbine.

Thus in practice, where a) the model of the turbine used in the design stage does not exactly have the same mechanical properties as the real machine or b) the software used to simulate the turbine in the design stage does not accurately model the actual physical phenomena, the automatically tuned DTD produced by the methodology developed in this research will, in most cases, be more effective than a normal DTD tuned at the design stage.

Moreover, an added advantage of the automatically tuned DTD became evident during this research. The best of the automatically tuned dampers seemed to show a reduction in loading of the drive-train in comparison to the manually tuned DTD, even when the model of the wind turbine used was exactly the same as the real wind turbine. Note that for the purposes of this research, the real wind turbine is the simulated turbine.

The following table shows the lifetime weighted Damage Equivalent Load (DEL) of the gearbox torque, achieved by a manually tuned and an automatically tuned damper, both based on a single band pass filter. The ten columns represent the ten models used for validation of the DTD performance as described in chapter 3.2.

Lifetime weighted DEL in kNm [inverse SN slope: 3, IEC IA conditions]										
	750k-a	750k-b	2M-a	2M-b	2M-c	2M-d	3M-a	3M-b	3M-c	3M-d
Manually tuned	37.0	40.4	192.2	181.8	180.3	175.8	301.6	278.8	276.9	288.9
Automatically tuned	37.9	40.6	190.3	180.7	179.2	174.4	299.7	278.5	278.1	284.6
Load Reduction	-2%	0%	1%	1%	1%	1%	1%	0%	0%	1%

Table 8: Gearbox torque Lifetime weighted DEL comparison of a manually tuned DTD and the best automatically tuned DTD.

The above results indicate that the automatically tuned dampers in all the 2 MW and most of the 3 MW models show a reduction in loading of the drive-train, even when the model of the wind turbine is exactly the same as the real wind turbine. Of course, this is not universally true, as in the 750 kW wind turbine models and the 3M-c wind turbine model the automatically tuned DTDs were at par or slightly worse than the manually tuned dampers.

Initially it was assumed that the automatically tuned DTDs were more efficiently damping the 3P excitation and thus were achieving less damage to the gearbox. The reasoning behind this, was that whilst the manually tuned dampers were tuned based on a linearized model that had no interference, the automatically tuned dampers were tuned based on a linearized model that had some experimental 3P interference, and thus the filter tuned would be biased to damping the oscillations at this frequency as well. Analysing the data, however, showed that the exact opposite was happening. This is explained by the use of an example. Figure 74 below, shows a power spectral density graph of the generator torque in the 2M-a model during a 10 minute power production simulation in a wind regime with an average free-stream velocity of 18m/s. As the label of the graph indicates, the solid blue line represents the wind turbine model running with the self-tuned DTD whilst the dotted green line represents the wind turbine model running with the manually tuned DTD.

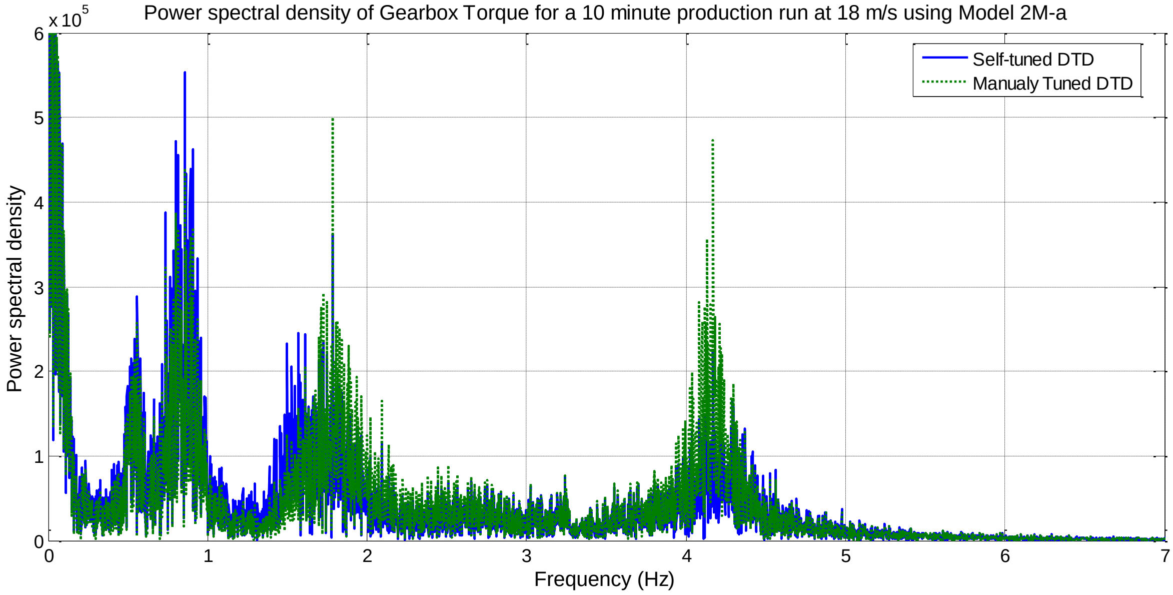


Figure 74: power spectral density graph of the generator torque in the 2M-a model using a self-tuned DTD and a manually tuned DTD.

From the above graph it is evident that the self-tuned DTD damps oscillations better at the two drive-train natural frequencies at 1.8 and 4.2 Hz (see also the Bode plot in Figure 27). However, the self-tuned DTD does not damp the oscillations of the drive-train at the 3P frequency (0.9 Hz) as well as the manually-tuned DTD. The result, nevertheless, is a small overall reduction in the DEL experienced by the gearbox while using the self-tuned DTD.

Finally, it was found that the manually tuned DTD was not optimally tuned by the author. As the tuning process is not straightforward, it is quite hard for the control engineer, and in this case the author, to always achieve an optimal result. To verify this, both the manual and the self-tuned DTD were introduced into the linear model created by GH Bladed for wind turbine model 2M-a. The self-tuned DTD seemed to damp the system response better than the manually tuned DTD thus validating the above hypothesis. This can be shown in the Bode response of the closed loop systems presented in Figure 75:

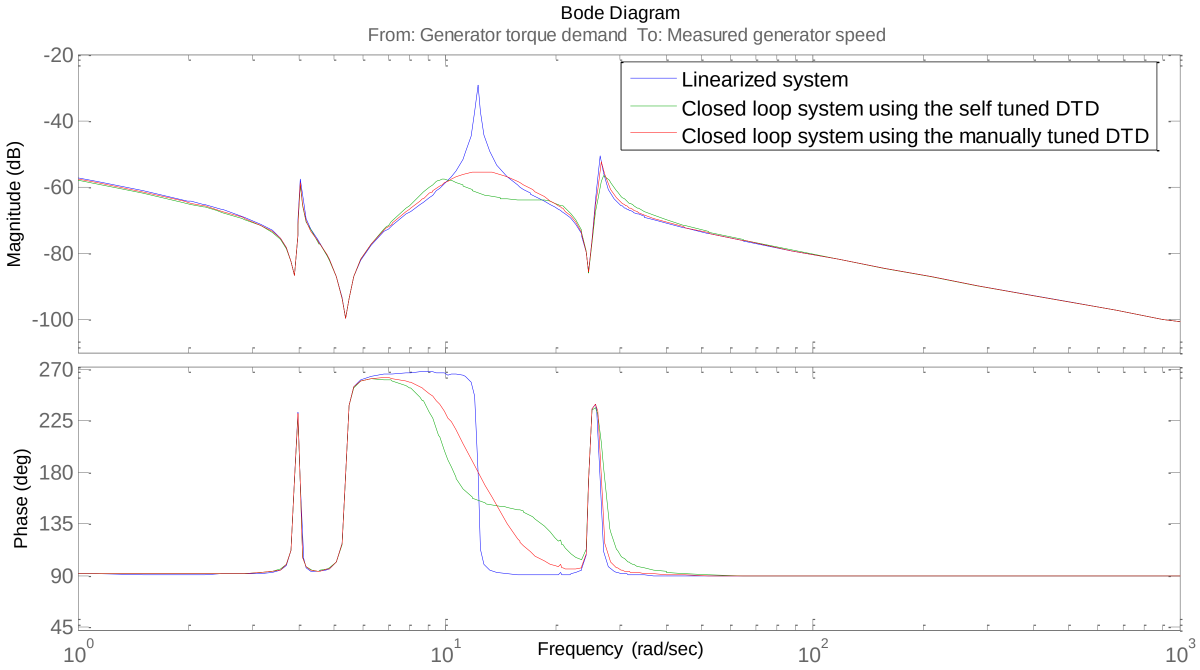


Figure 75: Closed loop system comparison of model 2M-a wind turbine’s drive-train systems using no DTD, the self-tuned DTD and the manually tuned DTD.

The Bode plot of Figure 75 clearly indicates that the self-tuned DTD damps both the first and the second drive-train natural frequency better than the manually tuned DTD.

Thus using the proposed self-tuning DTD methodology has the effect of eliminating human error, something that should not be considered trivial.

Chapter 6

In this chapter, the conclusions from this research are discussed. The results of the developed methodology are summarised and more general remarks and findings stemming from this research are presented. Finally, a number of suggestions for further work are outlined. What could be done by future researchers to further advance research on the self-tuning DTD, and the variety of interesting research subjects that have emerged during the author’s investigation, are identified.

6.1 Conclusions and Further Work

This research has shown that it is possible to create a procedure that automatically tunes a single band-pass filter that can be used as the wind turbine’s DTD.

The whole procedure needed for tuning the DTD should take no more than a few hours. The only prerequisites are that of favourable wind conditions and a torque-measuring sensor on the gearbox. The first step of the procedure needs to be run under an average wind speed of approximately 6 m/s, whilst the last step can be either run on similar or even higher wind speeds. A simple un-calibrated strain gauge can function as a torque-measuring sensor.

The benefits of using such a self-tuned DTD are multiple. Firstly, in the case where the model used by the control engineer to tune the manually tuned DTD was not an accurate representation of the actual wind turbine, the self-tuned DTD can be used to reduce the drive-train fatigue damage during the wind turbine’s operating life.

The causes of the inaccurate model representation, as discussed in the previous chapter of this thesis, are related to the following:

a) the differences of the modelled structural characteristics of the various parts of the wind turbine with their real counterparts,

b) the errors in the approximations applied by the simulation software which was used in order to first tune and then validate the DTD.

The amount of the drive-train fatigue damage reduction achieved by using the self-tuned DTD is not quantifiable, as it depends on how much deviation exists between the modelled wind turbine characteristics, and the real wind turbine characteristics. Based on the findings of this research, if the DTD used by a wind turbine produces more that 2% added fatigue damage to the wind turbine drive-train in comparison to an ideal single band pass filter based DTD, the use of the self-tuned DTD will reduce the fatigue loads on the gearbox.

Secondly, parts of the proposed tuning procedure and more specifically the third stage of the tuning procedure can also be alternatively employed.

Control engineers, instead of tuning the DTD manually, could use the third stage of the self-tuning DTD procedure to automate this task, or use it in parallel to manual tuning to ensure they have tuned the DTD in the most optimal way. This is done simply by using the linearized model they would use for manual tuning, and running the MATLAB code developed for this research. The author of this research found that for some of the models, the MATLAB code developed for this research tuned the DTDs equally well as the author. But for some other models this code tuned better DTDs than the author did by the manual tuning procedure. Of course, this is down to the control engineer's experience. The more experienced a control engineer is, the less the chances are that the automatic procedure will outperform them in tuning a better filter. Still however, even for the most experienced control engineers, having an automatic procedure should still be very helpful in reducing the controller tuning time, and the risk of tuning a suboptimal controller by mistake.

Moreover, and since this the MATLAB code tuning the DTD is fully automated, it can be run for a number of linearized wind turbine models, each one based on the wind turbine at different blade pitch angles. Thus, if a control engineer decided to use a pitch scheduled DTD, this procedure could be used to tune all the necessary parameters of this DTD much more efficiently.

Finally, apart from actually creating this useful self-tuning DTD procedure this research has made some more general contribution to the research field. It has shown that a relatively accurate model of the drive-train can be constructed with the help of system identification. Although this has also been shown by other researchers, this research verifies that this is feasible for a large number of wind-turbines ranging in size and structural characteristics. It has also shown that the system identification process is quite susceptible to 3P loading, as well as to numerous unmeasured disturbances. With some wind turbine models used, the system identification process is inherently more susceptible to such disturbances, and thus a large number of experiments need to be run to identify the drive-train model of such wind turbines correctly. Moreover, it was demonstrated that a relatively simple algorithm can be used to correctly and accurately tune a DTD once an accurate model of the drive-train has been created, by replicating the procedure a control engineer follows to tune the DTD. For the reader who is familiar with the DTD tuning procedure and is also familiar with the limitations of computer logic to replicate the human brain in some tasks, it is obvious that this task was not straight forward.

Apart from creating this self tuning DTD procedure, and the general contribution to the research field already discussed in the previous paragraph, this research has revealed one more interesting finding. In section 4.2.3 it was shown that a predictive 'load cancellation' technique can be employed to further reduce the gearbox loading. This was not extensively validated, as this was not the topic of this research. However, based on the very promising initial results it seems that an in-depth investigation of such a technique could be the topic of a dedicated research activity.

