Mechanical Engineering Design 3: Sustainable Energy Group Project Report
A Feasibility Study of a Micro Wind Turbine
Group 3

Authors: Max Benson, James Yates, Yidan Xue, Maurice Rahme, Tom Wilson, Murdo Lunn
Submission date: 06/04/2018

Table of Contents

Abstract

The aim of this project is to design and manufacture a small portable wind turbine that can power 2 1A USB ports to charge mobile devices. The Design must be IP67 rated (dust and wind proof) and include a protective shroud around the blades. This was to be achieved under an £100 budget.

The final design is a 3-blade HAWT, with a passive yaw system. BEM theory is used to determine local forces on the turbine blades, and from that the NACA 2415 profile was selected for the final blade. The final maximum mechanical power produced is 18.54W

A custom built axial flux generator is used with a rated power of 15W, allowing an Arduino controller and OLED screen to be powered without the need for an external battery.

The design is modular and can be broken down to a few sub-assemblies for portable transportation. And these sub-assemblies can also be broken apart, if any repairs are needed to be made. The design covers all the points in the customer design specification. The final cost of the project was £80.82, £19.18 under the specified budget.

1. Introduction

One of the greatest challenges facing current and future engineers today, is the effect of human impacted global warming/climate change. To combat this, many engineers are involved in transitioning from a dependence on fossil fuel energy to a solution with renewable, sustainable, green alternatives. Even on the smallest of scales, replacing one-time use batteries with a renewable alternative, such as solar or wind, is vital in this transition.

The greatest renewable source available on land in Scotland is the wind. Wind also has the potential for more versatile applications, ranging from very small micro turbines to large scale turbines, capable of generating mega-watts of energy.

In this report, the aim was to provide a wind-based solution (turbine), capable of charging a small mobile device. The design is aimed at travellers or hikers with little storage space and should be manufactured under a budget of £100. Specifically, the turbine should be able to provide enough energy at 12 m/s, to charge one USB device at 2.1A or two devices at 1A, with a cut-in speed to start charging at 8 m/s.

Due to the application requirement of using the device outside in temperate Scottish conditions, the device should conform to an IP67 rating, meaning it should be resistant to dust or water infiltration. For increased protection, a protective shroud around the blades should be installed, yet the design should also be easy to dismantle and flat-pack for efficient storage and transportation.

Finally, the turbine should be able to support itself without the use of guy-ropes or any supporting structures, by properly designing an appropriate tower to counter the effects of gravity and wind pressure. All parts of the design are to be tested with proper material selection, with structural stability and longevity key to the turbine.

2. The choice of wind turbine design

Our first major design decision was about the type of wind turbine we were going to make. Wind turbines can be split into two main categories: vertical axis wind turbines (VAWT) and horizontal axis wind turbines (HAWT). We needed to choose between the two different types then decide on the number of blades to use.

The initial research we did had us leaning towards the idea of designing a VAWT as they are more suitable to applications at lower heights. However, we were not convinced, and so made a decision matrix comparing different types of VAWT and HAWT.

The values in the decision matrix were decided on by evaluating the advantages/disadvantages of each wind turbine design using relative literature (Centurion Energy, 2013) (REUK, 2018). The outcome of this matrix is shown below in figure 1.

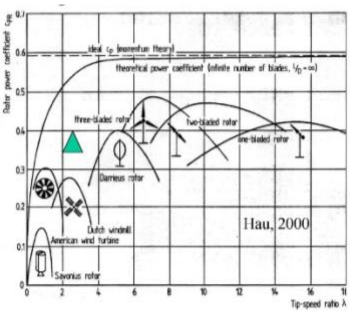
Factors	Weight	Single Blade HAWT	Two Blade HAWT	Three blade HAWT	Darrieus VAWT	Savonius VAWT	Giromill Darrieus VAWT
Modelling difficulty	5	10	8	6	1	2	1
Aeshetics	4	4	5	6	3	4	3
Portability	4	10	8	5	5	6	7
Efficiency	5	5	7	9	5	4	4
Complexity of design	8	2	5	7	9	10	8
Cost	10	9	7	5	2	2	1
Self starting capability	8	9	9	9	3	8	2
Stability	6	3	5	8	5	5	5
Number of parts	5	4	5	6	5	5	5
Appropriate TSR for generator	8	4	6	8	3	3	3
Sensivity to turbulence	5	5	5	5	3	10	3
Weighted totals		404	437	464	272	363	249

Figure 1. Decision matrix on type of wind turbine design

The decision matrix shows that we chose the 3-blade HAWT. Firstly, we chose a HAWT over a VAWT due to higher efficiencies and more appropriate tip speed ratios (TSR). Even though the

Savonius VAWT has very good self-starting capabilities and performs well in turbulent conditions, it is still less efficient than the common HAWT (Wikidot, 2007).

The choice of HAWT was the 3-blade and this was due to its high strength, working TSR and efficiency (Stein, 2018).



Having chosen a HAWT system, a yaw system would need to be implemented to orient the blades so that they are always perpendicular to the wind speed. This is to ensure the maximum power is always generated. An active system was deemed to be too bulky and expensive, as a yaw drive, brake, bearing and control system would be needed. Therefore, a passive system was selected. A downwind system provided sufficient protection from 'wind shade', allowing for easier and more precise power calculations.

Figure 2: Efficiency vs. TSR (Stein, 2018)

3. Blade Design

Blade Element Momentum Method, as described by Ingram (2011), was implemented in the blade design. Assumptions include: ignorance of the area of the hub, incompressible and inviscid flow, no wake rotation, no fluid whirl and no tip losses (Stein, 2018). Based on this theory and these assumptions, MATLAB code was developed to determine the shape of the whole blade. The function of the code will be explained in the following sections.

3.1 Base calculation

The energy extraction equation below (Manwell et al, 2009) is used to decide the tip radius of the blade:

$$P = \frac{1}{2} A C_P \eta \rho V^3$$

where P is the power generated, A is the area, C_P is the coefficient of power, η is the efficiency of the transmission system and generator, ρ is the density of air and V is the wind speed. Then, the circular area is divided into 100 sections from the hub to the tip. In each section, the local rotational speed can be calculated using TSR of 5. Additionally, the Reynolds Number at each section can be determined by:

$$Re = \frac{2\rho V R}{\mu}$$

where R is the local radius and μ is the dynamic viscosity of the fluid (Manwell et al, 2009).

