A load monitoring system for the 5kW wind turbine at the Renewable Energy Outdoor Test Area

ENG-470 Engineering Honors Thesis

2015

Thesis Report submitted to the school of Engineering and Information Technology, Murdoch University, as main part of ENG-470 Engineering Honours Thesis for majors in Industrial Computer Systems and Renewable Energy Engineering

Written By: Levi Pleiksna
Unit Co-ordinator: Prof. Parisa A. Bahri
Thesis Supervisors: Prof. J. Whale
Declaration

I declare that this thesis is my own work and have referenced any external material.

Word Count: 11815
Contents
Declaration ..........................................................................................................................i
Abstract ..........................................................................................................................v
Acknowledgements .........................................................................................................vi
Introduction .....................................................................................................................1
  Motivation .......................................................................................................................1
  Research Questions ........................................................................................................3
  Literature Review and Research ....................................................................................3
    Root Blade Load Estimation and other techniques .......................................................3
    Other Methods advantages and Disadvantages .........................................................3
    TUDelft University paper .............................................................................................3
    Strain measurements ...................................................................................................5
    Rainflow counting matrixes and load histograms .......................................................6
    Cyclic, Stochastic loading through Aerodynamic Loading ..........................................7
Methodology ..................................................................................................................9
  The Turbine and Location ..............................................................................................9
  Blade Strain Gauge Logging Transmitter Device Development ....................................9
    Sampling Rate .............................................................................................................9
The Device Schematic and Components ......................................................................10
  Freetronics Leostick ....................................................................................................10
  Buck Converter ..........................................................................................................11
  Schematic Transmitter System ....................................................................................11
  The building process ...................................................................................................11
  The program design .....................................................................................................13
  Battery Consumption ..................................................................................................13
  Schematic Receiver System .........................................................................................14
Strain Gauge Calibration ...............................................................................................14
  Method ..........................................................................................................................14
Datataker and LabVIEW Implementation ......................................................................15
  Datataker Device ..........................................................................................................15
  Labview Functionality ..................................................................................................15
9DOF Azimuthal Angle Tracking .................................................................................16
Abstract

It is necessary for small wind turbine development to conduct a “Root Blade Load Estimation” analysis which enables us to mathematically model the effects that turbulence has on fatigue of turbine blades and possibly in turn on the shaft it is coupled with.

Our results demonstrated that actual research into the effects of turbulence can be conducted on a small scale turbine using similar principles outlined in the methods of large scale testing. We were able to calibrate the strain gauges, overcome engineering issues through use of a transmitter receiver system and signal conditioning, minimize power consumption of this system, implement blade tracking as well as investigate the real time Labview interface.

Design considerations of the data logging are very important if the data captured is to be effective in research and development and this starts with the instrument and hardware design. For our project we looked at the TUDelft thesis Root Load Estimation Method and using the results we went back to the hardware to change some foundations and apply it to different scale turbines.
Acknowledgements

I would like to acknowledge my parents David and Nicolene and all the lecturers that pushed me through this Renewable Energy and Industrial Computer Systems Honors degree, as well as Dr. Jonathan Whale for being the mentor over the course of this thesis project.
Introduction

Motivation

Wind energy is the process by which the wind is harnessed to generate mechanical power or electricity. Wind turbines convert the kinetic energy in the wind into mechanical power. This mechanical power can be used for specific tasks, such as crushing or grinding grain or pumping ground water. More application is possible when a generator is installed to convert this mechanical power into electricity.

Wind energy turns propeller-style blades around a rotor which is connected to the main shaft. This shaft, in turn, spins a generator rotor to create electricity. To capture the most energy wind turbines are usually installed on tall towers. The turbine blades are often at heights of 20m or more depending on the power capacity.

In recent times smaller turbines of less than 20kW capacity have been built and installed in both developed urban areas and in more open sparse country areas. Smaller size turbines, while currently considered as less feasible than the large scale turbines, do have their benefits in that they have less effect on the grid, they do not need huge power storage systems and they are generally more environmentally friendly when referring to spatial land considerations of large scale turbines. (Pagnini n.d.).

Some of the places where small wind turbines may be installed at relatively low cost in urban regions include rooftops, within shopping centre grounds, and in car parks. (IEC 2013).

When a wind turbine is placed in an urban environment then because of surrounding objects, there is a generation of turbulence conditions. Turbulence is defined as the standard deviation of wind speed fluctuations. (Frandsen 2007). This is due to the fact that wind that is forced into an obstruction is then forced to change directions and acquires vorticity, which means it is rotating about a central axis of rotation. As a particle moves towards and away from that point of rotation it is becoming faster and slower respectively, hence generating turbulence. Due to the air-foil nature of the blades, which are made to produce lifting forces, these conditions generate a repeated cycle of loading on the blades, including in ways they were not designed for. (IEC 2013). Over time there is an increase in the blades’ inability to withstand load. This inability is known as fatigue damage and is derived from tiny cracks in the material.

There have been several reported cases of blades coming off these smaller turbines, which poses both safety and design issues. Every year around the world about 30 blade failures occur and this number is increasing with time. (Caithness Windfarm Information Forum 2015).

Although there is a reasonable amount of literature on large wind turbine blade loads, there is very little on small wind turbines, particularly in turbulent environments. (Whale 2015). In turn there is a problem with the design standard for smaller turbines being that it assumes that smaller turbines are generally situated in open fields. Since the turbines are designed for open fields, when they are sited in a turbulent flow environment, they may experience blade loads that they were not designed for. Hence the original wind design model is technically no longer valid or applicable under these actual other conditions. (IEC 2013).
The primary objective of this thesis paper is to undertake a preliminary exploration into the measurement methods that should be used to calibrate and log accurate and relevant quantitative measurement data for developing the Root Blade Load Estimation model of a smaller turbine. The focus in this paper is directed at the physical hardware design aspects of the recording device. It subsequently warrants an in-depth analysis into a prototype logging system which was developed and progressively assembled over the course of a 14 week Semester. The time spent in design considerations of this system is partially running in conjunction with overseas Root Blade Load Estimation research at Oldenburg University in Germany. This university was recently visited by Murdoch University lecturer Jonathan Whale to examine how the German project was conducted. In the Oldenburg project, turbulence conditions are being simulated by using a sophisticated wind tunnel. (Whale 2015)

One of the design issues that we need to know is at what azimuthal angle the blade is when it experiences the load. This will help produce an accurate model since down the line the data can be binned based on azimuthal angles. Binning is a data pre-processing technique used to control how different conditions should be considered exclusively of each-other. The idea of azimuthal tracking was originally, only meant to be a research project when we originally included the idea in the thesis proposal. However it was subsequently determined that implementing a measuring device as part of the prototype transmitter would address the blade azimuthal angle aspect and be a useful addition to the project.

Through the use of a technique called “Root Blade Load Estimation” we are able to mathematically model the effects that wind speed, wind direction and turbulence have on the load applied to the turbine blades and possibly on the shaft it is coupled with. This method requires measurement in two directions, edgewise and flap-wise, as shown below in figure 1.

Fig.1 Types of blade loads in Root Estimation

The system when ready will be installed on the Westwind 5kW turbine (Appendix 14) which is located in an outdoor test area at Murdoch University where turbulent conditions are prevalent. So far a company called “Pure Engineering” has installed the strain gauges on the blades so we can start calibrating these.
Research Questions
How is a Root Blade Load Estimation conducted?
What can we learn from other Root Blade Load Estimation projects?
How does implementing a blade monitoring system on a small turbine compare to a large turbine?
What should be the excitation voltage of a strain gauge?
What are the terms for types of loading experienced by blades subject to turbulence?

Literature Review and Research

Root Blade Load Estimation and other techniques
Since there is such a fast progression in technology for modern wind turbines, many of the initial studies mainly undertaken in the early 2000s, are now out of date. The rapid progress included changes in materials, components and technologies, which have corresponded to an increase in generation capacity. With an increase in generation capacity there has also been an increase in capacity in relation to physical size. Smaller wind turbines have become an economical means to generate power. Some of the early research needs to be updated for the new technologies, and one key area is through load modelling. The Root Blade Load estimation method is one of three methods that may help develop maintenance programs and load oriented operation on the blades of turbine fleets. The two other methods are “Regression techniques” and “Neural Networks”. (Algarin 2012).