Other suggestions to further improve and extend the implications of the present work are recommended in the following paragraphs.

This research is based on simulations on various software codes. Some parts of the procedure and especially the system identification procedure need to be validated using real wind turbine data. Step one of the developed procedure describes the experimental collection of data. For the purposes of this research, these data were collected by running an experiment on a simulated wind turbine and not a real wind turbine. In case the dynamics of the drive-train are not exactly the same as predicted by theory and simulated by the software codes used for this research, the procedure will have no problem adapting to these dynamics. After all, this is the purpose and advantage of this procedure: it will self-tune a DTD according to the real drive-train dynamics, and not the perceived drive-train dynamics calculated by some software. However, issues related to noise and sensor quality, and their effect on system identification, have not been explored. Thus in order to fully validate the proposed procedure, an experimental campaign using a number of different real wind turbines should be performed.

It is obvious from the results obtained from this research, an example being those presented in Figure 74, that some DTDs are better at alleviating the 3P loading than others. Counterbalancing such loading is not the main purpose of the DTD, but it seems there is good evidence to suggest that a DTD can be designed in a way that also alleviates 3P loading. Thus adding additional band pass filters, in order to alleviate 3P and 6P loading, might prove beneficial for the drive-train fatigue loads.

This research failed to create a procedure that self-tunes a double band pass filter based DTD. However, it may be possible to fine-tune a double band pass filter that has been already tuned by a control engineer. The methodology would be almost identical to the one explored in this research, with the only difference being that the starting point for tuning the filter parameters would be already known, and a much smaller range of variables would need to be tested in order to fine tune the filter.

Another very interesting expansion to this research, would be to couple a state-space pitch controller, the objective of which is to alleviate drive-train loads such as the one presented by [51], with the self tuned drive-train damper. By firstly identifying possible negative effects of the interaction between the pitch controller and the DTD, and then taking the appropriate actions to prevent these interactions, this combination of control strategies could lead to a very promising control methodology.

Finally, by showing that it is possible for a software code to replicate the quite demanding task of tuning the DTD, creating similar software codes to replicate the tuning of other filters or controllers used in a wind turbine, for example the tower feedback filter, or the pitch or torque PI controllers, seems to be the logical next step.

References

[1] Drachmann A. G., “Heron's Windmill”, Centaurus, vol. 7, pp. 145–151, 1961.

[2] Hunt V. D., “Windpower: A Handbook on Wind Energy Conversion Systems”, pp. 1 and 9-10,Van Nostrand Reinhold Company, New York., ISBN 0442273894, 1981.

[3] Price T. J., “James Blyth - Britain's first modern wind power engineer”, Wind Engineering 29:3, pp.191–200, 3 May 2005

[4] Spinato F., Tavner P. J., van Bussel G. J. W., Koutoulakos E., “Reliability of wind turbine subassemblies”, IET Renewable Power Generation, Vol. 3, Iss. 4, pp. 387–401, 2009.

[5] Iov F., Blaabjerg F., “Power electronics and control for wind power systems”, PEMWA 2009, IEEE, pp. 1–16, 24-26 June 2009.

[6] Khadraoui R., Elleuch M., “Comparison between OptiSlip and Fixed Speed wind energy conversion systems”, 5th IEEE International Multi-Conference on Systems, Signals and Devices, pp. 1-6, July 2008.

[7] Vestas V-47 Product Brochure, downloaded from http://www.creswindfarm.gr/site1/Articles/V47_US.pdf on 06/2010

[8] Hoffmann W., Thieme A., “Control of a doubly-fed induction generator for wind power plants”, Proc. Power. Conversion and Intelligent Motion, pp. 275-282, May 1998.

[9] Vestas V-52 Product Brochure, downloaded from http://www.vestas.com/en/media/brochures.aspx on 06/2010

[10] Zinger D., Muljadi E., “Annualized wind energy improvement using variable speed” IEEE Trans Industry Applications, 33:6, pp. 1444–1447, 1997.

[11] Hoffmann R., Mutschler P., “The Influence of Control Strategies on the Energy Capture of Wind Turbines” 2000 IEEE Industry Applications Society Annual Meeting, Rome, 8-12 October, 2000.

[12] Smith G., “A novel converter for VSCF wind turbines”, Renew Energy, **9**, pp. 853–857, 1996.

[13] Hansen A. D., Hansen L. H, “Wind turbine concept market penetration over the years”, http://130.226.56.153/rispubl/art/2007_136_paper.pdf, accessed on 6/2010 (note: this is a more up-to-date version of the same authors’ paper “Wind turbine concept market penetration over 10 years (1995-2004)”, Wind Energy, v.10, Iss. 1, pp. 81-97, 2007.)

[14] De La Salle S. A., Reardon D., Leithead W. E., Grimble M.J., “Review of wind turbine control”, International Journal of Control, 52:6, pp. 1295-1310, 1990

[15] Leithead W. E., De La Salle S. A., Reardon D., “Role and objectives of control for wind turbines, IEE Proceedings Part C, 138, pp. 135-148, 1991.

[16] Burton T., Sharpe D., Jenkins N., Bossanyi E., “Wind Energy Handbook”, John Wiley and Sons, ISBN 0 471 48997 2, 2001.

[17] Bossanyi E. A., “Electrical Aspects of Variable-Speed Operation of Horizontal Wind Turbine Generators”, ETSU W/33/00221/REP, Energy Technology Support Unit, Harwell, UK, 1994.

[18] Leithead W. E., Connor B., “Control of variable speed wind turbines: dynamic models”, International Journal of Control, 73: 13, pp. 1173-1188, 2000.

[19] Boukhezzar B., Siguerdidjane H., Maureen M., “Nonlinear Control of Variable-Speed Wind Turbines for Generator Torque Limiting and Power Optimization”, Journal of Solar Energy Engineering, Vol. 128, Nov. 2006

[20] Leithead W. E., Connor B., "Control of variable speed wind turbines: Design Task", International Journal of Control, 73: 13, pp. 1189-1212, 2000.

[21] Munteanu I., Bratcu A. I., Cutululis N. A., Ceanga E., "Optimal Control of Wind Energy Systems, Towards a Global Approach", Springer, ISBN 987-1-84800-080-3, 2008.

[22] Bossanyi E. A., "The Design of Closed Loop Controllers for Wind Turbines", Wind Energy 2000, John Wiley and Sons, 3:149–163, 2000.

[23] Oyague F., "Gearbox Modeling and Load Simulation of a Baseline 750-kW Wind Turbine Using State-of-the-Art Simulation Codes", Technical Report NREL/TP-500-41160, February 2009.

[24] Brebbia C.A., et al., "Vibrations of Engineering Structures", Springer-Verlag, ISBN 3540139591, 1985.

[25] Muyeen S. M., et al., "Comparative study on transient stability analysis of wind turbine generator system using different drive train models", I.E.T. Renewable Power Generation, vol. 1, issue 2, pp. 131-141, 2007.

[26] Salman S. K., Teo A. L. J., "Windmill Modeling Consideration and Factors Influencing the Stability of a Grid-Connected Wind Power-Based Embedded Generator", I.E.E.E. Transactions on Power Systems, vol. 18, no. 2, pp. 793-802, 2003.

[27] Hinrichsen E. N., "Controls for Variable Pitch Wind Turbine Generators", I.E.E.E. Transactions on Power Apparatus and Systems, 103, pp. 886-892, 1984.

[28] Wright A. D., "Modern Control Design for Flexible Wind Turbines", Technical report: NREL/TP-500-35816, National Renewable Energy Laboratory 2004.

[29] Mattson S. E., "Modeling and Control of Large Horizontal Axis Wind Power Plants, Ph.D. Thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1984.

[30] Hinrichsen E. N., Nolen P. J., "Dynamics and Stability of Wind Turbine Generators", I.E.E.E.
Transactions on Power Apparatus and Systems, vol. 101, No. 8, pp. 2640-2648, 1982.

[31] Kos J. M., "On-line control of a large horizontal axis wind energy conversion system and its performance in a turbulent wind environment", Proceedings of the 13th Intersociety Energy Conversion Engineering Conference, California, vol. 3, pp. 2064-2073, 1978.

[32] Hinrichsen E. N., "Controls for Variable Pitch Wind Turbine Generators", I.E.E.E. Transactions on Power Apparatus and Systems, vol. 103, No. 4, pp. 886-892, 1984.

[33] Wasynczuk O., Man D. T., Sullivan J. P., "Dynamic Behaviour of a Class of Wind Turbine Generators During Random Wind Fluctuations", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-lOO, No. 6, 1981.

[34] Doman G. S., "Wind Turbine with Drive train Disturbance Isolation". US Patent No: 4329117, 1982.

[35] Wright A. D., Fingersh L. J., Balas M. J., "Testing State-Space Controls for the Controls Advanced Research Turbine", Journal of Solar Energy Engineering, vol. 128, iss. 4, pp. 506-516, 2006.

[36] Bossanyi E. A., "Wind Turbine Control for Load Reduction", Wind Energy, vol. 6, pp. 229–244, 2003.

[37] Dixit A., Suryanarayanan S., "Towards Pitch-Scheduled Drive Train Damping in Variable-Speed, Horizontal-Axis Large Wind Turbines", 44th IEEE Conference on Decision and Control and European Control Conference, 2005.

[38] Van Engelen T.G., Schaak P., Lindenburg C., “Control for Damping the Fatigue Relevant Deformation Modes of Offshore wind Turbines”, Proceedings of the 2003 European Wind Energy Conference, EWEC 2003.

[39] Bossanyi E. A., “Adaptive pitch control for a 250 kW wind turbine’. Proceedings of the Ninth BWEA Conference, Mechanical Engineering Publications, Bury St. Edmunds, UK, 1987.

[40] Liebst B.S., “A Pitch Control System for the KaMeWa Wind Turbine”, Journal of Dynamic Systems and Control, Volume 107, Issue 1, pp. 47-52, 1985.

[41] Muhando E., Senjyu T., Urasaki N., Kinjo H., Funabashi T., “Reliability and Performance Analysis of a Variable Speed Pitch-Regulated Wind Turbine by a Hybrid Controller of LQG and Neurocontroller”, Journal of LATEX Class Files, vol. 1, No. 11, November 2002.

[42] Munteanu I., Cutululis N. A., Bratcu A. I., Ceanga E., “Optimization of variable speed wind power systems based on a LQG approach”, Control Engineering Practice, 13, pp. 903-912, 2005.

[43] Knudsen T., Andersen P., Töffner-Clausen S., “Comparing PI and robust pitch controllers on a 400 kW wind turbine by full-scale tests”, Proceedings of the European Wind Energy Conference, Dublin, pp. 546-550, 1997.

[44] Lescher F., Zhao J., Borne P., “Switching LPV Controllers for a Variable Speed Pitch Regulated Wind Turbine”, International Journal of Computers, Communications & Control, Vol. I, No. 4, pp. 73-84, 2006.

[45] Lescher F., Zhao J., Borne P., “Robust gain scheduling controller for pitch regulated variable speed wind turbine”, Studies in Informatics and Control, Vol 14, No. 4, 2005.

[46] Stol K., Balas M., “Periodic Disturbance Accommodating Control for Speed Regulation of Wind Turbines,” 21st ASME Wind Energy Conference, Nevada, 2002.

[47] Balas M. J., Wright A., Hand M., Stol K., “Dynamics and Control of Horizontal Axis Wind Turbines”, Proceedings of the American Control Conference, Colorado, vol.5, pp. 3781 - 3793, 2003.

[48] Balas M. J., Lee Y. J., Kendall, L., “Disturbance Tracking Control Theory with Application to Horizontal Axis Wind Turbines,” 17th ASME Wind Energy Conference, Nevada, 1998.

[49] Stol K., “Disturbance Tracking Control and Blade Load Mitigation for Variable-Speed Wind Turbines”, Journal of Solar Energy Engineering, Vol. 125, Iss. 4, pp. 396-403, 2003.

[50] Fingersh L., Johnson K., “Baseline Results and Future Plans for the NREL Controls Advanced Research Turbine”, 23rd ASME Wind Energy Conference, Nevada, 2004.

[51] Wright A. D., Fingersh L.J., Stol K.A., “Progress in Implementing and Testing State-Space Controls for the Controls Advanced Research Turbine”, 24th ASME Wind Energy Symposium Reno, Nevada, January 10–13, 2005.

[52] De Vries E., “Trouble spots: Gearbox failures and design solutions”, Renewable Energy World, 9(2), pp. 37–47, 2006.

[53] Manwell J. F., et al., “Wind Turbine Gearbox Evaluation”, EWEC 1999, pp. 695-698, 1999.

[54] Harrison L., et al., “The gearbox challenge”, Windpower monthly, November 2005.

[55] IEA Wind Annual Report 2004.

[56] Garrad Hassan and Partners Ltd, “GH Bladed”, Verstion 3.72, Bristol, UK, 2007.

[57] Øye S., FLEX 5, Thechnical University of Denmark, Lyngby, 1999.

[58] Øye S., et al., “Verification of European Wind Turbine Design Codes”, VEWTDC: Final report. ECN-C-01-055, Energy Research Centre of the Netherlands (ECN), 2003.