3.2 Profile selection

NACA 4 Digit profiles are compared as there is a large amount of experimental data made available. This includes the maximum Lift/Drag ratio at certain Reynold's Numbers (R_e) for each profile.
$\Longrightarrow \; R_e = 50000, 100000, 200000, 500000$

The blade can be discretised into small sections along its radii; this allows R_e to be calculated at each section for the design wind speed of $12ms^{-1}$.
$\Longrightarrow 44{,}297 \;\le\; R_e \;\le 236{,}250$

To determine which profile is most suited we compare the Lift/Drag ratio of each profile for each section, the profile with the largest ratio is selected.

We can assign the closest provided Lift/Drag ratio to each section; however, it is more accurate to make a linear approximation between the provided values. This allows for interpolation across the range of $50{,}000 \leq R_e \leq 500{,}000$.

We can extrapolate outside of this range for our lower R_e, this may result in a decreased accuracy but as these sections will be the smallest torque contributors it is assumed to be fine.

Profiles	Ratio	Radius
2408	35.666	0.03
2408	42.15	0.044444
6409	52.597	0.058889
6409	63.711	0.073333
6409	69.15	0.087778
6409	74.589	0.10222
6409	80.027	0.11667
6409	85.466	0.13111
6409	127.67	0.14556
6409	135.2	0.16

Figure 3. Initial - 10 Sections (Airfoil Tools, 2018)

Profiles	Ratio	Radius
2415	33.748	0.045
2415	41.03	0.057778
4415	49.349	0.070556
4415	53.631	0.083333
4415	57.914	0.096111
4415	62.197	0.10889
4415	66.48	0.12167
4415	70.763	0.13444
4415	101.5	0.14722
4415	106.39	0.16

Figure 4. Final - 10 Sections (Airfoil Tools, 2018)

Initially two sections were selected, however, discontinuities occurred resulting from large changes in chord length and relative wind angle across the profile transition. This led a redesign in which one profile was selected. Low chord length to thickness ratio profiles were excluded to help increase the strength of the blade.

Despite the selection of 4415 for most sections by the code, the 2415 profile was selected as it produced a thicker blade throughout.

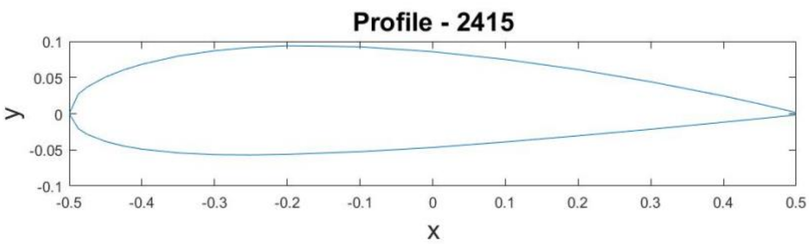


Figure 5. Profile – 2415 (Airfoil Tools, 2018

3.3 Lift and drag approximation

The NACA 4 Digit website provides data points to produce Lift and Drag plots against angle of attack for the R_e previously mentioned. Curves can be fitted to this data allowing both coefficients to be determined from an angle of attack, using this we can interpolate from the provided R_e to the actual R_e values.

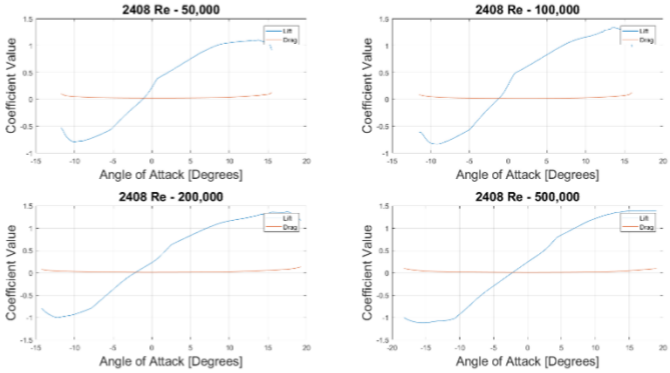


Figure 6. Lift and Drag Curves (Airfoil Tools, 2018)

3.4 Optimisation

After getting the curves of lift coefficient and drag coefficient against angle of attack, the optimal angle of attack at each section was decided by finding the maximum ratio of lift to drag coefficient at the certain Reynolds Number, which was reasonable because this wind turbine is a lift machine instead of a drag machine.

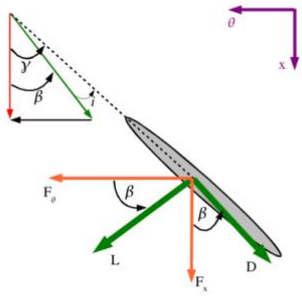


Figure 7. Angle relationship (Ingram, 2011)

Next, the initial guesses were made. Initial relative flow angle could be obtained by

$$\beta = 90^\circ - \frac{2}{3}\arctan(\frac{1}{\lambda_r})$$

where λ_r is the local tip speed ratio (ratio of local rotational speed to wind speed). Twist angle of each section is the sum of relative flow angle and the optimal angle of attack. Then, initial chord length was equal to

$$c = \frac{8\pi r \cos\beta}{3B\lambda_r}$$

where r is the local radius and B is the blade number (3) (Ingram, 2011). After that, the initial axial induction factor and angular induction factor (Liu et al, 2013) could be decided by

$$a = (1 + \frac{4\cos^2\beta}{\sigma(C_L\sin\beta + C_D\cos\beta)})^{-1}$$

$$a' = (1 + \frac{4\cos\beta\sin\beta}{\sigma(C_L\sin\beta + C_D\cos\beta)})^{-1}$$

where C_L is coefficient of lift, C_D is coefficient of drag and σ is the local solidity which is

$$\sigma = \frac{Bc}{2\pi r}$$

After determining the initial guesses of all parameters, the optimisation was carried out. Firstly, in each iteration, new relative flow angle (Ingram, 2011) was obtained by

$$\beta = \arctan(\frac{\lambda_r(1+a')}{1-a})$$

Then, due to the angle relationships as detailed in figure 7, angle of attack was recalculated by:

$$i = \gamma - \beta$$

where γ is the twist angle. There is a correction for angle of attack in order to make it between 0 and 10 degrees, because the coefficient curves are accurate in this range and the angle of attack should be kept away from the range of stall. After getting the new angle of attack, the new coefficients of lift and drag could be calculated from the coefficient curves (figure 6). Using the new lift coefficient, the new chord length (Liu et al, 2013) could be obtained by:

$$c = \frac{8\pi r}{3C_L}(1 - \sin\beta)$$

where β is the new relative flow angle. Next, the local solidity, axial induction factor and angular induction factor were recalculated by the same equation as used in initial guesses. Hence, an iteration loop of parameters was created. Fortunately, this iteration loop could converge in this problem.

When the approximate relative error of relative flow angle, axial induction factor and angular induction factor in a section were all less than 0.5%, the iteration would be stopped and the results of this section would be output. This iteration process only needs 7.8 seconds to reach a solution in 50 sections. In addition, it is more reliable than the ordinary process, for it takes both lift and drag coefficients of the profile into consideration.