For the root blade load estimation method, a load database is created through analysis and classifications of blade loads according to turbine status, mean wind speed and turbulence intensity. Once the load samples are reconstructed one can obtain Rain-flow Counting Matrices and load amplitude histograms and these are obtained for upwind and downwind directions and both edgewise and flap-wise moments.

Other Methods with their advantages and Disadvantages
Regression techniques use methods of correlation where the relationships between variables are proven to be statistically valid rather than having a physical model of the relationships. The accuracy of the regression function depends on the number of samples it takes to make the regression. This process applied to turbines has errors ranging from 2 to 23%.

Neural networks produce errors ranging from 12% to 22% depending on the number of nodes in the network (Algarin 2012). In this method mathematical operations between different variables are called nodes. One looks to develop a solution for relationships that are complicated without the development of a physical model but rather through making relationships between simple processes applied consecutively on each node. In one report equivalent loads, load amplitude and cumulative distributions were estimated. In order to find out more about neural network estimation please refer to the paper by (Cosack 2010).

TUDelft University paper
A clear explanation into the mathematical and statistical processes of the Root blade load estimation method is recorded in the 2012 Delft University thesis (Algarin 2012). The descriptions are
comprehensive for anyone looking into applying the Root Blade Estimation Method and gives scope for ways the process can be improved. The motivation for the paper was that by developing these models, more accurate project length analyses based on design parameters can bring the cost of wind turbine projects down and make it a more attractive way to meet loads on and off grid.

In projects analysed in the paper, the tests were done on two very large offshore turbines with 90m blades. They took six months’ worth of measurements at a sampling rate of 32Hz in 10min intervals. The data was sorted based on wind speed and turbulence intensity. The load measurements were then converted to the equivalent loads time series, Rainflow counting matrices then load histograms. Fatigue is the blades inability to withstand load over time due to tiny cracks in the material. The fatigue on the blades was calculated for the measurement database using the same material properties as the simulation.

The load estimation database results and distributions were compared to the simulation. This gave a means for validation of the model. The simulation was able to compare common operational cases which were the parked, normal shutdown, start up, power production modes and then split wind speeds into bins 2m/s apart.

Part of the damage assessment for the simulation was also used for the Database estimation. This was where they made general assumptions about the blades construction that it was a combination of fiberglass, reinforced epoxy and carbon fibres and obtained an S-N curve from a respected materials testing source. This curve looks at the stress load amplitudes vs. number of bending cycles to failure for the assumed material.

For the simulation, to transform the material properties from stresses to moments a known physical equation was used. This required an input of the geometric profile of the blades. The total stress at a given point on a blade could then be found.

From there a stress-moment conversion was developed and values of Ultimate Tensile Strength (UTS) and Ultimate Compressive Strength (UCS) were found. These values used in conjunction with the amplitude values of the bending moment cycle and the mean value of the bending moment cycle from the wind simulation then formed the equation to give the bending moments on the simulated blade in KNm.

Unfortunately, the paper did not go very much into the actual measurement installation or the SCADA system that logged it, however, comparisons of the post processed data and statistical models with the simulation it did give us an indication of what we might expect to occur in our model and how our logging system may be implemented. Some points to be considered are:

- A strong relationship was observed between load amplitude distributions and fatigue damage values. This included both edgewise and flap-wise directions. This damage increased linearly with turbulence for edgewise and exponentially for the flap-wise direction. This can be utilized in estimation of the maximum blade deflections and thrust force over the nacelle.
- A load behaviour effect due to gravity is observed between load amplitude distribution functions. This is observed when the wind speeds are close to cut in and idling conditions. For this reason we want to be able to compare these conditions with our data.
• The study had to apply a correction factor to the cumulative fatigue estimation generated from the database to bring it closer to the simulation calculations. This could be considered valid since they also applied this factor to another turbine with success.
• 20 minute intervals of turbine operation logging would reduce the probability of finding bins without data.
• To estimate long term loading it was proven that load characterization results for one turbine could be applied to another.
• It was proven that the same inflow condition generates similar load behaviour between turbines. At worst you can apply the load estimation to other turbines in the same row. At best these conditions could be applied to other turbines of the same model for simulation.
• Load safety factors and materials could be considered to account for uncertainties like manufacturing errors when it comes to determining fatigue damages from our measurements.

If we look at the graphs given in figure 2, which were taken from the TUDelft paper, we can note that it would appear that the flap-wise moment doesn’t correlate with wind speed, which means more complex turbulence or vortexes have a larger impact on the blades flap-wise moments rather than wind speed or blade rotation speed. We can also note the moment is in the order of 150 to 400 kNm for a slightly larger 14kW turbine.

![Image of graphs](images from Ladean R.)

**Fig.2 Delft Edgewise Root Bending Moments plots (appendix 13)**

**Strain measurements**

A strain gauge is a mechanical device that uses changes in the surface of a body at a position where it is useful to measure the strain on the object at that point and gives a measurement of the strain. The most common method has varying electrical conductance which changes when the surface shape changes. The quality of the strain gauge data will depend on the selection of the correct type of strain gauge but also that it is installed in a way with precision that it is only detecting the forces you are
interested in acquiring. It needs to be decided whether the gauge will measure pressure, yield to protect the surface, analyse cracks, analyse crack propagation, apply to internal surfaces in complex locations or if they need to be waterproof. In our instance we just needed a basic strain gauge in a configuration which doesn’t vary too much with temperature.

![Fig.3 Full Bridge Configuration](image)

The configuration on the blade is the full bridge configuration as shown in Fig3 above. This method uses four gauges in a Wheatstone bridge to make the effects of temperature on the output voltage lower. This is due to the fact that given all the materials are the same, the temperature will affect each strain gauge in the same way. The output voltage will not change much due to a change in temperature since the ratio of resistances is the same and so the temperature effects cancel out.

Once the strain gauge is installed, a design consideration is what excitation voltage should be applied to the strain gauge. Normally this is determined by experiment. The excitation voltage should be progressively raised with no load applied. Once the output voltage indicates instability through fluctuations around the zero mark you should return back to the region of stability. This can be an indicator of excitation voltage range. In our instance this will be less experimentation and more of decision based on the more important factor. Common practice suggests a voltage in the order of 5-15V is to be applied (National Instruments 2015). Having the lower bound of voltage will mean a trade off in resolution, particularly if the input to the logging device has a large offset. In our case, it is a requirement that we don’t use too much power since our system is supplied by a battery bank. We needed to strike a balance between these two factors and decided that 5V excitation voltage would do fine.

It is also common practice to use an operational amplifier, in case the microcontroller can’t detect small changes in lower order signals. During the project we will find out the calibration curve of the strain gauge and the voltage range of the output signal. It would be good to be able to see how temperature changes might affect the voltages as well.

**Rainflow counting matrixes and load histograms**

There are several different counting methods for cyclic loading however the rainflow method is considered the most accurate. As outlined in the TUDelft thesis (Delft, 26), it is necessary to use a method called Rain flow counting Matrices to analyse the number of cycles in each load range. In this method the strain gauge measurements are plotted against time. The process then can be likened to imagining the graph being turned it on its side so that there is water flowing down from each peak. This can be likened to gravity acting on the water and making it freefall straight down. If a flow is
interrupted by a free-fall stream it is then terminated. From here you can count the number of half cycles by counting the number of end points of the streams. The magnitude of each half cycle is the difference in kNm between the beginning of flow and termination. You do this for both the left and right sides of the plot as shown below. Cycles of equal magnitude but in the opposite side get paired up and are considered to be a whole cycle. From here you should have a table with the stress magnitudes and the number of half and full cycles that fall within those magnitudes.

**Fig. 4 Rainflow counting matrices illustration**

When this process is applied to a load regimen for example to a flap-wise load time series, a matrix plot or table can be obtained to give an idea of how the stress time series is distributed for flap wise loads.

Once you have the rain-flow matrices you can then progress to make load histograms. Depending on what conditions you wish to research you can bin data based on inflow conditions such as wind speed or turbulence intensity. The number of occurrences then gets plotted logarithmically against cycle load amplitude in kNm. These graphs can help identify the most common magnitudes of flapwise and edgewise moments by observing the peaks and can potentially help you design blades which withstand those amplitudes of bending moments.