[59] Buhl M. L., Wright A., Pierce K. G., “Wind Turbine Design Codes: A Comparison of the Structural Response”, Proceedings of the ASME Wind Energy Symposium, US, January 10-13, 2000.

[60] Passon P., Kühn M., Butterfield S., Jonkman J., Camp T., Larsen T. J., “OC3—Benchmark Exercise of Aero-elastic Offshore Wind Turbine Codes”, Journal of Physics: Conference Series, Volume 75, 2007.

[61] Jonkman J., et al., “Offshore Code Comparison Collaboration within IEA Wind Task 23: Phase IV Results Regarding Floating Wind Turbine Modeling”, Preprint to be presented in EWEC Warsaw, April 2010 (downloaded on 16/10/2010 from http://www.nrel.gov/docs/fy10osti/47534.pdf)

[62] Prandtl L., Tietjens O. G., “Applied Hydro and Aeromechanics”, Dover Publications, 1957.

[63] Pitt D. M., Peters D. A., “Theoretical prediction of dynamic inflow derivatives”, Vertica, Vol. 5, No 1, 1981.

[64] Leishman J. G., Beddoes T. S., “A semi-empirical model for dynamic stall”, Journal of the American Helicopter Society, July 1989.

[65] Bossanyi E.A., “GH Bladed Theory Manual”, Garrad Hassan and Partners Limited, Document No. 282/BR/009, April 2005.

[66] Stol K., Rigney B. Balas M., “Disturbance Accommodating Control of a Variable-Speed Turbine using a Symbolic Dynamics Structural Model”, Proceeding of the 2000 ASME Wind Energy Symposium, pp. 84–90, Nevada, 2000.

[67] Private Communication with Professor Bill Leithead (University of Strathclyde, UK) and Spyros Voutsinas (NTUA, GR).

[68] Adams V., Askenazi A., “Building Better Products with Finite Element Analysis”, OnWord Press, Santa Fe, N.M., 1999.

[69] Riziotis V. A., "Aerodynamic and Aeroelastic analysis of Stall on Wind Turbine Rotors", PhD thesis, NTUA (GR), 2003.

[70] IEC 61400-1, “Wind turbines-Part 1: Design requirements” 3rd edition, International Electrotechnical Commission, August 2005.

[71] Novak P., Ekelund T., Jovik I., Schmidtbauer B., “Modeling and Control of Variable-Speed Wind-Turbine Drive-System Dynamics”, IEEE control Systems, pp 28-38, August 1995.

[72] Isermann R., “Identifikation dynamischer Systeme”, Springer-Verlag Berlin, ISBN 0-387-12635-X, 1988.

[73] Ljung L., “System Identification: Theory for the User”, Prentice Hall, ISBN: 0-13-656695-2, 1997.

[74] Ljung L., “State of the Art in Linear System Identification: Time and Frequency Domain Methods”, Proceeding of the 2004 American Control Conference, 30 June-2 July 2004, vol. 1, pp. 650 - 660, 2004.

[75] Pintelon R., Schoukens J., “System identification: a Frequency Domain Approach”, IEEE Press, ISBN 0-7803-6000-1, 2001.

[76] Schoukens J., Pintelon R., “Identification of Linear Systems: A Practical Guideline to Accurate Modeling”, Pergamon Press, ISBN 0-08-040734-X, 1991.

[77] Brillinger D. R., “Time Series: Data Analysis and Theory”, Holden-Day (San Francisco), ISBN: 0-8162-1150-7, 1981.

[78] Jenkins G., Watts D., “Spectral Analysis and Its Applications”, Holden-Day (San Francisco), ISBN 0-8162-4464-2, 1968.
[79] Stenman A., Gustafsson F., “Adaptive smoothing methods for frequency-function estimation”, Automatica, v. 37, pp. 675-685, 2001
[80] Blackman R. B., Tukey J. W., “The measurement of power spectra”, Dover, ISBN: 0-486-60507-8, 1958.
[81] http://www.mathworks.com/help/toolbox/ident/ug/bq2fjka.html, accessed on 09/2010
[82] Favoreel W., De Moor B., Van Oversschee P., “Subspace State Space System Identification for Industrial Processes”, Journal of Process Control, Volume 10, Issues 2-3, April 2000, Pages 149-155, 2000.
[83] Van Overschee P., DeMoor B., “N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica, vol. 30, pp. 75-93, 1994
[84] The Mathworks Inc., “MATLAB Manual”, Version 6.5.
[85] Private Communication with David Witcher (control engineer at GL Garrad Hassan & Partners Ltd., Bristol, UK)
[86] Riziotis V. A., Voutsinas S. G., Politis E. S., Chaviaropoulos P. K., “Aeroelastic Stability of Wind Turbines: the Problem, the Methods and the Issues”, Wind Energy, vol. 7, no 4, pp. 373-392, 2004
[87] NWTC Design Codes (FAST by Jason Jonkman, Ph.D.), http://wind.nrel.gov/designcodes/simulators/fast/. Version 6.01.
[88] Narendra K. S., Mukhopadhyay S., “Adaptive Control Using Neural Networks and Approximate Models,’ IEEE Transactions on Neural Networks, vol. 8, 475-485, 1997
[89] Hagan M. T., Demuth H. B., De Jesus O., “An Introduction to The Use of Neural Networks in Control Systems”, International Journal of Robust and Nonlinear Control, Volume 12 Issue 11, Pages 959 – 985, 2002
[90] Hagan M. T., Demuth H. B., Beale M. H., “Neural Network Design”, PWS Publishing, Boston, Chapter 10, 1996
[91] Gerhard G. A., “Rating and sizing of precision low backlash planetary gearboxes for Automation Motion Control and Robotics applications”, www.neugartusa.com, accessed 29 March 2006
[92] Sonsino C. M., “Concepts and Required Materials Data for Fatigue Design of PM Components”, European Powder Metallurgy Association, European Congress and Exhibition on Powder Metallurgy, Vol.2, ISBN 1-89907-208-X, 2001
[93] Moroney M. J., “Facts from Figures”, Penguin Books, 1990

Appendix I

Simulation Parameters

(Same for all models)

PHYSICAL CONSTANTS

Air density	1.225	kg/m³
Air viscosity	1.82E-05	kg/ms
Gravitational acceleration	9.81	m/s²
Density of water	1030	kg/m³

AERODYNAMICS CONTROL

Tip loss correction	Prandtl	
Wake model	Dynamic	
Stall hysteresis model	Yes	
Starting radius for dynamic stall	50	%

TOWER SHADOW

Tower shadow model	Potential Flow
Fraction of tower diameter to use	1

VERTICAL WIND SHEAR

Wind shear model	Exponential
Wind shear exponent	0.11

TIME DEPENDENT WIND FIELD

Wind model type	Turbulent Wind created by BLADED
Wind Characteristics	Class I A winds as defined in *IEC61400-1*
Turbulence Spectrum type	Improved von Karman

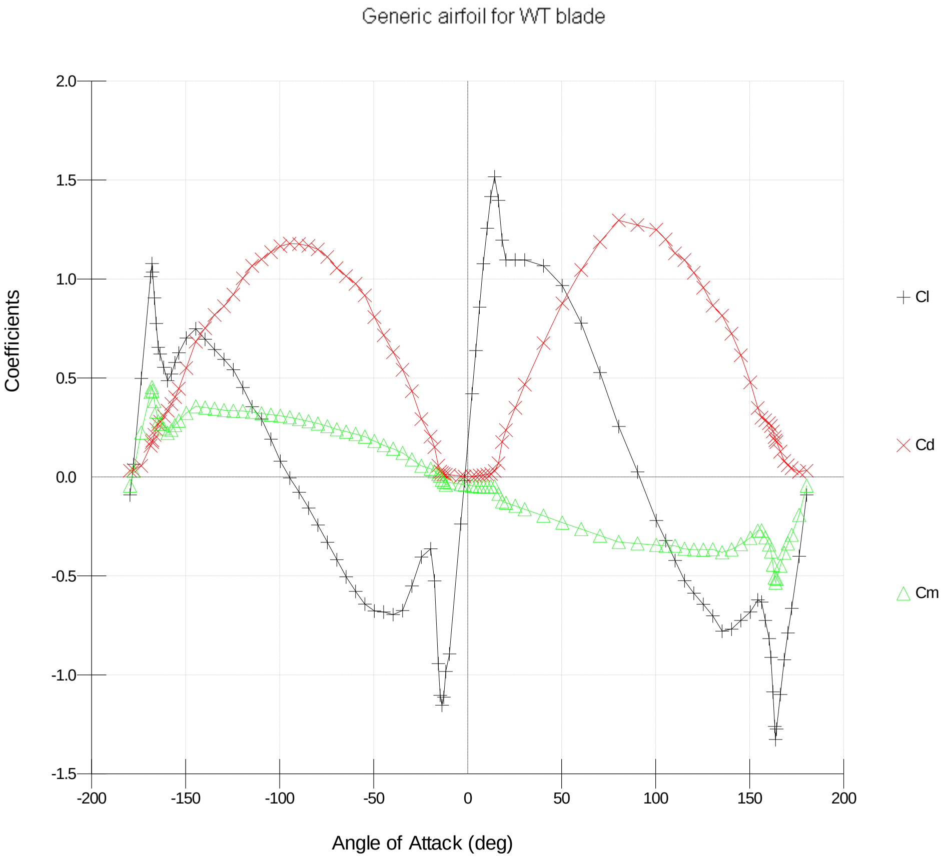
WT Model Parameters

A) Common parameters for all models:

AEROFOIL DATA

Description	Generic aerofoil for WT blade

Percentage thickness	12	%
Reynolds number	6.E+06	
Chordwise origin for pitch moments	25	%



B) Wind turbine model specific parameters:

750 kW models (750k-a, 750k-b)

GENERAL CHARACTERISTICS OF ROTOR AND TURBINE

Rotor diameter	50	m
Number of blades	3	
Tilt angle of rotor to horizontal	5	deg
Cone angle of rotor	2	deg
Rotor overhang	3	m
Rotational sense of rotor, viewed from upwind	Clockwise	
Position of rotor relative to tower	Upwind	
Transmission	Gearbox	
Aerodynamic control surfaces	Pitch	
Fixed / Variable speed	Variable	
Diameter of spinner	1.9	m
Radial position of root station	1	m

BLADE GEOMETRY

Blade length	24	m
Pitch control	Full span	

Distance from root (m)	**Chord (m)**	**Twist (deg)**	**Thickness (% chord)**	**Pitch Axis (% chord)**	**Aerofoil section reference**
0	1.3	0	100	50	cylinder
1.2	1.3	0	100	50	cylinder
2	1.3	0	85	50	cylinder
5	2.25	14	24	25	cylinder
6	2.2	11	23	25	Generic aerofoil
11	1.9	5	22	25	Generic aerofoil
15	1.5	2	20	25	Generic aerofoil
20	1	0.3	16	25	Generic aerofoil
22	0.7	0	16	25	Generic aerofoil
23	0.55	0	16	25	Generic aerofoil
24	0.05	0	16	25	Generic aerofoil

BLADE MASS DISTRIBUTION

Distance from root (m)	Centre of Mass (% chord)	Mass/unit length (kg/m)
0	50	700
1.2	50	250
2	50	225
5	33	200
6	33	180
11	33	100
15	33	70
20	33	45
22	33	30
23	33	10
24	33	1

Blade Mass Integrals

Blade Mass	3015.5	kg
First Mass Moment	21135.8	kgm
Second Mass Moment	253443	kgm²
Blade inertia about shaft	298730	kgm²

BLADE STIFFNESS DISTRIBUTION

Radial Position (m)	Stiffness about Chord Line (Nm²)	Stiffness perpendicular to Chord Line (Nm²)
0	6.9E+08	6.9E+08
1.2	6.9E+08	6.9E+08
2	5.75E+09	5.75E+09
5	8.05E+07	6.9E+08
6	5.75E+07	4.6E+08
11	2.3E+07	1.15E+08
15	6.9E+06	5.75E+07
20	1.15E+06	1.15E+07
22	241500	1.495E+06
23	115000	575000
24	1	1

HUB MASS AND INERTIA

Mass of hub	5000	kg
Mass centre of hub	0	m
Hub inertia: about shaft	3000	kgm²
perpendicular to shaft	0	kgm²
Total Rotor Mass	14046.5	kg
Total Rotor Inertia	898644	kgm²

NACELLE GEOMETRY

Nacelle width	2.5	m
Nacelle length	10	m
Nacelle height	2.5	m
Nacelle drag coefficient	1.1	

NACELLE MASS

Nacelle mass	25000	kg
Nacelle centre of mass lateral offset	0	m
Nacelle centre of mass above tower top	1.5	m
Nacelle centre of mass in front of tower axis	0	m
Total Tower-head Mass	39046.5	kg
Total Yaw Inertia: 0° azimuth	574241	kgm²
Total Yaw Inertia: 90° azimuth	574241	kgm²

DRIVE TRAIN

Gearbox ratio	41	
Position of shaft brake	High speed shaft	(Gearbox End)
Generator inertia	55	kgm²
Low speed shaft	Flexible	
Low speed shaft torsional stiffness	4.41E+07	Nm/rad
Low speed shaft torsional damping	39117.7	Nms/rad
High speed shaft	Flexible	
High speed shaft torsional stiffness	1.86E+06	Nm/rad
High speed shaft torsional damping	954.09	Nms/rad