3.5 Performance analysis

1.Torque

Implementing the Blade Element Momentum Equations (Ingram, 2011), the torque at each section could be derived as

$$dT = \sigma\pi\rho\frac{V^2(1-a)^2}{\cos^2\beta}(C_L\cos\beta - C_D\sin\beta)r^2$$

Then, two curves of torque against radius in design speed and cut-in speed were fitted as shown in figure 8.

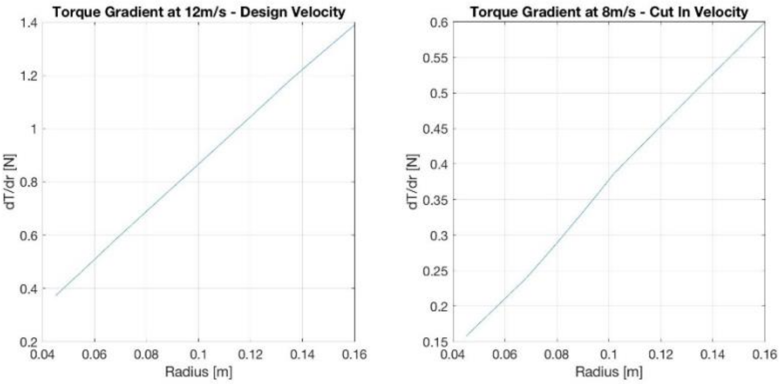


Figure 8. Torque gradients at design speed and cut-in speed

Next, the torque along the whole blade was integrated. This blade is able to generate 0.102 Nm at design speed and 0.044Nm at cut-in speed. At the same time, the starting torque of generator is 0.0016Nm. Hence, it is apparent that this product can start generating power at wind speed of 8m/s.

2.Power

By multiplying torque with angular velocity, the power generated by this blade is 38.27W, which has a power coefficient of 0.451. If the efficiency of transportation and generator is taken into consideration, the final power output will be 18.54W. Therefore, it is sufficient to power both USB charger and Arduino.

3.Stall point

In order to find the stall point, the blade was analysed from high rpm to low rpm. In each rpm, the relative flow angles were calculated again to obtain the new angle of attack. Based on the coefficient curves, the stall angle is about 14 degrees when Reynolds number is smaller than 100,000 and 17 degrees when Reynolds number is larger than 100,000. Hence, when the angle of attack exceeds the stall angle, the blade will reach the stall point. Using this process, we can get the blade will stall at 2540 rpm when the wind speed is 12m/s and 1690 rpm when the wind speed is 8m/s. These data are used in generator design.

4. Nose cone design

4.1 Theory

The nose cone of the turbine was designed from an aesthetic point of view due to the size of the turbine. The working wind speeds did not merit any deep thought into aerodynamic design of the nose cone profile.

A power series method was adopted to find and compare different possible nose cone profiles, and this was done in MATLAB for different values of n. The power series profiles are generated using the equation (The Descriptive Geometry of Nose Cones, 2011)

$$y = R\left(\frac{x}{L}\right)^n \quad , \quad 0 \leq n \leq 1$$

Where *n=the 'bluntness' of the nose cone*, *L=length of the nose cone*, *x=a distance along L*, *y=coordinate in the y-direction and R=radius of the nose cone*.

From the equation, the following profiles were obtained. The profile described by the red line was chosen by the group to be the most aesthetic shape.

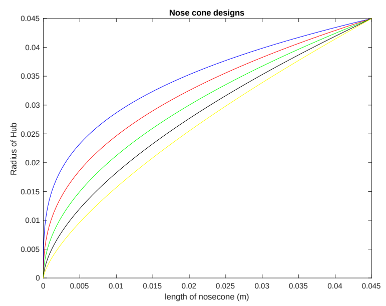


Figure 9. Nose cone profiles

Figure 10. 3D nose cone in solid edge

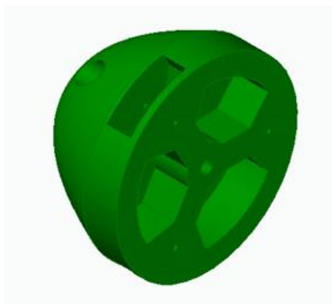


Figure 11. Final nose cone design

4.2 Final concept

The original concept was to have a separate nosecone and hub, where the hub would rotate, and the nosecone would be stationary. However, there would be no easily accessible way of removing the blades, without removing either the nacelle or the nosecone. Therefore, the design was changed to incorporate the hub and the nosecone as one piece.

Having decided on an integrated nosecone and hub part, the connections between the blade and the hub were to be designed. The final concept had the blades being slotted into the hub interior and thumbscrews used to fix them in place. The thumbscrews would therefore have to come through the front of the nosecone, adding some complexity to the design.

4.3 Strengths of chosen final design

Having gone through the iterative design process, the nosecone has improved drastically since its first iteration. The strengths of the selected design include;

- Integration of Hub and Nosecone. Only one piece means fewer failure modes from separate parts and their fastening mechanisms
- Less complexity in the mechanical fastening of parts. Simple but effective fastening, using readily available standard bolts, not specialised parts. Easy to replace if broken or lost
- Aesthetically pleasing design using power series representation
    - Hub can fit blade connections easily, whilst keeping continuity in its curvature, and the hubs radius perfectly matches the nacelle.

The nosecone also meets the customer design specifications, maintaining an IP67 rating. Its modular design also allows it to be disassembled quickly and easily, meeting the portability requirements.

5. Bearing selection

When operating at the design wind speed of $12ms^{-1}$; base calculations determine the main shaft speed to be 3581 RPM, numerical integration across the blade shows the thrust force to be 1.3395N.

As a passive yaw system is implemented, rotation of the nacelle is minimal; the main requirement of the tower bearing is to support the nacelle weight. In the prototype a 15mm inner diameter bearing was selected. In future manufacturing a larger bearing would likely be selected as the Tower Mount would be made the same diameter as the Tower Adapter for aesthetic reasons.

The decision to use Metal Shielded Bearings was made, as they can hold lubricant unlike Open Bearings and they have less friction losses than Rubber Shielded Bearings.

	Main Shaft Req.	6mm 626-ZZ Rating	Safety Factor	Tower Req.	15mm 61802-ZZ Rating	Safety Factor
Speed [RPM]	3581	3.50E+04	9.77	-	2.80E+04	-
Axial Load [N]	1.3395	1.08E+03	8.06E+02	1.24E+01	8.00E+02	6.44E+01

Figure 12. Bearing Requirements and Ratings (Dunlop BTL, 2018)

6. Nacelle

The main shape of the nacelle was made cylindrical to aid in reducing its drag coefficient, the diameter was based on that of the 60mm generator rotors. This gave an inner diameter of 80mm and outer diameter of 90mm.