**Cyclic, Stochastic loading through Aerodynamic Loading**

Cyclic loading and Stochastic loading are two different types of loading. Aerodynamic loading rather than being a type is actually a source. Cyclic occurs in two types: quasi static and dynamic cycling. Quasi static loads vary slowly so that the amount that the blade deflects is proportional to the load. For dynamic cycling the loading is based on the forces of the structure that are dampening the load and happens when the load application frequency is close to the natural vibration frequency of the structure. In Stochastic loading the load varies randomly and it happens predominantly due to wind turbulence although on average the load is fairly constant. These types of loads are all relevant to the blades fatigue response of the wind turbine (Ragheb 2008).
Aerodynamic loading is a source of loading. The turbine blades are actually designed for when the incoming loads on the blades are steady, continuous or of constant frequency and magnitude. It occurs when the wind’s kinetic energy is converted into useful mechanical work through aerodynamic effects. The most useful power is produced by forces in the rotor plane perpendicular to the oncoming stream. This can be determined by the equation of thrust. The aerodynamic loads that affect the fatigue of our blades can get broken down into categories cyclic and stochastic turbulence and cause structural failures.

**SCADA and a usual load database on larger turbines (Delft, 11 and 12):** Generally speaking only the prototype turbines have sensors to measure the loads on the wind turbine blades for the small systems. This is because for the smaller turbines the market is looking for lower cost solutions and there is much less risk when compared to a large scale turbine unless you do consider that these turbines are really more hazardous given they are around people and buildings.

If we consider an example of a full scale SCADA system we can observe that these turbines have 9 analogue measurement sensors and 5 status information sensors on the rotor alone (Kim 2014). This fact shows that the large scale turbine blades aren’t being monitored more heavily just because they are easy to implement but because there is more of a concern as to safety and efficiency because of the large investment in money and energy. Potentially larger turbine sensors could be installed by having electricians climb up the tower and go inside the nacelle which should make taking measurements such as rotational displacement easier given they can access the rotating shaft. That being said, a strain gauge measurement and anything put on the blades still needs to be decoupled from the nacelle since it is rotating. These large scale turbines have complex networks and topology’s including wireless systems to communicate between fleets of turbines as well as within the nacelle and are also often set up for this. Given that there is a place for small turbines in the world, smaller versions of these systems should be considered so that the efficiency and safety can improve and also the reputation of the small scale system.
Methodology

The Turbine and Location

The tower was installed by Westwind turbines in 2003 and was a new 5kW model at the time. The ROTA or Renewable Outdoor Test Area is an outdoor area located at the rural south side of the Murdoch University campus. This area has in place the required wind measurement systems given it has anemometer towers located very close by. There are also random surrounding objects such as buildings and bushes which generate the turbulence conditions required to cause both flap-wise and edgewise forces on the blades.

The project is a continuation of three shorter phases of Industrial Computer Systems Engineering project units where the students worked in small teams on the 5kW Turbine Monitoring system. The initial contributions to design were useful and helped to generate ideas and considerations for the thesis engineering project. The most important idea was the design and installation of the Li-Po battery powered transmitter and microcontroller placed inside the nose cone of the tower.

Daniel Jones from a small mechanical engineering company called Pure Engineering works on the turbines and tow them up and down when the University requires it. He was the person who installed the strain gauges at the root of the tower blades as well as on some other places on the tower. Daniel will help calibrate the Strain Gauges for my project.

Some key points of difference to that of previous Root Blade Load Estimation research is firstly that most of this system was built and customised with common low cost electronic components and applied to a smaller scale turbine. Secondly, there was a requirement for a low power consumption microcontroller process due to the turbine mechanics and thirdly that our system includes rotational real time tracking of the blades, which are logged and synced with the blade strain gauge values.

Blade Strain Gauge Logging Transmitter Device Development

It has been decided that a wireless system is required to send the strain gauge data from the spinning part of the turbine. This is because cables cannot run from a rotating part through to a stationary part without damaging anything. This piece of equipment needs to be specialized and designed specifically for the type of data it will be receiving. Previous ICSE students have done some work on building a prototype transmitter and receiver system, however it became clear I had to build my own Transmitter due to lack of documentation and too much troubleshooting without any resources. I made sure to document everything I did within this thesis. There were also new design aspects that needed to be considered.

Sampling Rate

The basic system comprises the components in the flow sheet diagram shown below. The strain gauge is outputting an analogue electrical signal. The Microcontroller then samples this signal and transmits it as an array of integers. The receiver then converts the integers into a PWM output which gets filtered by capacitors into an analogue waveform that’s approximately the same as the original value entering the microcontroller. Whether the Strain Gauge signal needs to be amplified or not will be determined once we have calibrated the strain gauges.
Since the system contains two devices that are sampling values, there needs to be sample rates selected for both. Normally as a general rule both devices would sample at the same rate since the resolution is only as good as the weakest link. In our case though, the transmitter while it might be able to sample the analogue inputs at a fairly slow rate, actually needs to send data at a much faster rate to increase the chances of the receiver picking up the signal. In our system, the Data-taker DT80 has a maximum sampling rate of 25Hz, since it uses physical relays.

Because part of our project involves getting the rotational locations of the strain gauge blades and the blades are spinning at a very fast rate, we want to set our sampling as fast as possible so that by the end of the logging we have a whole spectrum of different angles where the strain gauge measurement has been logged. If we have this full spectrum of angles the data can then be binned into loggings in around 30 degree intervals to find out which stages of the blades cycle it is under the most force.

![Diagram of system components](image)

**Fig. 5 The process of the Transmitter Section to Receiver Section, Left to Right**

**The Device Schematic and Components**

**Freetronics Leostick**

The base controller for the design of the transmitter section was the Freetronics Leostick which utilizes an ATmega32U4 microcontroller (appendix 1) and is similar to the Arduino Leonardo. It was important to consider the power drawn from the microcontroller and its connected devices given that the transmitter was running on a Li-Po battery. The attraction of this device was the fact it could be supplied from a 3.7V rather than 5V DC supply and the processor set up is designed to consume little power compared to other full-fledged instrumentation microcontrollers (appendix 6).

Given another requirement was to track the rotational location of the blade at any given time (J. Whale 2005) while logging in order to help produce a spatial model of the effects of turbulence on the blade, we chose the 9DOF tracking chip. This was made by Freetronics and integrated well with the microcontroller and the I2C interface.

A 5V excitation will provide 8.96 mV output signal at 1/1000 of the maximum strain resistance. The leostick analogue inputs are 10 bits or 0-1023. An acceptable external reference voltage AREF is at 5V which registers approx 4.88mV changes. This is a great resolution as it would register roughly 1 / 2000
changes in strain over the range of resistances. By default the analogue inputs measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the `analogReference()` function. (Appendix 9) This will not be necessary as we are satisfied with the resolution provided for 0 to 5V.

**Buck Converter**

A buck converter is required if we are to externally supply the Leo-stick through the terminals and the voltage is over 5V. Most non USB Li-Po batteries, if they have more than 1 cell, exceed 5V, so it is likely this will become an issue. A buck converter is a device which can step down the DC voltage with extremely high efficiency. For more information about how a buck converter works, see the Art of Electronics (Hill 2015).

**Schematic Transmitter System**

Below is the schematic for the version of the transmitter I built. It should be noted that the whole switching and buck converter setup is not necessary if power is supplied to the Leo-stick via USB. The screw terminal strip of 9 pins is actually representative of a terminal of 3 pins plus a terminal of 6 pins going from left to right.

![Schematic Transmitter System](image)

**Fig.6 The schematic for the transmitter board without any amplification**

**The building process**

To build the transmitter we used a copper board with conducting rows and separated columns. Every component and foundation in the build was designed to be easily replaceable to help with troubleshooting. The first part that was installed was the Freetronics shield sitting on headers with wires from the circuit poking through the holes in the copper board and forming connections within the headers. The Freetronics Leostick sat on the pins protruding from the top part of the headers through the shield and the Leostick could then be pulled off those pins. As seen in the transmitter images shown in figures 7 and 8, the terminal strips were linked to conductive lines on the shield and these were connected to the Leostick as shown in the transmitter schematic above in figure 6. Most the wire
connections were done at the bottom of the copper board. The transmitter was installed on headers as well so it could be changed over. For regulated supply of the Leo-stick the red connectors can be pulled apart and a buck convertor can be put in between them. Most of the things located on top of the shield such as the terminals actually have wires directly from the header pins and the ICSP interface, for the transmitter has wires running down the bottom of the copper board and coming up to connect to the very top of the Leo-stick. Apart from the removable components every wire was carefully soldered and copper strips if they were meant to be severed need to be tested for conductivity.