GENERATOR CHARACTERISTICS

Generator model	Variable-speed	
Maximum generator torque	6500	Nm

ELECTRICAL LOSSES

No load power loss	5	kW
Efficiency	95	%

IMBALANCES AND FAILURE MODES

Out of balance mass	28	kg
Radius of out of balance mass	1	m
Azimuthal position of out of balance mass	0	deg

Blade	Error in Blade Set Angle (deg)
1	0
2	-0.3
3	0.3

ROTOR MODES at 0.0 degrees pitch

Mode	Frequency at 31.7 rpm (Hz)	Non-rotating frequency (Hz)
Out of plane 1	1.651	1.467
Out of plane 2	1.651	1.467
Out of plane 3	1.651	1.467
Out of plane 4	4.931	4.724
Out of plane 5	4.931	4.724
Out of plane 6	4.931	4.724
In plane 1	3.397	3.325
In plane 2	3.397	3.325
In plane 3	6.134	6.036
In plane 4	10.610	10.541
In plane 5	10.610	10.541

TOWER DETAILS

For Model 750k-a:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5	2500	Concrete	48419.8	1.135E+12
2	0.5	5	2500	Concrete	48419.8	1.135E+12
3	0.5	3.8	19	Steel	1771.66	8.47E+10
4	9	3.51	19	Steel	1635.78	6.666E+10
5	9	3.51	14	Steel	1207.04	4.933E+10
6	23	3.04	14	Steel	1044.76	3.199E+10
7	23	3.04	12	Steel	896.097	2.747E+10
8	30	2.8	12	Steel	825.072	2.145E+10
9	30	2.8	11	Steel	756.592	1.968E+10
10	40	2.46	11	Steel	664.358	1.332E+10
11	40	2.46	10	Steel	604.206	1.213E+10
12	53.5	2	10	Steel	490.763	6.499E+09

Total Tower Mass	74974	kg
Total Turbine Mass	114021	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.636	0.0050	0.0380
Fore-aft 2	4.305	0.0050	-1.7751
Side-side 1	0.633	0.0050	0.0383
Side-side 2	3.918	0.0050	-1.1084

For Model 750k-b:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5.2	2600	Concrete	52370.8	1.328E+12
2	0.5	5.2	2600	Concrete	52370.8	1.328E+12
3	0.5	4.2	23	Steel	2369.25	1.382E+11
4	6	4.04	23	Steel	2278.5	1.229E+11
5	6	4.04	19	Steel	1884.12	1.019E+11
6	14	3.8	19	Steel	1771.66	8.47E+10
7	14	3.8	15	Steel	1400.15	6.708E+10
8	28	3.39	15	Steel	1248.49	4.756E+10
9	28	3.39	12	Steel	999.675	3.815E+10
10	35	3.19	12	Steel	940.488	3.176E+10
11	35	3.19	11	Steel	862.39	2.914E+10
12	45	2.9	11	Steel	783.72	2.187E+10
13	45	2.9	10	Steel	712.716	1.991E+10
14	58.5	2.5	10	Steel	614.07	1.273E+10

Total Tower Mass	96107	kg
Total Turbine Mass	135154	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.700	0.0050	0.0338
Fore-aft 2	4.570	0.0050	4.3026
Side-side 1	0.697	0.0050	0.0339
Side-side 2	4.292	0.0050	-10.6070

2 MW models (2M-a, 2M-b, 2M-c, 2M-d)

GENERAL CHARACTERISTICS OF ROTOR AND TURBINE

Rotor diameter	75	m
Number of blades	3	
Tilt angle of rotor to horizontal	4	deg
Cone angle of rotor	0	deg
Rotor overhang	4	m
Rotational sense of rotor, viewed from upwind	Clockwise	
Position of rotor relative to tower	Upwind	
Transmission	Gearbox	
Aerodynamic control surfaces	Pitch	
Fixed / Variable speed	Variable	
Diameter of spinner	2.5	m
Radial position of root station	1.25	m

BLADE GEOMETRY

Blade length	36.25	m
Pitch control	Full span	

Distance from root (m)	**Chord (m)**	**Twist (deg)**	**Thickness (% chord)**	**Pitch Axis (% chord)**	**Aerofoil section reference**
0	1.93	13	100	50	cylinder
1.07	1.93	13	100	50	cylinder
2.15	2.15	13	85	47	cylinder
3.22	2.58	13	64	38	cylinder
4.3	3.01	13	50	33	cylinder
5.37	3.22	13	40	30	Generic aerofoil
8.59	3.22	11	30	30	Generic aerofoil
11.81	2.9	9.5	25	30	Generic aerofoil
15.04	2.58	7.8	22	30	Generic aerofoil
18.26	2.15	6.2	19	30	Generic aerofoil
21.48	1.93	4.7	17	30	Generic aerofoil
24.7	1.72	3.3	15	30	Generic aerofoil
27.39	1.5	2.3	13	30	Generic aerofoil
29.54	1.4	1.5	12	30	Generic aerofoil
31.69	1.18	0.8	12	30	Generic aerofoil
33.3	1.07	0.3	11.5	30	Generic aerofoil
34.37	0.97	0	11	30	Generic aerofoil
35.77	0.64	2.75	11	30	Generic aerofoil
36.25	0.03	4	11	30	Generic aerofoil

BLADE MASS DISTRIBUTION

Distance from root (m)	**Centre of Mass (% chord)**	**Mass/unit length (kg/m)**
0	50	949
1.07	50	324
2.15	47	291
3.22	38	243
4.3	33	205
5.37	30	205
8.59	30	183
11.81	30	173
15.04	30	151
18.26	30	129
21.48	30	108
24.7	30	91
27.39	30	72
29.54	30	60
31.69	30	52
33.3	30	49
34.37	30	47
35.77	30	36
36.25	30	22

Blade Mass Integrals

Blade Mass	5320	kg
First Mass Moment	63515	kgm
Second Mass Moment	1.252E+06	kgm^2
Blade inertia about shaft	1.419E+06	kgm^2

BLADE STIFFNESS DISTRIBUTION

Radial Position (m)	Stiffness about Chord Line (Nm^2)	Stiffness perpendicular to Chord Line (Nm^2)
0	5.72E+09	5.72E+09
1.07	1.86E+09	2.E+09
2.15	1.6E+09	1.86E+09
3.22	1.08E+09	1.6E+09
4.3	7.59E+08	1.28E+09
5.37	6.39E+08	1.09E+09
8.59	4.26E+08	9.85E+08
11.81	2.66E+08	6.79E+08
15.04	1.6E+08	4.33E+08
18.26	8.38E+07	2.79E+08
21.48	4.52E+07	1.66E+08
24.7	2.26E+07	9.32E+07
27.39	1.24E+07	5.59E+07
29.54	7.19E+06	3.99E+07
31.69	4.19E+06	2.99E+07
33.3	1.73E+06	1.86E+07
34.37	506000	1.25E+07
35.77	87200	3.46E+06
36.25	2400	6260

HUB MASS AND INERTIA

Mass of hub	17000	kg
Mass centre of hub	0	m
Hub inertia: about shaft	12000	kgm^2
perpendicular to shaft	0	kgm^2
Total Rotor Mass	32961	kg
Total Rotor Inertia	4.268E+06	kgm^2

NACELLE GEOMETRY

Nacelle width	2.5	m
Nacelle length	6	m
Nacelle height	2.5	m
Nacelle drag coefficient	1.4	

NACELLE MASS

Nacelle mass	65000	kg
Nacelle centre of mass lateral offset	0	m
Nacelle centre of mass above tower top	1.3	m
Nacelle centre of mass in front of tower axis	-0.6	m
Total Tower-head Mass	97961	kg
Total Yaw Inertia: 0° azimuth	2.959E+06	kgm²
Total Yaw Inertia: 90° azimuth	2.959E+06	kgm²

DRIVE TRAIN

Gearbox ratio	84.15	
Position of shaft brake	High speed shaft	(Gearbox End)
Generator inertia	110	kgm²
High speed shaft inertia:	0	kgm²
Low speed shaft	Flexible	
Low speed shaft torsional stiffness	1.724E+08	Nm/rad
Low speed shaft torsional damping	214160	Nms/rad
High speed shaft	Flexible	
High speed shaft torsional stiffness	1.723E+06	Nm/rad
High speed shaft torsional damping	2544.98	Nms/rad

GENERATOR CHARACTERISTICS

Generator model	Variable-speed	
Maximum generator torque	14400	Nm

MECHANICAL LOSS TORQUE (kNm, referred to low speed shaft)

Low speed shaft torque (kNm)	Loss torque (kNm)
0	23
960	37
1280	49

ELECTRICAL LOSSES

No load power loss	5	kW
Efficiency	95	%

IMBALANCES AND FAILURE MODES

Out of balance mass	130	kg
Radius of out of balance mass	1	m
Azimuthal position of out of balance mass	0	deg

Blade	Error in Blade Set Angle (deg)
1	0
2	-0.3
3	0.3

ROTOR MODES at -2.0 degrees pitch

Mode	Frequency at 17.83 rpm (Hz)	Non-rotating frequency (Hz)
Out of plane 1	1.059	0.977
Out of plane 2	1.059	0.977
Out of plane 3	1.059	0.977
Out of plane 4	2.910	2.812
Out of plane 5	2.910	2.812
Out of plane 6	2.910	2.812
In plane 1	1.585	1.541
In plane 2	1.585	1.541
In plane 3	3.198	3.137
In plane 4	4.971	4.921
In plane 5	4.971	4.921

TOWER DETAILS

For Model 2M-a:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	4.5	50	Steel	5487.17	3.634E+11
2	2.5	4.42475	48.5691	Steel	5241.74	3.357E+11
3	5	4.34951	47.1382	Steel	5001.49	3.096E+11
4	10	4.19901	44.2763	Steel	4536.63	2.619E+11
5	20	3.89802	38.5527	Steel	3669.45	1.828E+11
6	30	3.59703	32.829	Steel	2885.62	1.226E+11
7	40	3.29604	27.1054	Steel	2185.15	7.809E+10
8	50	2.99505	21.3817	Steel	1568.03	4.637E+10
9	63	2.6	13.8694	Steel	884.56	1.978E+10

Total Tower Mass	184081	kg
Total Turbine Mass	282042	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.657	0.0050	0.0326
Fore-aft 2	3.641	0.0050	-3.5036
Side-side 1	0.653	0.0050	0.0329
Side-side 2	3.287	0.0050	-1.2065

For Model 2M-b:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5.6	2800	Concrete	60737.7	1.786E+12
2	0.5	5.6	2800	Concrete	60737.7	1.786E+12
3	0.5	5	50	Steel	6103.71	5.002E+11
4	10	4.72	50	Steel	5758.45	4.2E+11
5	10	4.72	44	Steel	5073.94	3.71E+11
6	16	4.54	44	Steel	4878.62	3.298E+11
7	16	4.54	39	Steel	4329.06	2.933E+11
8	21	4.4	39	Steel	4194.4	2.668E+11
9	21	4.4	35	Steel	3767.67	2.401E+11
10	26	4.25	35	Steel	3638.19	2.162E+11
11	26	4.25	29	Steel	3018.79	1.799E+11
12	34	4.01	29	Steel	2847.15	1.509E+11
13	34	4.01	25	Steel	2456.9	1.305E+11
14	48	3.6	25	Steel	2204.12	9.42E+10
15	48	3.6	21	Steel	1853.54	7.94E+10
16	55	3.39	21	Steel	1744.78	6.622E+10
17	55	3.39	19	Steel	1579.55	6.002E+10
18	65	3.1	19	Steel	1443.66	4.583E+10
19	65	3.1	15	Steel	1141.21	3.632E+10
20	78.5	2.7	15	Steel	993.24	2.395E+10

Total Tower Mass	254604	kg
Total Turbine Mass	352564	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0 deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.513	0.0050	0.0274
Fore-aft 2	2.948	0.0050	1.9964
Side-side 1	0.511	0.0050	0.0276
Side-side 2	2.732	0.0050	-86.9390

For Model 2M-c:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5.6	2800	Concrete	60737.7	1.786E+12
2	0.5	5.6	2800	Concrete	60737.7	1.786E+12
3	0.5	5	33	Steel	4042.29	3.335E+11
4	10	4.72	33	Steel	3814.42	2.802E+11
5	10	4.72	29	Steel	3354.93	2.469E+11
6	16	4.54	29	Steel	3226.2	2.195E+11
7	16	4.54	26	Steel	2894.37	1.972E+11
8	21	4.4	26	Steel	2804.61	1.794E+11
9	21	4.4	24	Steel	2590.04	1.659E+11
10	26	4.25	24	Steel	2501.26	1.494E+11
11	26	4.25	21	Steel	2190.15	1.31E+11
12	34	4.01	21	Steel	2065.87	1.099E+11
13	34	4.01	19	Steel	1870.06	9.961E+10
14	48	3.6	19	Steel	1677.95	7.195E+10
15	48	3.6	17	Steel	1502.15	6.449E+10
16	55	3.39	17	Steel	1414.11	5.38E+10
17	55	3.39	16	Steel	1331.33	5.068E+10
18	65	3.2	16	Steel	1256.36	4.259E+10
19	65	3.2	16	Steel	1256.36	4.259E+10
20	78.5	3	16	Steel	1177.44	3.506E+10