There are two caps at either end of the nacelle. Both contain curved geometries to attempt to limit fluid separation, but in reality, this is mainly aesthetic.

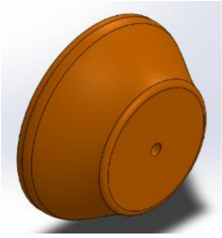


Figure 12. Initial Hub Cap

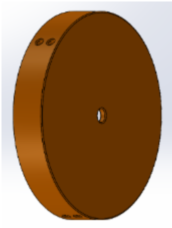


Figure 13. Final Hub Cap

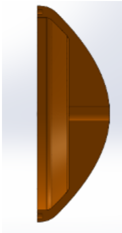


Figure 14. Final Tail Cap

The initial Hub Cap design was the result of the nacelle and hub having different radii, this was not the case in the final design, meaning a flat section could be used. Holes were included at 120° apart to mount the shroud. The Tail Cap has a 16mm threaded hole to support the rudder arm.

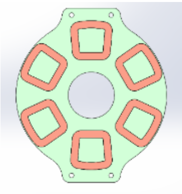


Figure 15. Stator including tabs

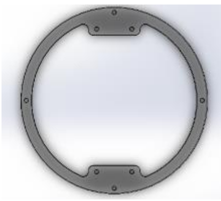


Figure 16. Stator Mounts

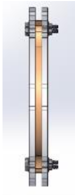


Figure 17. Bolted Stator

Tabs were added to the stator and stator mounts; the stator is to be bolted between two mounts. This allows the nacelle to be split into two sections for prototype maintenance. Bearing stress is the most likely failure type here, holes are positioned 1.5 times their diameter from edges to mitigate this.

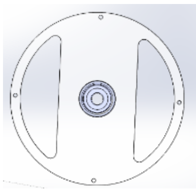


Figure 18. Bearing Mount

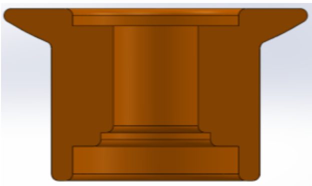


Figure 19. Cross Section of Tower Mount

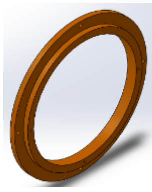


Figure 20. Connection

Bearing mounts are of a similar thickness to the bearings to maximise contact area. Cut outs have been included to reduce their mass, but when further developing should be solid to help maintain IP67 rating more effectively.

The tower mount has been designed to accommodate a slip ring in its top and the tower bearing in its base, the curved surface in the top also allows for it to sit nicely inside the curved nacelle geometry.

The connection piece is designed with a lip that allows silicon to be used in place of an O-Ring, meaning the caps are removable to access the stator bolts and the seal remains water tight.

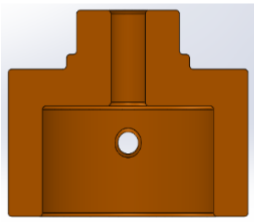


Figure 21. Cross Section of Tower Adapter

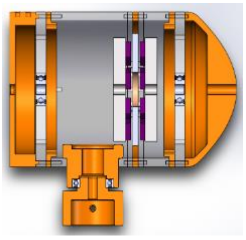


Figure 22. Cross Section of Nacelle Prototype

A Tower Adapter is fitted into the tower bearing, this then fits over the tower itself and is pinned in place allowing the nacelle to be removed.

The clear sections of the Nacelle prototype were laser cut from acrylic and the orange sections were 3D printed. These were glued together and were aligned using four pin holes found around their edges. The bearings were epoxied in place.

When upscaling the manufacturing process it is recommended that the nacelle be injection moulded in the three sections shown below. The recommended material being polypropylene as it is naturally clear allowing the generator to be seen, it is chemically inert, has excellent impact strength in case of dropping, and is relatively cheap at £0.47/kg. O-Rings should be used in place of the silicon.

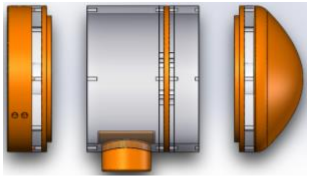


Figure 23. Injection Mould Sections (Plasticker, 2018)

7. Tail design

7.1 Theory

For the tail design we propose the following geometry:

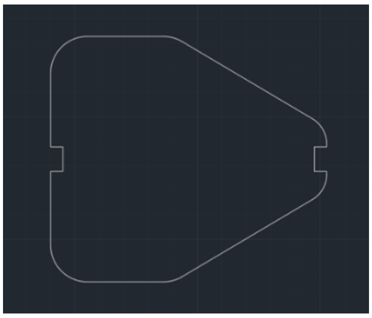


Figure 24. Tail fin geometry

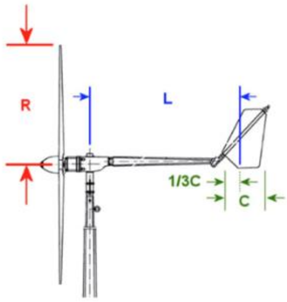


Figure 25. Dimensions of tail system (Windy Nation, 2010)

This shape of tail was justified for primarily aesthetic reasons. Some brief research was done on different tail shapes and we found that the choice of shape had little effect on the performance of the system. Therefore, we picked a tail shape that was aesthetically appealing. The dimensions of the tail were decided on through the area the shape was required to have.

The tail area required was taken to be (Windy Nation, 2010):

$$Tail\ area = 10\% \times sweep\ area\ of\ rotor = 0.1\pi R^2$$

Where *R=radius of rotor*. The tail boom length, *L*, was defined using the simple equation (Windy Nation, 2010):

$$L = 1.2 \times R$$

7.2 Manufacture considerations

Our plan was to 3D print the tail system. However, after consulting with the lab technician Andrew we discovered that our concept would take up too much printing time, so this was not a feasible method.

The alternate solution was to use 2mm thick sections of laser cut acrylic. These sheets were connected together from each end via 3D printed parts that provide a male-female connection to the tail sections.

Acrylic was the chosen material due to its strength as well as being lightweight and aesthetically pleasing.

A piece of 6mm steel rod threaded at each end connects the tail to the nacelle. This was a key element of our design as we wanted to keep everything as modular as possible. Having the ability to screw out the tail from the nacelle is a major advantage of our design.

8. Shroud design

The shroud we chose was inspired by simply looking on Grab Cad at wind turbine shrouds. Since the product specification only mentioned the shroud acting as protection for the blades we decided that a simple static shroud surrounding the blades would be sufficient.