Fig. 7 Transmitter System
The program design
To program the Leostick we required libraries with call functions dedicated to the particular microcontroller and the physical pin outs and inbuilt registers. The Leostick requires a board profile download as well as a USB driver download to recognise it on any COM port (appendix 1). The NRF24L01 transmitter by Nordic semi-conductors has open source libraries with example programs (appendix 3). The 9DOF also has a library (Appendix 2). Once each component’s code is working which is the transmitter and 9DOF they can be combined. The transmitter program will be set up to read in the required number of analogue voltages which is 1 at this stage, and then put this into a 1D array and then also put the 9DOF values into the same array. The transmitter will have to work with the previous receiver code (appendix 16) that was written by Aharon Cunta, a Murdoch University Engineering Student who worked on this project previously (Aharon Cunta 2014). The program receives this array and breaks it down into individual values followed by outputting the integers as scaled PWM.

Battery Consumption
The Nordic transmitter and 9DOF will operate at 3.7V. The Leostick could potentially be run at 5V or 3.7V, so we will see what’s economical. The strain gauge will be excited at 5V and a super low power remote battery switch will be able to connect and disconnect the batteries from the devices hence switching the whole system on and off at any given time (appendix 4). The maximum current draw at full operation is approximately 46.5 mA, as shown in the table below. This is divided into the Strain Gauge circuit of 20mA draw and the Transmitter Circuit which will be around 26mA.

<table>
<thead>
<tr>
<th>Leostick Power 6 channels 3.7V supply</th>
<th>3.6V Nordic Transmitter typical transmission 10.5mA peak</th>
<th>Strain Gauge Power</th>
<th>Similar 9-DOF sensor 4.25mA (full power, gyro at all rates, accel at 1kHz sample rate, compass at 8Hz rate)</th>
<th>&quot;50uA&quot; Low Power RF Control Switch</th>
<th>Total mA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistance</td>
<td>11.72</td>
<td>10.5</td>
<td>20</td>
<td>4.25</td>
<td>0.0009</td>
</tr>
</tbody>
</table>

The current draw was calculated using a different brand 9DOF than the Freetronics brand since this information was not available. Since we do not currently know if the strain gauge requires an Op amp we have not considered this power consumption or battery configuration. There is however low quiescent current op amps available and hopefully this current draw can be kept to a minimum. We will later test the current consumption when the device is in operation by running the transmission program and opening the circuit for current measurement.

There are two batteries to be purchased. Two examples of batteries that could work are a ROMOSS 20000mAh 5V USB bank for the transmitter side and perhaps a 15000 mAh USB bank for the strain gauge. Allowing 85% discharge gives 17000mAh and 12750mAh per battery respectively.

\[
17,000 / (11.72+10.5+4.25) = 642 \text{ hrs.} = 26 \text{ days}
\]

\[
(12,750 / 20) = 637.5 \text{ hrs.} = 26 \text{ days}
\]
These calculations are for full rate transmission operation without any sleep modes. The ROMOSS battery banks are very compact and light hence will have negligible influence on the rotational moments within the nose cone although they should be mounted centrally. Compared to other packages of the same capacity, the ROMOSS banks have a great capacity to weight ratio. They also have built in discharge protection circuits, which means that once they get below a certain level they stop outputting which prevents damage to the batteries. But the sensing electronics on board these batteries is unknown and this could affect our decision to use these banks within the turbine. The sensing may make the device not be compatible for use as a DC supply. A modified USB cable could possibly be used to bypass any sensing circuitry.

**Schematic Receiver System**
The previous receiver system I was provided is working and the schematic provided in ENG454 phase 3 has been corrected to reflect the actual connections for the radio transmitter/receiver which is shown in figure 10.

![Receiver Schematic](image)

**Fig.10 Receiver Schematic**

**Strain Gauge Calibration**

**Method**
The strain gauges need to be calibrated so that flapwise downwind and upwind forces corresponded to voltages. To apply known forces both upwind and downwind and obtain accurate information, the forces need to be applied perpendicular to the gauge installation. Either a pulley system or vector calculations could be used to apply a known force “flapwise” to the blade. The easiest method was to position the blades and have them fixed so that the moments corresponded perfectly to the weight. If the hanging weights were to be applied at an angle to the blades, this would mean the angles would need to be measured precisely in order to apply vector calculations. One concern however because of the way the strain gauges were lodged at the base of the blade was that an applied force will register
an output voltage with a slight component attributed to edgewise moment. We needed to test the magnitude of edgewise forces that the strain gauge picked up by applying a large edgewise component. We considered the range of weights we may require given that for a large turbine in the TUDelft thesis the flapwise bending moment was roughly in the order of 150 to 400 kNm for a much larger 14kW turbine. Given our blade lengths of only 2m we assumed a maximum bending moment of 200kNm and even if we didn’t hit the upper limit in the calibration tests we could still extrapolate. A rough equation to find the total weight is:

\[
\frac{200 \text{ kNm}}{2 \text{ m}} = 100 \text{ kN},
\]

\[
100/9.8 = 10.2 \text{ kg}.
\]

Based on these calculations, the weights in steps of 50g we have available will do fine to get enough data for the calibration curve.

**Datataker and LabVIEW Implementation**

**Datataker Device**
The Datataker DT80 is a robust low power data logger that is a stand-alone solution for logging in remote environments. We are only concerned with the analogue inputs in which we will wire the receiver terminals to the positive and negative contacts on the channels. The DT80 has a maximum sample speed of 25Hz and 18 bit resolution for the analogue inputs (appendix 12). It also has serial sensor channels which could be utilized for sending commands to the receiver transmitter circuit. Digital inputs can also be used to trigger the DT80 to start logging. The DT80 has a plug and play web interface allowing the user to get logged data, set scheduling for logging or see the current measurements visually as “mimics” albeit with a slow refresh rate. Potentially once a wind anemometer is connected into a serial channel of the Datataker, a particular message can be set and used as a trigger in order to make the Datataker send digital output to the switching circuit and make the transmitter log when the Datataker wants data and turn off once it’s finished.

**Labview Functionality**
Labview is a graphical user interface that is very expansive in what it can potentially accomplish for any acquisition system. It has a lot of functionality to link with different technologies and build a customizable interface that can then be used to apply more complex methods of data manipulation. For example we could set it up so that the 9DOF tracking values can be converted into a 3D view of where the blade azimuthal angle and nacelle direction is in the 360 degree planes. You could also potentially set up simple buttons which in the background send serial messages to the receiver system and give it commands. You can set status indicators so that when serial messages come back from the receiver system it tells you if it’s working properly. If the Datataker supports real time streaming Labview is basically able to be customised into a mini SCADA system as on the larger turbine projects. The Data-taker device actually supports Labview in that they have provided a library of different Labview programs both Low and High Level. As part of this project I will investigate how Labview can be implemented as an interface for the DT80 since it is so versatile and expandable.
9DOF Azimuthal Angle Tracking

Design Problem
In order to form a detailed model of the root blade load estimation we need to organise our data into bins of the rotational location of the blade at any given point in time given the three sixty degree rotation of the blades. In order to find this we need this data but also it needs to sync with the strain gauge logging. There were several options that each had their own design issues.

Alternatives
One method I considered was setting up a video camera and syncing this with the data. I realised soon that the syncing of the data with the transmitter system would be very difficult since they could not be connected.

The other option was some sort of field sensor which utilizes electrodes to create an area that a microchip can actually read what is moving through the field as well as the specific coordinates. Unfortunately this would have suffered similar issues to the video camera and it was too difficult to mount an electrode close enough to the blades that it would have enough depth of field to register the blades.

Both these ideas would have failed due to the fact they could not be located inside the nose cone and been able to sync easily with the strain gauge logging data.