Total Tower Mass	195841 kg
Total Turbine Mass	293802 kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.441	0.0050	0.0252
Fore-aft 2	2.940	0.0050	1.4487
Side-side 1	0.439	0.0050	0.0253
Side-side 2	2.745	0.0050	4.1568

For Model 2M-d:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	7	3500	Concrete	94902.7	4.361E+12
2	0.5	7	3500	Concrete	94902.7	4.361E+12
3	0.5	5.7	50	Steel	6966.86	7.438E+11
4	10	5.42	50	Steel	6621.6	6.386E+11
5	10	5.42	45	Steel	5965.01	5.763E+11
6	20	5.12	45	Steel	5632.08	4.851E+11
7	20	5.12	41	Steel	5135.47	4.43E+11
8	30	4.83	41	Steel	4842.24	3.714E+11
9	30	4.83	36	Steel	4256.18	3.271E+11
10	36	4.65	36	Steel	4096.37	2.916E+11
11	36	4.65	32	Steel	3644.38	2.599E+11
12	41	4.5	32	Steel	3526.01	2.354E+11
13	41	4.5	29	Steel	3197.59	2.138E+11
14	46	4.35	29	Steel	3090.31	1.93E+11
15	46	4.35	24	Steel	2560.45	1.602E+11
16	54	4.12	24	Steel	2424.32	1.36E+11
17	54	4.12	21	Steel	2122.83	1.193E+11
18	68	3.7	21	Steel	1905.32	8.624E+10
19	68	3.7	18	Steel	1634.47	7.41E+10
20	75	3.5	18	Steel	1545.68	6.267E+10
21	75	3.5	16	Steel	1374.73	5.58E+10
22	85	3.3	16	Steel	1295.82	4.673E+10
23	85	3.3	15	Steel	1215.19	4.385E+10
24	98.5	3.2	15	Steel	1178.2	3.997E+10

Total Tower Mass	367349	kg
Total Turbine Mass	465309	kg

Aerodynamic Details

Drag coefficient for tower	0.6	

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.424	0.0050	0.0223
Fore-aft 2	2.293	0.0050	0.4303
Side-side 1	0.422	0.0050	0.0224
Side-side 2	2.209	0.0050	0.5353

3 MW models (3M-a, 3M-b, 3M-c, 3M-d)

GENERAL CHARACTERISTICS OF ROTOR AND TURBINE

Rotor diameter	96	m
Number of blades	3	
Tilt angle of rotor to horizontal	4	deg
Cone angle of rotor	0	deg
Rotor overhang	5	m
Rotational sense of rotor, viewed from upwind	Clockwise	
Position of rotor relative to tower	Upwind	
Transmission	Gearbox	
Aerodynamic control surfaces	Pitch	
Fixed / Variable speed	Variable	
Diameter of spinner	4	m
Radial position of root station	1.5	m

BLADE GEOMETRY

Blade length	46.5	m
Pitch control	Full span	

Distance from root (m)	Chord (m)	Twist (deg)	Thickness (% chord)	Pitch Axis (% chord)	Aerofoil section reference
0	2.37	13	25	100	cylinder
1.37	2.37	13	25	100	cylinder
2.76	2.64	13	25	85	cylinder
4.13	3.17	13	25	64	cylinder
5.52	3.7	13	25	50	cylinder
6.89	3.96	13	25	40	Generic aerofoil
11.02	3.96	11	25	30	Generic aerofoil
15.15	3.56	9.5	25	25	Generic aerofoil
19.29	3.17	7.8	25	22	Generic aerofoil
23.42	2.64	6.2	25	19	Generic aerofoil
27.55	2.37	4.7	25	17	Generic aerofoil
31.68	2.11	3.3	25	15	Generic aerofoil
35.13	1.84	2.3	25	13	Generic aerofoil
37.89	1.72	1.5	25	12	Generic aerofoil
40.65	1.45	0.8	25	12	Generic aerofoil
42.72	1.31	0.3	25	11.5	Generic aerofoil
44.09	1.19	0	25	11	Generic aerofoil
45.88	0.79	2.75	25	11	Generic aerofoil
46.5	0.04	4	25	11	Generic aerofoil

BLADE MASS DISTRIBUTION

Distance from root (m)	Centre of Mass (% chord)	Mass/unit length (kg/m)
0	50	1529.63
1.37	50	522.23
2.76	47	469.04
4.13	38	391.68
5.52	33	330.43
6.89	30	330.43
11.02	30	294.97
15.15	30	278.85
19.29	30	243.39
23.42	30	207.93
27.55	30	174.08
31.68	30	146.68
35.13	30	116.05
37.89	30	96.71
40.65	30	83.82
42.72	30	78.98
44.09	30	75.76
45.88	30	58.03
46.5	30	35.46

Blade Mass Integrals

Blade Mass	10998.4	kg
First Mass Moment	168439	kgm
Second Mass Moment	4.258E+06	kgm²
Blade inertia about shaft	4.788E+06	kgm²

BLADE STIFFNESS DISTRIBUTION

Radial Position (m)	Stiffness about Chord Line (Nm²)	Stiffness perpendicular to Chord Line (Nm²)
0	1.472E+10	1.472E+10
1.37	4.788E+09	5.149E+09
2.76	4.113E+09	4.788E+09
4.13	2.774E+09	4.114E+09
5.52	1.957E+09	3.296E+09
6.89	1.643E+09	2.802E+09
11.02	1.092E+09	2.536E+09
15.15	6.84E+08	1.748E+09
19.29	4.113E+08	1.112E+09
23.42	2.157E+08	7.173E+08
27.55	1.159E+08	4.265E+08
31.68	5.814E+07	2.394E+08
35.13	3.192E+07	1.434E+08
37.89	1.853E+07	1.026E+08
40.65	1.074E+07	7.695E+07
42.72	4.446E+06	4.788E+07
44.09	1.302E+06	3.211E+07
45.88	224200	8.902E+06
46.5	6175	16055

HUB MASS AND INERTIA

Mass of hub	27000	kg
Mass centre of hub	0	m
Hub inertia: about shaft	30000	kgm²
perpendicular to shaft	0	kgm²
Total Rotor Mass	59995.2	kg
Total Rotor Inertia	1.439E+07	kgm²

NACELLE GEOMETRY

Nacelle width	3.5	m
Nacelle length	8	m
Nacelle height	3.5	m
Nacelle drag coefficient	1.4	

NACELLE MASS

Nacelle mass	100000	kg
Nacelle centre of mass lateral offset	0	m
Nacelle centre of mass above tower top	1.3	m
Nacelle centre of mass in front of tower axis	-0.6	m
Total Tower-head Mass	159995	kg
Total Yaw Inertia: 0° azimuth	9.077E+06	kgm²
Total Yaw Inertia: 90° azimuth	9.077E+06	kgm²

DRIVE TRAIN

Gearbox ratio	75.4	
Position of shaft brake	High speed shaft	(Gearbox End)
Generator inertia	255	kgm²
High speed shaft inertia:	0	kgm²
Low speed shaft	Flexible	
Low speed shaft torsional stiffness	4.564E+08	Nm/rad
Low speed shaft torsional damping	493348	Nms/rad
High speed shaft	Flexible	
High speed shaft torsional stiffness	5.683E+06	Nm/rad
High speed shaft torsional damping	6543.23	Nms/rad

GENERATOR CHARACTERISTICS

Generator model	Variable-speed	
Maximum generator torque	27650	Nm

MECHANICAL LOSS TORQUE (kNm, referred to low speed shaft)

Low speed shaft torque (kNm)	Loss torque (kNm)
0	30
960	48
1280	84

ELECTRICAL LOSSES

No load power loss	5	kW
Efficiency	95	%

IMBALANCES AND FAILURE MODES

Out of balance mass	130	kg
Radius of out of balance mass	1	m
Azimuthal position of out of balance mass	0	deg

Blade	Error in Blade Set Angle (deg)
1	0
2	-0.3
3	0.3

ROTOR MODES at -2.0 degrees pitch

Mode	Frequency at 15.92 rpm (Hz)	Non-rotating frequency (Hz)
Out of plane 1	0.833	0.750
Out of plane 2	0.833	0.750
Out of plane 3	0.833	0.750
Out of plane 4	2.259	2.159
Out of plane 5	2.259	2.159
Out of plane 6	2.259	2.159
In plane 1	1.229	1.183
In plane 2	1.229	1.183
In plane 3	2.478	2.415
In plane 4	3.827	3.776
In plane 5	3.827	3.776

TOWER DETAILS

For Model 3M-a:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5.2	2600	Concrete	60737.7	1.79E+12
2	0.5	5.2	2600	Concrete	60737.7	1.79E+12
3	0.5	4.6	42	Steel	4721.1	3.28E+11
4	2.5	4.6	42	Steel	4721.1	3.28E+11
5	2.5	4.6	40	Steel	4498.25	3.128E+11
6	5	4.6	40	Steel	4498.25	3.128E+11
7	5	4.6	38	Steel	4275.23	2.975E+11
8	10	4.45	38	Steel	4134.66	2.692E+11
9	10	4.45	34	Steel	3702.77	2.415E+11
10	20	4.16	34	Steel	3459.6	1.97E+11
11	20	4.16	29	Steel	2954.43	1.686E+11
12	30	3.86	29	Steel	2739.87	1.345E+11
13	30	3.86	23	Steel	2176.4	1.072E+11
14	40	3.57	23	Steel	2011.91	8.465E+10
15	40	3.57	19	Steel	1663.89	7.016E+10
16	50	3.27	19	Steel	1523.32	5.384E+10
17	50	3.27	16	Steel	1283.98	4.546E+10
18	66	2.8	16	Steel	1098.52	2.847E+10

Total Tower Mass	192302	kg
Total Turbine Mass	352298	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.478	0.0050	0.0317
Fore-aft 2	2.663	0.0050	-0.5063
Side-side 1	0.472	0.0050	0.0322
Side-side 2	2.214	0.0050	-0.3304

For Model 3M-b:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5.6	2800	Concrete	60737.7	1.786E+12
2	0.5	5.6	2800	Concrete	60737.7	1.786E+12
3	0.5	5	50	Steel	6103.71	5.002E+11
4	10	4.72	50	Steel	5758.45	4.2E+11
5	10	4.72	44	Steel	5073.94	3.71E+11
6	16	4.54	44	Steel	4878.62	3.298E+11
7	16	4.54	39	Steel	4329.06	2.933E+11
8	21	4.4	39	Steel	4194.4	2.668E+11
9	21	4.4	35	Steel	3767.67	2.401E+11
10	26	4.25	35	Steel	3638.19	2.162E+11
11	26	4.25	29	Steel	3018.79	1.799E+11
12	34	4.01	29	Steel	2847.15	1.509E+11
13	34	4.01	25	Steel	2456.9	1.305E+11
14	48	3.6	25	Steel	2204.12	9.42E+10
15	48	3.6	21	Steel	1853.54	7.94E+10
16	55	3.39	21	Steel	1744.78	6.622E+10
17	55	3.39	19	Steel	1579.55	6.002E+10
18	65	3.1	19	Steel	1443.66	4.583E+10
19	65	3.1	15	Steel	1141.21	3.632E+10
20	78.5	2.7	15	Steel	993.24	2.395E+10

Total Tower Mass	254604	kg
Total Turbine Mass	414599	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.410	0.0050	0.0281
Fore-aft 2	2.261	0.0050	-0.7522
Side-side 1	0.405	0.0050	0.0286
Side-side 2	1.902	0.0050	-0.4345

For Model 3M-c:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	5.6	2800	Concrete	60737.7	1.786E+12
2	0.5	5.6	2800	Concrete	60737.7	1.786E+12
3	0.5	5	33	Steel	4042.29	3.335E+11
4	10	4.72	33	Steel	3814.42	2.802E+11
5	10	4.72	29	Steel	3354.93	2.469E+11
6	16	4.54	29	Steel	3226.2	2.195E+11
7	16	4.54	26	Steel	2894.37	1.972E+11
8	21	4.4	26	Steel	2804.61	1.794E+11
9	21	4.4	24	Steel	2590.04	1.659E+11
10	26	4.25	24	Steel	2501.26	1.494E+11
11	26	4.25	21	Steel	2190.15	1.31E+11
12	34	4.01	21	Steel	2065.87	1.099E+11
13	34	4.01	19	Steel	1870.06	9.961E+10
14	48	3.6	19	Steel	1677.95	7.195E+10
15	48	3.6	17	Steel	1502.15	6.449E+10
16	55	3.39	17	Steel	1414.11	5.38E+10
17	55	3.39	16	Steel	1331.33	5.068E+10
18	65	3.2	16	Steel	1256.36	4.259E+10
19	65	3.2	16	Steel	1256.36	4.259E+10
20	78.5	3	16	Steel	1177.44	3.506E+10

Total Tower Mass	195841	kg
Total Turbine Mass	355836	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.353	0.0050	0.0256
Fore-aft 2	2.307	0.0050	-0.9307
Side-side 1	0.350	0.0050	0.0259
Side-side 2	1.961	0.0050	-0.5123

For Model 3M-d:

Structural Details

Station Number	Height (m)	Diameter (m)	Wall thickness (mm)	Material	Mass/unit length (kg/m)	Stiffness (Nm²)
1	0	7	3500	Concrete	94902.7	4.361E+12
2	0.5	7	3500	Concrete	94902.7	4.361E+12
3	0.5	5.7	50	Steel	6966.86	7.438E+11
4	10	5.42	50	Steel	6621.6	6.386E+11
5	10	5.42	45	Steel	5965.01	5.763E+11
6	20	5.12	45	Steel	5632.08	4.851E+11
7	20	5.12	41	Steel	5135.47	4.43E+11
8	30	4.83	41	Steel	4842.24	3.714E+11
9	30	4.83	36	Steel	4256.18	3.271E+11
10	36	4.65	36	Steel	4096.37	2.916E+11
11	36	4.65	32	Steel	3644.38	2.599E+11
12	41	4.5	32	Steel	3526.01	2.354E+11
13	41	4.5	29	Steel	3197.59	2.138E+11
14	46	4.35	29	Steel	3090.31	1.93E+11
15	46	4.35	24	Steel	2560.45	1.602E+11
16	54	4.12	24	Steel	2424.32	1.36E+11
17	54	4.12	21	Steel	2122.83	1.193E+11
18	68	3.7	21	Steel	1905.32	8.624E+10
19	68	3.7	18	Steel	1634.47	7.41E+10
20	75	3.5	18	Steel	1545.68	6.267E+10
21	75	3.5	16	Steel	1374.73	5.58E+10
22	85	3.3	16	Steel	1295.82	4.673E+10
23	85	3.3	15	Steel	1215.19	4.385E+10
24	98.5	3.2	15	Steel	1178.2	3.997E+10

Total Tower Mass	367349	kg
Total Turbine Mass	527344	kg

Aerodynamic Details

Drag coefficient for tower	0.6

Tower Modes

Rotor azimuth angle	0	deg

Mode	Frequency (Hz)	Damping factor	Tower top slope
Fore-aft 1	0.342	0.0050	0.0228
Fore-aft 2	1.974	0.0050	156.5500
Side-side 1	0.339	0.0050	0.0230
Side-side 2	1.755	0.0050	-1.1443

Appendix II

Stage 1, Experimental Procedure Code, scripted in C++

1. Chirp signal Experiment

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <sstream>

#define NINT(a)  ((a) >= 0.0 ? (int)((a)+0.5) : (int)((a)-0.5))


extern "C"
{
void __declspec(dllexport) __cdecl DISCON (float *avrSwap,
                                           int *aviFail,
                                           char *accInfile,
                                           char *avcOutname,
                                           char *avcMsg);
}

//Main DLL routine START
void __declspec(dllexport) __cdecl DISCON (float *avrSwap,
                                           int *aviFail,
                                           char *accInfile,
                                           char *avcOutname,
                                           char *avcMsg)

{
char Message[257], InFile[257], OutName[1025];
static float TorqueDemand;
static float Amplitude, Frequency, InitialTorque;
double pi, Time;
static float Timedif, TorqueDemandNew, Hysteresis,
static float PrevGenSpeed, GenSpeedTimer;
static int HystFlag, Droping, StartFlag;
int     decimal = 2;
int     outStart,ValidExp;
int     precision = 10;
FILE *file;

//Take local copies of strings
memcpy(InFile,accInfile,256);
InFile[NINT(avrSwap[49])+1] = '\0';
memcpy(OutName,avcOutname,1024);
OutName[NINT(avrSwap[63])] = '\0';

//Set message to blank
memset(Message,' ',257);
Message[256] = '\0';
```

```
pi = 4*atan(1.0);

//Read any parameters from the DISCON.IN file
//(which contains the "External controller parameters" text
//from the GH Bladed interface
//In this implementation this contains the amplitude,
//frequency and average of the torque demand chirp signal
if (NINT(avrSwap[0]) == 0)    //First time only
   {
   if ((file = fopen(InFile, "r")) == NULL)
      {
      strcpy(Message, "Could not open file ");
      strcat(Message, InFile);
      *aviFail = -1;    //Set error flag
      return;
      }
   if (fscanf(file, " %f ", &Amplitude) != 1)
      {
      strcpy(Message, "Could not read Amplitude");
      *aviFail = -1;    //Set error flag
      return;
      }
   if (fscanf(file, " %f ", &Frequency) != 1)
      {
      strcpy(Message, "Could not read Frequency");
      *aviFail = -1;    //Set error flag
      return;
      }
   if (fscanf(file, " %f ", &InitialTorque) != 1)
      {
      strcpy(Message, "Could not read InitialTorque");
      *aviFail = -1;    //Set error flag
      return;
      }
   }

// set initial values
if (NINT(avrSwap[0]) == 0)    //First time only
   {
   TorqueDemand = InitialTorque;
   Timedif = 0;
   HystFlag = 0;
   Hysteresis = 5;
   Droping = 0;
   StartFlag = 1;
   ValidExp = 0;
   }


if ((avrSwap[19] > (0.9 * avrSwap[16])) && (avrSwap[19] < (1.1 *
avrSwap[18])) && (Hysteresis >= 5) && (StartFlag==1))
   {
   //  ----- [[ Valid Experiment ]]   -----------------------


   if ((avrSwap[44] == avrSwap[5]) && (TorqueDemand <=
InitialTorque+(2* Amplitude)) && (TorqueDemand >= InitialTorque-(2*
Amplitude)))
      {
      // if ready to start experiment
      // set time variable
      Time = avrSwap[1] - Timedif;
      if ((Time>=70)&&(Time<140))
         {
         Time=Time-69;
         }
      if ((Time>=140)&&(Time<210))
         {
         Time=Time-139;
         }
      if ((Time>=210)&&(Time<280))
         {
         Time=Time-209;
         }
      if (Time>=280)
         {
         Time=Time-279;
         }

      // set control flags
      HystFlag = 0;
      ValidExp = 1;

      // calculate correct torque (chirp Signal)
      TorqueDemandNew = InitialTorque +
Amplitude*(float)sin(2*pi*Frequency*Time*Time/180);

      //---  ramp torque if needed to avoid
      // transients at initiation of valid experiment ---

      if (TorqueDemandNew<(TorqueDemand- (2* Amplitude)))
         {
         TorqueDemand= TorqueDemand-(2* Amplitude);
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+(2* Amplitude)))
            {
            TorqueDemand= TorqueDemand+(2* Amplitude);
            }
         else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }
      }
   else // if not ready to start experiment
      {
      // set control flags
      ValidExp = 0;
      TorqueDemandNew = InitialTorque;
      //---  ramp torque if needed ---

      if (TorqueDemandNew<(TorqueDemand- (2* Amplitude)))
         {
         TorqueDemand= TorqueDemand-(2* Amplitude);
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+(2* Amplitude)))
            {
            TorqueDemand= TorqueDemand+(2* Amplitude);
            }
         else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }

      // set the pitch angle to minimum (Ramp at 6deg/s)
      if ((avrSwap[3] - (0.1047 * avrSwap[2])) > avrSwap[5])
         {
         avrSwap[44] =  (avrSwap[3] - (0.1047 * avrSwap[2]));
         }
      else
         {
         avrSwap[44] = avrSwap[5];
         }
      }

   }

else
   {
   // ______ --- Non Valid Experimental Region --- ________

   ValidExp = 0;
   if ((avrSwap[19] < (0.8 * avrSwap[18])) && (avrSwap[19] > (1.1 *
avrSwap[16])))
      {
      StartFlag = 1;
      }
   else
      {
      StartFlag = 0;
      }

   Timedif = avrSwap[1];



   if ((avrSwap[19] > (1.1 * avrSwap[18])) ||  (Hysteresis < 5))
      {
      //  ---- [[ above rated ]]   ----
      // ---------  Pitch Control ----------------

      if (avrSwap[19] > (0.7 * avrSwap[18]))
         {
         //set the pitch angle to .2 of the maximum
         // allowable (Ramp at 6deg/s)

         if ((avrSwap[3] + (0.1047 * avrSwap[2])) < (avrSwap[6]*0.2))
            {
            avrSwap[44] =  (avrSwap[3] + (0.1047 * avrSwap[2]));
            }
         else
            {
            avrSwap[44] = avrSwap[6]*0.2;
            }
         }
      else
         {
         if ((avrSwap[3] - (0.1047 * avrSwap[2])) > avrSwap[5])
```

```
            {
            avrSwap[44] =  (avrSwap[3] - (0.1047 * avrSwap[2]));
            }
         else
            {
            avrSwap[44] = avrSwap[5];
            }
         }



      if (HystFlag == 0)
         {
         HystFlag = 1;
         Hysteresis = 0;
         }

      if (avrSwap[19] < (0.9 * avrSwap[18]))
         Hysteresis = Hysteresis + avrSwap[2];

      // torque needs to be 1) equal to max demanded torque above
      // rated (under normal conditions)
      //                or 2) the demanded torque below rated
      // (under normal conditions)

      TorqueDemandNew = avrSwap[19] * avrSwap[19] * avrSwap[15];

      if (TorqueDemandNew > (avrSwap[17] * avrSwap[17] *
      avrSwap[15]))
         TorqueDemandNew = avrSwap[17] * avrSwap[17] * avrSwap[15];

      // ramp torque if needed

      if (TorqueDemandNew<(TorqueDemand-10))
         {
         TorqueDemand= TorqueDemand-10;
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+10))
            {
            TorqueDemand= TorqueDemand+10;
            }
         else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }


      }
   else
      {
      //  --- [[ below rated ]]   ---
      // 1) set the pitch angle to minimum (0.001 below
      // minimum is used here for identification of non
      // valid experiment regions) (Ramp at 6deg/s)

      if ((avrSwap[3] - (0.1047 * avrSwap[2])) > (avrSwap[5] -
0.001))
         {
```

```
         avrSwap[44] =  (avrSwap[3] - (0.1047 * avrSwap[2]));
         }
      else
         {
         avrSwap[44] = avrSwap[5] - 0.001;
         }


      // 2) set the Torque demand acording to the optimal
      // mode gain

      TorqueDemandNew = avrSwap[19] * avrSwap[19] * avrSwap[15];

      //--------  ramp torque if needed ---------

      if (TorqueDemandNew<(TorqueDemand-10))
         {
         TorqueDemand= TorqueDemand-10;
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+10))
            {
            TorqueDemand= TorqueDemand+10;
            }
            else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }

      // 3) update flags
      Hysteresis = Hysteresis + avrSwap[2];
      }

   }

strcpy(OutName, "validFlag:-;");

//Output new Generator Torque demand to wind turbine controller
avrSwap[46] = TorqueDemand;
return;