Figure 7 shows the design that inspired the type of shroud. We also had to find a good method of connecting the shroud to the nacelle. Our solution was to have three connections from the nacelle to the shroud, each placed 120 degrees from the next. The connectors had two 3mm by 3mm pins sticking out from either end and these slot into the shroud and nacelle to hold the assembly together.

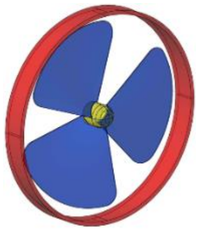


Figure 26. Shroud inspiration (Rehman, 2017)

As per the tail, we wanted to keep the shroud modular to make it easy for the customer to assemble the turbine. Our idea for the shroud was to split it into six sections that would slot together to form the completed part.

8.1 Manufacture considerations

We planned to have each section of the shroud 3D printed, however, after discussing our idea with Andrew he told us that the printing would use up far too much time and material to be feasible. He suggested that we opt towards using acrylic sheets instead.

Our final design incorporated both modularity as well as the use of acrylic. Each section of the shroud was made from five pieces of 8mm acrylic. The outside pieces were normal curved sections and the middle pieces had sections cut out/added to either end to provide a connection mechanism between consecutive shroud pieces.

Figure 27. Middle section of shroud

Figure 28. End section of shroud

Figure 29. Full shroud assembly

After assembling the acrylic, the connections between parts were not tight enough to have a modular shroud. Therefore, all sections were epoxied together to give a stronger and more stable shroud for the prototype. Also, the connections from the nacelle were not strong enough. Our solution was to drill into the connector pieces and use steel rod to give a stronger connection.

9. Electronics box

The electronics box is very important, as all the main circuitry and electronics are to be housed within it. It is especially important that it conforms to the IP67 rating required, being protected against both water and dust infiltration. This required every part of the box to be precision manufactured and therefore, 3D printing was used to achieve this.

Initial research suggested two possible solutions on where the electronics should be housed;

1. Electronics to be housed with/next to/in the nacelle itself
2. Electronics to be housed on the tower.

The second concept was preferred because of the need for sufficient space for electronics. However, as the yaw system allows for rotation of the nacelle, any wires leading into the tower are in danger of twisting and breaking. Furthermore, we wanted to place a screen on the housing with relevant information (wind speed/RPM etc); if this screen were constantly rotating, it would be hard to read. Therefore, a slip ring was used to avoid tangling of the wires in the tower as the nacelle rotated, while keeping the electronics housing stationary.

As mentioned before, a constant theme of our design is that it is portable and highly modular. This theme is also present within the housing. A 'shelf system' was designed to house each of the separate parts of the electronics (Battery/temp sensor → Arduino → Circuitry PCB → Screen). Each 'shelf' is connected and held in place with a series of small pins in the walls, aiding alignment. This meant in the event of an electrical failure during prototyping, the problem component could be identified and fixed, without disturbing the other parts of the electronics. This is shown via the exploded view of the housing.

Figure 30. Exploded view

10. Electronics

10.1 Generator

Multiple generator configurations were available for this application. Two prominent ones are Axial Flux (AFPM) and Radial Flux (RFPM) permanent magnet generators. The AFPM setup was chosen as it is coreless, and hence has negligible starting torque, no eddy current losses and near-noiseless output signals, which are essential when measuring frequency in real time, as is done for this product. The generator's starting torque was estimated at **0.00167Nm** as in the report FEMM simulation. Star configuration was chosen as it simplifies testing via the neutral output, and has a higher line voltage than phase voltage, which is more compatible with the buck-boost converter, which encounters faults if fed below 3.5V.

To meet the specification of providing a cumulative 10W at 5V, 2A to the USB outputs, some contingency was required, taking into consideration the presence of a control circuit operated by an Arduino Uno, which also outputs some user information to an OLED display, and grants the user control over the system's operation. In considering these variables, a more suitable generator target output was found; noting that some components consume negligible power and thus are not accounted for:

Table 1. Electrical Component List

Component	Current (Typical)	Voltage	Power Consumption	Datasheet Link
Arduino Uno	250mA	5V	1.2W	https://goo.gl/nD5gDk
I2C OLED	10mA	3.3V	33mW	https://goo.gl/LiyNfa
Push Button	NEG	NEG	NEG	N/A
Switch	NEG	NEG	NEG	N/A
MOSFET	250uA	5V	1.25mW	https://goo.gl/XzDit8
Relay	40mA	5V	200mW	https://goo.gl/dJpPpW
INA219 Sensor	5mA	5V	17mW	https://goo.gl/MXgNzS
DHT11 H/T Sensor	2.5mA	3.3V	8.25mW	https://goo.gl/t8yS1t

Additionally, the following factors were considered; Buck-Boost Converter Efficiency: 90% (TPS63036 High-Efficiency Single Inductor Buck-Boost Converter in Tiny WCSP, 2015). Generator Efficiency: assumed 80% for Coreless Axial Flux setup (DESIGN, PROTOTYPING AND ANALYSIS OF A LOW-COST DISK PERMANENT MAGNET GENERATOR WITH RECTANGULAR FLAT-SHAPED MAGNETS, 2008).

This amounts to an insufficient net output power of **7.85W** left for the phones, assuming a 10W generator design. Thus, to comfortably achieve the **10W** net supply, the minimum design specification was found to be **20W**, which would provide a net output of **14.14W** without compromising on size, allowing for some contingency, notably at the cut-in RPM. This even leaves room to power the Arduino without an external battery, although this is difficult logistically, as the pre-DC-DC-converter start-up stage is controlled by said Arduino and does not yet power any devices at this point for the safety of the circuit, as detailed later.

A comprehensive generator design MATLAB script was created, which allows for altering parameters and scaling as per the customer's needs. (3-D FEM Analysis, Prototyping and Tests of an Axial Flux Permanent-Magnet Wind Generator, 2017) (Axial Flux Permanent Magnet Generator Design for Low Cost Manufacturing of Small Wind Turbines, 2012).

Here are some key parameters. Given 3600RPM working speed, the number of poles was chosen with this equation:

$$p = 120\frac{f_{nom}}{n_{nom}}$$

Where f_{nom} is nominal frequency and n_nom is nominal RPM. And hence, the number of coils:

$$Q = \frac{3}{4}p$$

Finally, the number of magnets per rotor was selected:

$$n_m = \frac{4}{3}Q$$

Next, the steel rotor plate thickness was chosen as 7mm in an iterated process carried out in FEMM, which delivered a flux density at the rotor plate just below 1Tesla, as per the specification. Finally, the magnet selection was possible, with both performance and cost being considered as these are the largest generator expense. The aim was to minimize the generator's size whilst meeting both of these requirements. 10x10x3mm N42 Neodymium magnets were selected since although the 10x10x5 magnets performed 30% better, according to their remanence and coactivity, they came at twice the price tag (FIRST4MAGNETS, n.d. 1) (FIRST4MAGNETS, n.d. 2). A 1mm airgap between the coil and magnet faces justified this choice.