Implementation
The Freetronics 9DOF interface combines an accelerometer, a magnetometer, and a gyroscope into a small chip and integrates with an I2C interface. (Appendix 1) Some things this chip has been used for are building an autopilot for a quad-copter, or tracking the position of a robot, or logging data in a rocket. Combining the three axis data from each of the sensors allowed for a total of 9 degrees-of-freedom (DOF) of tracking. (Appendix 1)

It became clear that this was a perfect choice for our tracking system since we could actually put this device within the nose cone. The gyroscope would allow for the rotational tracking, the accelerometer may prove useful given if the magnitude of rotational acceleration can be identified from vibrations of the tower it could be used in conjunction with the accelerometer located in the nacelle that was installed by Pure Engineering. Given that a turbulence condition is classified as the standard deviation of wind speed, a turbulence condition might be able to be picked up through acceleration of the blades. It’s possible that if the microcontroller can identify this condition it might be able to use this as an interrupt, as discussed later.
Results

Strain Gauge Calibration

Daniel Jones from Pure Engineering came in to help calibrate the strain gauges. The blades were positioned so that they wouldn’t move through use of object anchoring on the non-measured blades. The root of measured blade was positioned parallel to the ground and the length of the blade was also levelled so that it was parallel with the ground. A spirit level was used to make sure the blade was in a position that correctly reflects the installation angle of the gauge and in turn the actual weights and forces that the root of the blade will experience during its cycling. Once the known weights were applied up and down at a span-wise location along the blade of two meters from the hub/blade root intersection we were then able to plot the voltage vs. weight graph for the flap-wise measurements. When we refer to upwind direction we refer to when the blade is bending backwards and conversely downwind referring to the wind causing the blade to bend forwards.

Fig.11 Pulley weight system to apply load perpendicular to root
For the edgewise calibration tests we applied a higher excitation voltage of 11.44V to the strain gauges since we weren’t expecting to have much output. With no force the gauge was registering -0.002V. A very large force applied edgewise output -0.004V. This is enough data to allow us to determine the impacts edgewise loads have on the strain gauge readings.

**Implementation into Datataker Device**

We were able to produce a linear regression that can be implemented into the Datataker device. Since the output of the strain gauge is a continuous piecewise linear function going from positive to negative, the high gain op amp feeding a positive signal to a transmitter channel and the split off into an inverting op amp at unity gain will allow the negative signal to be read positively in a separate channel. For the Datataker, we will read in these two signals into separate analogue input channels. The HTML interface allows the readings to be broken into two separate equations. The validity of the equation will be

\[ y = -0.0229x + 0.9518 \]

\[ y = 0.0293x + 0.9221 \]
determined by the voltage range being output. Appendix 18 demonstrates the actual Datataker HTML read in interface and how calibration scaling can be implemented as a polynomial.

**Signal Conditioning for Transmitter Strain Gauge Interface**

Given the output of the strain gauge in the order of approximately -3 to 6 mV, it has become clear that some signal conditioning is required in order for the Leostick to register these small changes in voltage. This will in turn bring the analogue input to a voltage range where the Leostick channel uses its full resolution. The Leostick is designed for a 1023 bit resolution over the range 0 to 5V. If we extrapolate the downwind curve to around a 120Nm moment this would give around a 4.5mV output. We need to allow for an extreme case though so that we don’t damage the Leo-stick’s analogue inputs. We will spread 8mV over range of 0 to 5V. $5/0.008$ gives a gain of 625.

Through the use of the design factors considered in The Art of Electronics (Hill 2015) we were able to select the OPA333 Op Amp which is a low offset voltage op amp. This essentially means that the zero value will be the offset voltage at the input and given the massive gain of around 600 will be around 600 times the offset voltage. On top of that, this op amp has a low quiescent current and an operating supply voltage of 1.8V to 5.5V. At a gain of 625 the resistor in series with the op amp is 16ohm and the resistor in parallel is 10k as shown in figure 14. To make the Op amp amplify the positive signal positively and negative signal negatively we require +ve as well as negative supply. Not all op amps do this however the OPA has a feature called “swing to supply” which allows it to amplify –ve and +ve signals and output them over the –ve to +ve range of supply. Unfortunately we can’t use the ROMOSS banks since they are both supplying separate circuits. We will need two low capacity 5V batteries as shown in the strain gauge circuit configuration schematic in figure 15 and have them connected in series with a grounding in the middle.

![Fig.14 OPA333 Configuration from spec sheet appendix(10)](image)

Once the OPA333 chip arrives it will then need to be calibrated by looking at the linear response of the input signal to the output signal. This calibration should be done with the actual resistors being used in the circuit.
Since the Freetronics Leostick only reads positive signals, it is necessary to split the signal off into another channel going through an inverting op amp with no gain. Since there is no gain the op amp can be generic. What will happen is that when the wind is downwind and the blade is bending forwards, the signal will be positive and it will be amplified positively into channel A2 as in figure 13. The A1 channel won’t register since it will be inverted into a negative signal. The opposite will happen for the other channel. This method only has one channel registering at a time depending on upwind or downwind condition. This is effective for our output as it is essentially making a piecewise function. Another OPA333 could be selected for this purpose.

**Strain Gauge Circuit Configuration**

The strain gauge circuit configuration can be observed below in figure 15. I have had issues using my own Romoss banks to power the Leo-stick in that it powers for about 20 seconds and seems to realise that it isn’t a battery capacity. If the ROMOSS bank sensing electronics can’t be removed there will have to be another high capacity bank installed. In the image below we can see the configuration of the strain gauge and the colours of the wires. The output curves in the results will help give data about what’s actually being read.

**Transmission Tests**

**Accuracy**

Using Maniacbugs NRF library I was able to make sure the transmitter was working properly without the need for inputting voltages. This first program allowed each node to be set up either as the transmitter or receiver by sending in the T or R command demonstrating how the Leostick can potentially be given commands and ping back to the user via serial. As you can see below, with the
transmitter serial port on the left and the receiver serial port on the right they are talking to each other in that the receiver is identifying the payload correctly. The roles could also be switched around.

Fig. 16 Maniacbug NRF library test program through serial

The actual transmitter program that I wrote using the Maniacbug Library is shown below. This linked in with the receiver program written by Aharon (Aharon Cunta 2014) (Appendix 16). My transmitter program (appendix 15) was written so that it sends across an array of Long Long integer characters in a 7*1 matrix and the receiver program breaks this down into the individual values and outputs them as PWM signals through the analogue outputs. In the image below the variable names are different but they are the same values. Actually I did need to modify the previous receiver code that I was provided so that it refreshed the array every processing cycle. This way it is always getting the most current data from the transmitter.

Fig. 17 Transmission results from Transmitter and Receiver through serial

Range and Interference
We tested the transmitter and receiver circuit in an open outdoor area with similar terrain and conditions to the ROTA. We did this using the booster antenna on the transmitter and it was found that when the two units were in an open area with the antenna in line of sight the transmission worked up to 27m. When the transmitter was inside the nose cone without the antenna in line of sight it only transmitted 9m. For this reason it is necessary for the transmitter antenna to pop out the centre of the nose cone and that a hole is drilled at this point. This way the transmitter will be in line of sight of the receiver at most times. This has not been done yet and will have to be done carefully with a metal drill bit that is a little bit larger than the radius of the antenna. Since for the turbulence condition we are most interested in logging data when the wind is coming from the east side where there are more objects creating turbulence; it doesn’t matter if the transmission drops out in the opposite direction perhaps due to the tower being in the way or losing line of sight. I would however recommend moving the receiver up the tower as far as the USB cable can go if it is going to be a permanent setup mounted to the tower. Otherwise it wouldn’t be too difficult to build another intermediate location transmitter receiver to boost the signal into a wider area. For now having the antenna pointing out the nose and maybe putting a laptop with the USB powered receiver a bit towards the nearby shed on the east side will work. It is still uncertain how the turbine electronics such as the generator can affect the transmitter signal.