//Main DLL routine END
}
```

2. Chirp signal Experiment

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <sstream>
#include <iostream>
```

```
#include <iomanip>
#include <fstream>
using namespace std;

#define NINT(a) ((a) >= 0.0 ? (int)((a)+0.5) : (int)((a)-0.5))


extern "C"
{
void __declspec(dllexport) __cdecl DISCON (float *avrSwap,
                                           int *aviFail,
                                           char *accInfile,
                                           char *avcOutname,
                                           char *avcMsg);
}

//Main DLL routine START
void __declspec(dllexport) __cdecl DISCON (float *avrSwap,
                                           int *aviFail,
                                           char *accInfile,
                                           char *avcOutname,
                                           char *avcMsg)

{
char Message[257], InFile[257], OutName[1025];
static float TorqueDemand;
static float Amplitude, Frequency, InitialTorque;
double pi, Time;
static float Timedif, TorqueDemandNew, Hysteresis,
static float PrevGenSpeed, GenSpeedTimer;
static int HystFlag, Droping, StartFlag, No;
int     outStart,ValidExp;
int x;
int i;
static int PRBSsignal[12001];
FILE *file;

//Take local copies of strings
memcpy(InFile,accInfile,256);
InFile[NINT(avrSwap[49])+1] = '\0';
memcpy(OutName,avcOutname,1024);
OutName[NINT(avrSwap[63])] = '\0';

//Set message to blank
memset(Message,' ',257);
Message[256] = '\0';

pi = 4*atan(1.0);

//Read any parameters from C:\\BladedPhD\\PRBSsignal.txt file
//which contains the PRBS experimental controller base signal. This
//signal is a PRBS signal created by MATLAB

if (NINT(avrSwap[0]) == 0)    //First time only
   {
   i=0;
   ifstream inFile2;

   inFile2.open("C:\\BladedPhD\\PRBSsignal.txt");
   if (!inFile2)
      {
```

```
      strcpy(Message, "Could not read PRBSsignal.txt");
      exit(1); // terminate with error
      }

   while (inFile2 >> x)
      {
      i=i+1;
      PRBSsignal[i] =  x;
      }

   inFile2.close();
   }

// set initial values

if (NINT(avrSwap[0]) == 0)                //First time only
   {
   TorqueDemand=  avrSwap[16] * avrSwap[16]* avrSwap[15];
   Timedif = 0;
   HystFlag = 0;
   No=-60;
   Hysteresis = 5;
   Droping = 0;
   StartFlag = 1;
   ValidExp = 0;
   }

No=No+1;

if ((avrSwap[19] > (0.9 * avrSwap[16])) && (avrSwap[19] < (
avrSwap[18])) && (Hysteresis >= 5) && (StartFlag==1))
   {
   //  ----- [[ Valid Experiment ]]   -----------------------

   BaseTorque = avrSwap[19] * avrSwap[19] * avrSwap[15];
   AddedTorqueAmpl= BaseTorque*0.05;
   if (No>0)
      {
      // if ready to start experiment
      AddedTorque= PRBSsignal[No]*AddedTorqueAmpl;
      TorqueDemandNew = BaseTorque + AddedTorque;
      }
   else
      {
      TorqueDemandNew = TorqueDemand;
      }



   if ((avrSwap[44] == avrSwap[5]) && (TorqueDemandNew <=
   TorqueDemand +(2* AddedTorqueAmpl)) && (TorqueDemandNew >=
   TorqueDemand-(2* AddedTorqueAmpl)))
      {
      //--- valid experiment
      HystFlag = 0;
      ValidExp = 1;
      TorqueDemand = TorqueDemandNew;
      }

   else
      {
```

```
      // if not ready to start experiment
      // set control flags
      ValidExp = 0;

      //---  ramp torque if needed ---

      if (TorqueDemandNew<(TorqueDemand- (2* AddedTorqueAmpl)))
         {
         TorqueDemand= TorqueDemand-(2* AddedTorqueAmpl);
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+(2* AddedTorqueAmpl)))
            {
            TorqueDemand= TorqueDemand+(2*AddedTorqueAmpl);
            }
         else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }

      // set the pitch angle to minimum (Ramp at 6deg/s)

      if ((avrSwap[3] - (0.1047 * avrSwap[2])) > avrSwap[5])
         {
         avrSwap[44] =  (avrSwap[3] - (0.1047 * avrSwap[2]));
         }
      else
         {
         avrSwap[44] = avrSwap[5];
         }
      }
   }
else
   {
   // ______ --- Non Valid Experimental Region --- ________

   ValidExp = 0;

   if ((avrSwap[19] < (0.8 * avrSwap[18])) && (avrSwap[19] > (1.1 *
avrSwap[16])))
      {
      StartFlag = 1;
      }
   else
      {
      StartFlag = 0;
      }
   Timedif = avrSwap[1];

   if ((avrSwap[19] > (avrSwap[18])) ||  (Hysteresis < 5))
      {
      //  ---- [[ above rated ]]   ----
      // ---------  Pitch Control ----------------
      if (avrSwap[19] > (0.7 * avrSwap[18]))
         {
         //set the pitch angle to .2 of the maximum
         // allowable (Ramp at 6deg/s)

         if ((avrSwap[3] + (0.1047 * avrSwap[2])) < (avrSwap[6]*0.2))
            {
            avrSwap[44] =  (avrSwap[3] + (0.1047 * avrSwap[2]));
            }
         else
            {
            avrSwap[44] = avrSwap[6]*0.2;
            }
         }
      else
         {
         if ((avrSwap[3] - (0.1047 * avrSwap[2])) > avrSwap[5])
            {
            avrSwap[44] =  (avrSwap[3] - (0.1047 * avrSwap[2]));
            }
         else
            {
            avrSwap[44] = avrSwap[5];
            }
         }

      if (HystFlag == 0)
         {
         HystFlag = 1;
         Hysteresis = 0;
         }

      if (avrSwap[19] < (0.9 * avrSwap[18]))
         Hysteresis = Hysteresis + avrSwap[2];

      //torque needs to be 1) equal to max demanded torque above
      //                      rated (under normal conditions)
      //                or 2) the demanded torque below rated (under
      //                      normal conditions)

      TorqueDemandNew = avrSwap[19] * avrSwap[19] * avrSwap[15]*1.1;

      if (TorqueDemandNew > (avrSwap[17] * avrSwap[17] *avrSwap[15]))
         TorqueDemandNew = avrSwap[17] * avrSwap[17] * avrSwap[15];

      //--------  ramp torque if needed ---------

      if (TorqueDemandNew<(TorqueDemand-10))
         {
         TorqueDemand= TorqueDemand-10;
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+10))
            {
            TorqueDemand= TorqueDemand+10;
            }
         else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }

      }
   else
      {
      //  --- [[ below rated ]]   ---
```

```
      // 1) set the pitch angle to minimum (0.001 below minimum is
      // used here for identification of non valid experiment
      // regions)(Ramp at 6deg/s)

      if ((avrSwap[3] - (0.1047 * avrSwap[2])) > (avrSwap[5] -
      0.001))
         {
         avrSwap[44] =  (avrSwap[3] - (0.1047 * avrSwap[2]));
         }
      else
         {
         avrSwap[44] = avrSwap[5] - 0.001;
         }


      // 2) set the Torque demand according to the optimal mode gain
      TorqueDemandNew = avrSwap[19] * avrSwap[19] * avrSwap[15];

      //--------  ramp torque if needed ---------
      if (TorqueDemandNew<(TorqueDemand-10))
         {
         TorqueDemand= TorqueDemand-10;
         }
      else
         {
         if (TorqueDemandNew>(TorqueDemand+10))
            {
            TorqueDemand= TorqueDemand+10;
            }
         else
            {
            TorqueDemand = TorqueDemandNew;
            }
         }

      // 3) update flags
      Hysteresis = Hysteresis + avrSwap[2];
      }

   }

strcpy(OutName, "validFlag:-;");

//Output new Generator Torque demand to wind turbine controller
avrSwap[46] = TorqueDemand;
return;

//Main DLL routine END

}
```

Stage 2, Linearization Procedure Code, scripted in MATLAB

A number of batch files have been created to handle the 990 experimental runs (99 experimental runs per each of the 10 wind turbine models). As these are not of any benefit to the reader, they have been excluded.

The following code presents a simplified version of the code used to collect the experimental data and system identify the drive-train model as described in Chapter 4 of the thesis.

```
% Stage 2, Identification Code [START]


clear all
% load gearbox ratio (this is used to calculate 1P from the generator
% speed
load('GBratio.mat');

% a loop is first run to import all the experimental data
% The loop is not shown here for clarity purposes.
% The data imported are the following:
% 1) Generator Rotor azimuth. Variable name: Temp_RotorAzimuth
% 2) Generator Torque Demand. Variable name: Temp_genTorque
% 3) Generator Speed. Variable name: Temp_genSpeed
% 4) Controller time step. Variable name: TimeStep
% 5) Pitch Angle. Variable name: PitchAngle

temp= size(PitchAngle);
DataSize = temp(1,1);

% discard non valid data
FinePitchAngle = PitchAngle(1);
for i=1:DataSize
    if or(PitchAngle(i)>FinePitchAngle,PitchAngle(i)<FinePitchAngle)
        Discard(i)=1;
    else
        Discard(i)=0;
    end
end

% Search within the 600 s experiment for the longest valid
% experimental period

EndData = 0;
i = 0;
Maxlength = 0;
dlength= 0;
tempStartData = 1;
for i=1:DataSize
    if Discard(i)== 0
        dlength= dlength +1;
        if dlength > Maxlength
            Maxlength = dlength;
            StartData = tempStartData;
            EndData = i;
        end
     else
        dlength = 0;
        tempStartData = i+1;
     end
end

% extract longest experimental period

for i= StartData : EndData
    genTorque(i- StartData+ 1 ) = Temp_genTorque(i);
    genSpeed(i- StartData+ 1 ) = Temp_genSpeed(i);
    RotorAzimuth(i- StartData+ 1 ) = Temp_RotorAzimuth(i);
end
genTorque = genTorque'; genSpeed = genSpeed';
RotorAzimuth = RotorAzimuth';
ExpTime = TimeStep*(EndData-StartData);

% create the Azimuth load indicator

for i= 1:EndData-StartData+1
    if RotorAzimuth(i)<=pi
       AzimuthLoading(i)=RotorAzimuth(i)/pi;
    else
       AzimuthLoading(i)=2-RotorAzimuth(i)/pi;
    end
end

% create the input signals for system identification

InputArray = [genTorque,AzimuthLoading'];



%-------------------
%- Identification -
%-------------------

% Identification is only run if the longest continuous valid
% experimental period is at least 150 seconds

if ExpTime>=150

% record the average 3P frequency. This is later used in the 3rd
% stage of the self tuning DTD procedure.

   ThreeP = 3*mean(genSpeed)/GbRatio;

% create an ‘iddata’ object from the input and output data.

   mydata = iddata(genSpeed,InputArray,TimeStep);

% import the experimental data

   mydatad = dtrend(mydata,0);                     %removes means
   mydatadd = dtrend(mydatad,1);                  %removes trends
```

```
% remove the effect of wind variation

   filter_below = 1.5;                % limit selected (in rad/s)

% converted to a in fraction of Nyquist freq:
   filter_below_N = filter_below / (pi/TimeStep);

% filter below selected frequency
   mydataddff = idfilt(mydatadd,5,filter_below_N,'high');


% crop data if needed (due to distortion from filtering)

   Temp_Size = size(mydatadd.y);
   NormMax = max(mydataddff.y(round(20/TimeStep):Temp_Size(1)));
   NormMin = min(mydataddff.y(round(20/TimeStep):Temp_Size(1)));

% check if cropping is needed and crop distorted data

   StartMax = max(mydataddff.y(1:round(20/TimeStep)));
   StartMin = min(mydataddff.y(1:round(20/TimeStep)));

   if or(StartMax>NormMax*1.2,StartMin<1.2*NormMin)

     for i=1:5:round(20/TimeStep)
        AveragedStart(i) = (mydataddff.y(i)+ mydataddff.y(i+1)
        +mydataddff.y(i+2)+mydataddff.y(i+3)+mydataddff.y(i+4))/5;
     end
     CropPoint = 1;
     for i=round(20/TimeStep)-5:-1:1
         if or(AveragedStart(i)>NormMax,AveragedStart(i)<NormMin)
             CropPoint = i;
             break;
         end
     end
     clear AveragedStart

% add 4 seconds to cropping point to ensure that all the initial
% resonance is discarded

     if and(CropPoint < round(16/TimeStep),CropPoint >
round(1/TimeStep))
         CropPoint = CropPoint + round(4/TimeStep);
     end

     mydataCropped = mydataddff([CropPoint:Temp_Size(1)]);

% create the state-space model with auto order (focus: stability)

     SSauto = n4sid(mydataCropped);
     clear CropPoint mydataCropped

% if cropping is not needed, no data is cropped

   else

% create the state-space model with auto order (focus: stability)
```

```
     SSauto = n4sid(mydataddff);

   end

% transforming the multi-chanel discrete SSauto into a continuous
% SISO TF:

   temp=tf(SSauto); temp=d2c(temp);
   [num,den]=tfdata(temp); sysdd=tf(num{1},den{1});

   clear mydata mydatad mydatadd mydataddff filter_below_N
   clear SSauto temp num den filter_below Temp_Size NormMax
   clear NormMin StartMax StartMin

% if the longest continuous valid experimental period is below 150
% seconds, the experiment is considered unsuccessful


else
sysdd='insuficient experimental length';
ThreeP=0;
end

% Stage 2, Identification Code [END]
```

Stage 3, DTD Tuning Code, Scripted in MATLAB

Again, a number of batch files have been created to handle the large number of identified models (99 experiments have been used for system identification, and thus a large number of identified models for each of the 10 wind turbine models have been created). As these are not of any benefit to the reader, they have been excluded.