The coil dimensions were then set as follows:

The stator's axial thickness was a function of the magnet's thickness (h_m), airgap (g), vacuum (mew_0) and recoil permeability (mew_{rrec}), remnant magnetic flux density (B_r), corrective field strength(H_c) and saturation factor (rk_{sat} - equal to 1 since the setup is coreless)

$$t_w = 2(\frac{h_m}{mew_{rrec} rk_{sat}} - g)$$

Where: $$mew_{rrec} = \frac{B_r}{mew_0 H_c}$$

The number of turns per coil is a function of cut-in EMF (E_{fcutin}) and RPM (n_{cutin}), coils per phase (q), maximum magnetic flux per pole (phi_{max}), and winding coefficient (rk_w).

$$N_c = \frac{\sqrt{2}E_{fcutin}}{2q\pi rk_w phi_{max} n_{cutin} \frac{p}{120}}$$

Alongside this operation, the coiling wire's diameter is chosen based on a function of maximum AC current, and a current density set to 6A/mm^2, which is appropriate for this application, as the spinning rotors will help cool the coils with the holes drilled inside them (3-D FEM Analysis, Prototyping and Tests of an Axial Flux Permanent-Magnet Wind Generator, 2017) (Axial Flux Permanent Magnet Generator Design for Low Cost Manufacturing of Small Wind Turbines, 2012). This then sets the coil's radial leg width, which is also a function of the number of turns (N_c), the stator's axial thickness (t_w), and the coil fill factor (rk_f), which can vary between 0.55 and 0.78; 0.65 was chosen as a median value.

Next, the coil's radial leg thickness is set as follows:

$$w_c = \frac{s_c N_c 10^{-6)}}{rk_f t_w}$$

Finally, the generator's inner and outer radii were found according to the following physical dimensioning parameters, based on the magnet and coil sizes

$$R_{in} = (\frac{2Qw_c + pw_m}{2\pi} - w_c) * 1000$$

$$R_{out} = R_{in} + (l_a + w_c) * 1000$$

The diagram below illustrates these parameters and how they play a role in dimensioning the setup:

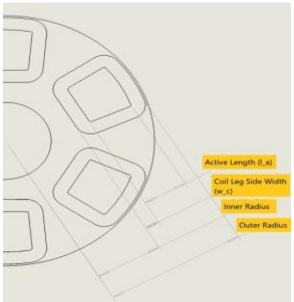


Figure 31. Generator parameters

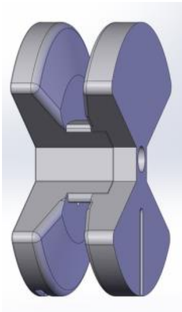


Figure 32. Coil wire jig

To manufacture the generator, the 'a' winding jig was made for repeatability, with the outer insert encompassing a tolerance fit to allow the inner to fit snugly. This jig allows for a simple setup and wire securing system, and also leaves an inner gap for winding electrical tape around the finished coils to retain their shape for further work.

The below images illustrate the connections required for star configuration. Leads that are soldered together are first heated and scraped to expose the copper, and heat shrink is applied to insulate the connection, and avoid phases touching. Electrical tape is also added where necessary, and the whole assembly is cast in resin for robust protection.

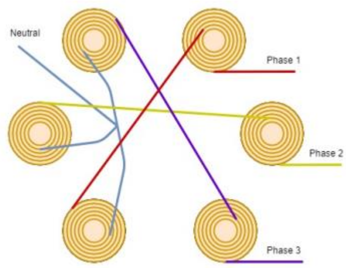


Figure 33. Star configuration connection

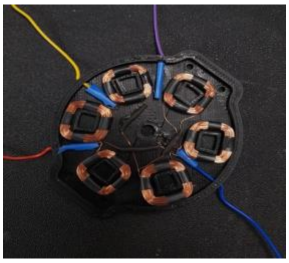


Figure 34. Stator with star configuration

In retrospect, a simpler solution would have been to wind pairs of coils together, so as to cut down on soldering and potential shorts. This would limit the soldering job to just the neutrals and the final output phases and hasten manufacturing while also improving quality. The stator can be cast using the existing mould, which must be waxed first to allow for easy removal of the final product. In the event that this can't be done, the bottom of the mould should be ground down as far as possible to reduce the resultant air gap

As for the rotors, the assembly is simple. A laser-cut acrylic jig is epoxied to the rotor, magnets are placed in their respective slots. It is essential to ensure that the magnets have opposite polarity adjacently. This can easily be verified by holding the magnet above its adjacent pair and ensuring that it is repelled.

Final Generator Specifications:

Table 2. General generator specifications

Rated Power	Pole Pairs	Coils	Phases	Air Gap	Inner Diameter (Net)	Outer Diameter (Net)
15W	8	6	3	1mm		

Table 3. Stator specifications

Stator Axial Thickness	Coil Radial Leg Width	Turns Per Coil	Copper Diameter (with insulation)
3.31mm	**3.43mm**	**96**	**0.28mm**

Table 4. Rotor specifications

Rotor Plate Thickness	Magnet Axial Thickness	Magnet Radial Width	Magnet Active Length
7mm	**3mm**	**10mm**	**10mm**

10.2 Control circuit

1- Rectifier via Slip Ring:

Schottky diodes were used for the 3-phase AC → DC rectifier as they have a low voltage drop (0.5V as opposed to 0.7V for conventional diodes) and allow for maximal power conservation. A smoothing capacitor is used to produce a more constant signal. It was specified as per the following equation:

$$C_{smooth} = \frac{P_{nom}}{V_{max}^2 - V_{min}^2} * \frac{1}{f}; \frac{\left(\frac{15}{10^2 - 3.5^2}\right)}{240} = 712\mu F$$ ($680\mu F$ chosen).

V_{max}, f, and V_{min} are determined during testing **[3- Power Electronics 3]**

2- Relay:

An Arduino-powered relay is used to connect or disconnect the buck-boost converter. This was used instead of a MOSFET switch as the buck-boost has inverting properties which would then operate outside the switch's control characteristics at generator input voltages above 5 Volts, which is the maximum the Arduino controller can supply.