**Transmitter Power Consumption**

**Consumption Breakdown**

When the multimeter was connected in series with the transmitter and the transmission program was running, the current draw was 33mA. This was with the Leostick supplied with 5V and the 9DOF program not running and hence drawing no power. We can probably expect that with the 9DOF program running the current draw will be in the range of 36-40mA. While running the Leostick at 3.7V could potentially save some power, given that we have acceptable consumption while at 5V, we would rather run it at 5V so that the transmitter pipes are switched at a maximum rate, thereby increasing the quality of received data. This will also improve the range as the probability of the receiver getting the signal is increased. I tested the transmitter at a low speed and what can potentially happen if the transmitter is cycling too low is that the receiver has periods where it just doesn’t detect the signal, which is terrible for PWM measurements since there are zero values in between the on values. There is no need to test power consumption of the strain gauge because it is just a known resistance input.

**Supply Configuration**

For our supply design it will be the same as the signal conditioning circuit as shown in figure 15, however the voltage sources will need to be connected to the Abacus 90micro amp RF controlled switches in the configuration below. This will allow the circuit to be completely switched off and on and save power so that the turbine can stay up for a long period of time before the transmitter batteries run out.

It should be noted that certain high capacity consumer electronics battery banks that are generally used for charging mobile devices such as cell phones, may not allow actual supply of devices that don’t contain battery capacity. Unfortunately the Leostick doesn’t have a battery capacity and may be the reason for what we discovered when using the ROMOSS banks. The bank would power the device for a few seconds and then turn off completely in what appeared to be the bank electronics realising there is
no capacity. This feature for the ROMOSS branded battery bank can probably be bypassed with usb or circuit modification. It is better though if a different portable high capacity battery bank can be used as a DC supply instead. Something that will also need to be considered is the discharge protection. Most commercial USB banks have this feature built in, however in the case of using hobby li-poly batteries most don’t have this automatic protection circuit. The protection can however be implemented with an external protection circuit board. This type of device will prevent over charging, over discharging and limit output current when connected to the battery terminals and will generally be selected based on the voltage output and the number of cells.

Fig. 18 RF Switching of main high capacity supplies

9DOF Tracking

Installation
The Freetronics chip was mounted on the edge of the transmitter as shown in Fig.19. This is a temporary location for the chip.

Fig.19 Pointing out current 9DOF placement

The pose program of the 9DOF library is a combination of the gyroscope, magnetometer, and accelerometer instruments and programs and it gives the rotational tracking in three variables roll, pitch and yaw. This program can be obtained from the source at appendix 2 which utilizes a library specifically for this chip. The pose program is basically a fusion of all these other instruments. These
three types of rotation can be visualised by an aircraft. An aircraft which rolls is activating the ailerons on the wings to rotate around the axis the aircraft is moving in. The yaw movement is as if the aircraft rudder is being controlled and the aircraft is rotating around an axis from the top and through the fuselage. The pitch axis is as if the elevators on the tail end are moving up and down and the nose of the aircraft is increasing or decreasing the incline causing the plane to ascend or descend.

Fig20. Illustration of 9DOF tracking in terms of aeroplane motion

These three variables can be implemented inside the nose cone mount as shown later in Fig24. By setting the 9DOF tangentially to the axis of rotation via a ribbon connected to the I2C interface. Further testing will determine how the values can correspond to the rotation and how the data can be calibrated. Figures 21, 22 illustrate the integer values being outputted when the 9DOF is working. The serial output can be restricted to showing some or all of Gyro, Accel, Mag and Pose matrices by commenting or uncommenting the values with “//”. Figure 23 shows the type of output when the device was subjected to particular types of motion.

Values

<table>
<thead>
<tr>
<th>Sample rate</th>
<th>Gyro bias valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pose roll 4.47 pitch 9.36 yaw 139.71</td>
<td></td>
</tr>
<tr>
<td>Pos  roll 4.47 pitch 9.36 yaw 139.69</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.47 pitch 9.35 yaw 139.56</td>
<td></td>
</tr>
<tr>
<td>Sample rate 50, gyro bias valid</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.47 pitch 9.35 yaw 139.56</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.47 pitch 9.36 yaw 139.45</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.47 pitch 9.36 yaw 139.45</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.48 pitch 9.30 yaw 139.41</td>
<td></td>
</tr>
<tr>
<td>Sample rate 50, gyro bias valid</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.48 pitch 9.36 yaw 139.56</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.48 pitch 9.37 yaw 139.63</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.48 pitch 9.36 yaw 140.07</td>
<td></td>
</tr>
<tr>
<td>Sample rate 50, gyro bias valid</td>
<td></td>
</tr>
<tr>
<td>Pose roll 4.48 pitch 9.36 yaw 140.23</td>
<td></td>
</tr>
</tbody>
</table>

Fig21. 9DOF Pose Program fusion values
Fig 22. 9DOF Gyroscope, Accelerometer, Magnetometer and Pose fusion values

The 9DOF could be set up for more precise calibration via a ribbon extension where it can be positioned in an upright position off set from the central axis but rotating tangentially as shown in fig 24. The device will need to be pre calibrated perhaps on spinning wheel that’s mounted to the nose cone then re-calibrated once on the tower with the proper orientation and electrical and radio interference that is surrounding it.
Fig 24. Recommended positioning of 9DOF in nose cone

The 9DOF undertaking was successful given as we can already see how the location is represented by integers and in turn output voltages. The pose program will work fine if the 9DOF is positioned as recommended in fig. 24. The roll values shown in fig. 23 will indicate where the strain gauge blade is rolling at any given time via the voltage received by the transmitter and in turn receiver and datataker. The values that are currently being received by the microcontroller go from negative values to positive values. Due to the nature of transmission and corresponding PWM output which requires positive output it may be better to convert that scale to purely positive values by shifting the boundaries of the integer scale.

Having the 9DOF in the suggested configuration will also allow for the yaw values which will correspond directly to the yaw of the wind turbine Nacelle. This in turn gives us rough information about the wind direction which is influencing the direction of the nacelle due to the tail on the turbine. It is designed to make the nose cone face the direction of the prevailing head wind.

I recommend putting the 9DOF on a ribbon and positioning it like in fig. 24 onto a bicycle wheel and plotting the roll values when the wheel is spinning at a variety of speeds to see if this has any influence on the output. It shouldn’t, but at least doing this will simulate what the tracking device will be doing inside the turbine cone. Doing pre calibration and positioning the 9DOF in the same way it will be in the turbine will make calibration easier once it is inside turbine.

Once the 9DOF is sitting on the tower, it can no longer be calibrated using the structured vector serial integer values that the Pose program outputs. The transmit array values will only be seen once they are broken up into channels at the receiver end. Providing the receiver values are printed on serial correctly though, the values will be the same and you can access the receiver via USB through the Arduino software. Like the strain gauge voltages, the integers at the receiver will have to be scaled to be represented by the PWM voltage output. At the Datataker however you will be monitoring this representative PWM that’s become a continuous waveform through capacitor charge and discharge. Mimic plots can be used in the Datataker software or the stored logging table data can be used to tell the positions and values of the strain gauge.
**LabVIEW Tests**

**Logging the analogue signal real time with Datataker**
Using the High level Lab-view program and logging into the IP address signed to the Data-taker we can use Lab-view to take data and also directly apply calibration factors to all the channels of the Data-taker. This is demonstrated in appendix 19 where several channels can be run at once. In this image we are only sending one voltage, meaning the other channels are at zero. The user will have to set the sample rate including the trigger time and the total number of samples. This should be limited to the Datatakers maximum sampling rate of 25 Hz. The HV channel type will scale the input to the order of V or the “V” channel type to mV.

**Transmission Program**
My Transmission program utilized Maniac Bugs NRF24L01 radio library. It consisted of creating a 7 value array which are all currently configured as read in analogue voltages and represented as integers. There are certain function calls that the radio requires. The main aspect is assigning addresses or “pipes”, one for the transmitter to receive and the other to transmit. The pipe then needs to be opened and a function needs to be called to write the array, called “analoghold” in this instance to the write pipe. The receiver needs to have this same pipe address open in order to read it in, and using an array sizing function can allow it to break down the array. Below is the Maniacbug Library Convention to Assign Pipe Addresses.

```c
const uint64_t pipes[2] = { 0x00F0F0F02122, 0x00F0F0F0112111L };
```

**Fig25. C RF transmitter library initializing read and write pipes (appendix14)**

The void loop is a continuous executing loop which has been set to keep opening the pipes and writing the array every single cycle without any delays, which means it is happening at the maximum sampling speed. This is discussed in the transmission section.

```c
void loop(void)
{
    analoghold[0] = analogRead(analoginone);
    analoghold[1] = analogRead(analogintwo);
    analoghold[2] = analogRead(analoginthree);
    analoghold[3] = analogRead(analoginfour);
    analoghold[4] = analogRead(analoginfive);
    analoghold[5] = analogRead(analoginsix);
    analoghold[6] = analogRead(analoginseven);

    radio.openWritingPipe(pipes[0]);
    radio.openReadingPipe(1,pipes[1]);
    radio.write(analoghold , sizeof(analoghold) );
}
```

**Fig26. C RF transmitter library sampling loop and writing to pipes every loop (appendix14)**

It is possible that the transmitter could be set to sleep normally but when a sensor condition is met either from the Leostick reading an acceleration value from the 9DOF or the receiver sending it a signal
to wake up it starts the logging process. This would allow the device to capture wind turbulence periods when nobody is around and actually gather more useful data.