The following code presents the code used to tune the single band pass filter acting as the DTD, as described in Chapter 4 of the thesis.

```
% Stage 3, DTD Tuning Code [START]

% initialise variables

lowestcostf1= 999999; lowestcostf2= 999999; lowestcostf3= 999999;
lowestcostf4= 999999;

for x=1:4
    ddK(x)=0;ddfreq(x)=0;ddzita(x)=0;
end

% sysdd is the system Identified model of the wind turbine drive-
% train. This was created by the code presented in the previous
% section of this Appendix.

% findpeaks is a custom built MATLAB function that finds the peaks of
% a system's Bode plot. See end of this section for the code.

[peakm,peakf,Npeaks] = findpeaks(sysdd);

SysPeakSum = sum(peakm);

% ddpeaks is a custom built MATLAB function that distinguishes and
% selects the peaks that are created by the drive-train resonant
% frequencies. It also creates the min and max frequency boundaries
% for the DTD filter. See end of this section for the code.

[ddpeakm,ddpeakf,ddNpeaksNoFeedback,minf,maxf] =
ddpeaks(peakm,peakf,Npeaks,ThreeP*1.1);


if minf == maxf
    minf = minf*0.9;
    maxf = maxf*1.1;
end

% running the procedure for all DTD filter parameters

for zita=0.29:0.01:0.45

   if and(zita>0.3,zita<0.44)
      continue    %exclude erroneous results
   end
   drawnow  %makes application of CTRL+C (stopping execution) easier
   for freq=minf:(maxf-minf)/80:maxf
      for K=0.5:100:7000.5

% set the filter
         num = K*conv([2*zita/freq 0],[0 1]);
         den = [1/freq^2 2*zita/freq 1];
         [a,b,c,d]=tf2ss(num,den);
         dd = ss(a,b,c,d);

% add the filter to the system

         sysddcl = feedback(sysdd,dd,1);
         [Wn,Z] = damp(sysddcl);
         Npoles= size(Z);


% create the cost functions


         medZ= 0; highZ = 0; vhighZ= 0; vlowZ=0;
         lowZ=0; costf3=0; costf4=0;


         for x=1:1:Npoles(1)
          if and(Wn(x)>(0.5*minf),Wn(x)<(1.5*maxf))

% create cost functions 3 and 4

            if and(Z(x)>0,Z(x)<0.2)
              costf3= costf3+ 3200*exp(-58.13*Z(x));
              costf4= costf4+ 3000*exp(-2*Z(x));
            end
            if and(Z(x)>=0.2,Z(x)<1)
              costf3= costf3+ 3200*exp(-58.13*0.2);
              costf4= costf4+ 3000*exp(-2*0.2);
            end

            if Z(x)<=0
              costf3=costf3+5000;
              costf4=costf4+5000;
            end


% categorise the poles with respect to their damping
% values in order to create the rest of the cost
% functions

            if Z(x)<.04
              vlowZ= vlowZ + 0.5;
            end
            if and(Z(x)>=.04,Z(x)<.07)
              lowZ= lowZ + 0.5;
            end
            if and(Z(x)>=.07,Z(x)<.10)
```

```
              medZ= medZ + 0.5;
            end
            if and(Z(x)>=.10,Z(x)<.15)
              highZ= highZ + 0.5;
            end
            if and(Z(x)>=.15,Z(x)<1)
              vhighZ= vhighZ + 0.5;
            end
          end
         end

% created sumZ variable in order to distinguish between similarly
% performing DTDs

         sumZ=sum(Z);

         costf1 =  .1* vhighZ + 1* highZ + 10*medZ +100*lowZ
         +1000*vlowZ - 0.001*sumZ;
         costf2 =  1* vhighZ + 1* highZ + 10*medZ +100*lowZ
         +1000*vlowZ - 0.001*sumZ;


         if costf1 < lowestcostf1
           lowestcostf1 = costf1;
           temp_ddK(1) = K; temp_ddfreq(1)=freq;
           temp_ddzita(1) = zita;
         end
         if costf2 < lowestcostf2
           lowestcostf2 = costf2;
           temp_ddK(2) = K; temp_ddfreq(2)=freq;
           temp_ddzita(2) = zita;
         end
         if costf3 < lowestcostf3
           lowestcostf3 = costf3;
           temp_ddK(3) = K; temp_ddfreq(3)=freq;
           temp_ddzita(3) = zita;
         end
         if costf4 < lowestcostf4
           lowestcostf4 = costf4;
           temp_ddK(4) = K; temp_ddfreq(4)=freq;
           temp_ddzita(4) = zita;
         end




      end
   end
end

% fine tuning of the filter parameters found according to the cost
% functions, for each of the cost functions

ddK = temp_ddK; ddfreq = temp_ddfreq; ddzita= temp_ddzita;

for i=1:7

    if temp_ddK(i)-100<0.5
        ddKlow=0.5;
    else
```

```
        ddKlow=temp_ddK(i);
    end

    if temp_ddzita(i)+0.01 >0.45
        ddZhigh=0.45;
    else
        ddZhigh=temp_ddzita(i)+0.01;
    end

% avoid second loop for badly identified systems

    if temp_ddzita(i)=0.45

% running the procedure for a range of DTD filter parameters, close
% to the filter parameters identified from the first step

    zita=temp_ddzita(i)
      for freq=temp_ddfreq(i)-(maxf-minf)/50:(maxf-
      minf)/250:temp_ddfreq(i)+(maxf-minf)/50
         for K=ddKlow:25:temp_ddK(i)+100

% set the filter

         num = K*conv([2*zita/freq 0],[0 1]);
         den = [1/freq^2 2*zita/freq 1];
         [a,b,c,d]=tf2ss(num,den);
         dd = ss(a,b,c,d);

% add the filter to the system

         sysddcl = feedback(sysdd,dd,1);
         [Wn,Z] = damp(sysddcl);
         Npoles= size(Z);

% create the cost functions

         medZ= 0; highZ= 0; vhighZ= 0; vlowZ= 0;
         lowZ= 0; costf3= 0; costf4= 0;

         for x=1:1:Npoles(1)

% create cost functions 3 and 4

            if and(Wn(x)>(0.5*minf),Wn(x)<(1.5*maxf))
               if and(Z(x)>0,Z(x)<0.2)
                costf3= costf3+ 3200*exp(-58.13*Z(x));
                costf4= costf4+ 3000*exp(-2*Z(x));
               end
               if and(Z(x)>=0.2,Z(x)<1)
                costf3= costf3+ 3200*exp(-58.13*0.2);
                costf4= costf4+ 3000*exp(-2*0.2);
               end
               if Z(x)<=0
                 costf3=costf3+5000;
                 costf4=costf4+5000;
               end

% categorise the poles with respect to their damping
% values in order to create the rest of the cost
% functions

               if Z(x)<.04
                  vlowZ= vlowZ + 0.5;
               end
               if and(Z(x)>=.04,Z(x)<.07)
                  lowZ= lowZ + 0.5;
               end
               if and(Z(x)>=.07,Z(x)<.10)
                  medZ= medZ + 0.5;
               end
               if and(Z(x)>=.10,Z(x)<.15)
                  highZ= highZ + 0.5;
               end
               if and(Z(x)>=.15,Z(x)<1)
                  vhighZ= vhighZ + 0.5;
               end
            end
         end

% created sumZ variable in order to distinguish between similarly
% performing DTDs

         sumZ=sum(Z);

         costf1 =  .1* vhighZ + 1* highZ + 10*medZ +100*lowZ
         +1000*vlowZ - 0.001*sumZ;
         costf2 =  1* vhighZ + 1* highZ + 10*medZ +100*lowZ
         +1000*vlowZ - 0.001*sumZ;

         if costf1 < lowestcostf1
            lowestcostf1 = costf1;
            ddK(1) = K; ddfreq(1)=freq; ddzita(1) = zita;
         end
         if costf2 < lowestcostf2
            lowestcostf2 = costf2;
            ddK(2) = K; ddfreq(2)=freq; ddzita(2) = zita;
         end
         if costf3 < lowestcostf3
            lowestcostf3 = costf3;
            ddK(3) = K; ddfreq(3)=freq; ddzita(3) = zita;
         end
         if costf4 < lowestcostf4
            lowestcostf4 = costf4;
            ddK(4) = K; ddfreq(4)=freq; ddzita(4) = zita;
         end

% ends for (k,freq)
         end
      end


% end for (avoiding second loop for badly identified systems)
end

% end for (i)
end

% create filters from filter parameters
```

```
dd1 = bandpass(ddK(1),ddfreq(1),ddzita(1),0,'Measured generator
speed','Generator torque demand');
dd2 = bandpass(ddK(2),ddfreq(2),ddzita(2),0,'Measured generator
speed','Generator torque demand');
dd3 = bandpass(ddK(3),ddfreq(2),ddzita(3),0,'Measured generator
speed','Generator torque demand');
dd4 = bandpass(ddK(4),ddfreq(4),ddzita(4),0,'Measured generator
speed','Generator torque demand');

%__________  Find Best DTD ___________

% create closed systems for comparison

sysddcl1 = feedback(sysdd,dd1,1);
sysddcl2 = feedback(sysdd,dd2,1);
sysddcl3 = feedback(sysdd,dd3,1);
sysddcl4 = feedback(sysdd,dd4,1);

% compare closed systems magnitude peaks and select best one
% Note: BestCostFunction variable identifies the best DTD tuned



w= ThreeP*1.1:(50-ThreeP*1.1)/5000:50;
tempsize=size(w);

[mag1,phase] = bode(sysddcl1,w);
[mag2,phase] = bode(sysddcl2,w);
[mag3,phase] = bode(sysddcl3,w);
[mag4,phase] = bode(sysddcl4,w);


for z=1:tempsize(2)
    magnitude1(z)=20*log10(mag1(:,:,z));
    magnitude2(z)=20*log10(mag2(:,:,z));
    magnitude3(z)=20*log10(mag3(:,:,z));
    magnitude4(z)=20*log10(mag4(:,:,z));
end



Npeaks1=0; Npeaks2=0; Npeaks3=0; Npeaks4=0;
nsamples=tempsize(2)-1;
SumPeakWeighted1=0; SumPeakWeighted2=0; SumPeakWeighted3=0;
SumPeakWeighted4=0;

for i=2:nsamples
if and(magnitude1(i)>magnitude1(i+1),magnitude1(i)>magnitude1(i-1))
    Npeaks1 = Npeaks1 +1;
SumPeakWeighted1=SumPeakWeighted1+magnitude1(i)*(1/w(i));
end
if and(magnitude2(i)>magnitude2(i+1),magnitude2(i)>magnitude2(i-1))
    Npeaks2 = Npeaks2 +1;
SumPeakWeighted2=SumPeakWeighted2+magnitude2(i)*(1/w(i));
end
if and(magnitude3(i)>magnitude3(i+1),magnitude3(i)>magnitude3(i-1))
    Npeaks3 = Npeaks3 +1;
SumPeakWeighted3=SumPeakWeighted3+magnitude3(i)*(1/w(i));
end
```

```
if and(magnitude4(i)>magnitude4(i+1),magnitude4(i)>magnitude4(i-1))
    Npeaks4 = Npeaks4 +1;
SumPeakWeighted4=SumPeakWeighted4+magnitude4(i)*(1/w(i));
end

end

BestCostFunction=1;
MinSumPeakWeighted=9999;

if and(SumPeakWeighted1<MinSumPeakWeighted,ddzita(1)>0.44)
    MinSumPeakWeighted=SumPeakWeighted1; BestCostFunction =1;
end
if and(SumPeakWeighted2<MinSumPeakWeighted,ddzita(2)>0.44)
    MinSumPeakWeighted=SumPeakWeighted2; BestCostFunction =2;
end
if and(SumPeakWeighted3<MinSumPeakWeighted,ddzita(3)>0.44)
    MinSumPeakWeighted=SumPeakWeighted3; BestCostFunction =3;
end
if and(SumPeakWeighted4<MinSumPeakWeighted,ddzita(4)>0.44)
    MinSumPeakWeighted=SumPeakWeighted4; BestCostFunction =4;
end

% Stage 3, DTD Tuning Code [END]
```

Custom functions used in "Stage 3, DTD Tuning Code" previously described are presented here:

Function: "findpeaks"

```
function [peakm,peakf,Npeaks] = findpeaks(siso);

% finds all peaks
% peakm variable holds the magnitude value of all peaks found in the
% SISO system
% peakf variable holds the frequency value of all peaks found in the
% SISO system
% Npeaks variable holds the number of peaks found in the SISO system


[magnitude,PHASE,freq] = bode(siso);

peak(1)=0;
Npeaks=0;
nsamples=size(freq);
nsamples=nsamples(1,1)-1;

for i=2:nsamples
```

```
if
and(magnitude(1,1,i)>magnitude(1,1,i+1),magnitude(1,1,i)>magnitude(1,
1,i-1))
    Npeaks = Npeaks +1;
    peakf(Npeaks)=freq(i);
    peakm(Npeaks)=magnitude(1,1,i);
end
end

if Npeaks == 0
    peakm = 99;
    peakf = 99;
end
```

Function: “ddpeaks”

```
function [ddpeakm,ddpeakf,ddNpeaks,minf,maxf] =
ddpeaks(peakm,peakf,Npeaks,minf);

% finds the peaks that interest us in drive-train damper design
% peakm, peakf and Npeaks variables are created by “findpeaks”
% function.
% minf is the lowest frequency limit above which this function tries
% to find drive-train resonance frequencies. In the final
% implementation of the “Stage 3, DTD Tuning Code” this frequency is
% set to 110% of the experimental 3P frequency. This ensures that the
% magnitude peak created at this frequency is discarded from the
% selection of drive-train natural frequencies


%find the peaks not needed for DTD design

Nclean=0;
highestm=0;

for i=1:Npeaks

%find the peaks below the minf frequency

    if peakf(i)< minf
        Nclean = Nclean +1;
        clean(Nclean)=i;
    else

%find the highest magnitude of any peak

        if peakm(i)>highestm
            highestm=peakm(i);
        end
    end
end

% find very low magnitude peaks. These are peaks that are not related
% to the system’s natural frequencies

for i=1:Npeaks
    if peakm(i)<(0.02*highestm);
% note: the 0.02 limit was set according to a large number of
% experiments, performed on all 10 WT models
        Nclean = Nclean +1;
        clean(Nclean)=i;
    end
end

% discard the peaks that are not related to the DT natural
% frequencies

Npeakpassed=0;

if Nclean>0
    for i=1:Npeaks
        peakpassed=1;
        for y=1:Nclean
            if i == clean(y)
                peakpassed =0;
            end
        end
        if peakpassed ==1
            Npeakpassed = Npeakpassed + 1;
            uddpeakm(Npeakpassed)=peakm(i);
            uddpeakf(Npeakpassed)=peakf(i);
        end
    end
    ddNpeaks=Npeakpassed;
else
    uddpeakm=peakm;
    uddpeakf=peakf;
    ddNpeaks=Npeaks;
end


minf= uddpeakf(1);
maxf= uddpeakf(ddNpeaks);

[ddpeakm,I]=sort(uddpeakm);
for i=1:ddNpeaks
    ddpeakf(i)=uddpeakf(I(i));
end
```