3- Buck-Boost DC-DC Regulator:

This was tuned to output a constant 5 Volts irrespective of the input voltage, a 5.6V Zener Diode and a 2.5-amp Fuse were added as protection circuitry, since it was found during testing that feeding the Buck-Boost sub-3.5V resulted in an unstable 30+V output. A switching regulator was preferred over a linear regulator as it has up to 94% efficiency and low heat output, although this one is assumed to have 80% efficiency for design and specification purposes.

4- Dump Load 1:

This dump load serves to slow the generator down in the event that it exceeds the specified operating range (>3900 RPM). A MOSFET switch is used to control this dump load (15W Power Resistor bolted to a heatsink) as the drain voltage is known to be 5V as supplied by the Buck-Boost Converter. This is also controlled by the Arduino via the MOSFET gate, as shown in the circuit diagram.

5- Battery → Phone:

A 5V supercapacitor is used to power the phone in case of inconsistent wind. It is connected to the USB outputs in parallel so that the phones are always able to draw power when plugged in. 100F, 5V Supercapacitors were found for £2 from China on eBay, however they did not arrive on time. The demo uses a 7.5F 5V capacitor, capable of providing 10W for approximately 9 seconds. This is found by calculating the energy stored in the capacitor: $\frac{CV^2}{2}$, converting this to Watt-hours, and dividing by Watts to find supply provision time. By comparison, the design-specified 100F capacitors can provide power for approximately two minutes. The supercapacitor feeds a dual USB A 3.0 Port configured to pull 2 amps when connected. Supercapacitors were chosen over Li-Ion or Li-Po for the reasons detailed below:

Function	Supercapacitor	Lithium-ion (general)
Charge time	1–10 seconds	10–60 minutes
Cycle life	1 million or 30,000h	500 and higher
Cell voltage	2.3 to 2.75V	3.6V nominal
Specific energy (Wh/kg)	5 (typical)	120–240
Specific power (W/kg)	Up to 10,000	1,000–3,000
Cost per kWh	$10,000 (typical)	$250–$1,000 (large system)
Service life (industrial)	10-15 years	5 to 10 years
Charge temperature	–40 to 65°C (–40 to 149°F)	0 to 45°C (32°to 113°F)
Discharge temperature	–40 to 65°C (–40 to 149°F)	–20 to 60°C (–4 to 140°F)

Table 3: Performance comparison between supercapacitor and Li-ion.
Source: Maxwell Technologies, Inc.

Figure 35. Supercapacitor data

6- Dump Load 2:

This dump load is connected *before* the DC-DC converter, and hence the relay, and serves to slow the generator down to a complete stop once the user decides to shut it off for disassembly. It also prevents the turbine blades from rotating prematurely, before the user has completed their assembly. A MOSFET Switch is used here since it is intended for use below 2400RPM, where the voltage is known to be below 5V.

7- Schmitt Trigger:

The Schmitt trigger's purpose is to cut down on calibration required for mass-production. It allows the Arduino to read the frequency off one of the generator's phases, and hence derive RPM in real time. The phase input fed into the trigger is stepped down by a potential divider as the maximum input that can be read is 5.5V, as per its supply rail. The input is halved as our maximum measured voltage is approximately 10V. The trigger outputs a 0-5V square wave that is fed into the Arduino. The Freq.Count library, optimized for 0.1Hz-1kHz is used to read frequency from the schmitt trigger output. It is used instead of an op-amp as it is immune to noise, which can be expected from a generator signal. RPM can be calculated from the frequency from equation (1) as the pole pairs are known**.** Wind Speed can also be derived from RPM, as done in the initial design process.

8- Current and Voltage Sensor:

The INA219 current and voltage sensor is connected to the smoothing capacitor's terminals to measure the voltage and current produced by the generator. This will read actual power produced, and although an argument for measuring available power could be made, it was found more beneficial to read prior to the DC regulator as this would allow for displaying user information even before reaching cut-in RPM. Furthermore, reading prior to the buck-boost allows the code to set a condition for the relay to disengage the buck-boost connection if voltage falls below 3.5V, as a protection measure. It is powered by the Arduino and sends data via its SDA and SCL pins.

10.3 Software

1- Arduino Uno

This is the microcontroller used for the charging and display circuitry. A nano was proposed as an alternative to save space and power, but external memory needs to be added to accommodate our display-heavy code.

2- Temperature and Humidity Sensor

This feeds relevant environmental information to the Arduino.

3- OLED

The I2C OLED is the centrepiece of the user interface, it displays critical information such as RPM, Wind Speed, Current, Voltage, Frequency, Power Output **(and Energy generated since the last boot)**, as well as Temperature and Humidity, System information, and the names of our group members. Upon start-up, the user is greeted with an animation of our logo. It is powered by the Arduino and is fed data via its SCL and SDA pins.

4- Push Button

The push button allows the user to cycle through different screens to display different information. It connects to pin 8 on the Arduino, and ground.

5- Two-Way Switch

The switch serves to start up and shut down the generator. By default, when the switch is off, dump load #2 is engaged, and the Arduino is put to sleep mode, drawing reduced current from its 9V battery source. When it is switched on, this dump load is disengaged to allow the generator to start up with minimal current draw, and hence torque. This feature is there for user safety, as it would be a critical danger to have the turbines rotating while the user is still mounting the shroud. If the switch is turned off during normal operation, dump load 1 will engage until the relay is turned off, which is when dump load 2 comes in. Once the Arduino detects that the generator is no longer spinning, it goes into sleep mode to conserve power, and dump load 2 is left engaged as an electrical lock to prevent accidental start-up. The switch is recognized as it feeds a signal from the Arduino to itself from adjacent digital pins when it is switched on.

11. Analysis

11.1 Analysis at shaft

Firstly, the analysis at shaft was used to determine the shaft diameter during the initial design. If the safety factor is set to be 3, the diameter required for the shaft will be 3.5mm. If the stress concentration is also considered, the diameter required will be about 5mm. Hence, a 6mm shaft was chosen. During the manufacture, some changes were made to the initial design. Therefore, a final analysis at shaft was completed in order to make sure it is suitable for the final design.

The torque at the shaft can be simplified as figure 36 shows, if we assure all the torques are point torque.

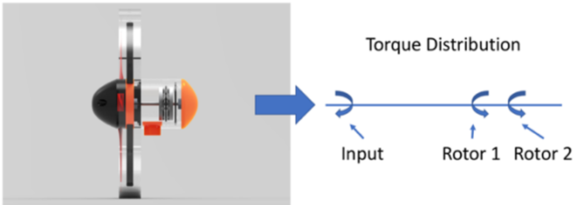


Figure 36. Torque distribution

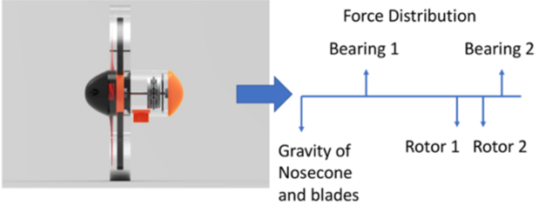


Figure 37. Force distribution

The force at the shaft can be simplified as figure 37 shows, if we assume all the forces are point load.