Initially I was just exploring the option of Labview because of the potential benefits of an easy to use and highly customisable interface. I wanted to see whether it would interface well with the Datataker and it appeared to have a lot of potential given that Datataker provided Labview support.

Unfortunately while the data-taker provides real time logging, the data-taker device does not provide true real time streaming, as the data is only displayed once logging is complete, which would defeat the purpose of having a lab-view interface. The data-taker also tends to get stuck into its process and lab-view has trouble telling the data-taker to stop doing what it is currently doing.

For these reasons I would recommend just using the inbuilt html browser for the Datataker to set up scheduling and analogue logging.

The inbuilt browser also provides scaling, however I think it is better to grab all the values at 25 Hz from the logging, put them into a spread sheet and do post processing of the voltage scaling. If scheduled control of the transmitter battery banks was wanted then the Datataker could send a digital signal to the switching circuit to connect the batteries and turn the whole transmitter logging system on.

If real time streaming was required, for example to track the blades live, which would help calibrate the 9DOF once it’s in the tower, then a different logging device such as a National Instruments USB DAQ could be used. This would likely make calibrating the 9DOF once it’s in the nose cone sitting up on the turbine easier.
Conclusion

Our results demonstrated that actual research into the effects of turbulence can be conducted in a small scale on a small scale turbine using similar principles outlined in the methods of large scale testing. Design considerations of the data logging are very important if the data captured is to be effective in research and development and this starts with the instrumentation and hardware design. The TUdelft thesis laid a foundation for how data should be processed once it’s there and using their results we could do an iteration and go back to change some foundations to enhance this method but apply it to a different scale which is just as important for progress in Wind Turbine Technology.

While my thesis paper focused on the engineering considerations in the implementation of hardware rather than the post processing and research it did delve into an effective method of tracking the blades at any given time for a smaller wind turbine. This method will affect the post processing of data drastically in that it will allow for binning values based on rotational regions of the blades and then Load histograms can be generated for each bin. The TUDelft thesis database actually only binned the moment data through more general values such as the wind speed and Turbulence intensity. Our method and design for this tracking is very different to what would be implemented for a larger turbine since for a larger turbine the shaft can be accessed on the nacelle side of the tower which opens up many more possibilities due to relative motion. We needed to find an effective method of tracking using the resources we had. The 9DOF tracker was installed on the transmitter and demonstrated that it can be implemented effectively as part of a well thought out portable system.

We did observe some limitations of our system for the purpose of real time streaming and monitoring. The Datataker is an exceptional bit of hardware for the purpose of leaving the device in a remote area to grab the data when it needs to and store it. It does however suffer from limited sampling rate and the internals don’t allow us to look at what’s going on with the turbine while it’s actually happening in a smooth manner. For this reason it could not be a replacement device for the safety and alarms monitoring of a large turbine SCADA system with regards to the smaller system.

Our actual results were limited to the design and construction and testing of the transmitter tracking system. Our results would have been much more interesting if we had the system installed on the West-wind turbine and could actually see the Root Blade Load Estimation data. Given a longer thesis period this may have been possible however the foundations have all been laid out for a successful system and a detailed model.

The objectives of the project were achieved since we were able to design a system to implement Root Blade Load estimation on a small turbine. It is a step in the right direction for successful analysis which will in the future generate a model using Rainflow counting and Histograms with detailed binning of load data.

As far as the project goes we were able to calibrate the strain gauges, overcome engineering issues through use of a transmitter receiver system and signal conditioning, minimize power consumption of this system, implement blade tracking as well as investigate the real time Labview interface.
Recommendations
For anyone wanting to design, build and install a transmission system or one similar to the one covered in this paper we can make some recommendations based on our results:

1. Close attention needs to be paid to the signal and power requirements. Try to obtain a long range signal with low power consumption requirements. To install the transmitter inside the nose cone the antenna should be put so it is hanging out the front to give field of view to the receiver.
2. To purchase battery banks make sure it is high capacity but also make sure it doesn’t have sensing elements to let it know if it is charging a capacity or not. Check Supply configuration subheading under Transmitter Power Consumption.
3. As outlined in the Delft paper, 20min logging intervals would be beneficial. This will however mean double the power consumption of the transmitter which is why it needs to be carefully considered. See consumption breakdown and supply configuration subheadings for more information.
4. To calibrate the 9DOF try the outputs of other types of motion and with the chip in a different orientation. It should become clear what orientation to use by the signal smoothly increasing and decreasing when the chip is rolling as with the motion of an actual turbine. Shift the upper and lower boundaries of this data numerically so the lowest point is at zero and all the numbers are positive.
5. For the transmitter receiver to detect values successfully it is more successful to open and close the writing and receiving pipes every cycle so the data gets refreshed at a high speed.
6. If you want real time streaming monitoring do not use a portable logging system but something like a National Instruments USB DAQ. In remote areas however the Datataker is suitable as it can be left running and is robust. Cost wise the Datataker is a more expensive hardware than the NI USB DAQs since it works as a standalone system, whereas the NI devices need a PC in the vicinity.
7. Pay close attention to the signal conditioning section of this thesis and look at schematic fig.15. This is because the output voltages of the Strain Gauges are very small and most microcontrollers unless specially designed for instrumentation only measure with around 5mV resolutions. Also the section demonstrates a way to split positive and negative voltages so that the Leostick can understand them.

Future Works
Points for people continuing this project:

- It is better to use a USB cable to power the Leostick from the battery bank and it is safer. The terminals were previously configured to link in the original 3 cell hobby Li-Po, however that battery wasn’t suitable due to capacity and its physical size so I ended up altering the setup including removing the buck converter section and just using USB instead. I was previously hoping to use the ROMOSS bank for USB supply however this didn’t work.
  If you need to supply the device with a DC supply or hobby LiPo without USB connection, the terminals need to be rewired back to being as in figure 6 (by supplying across Vin and Ground) and connecting the buck converter properly. Test it using a DC supply with <5V for shorts before connecting a non USB Li-Poly since they can be dangerous when shorted.
Combine the 9DOF Pose and Transmission Program (appendix 2 and 16). You must use the USB to do this. The Arduino software doesn’t work properly for these models on Murdoch university computers so use a separate PC for this with the latest Arduino software. Check the “program design” subheading for the required drivers.

Wire up the op amps as in fig.15 and 14. Then plot the output response for voltages ranges similar to figures 12 and 13 in the paper. The 2\textsuperscript{nd} op amp can be the same as the OPA333 however a cheaper one may be used with low noise since it is only inverting an already amplified signal at gain of 1.

Purchase:
1. Two roughly 20,000mAh batteries which are required to supply the transmitter and strain gauge at 5V separately. When I tested the ROMOSS bank it would not supply properly as mentioned in the “Supply configuration” subheading within the paper. You could consider testing it with a brand new Leostick to make certain it’s not just a faulty microcontroller however I believe it’s to do with a sensor inside the ROMOSS circuit checking for feedback from an actual battery. If it becomes too difficult to find a suitable high capacity 5V supply with USB then there are also hobby Li-Poly batteries which are generally of lower capacity but higher voltage and are of larger physical size and weight. Several of these will be required and need to be wired in parallel. A buck converter will be required too as in figure 6. A protection circuit board and battery charger for the corresponding number of cells and voltage of the chosen Li poly battery is required. Fig.15, fig 18
   Do not connect a non USB Li Poly to the terminals in the current configuration. Re wire it to being as in figure 6 with the Buck converter and test for shorts first. You can use fig.6 as a guide to trace the wires on which the terminals are viewed from left to right.
2. Two small low capacity 5V batteries to supply op amps Fig15
3. Two low power RF switches as in appendix 4. Fig18

Do an external calibration of the 9DOF as described in the discussion.

Install and Test system with Daniel Jones from pure engineering.
References


Pagnini, Luisa C. Experimental power curve of small-size wind turbines in turbulent. Report, Montallegro 1, 16145 Genova, Italy: Applied Energy and Department of Civil, Chemical and Environmental Engineering, Polytechnic School, University of Genova, n.d.


Whale, Jonathan, interview by Levi Pleiksna. Dr (October 27, 2015).

Appendix


(2) Libraries for the 9DOF [online] https://github.com/richards-tech/RTIMULib-Arduino

(3) Libraries for Nordic Transmitter [online] https://github.com/niacinbug/RF24


(9) https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage

(10) http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=opa333&fileType=pdf

(11) 1.8-V, micropower, CMOS Op Amps, Zero Drift Series OPA333

| Low offset voltage: 10 microvolts max |
| Zero drift: 0.05 microvolts / Degree Celsius |
| Low noise |
| Quiescent Current: 17 micro amps |
| Single Supply operation |
| Supply Voltage: 1.8V to 5.5V |
| Rail to Rail Input/Output |

(12) Datataker DT80 analogue channels

| 5 analog input channels (expandable to 100*) Each channel is independent and supports: one isolated 3-wire or 4-wire input, or two isolated 2-wire inputs, or three common referenced 2-wire inputs. |
| The fundamental inputs that the DT80 can measure are voltage, current, resistance and frequency. All other measurements are derived from these. |

| Sampling Integrates over 50/60Hz line period for accuracy and noise rejection Maximum sample speed: 25Hz Effective resolution: 18 bits |
| Inputs Inter-Channel Isolation: 100V (relay switching) Analog Section Isolation: 100V (opto-isolated) Input impedance: 100KΩ, >100MΩ Common mode range: ±3.5V or ±35V on 30V range |
| Sensor Excitation (Supply) Analog channels: selectable 250μA or 2.5mA precision current source, |
4.5V voltage source, or switched external supply General Purpose: Switchable 12V regulated supply for powering sensors & accessories (max 150mA) Switchable 5V regulated supply for powering analog sensors (max 25mA)

Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start-up wind speed</td>
<td>2m/s</td>
</tr>
<tr>
<td>Cut in wind speed</td>
<td>3m/s</td>
</tr>
<tr>
<td>Rated wind speed</td>
<td>14m/s</td>
</tr>
<tr>
<td>Cut out wind speed</td>
<td>none</td>
</tr>
<tr>
<td>Furling Wind Speed</td>
<td>16m/s</td>
</tr>
<tr>
<td>Rated Power</td>
<td>5.5kW</td>
</tr>
<tr>
<td>Rotor Speed</td>
<td>120-750</td>
</tr>
<tr>
<td>Rotor Diameter</td>
<td>5.1m</td>
</tr>
<tr>
<td>Weight</td>
<td>200kg</td>
</tr>
<tr>
<td>Overspeed Protection</td>
<td>Auto tail furl</td>
</tr>
<tr>
<td>Blade Material</td>
<td>Pultruded fiberglass</td>
</tr>
<tr>
<td>Winding Type</td>
<td>3 star connected</td>
</tr>
<tr>
<td>Winding</td>
<td>27 turns per coil</td>
</tr>
<tr>
<td>DC Voltage</td>
<td>120V</td>
</tr>
<tr>
<td>Magnetic Material</td>
<td>Ne-Fe-B</td>
</tr>
</tbody>
</table>

//Transmitter Code Written by Levi Pleiksna using Maniacbugs Library

```c
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include "printf.h"
#define CE_PIN 5
#define CSN_PIN 8
RF24 radio(CE_PIN, CSN_PIN);

// Radio pipe addresses for the 2 nodes to communicate.
const uint64_t pipes[2] = { 0xF0F0F0F0D2LL, 0xF0F0F0F0E1LL };```

(13) TU delft paper graph pg19
(14) Murdoch Turbine Facility Handbook

(15)

//Radio pipe addresses for the 2 nodes to communicate.
const int analoginone = A0;
const int analogintwo = A1;
const int analoginthree = A2;
const int analoginfour = A3;
const int analoginfive = A4;
const int analoginsix = A5;
const int analoginseven = A6;
int analoghold[7];

void setup(void)
{
    Serial.begin(57600);
    radio.begin();
}

void loop(void)
{
    analoghold[0] = analogRead(analoginone);
    analoghold[1] = analogRead(analogintwo);
    analoghold[2] = analogRead(analoginthree);
    analoghold[3] = analogRead(analoginfour);
    analoghold[4] = analogRead(analoginfive);
    analoghold[5] = analogRead(analoginsix);
    analoghold[6] = analogRead(analoginseven);

    radio.openWritingPipe(pipes[0]);
    radio.openReadingPipe(1,pipes[1]);
    radio.write(analoghold , sizeof(analoghold) );

    Serial.print("analogue pin 0 = ");
    Serial.print(analoghold[0]);
    Serial.print("analogue pin 1 = ");
    Serial.print(analoghold[1]);
    Serial.print("analogue pin 2 = ");
}
Serial.print(analoghold[2]);
Serial.print("analogue pin 3 = ");
Serial.print(analoghold[3]);
Serial.print("analogue pin 4 = ");
Serial.print(analoghold[4]);
Serial.print("analogue pin 5 = ");
Serial.print(analoghold[5]);
Serial.print("analogue pin 6 = ");
Serial.println(analoghold[6]);

//Reciever Code Written by Aharon Cunta and edited by Levi Pleiksna using Maniacbugs Library

#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include "printf.h"

//
// Hardware configuration
//
// Set up nRF24L01 radio on SPI bus plus pins 9 & 10

RF24 radio(48,53);

//
// Topology
//

// Radio pipe addresses for the 2 nodes to communicate.
const uint64_t pipes[2] = { 0xF0F0F0F0D2LL, 0xF0F0F0F0E1LL };

//
// Role management
//

// Set up role. This sketch uses the same software for all the nodes
// in this system. Doing so greatly simplifies testing.
//
int analoghold[8];

void setup(void)
{
    Serial.begin(57600);
    radio.begin();

    radio.openWritingPipe(pipes[1]);
    radio.openReadingPipe(1, pipes[0]);
    radio.setRetries(1000,1000);
    radio.setPayloadSize(sizeof(analoghold));
    radio.startListening();
}

void loop(void)
{
    if ( radio.available() )
    {
        // Read the data payload until we've received everything
        //     bool done = false;
        //     while (!done)
        {
            radio.read( analoghold, sizeof(analoghold) );
            radio.stopListening();
            radio.startListening();
            // Fetch the data payload
        }
    }
// This section is for testing

// Write the values to the serial monitor
Serial.print("Device -> ");
Serial.print(analoghold[0]);
Serial.print(" LY= ");
Serial.print(analoghold[1]);
Serial.print(" LR= ");
Serial.print(analoghold[2]);
Serial.print(" LP= ");
Serial.print(analoghold[3]);
Serial.print(" RY= ");
Serial.print(analoghold[4]);
Serial.print(" RR= ");
Serial.print(analoghold[5]);
Serial.print(" RP= ");
Serial.println(analoghold[6]);

// This is where the PWM is created

// The analogWrite function works like this:
// analogWrite(*PIN , *VALUE);
// *PIN is a PWM enabled Pin
// *VALUE is a number between 0 and 255

analogWrite(2, analoghold[0] / 4);
analogWrite(3, analoghold[1] / 4);
analogWrite(4, analoghold[2] / 4);
analogWrite(5, analoghold[3] / 4);
analogWrite(6, analoghold[4] / 4);
analogWrite(7, analoghold[5] / 4);

} 

} 

else 
{
    Serial.println("No radio available"); 
}
(17) Schematic images generated using schematics.com
(18) Datataker polynomial calibration analogue input print screen taken from Datataker HTML software

(19) Labview print screen with datataker Example program acquisition interface