Then, the bending moments and torsional moments distribution on the shaft can be calculated and plotted using MATLAB as figure 38 presents.

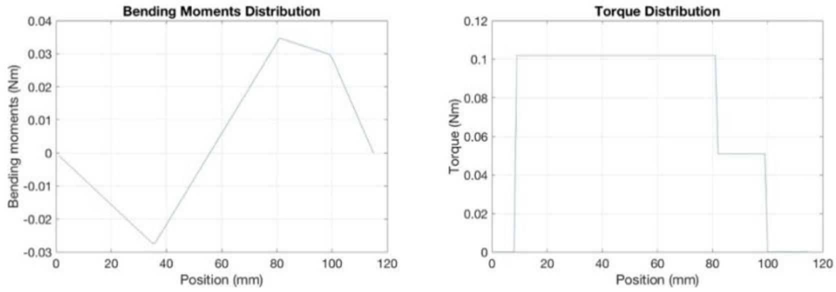


Figure 38. Bending moments and torque distribution along the shaft

Based on maximum shear stress theory, the diameter required along the shaft can be expressed as:

$$d = \left\{ \frac{5.1}{\tau_p} [(C_m M)^2 + (C_t T)^2]^{1/2} \right\}^{1/3}$$

Where τ_p is the permissible stress of the material, C_m is the factor of bending moments and C_t is the factor of torsional moments. In our case, the loads are steady, and the shaft is rotating. Hence, the C_m and C_t should be 1.5 and 1.0 (Shigley, 1986). If the permissible stress is set to be 215MPa (yield stress of 0.20C steel) and the safety factor is set to be 3, the required diameter against position on the shaft can be plotted by MATLAB as figure 39 shows.

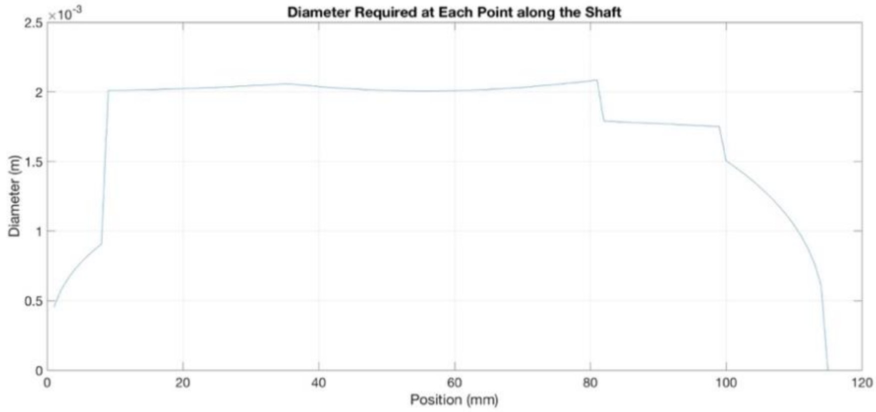


Figure 39. Diameter required for the shaft

11.2 Analysis at tower

Analysis of forces at the tower was completed during the design process. For the final design, the torque at tower can be simplified as the sum of torque of gravity and torque of axial wind force. The axial wind force is relatively small compared with gravity. So, it does not contribute much to the final torque. Thus, in order to balance the torque at the tower, the mass centre should be placed close to the axis of tower cross section. In the initial design, the generator was in front of the tower. However, including the mass of shroud, the mass centre would be about three centimetres in front of the tower and would cause a large torque. Therefore, the generator was moved behind the tower. In this way, the mass centre is very close to the axis of the tower. In addition, implementing the BEM theory (Ingram, 2011) and MATLAB code, the axial wind force can be obtained as 0.536N at cut-in speed and 1.201N at design speed.

After the calculation, the torque at the tower will be 0.047Nm when not operational, 0.017Nm at cut-in speed and 0.021Nm at design speed.

If we calculate the maximum stress at the bottom of the tower, it will be less than 0.2MPa. It is definitely smaller than the yield stress of aluminium (Gere and Goodno, 2009).

11.3 FMEA

In order to prevent the possible failures, Failure Modes and Effects Analysis (FMEA) was carried out throughout the process of design, manufacture and testing. Severity, occurrence and detection were analysed separately by criteria from Siemens (2016) and the product of them is obtained as the Risk Priority Number (RPN). The failure modes with RPN greater than 50 is listed in table. 1 (For the whole FMEA, please find attached appendix).

Table 1. Failure modes with RPN greater than 50

	Function	Failure Mode	Effects	Causes	S	O	D	RPN
1	Blade	Broken in the middle	The product cannot continue working (only two blades remaining). It may also cause injuries.	The strength of material (3D printing) is not enough	9	2	3	54
2	Blade	Broken at the connection	The product continues working (only two blades remaining). It may also cause injuries.	The connection is thin and the moment there is large	9	3	4	108
3	Nosecone	Broken	The product cannot continue working	The strength of material (3D printing) is not enough	9	3	3	81
4	Shroud	Connection broken at the cap	Shroud will fall	The strength of material (3D printing) is not enough	5	4	3	60
5	Generator stator	The generator phases touch	The generator will stop supplying power until the phases are uncoupled	Insufficient/deteriorating insulation or improper contacting	9	3	3	54
6	Circuit	water infiltration	Damage to circuit/short circuit	USB ports, or push button/toggle switch having poor elemental protection	8	2	5	80

This table shows the riskiest failures are those in 3D printed parts and electrical parts. Several actions were taken to reduce the possibilities of these failures. Firstly, parts made by 3D printing are often not strong enough because of their layered and hollow structures. Hence, some parts are redesigned with a larger area of cross section, some are printed with thicker surface and higher resolution. In addition, nearly all of them are epoxied in order to make the connections between each layer stronger. Next, in order to prevent the generator phases from touching, each phase is isolated with electrical tape inside the generator stator. Epoxy is also used to seal any possible holes near the USB ports and push buttons so that no water will infiltrate the circuit.

12. Conclusion

The final design for our turbine was a HAWT, with a downwind yaw system and an axial flux generator. The design is modular and can be broken down and flat-packed into a small compact travel case. It is designed to break down into the following sub-assemblies; nosecone/nacelle assembly, three blades, tailfin, rudder arm, and tower and electronics box. In troubleshooting errors or making repairs, each of the sub-assemblies are designed to break down further as well. For example, accessing the generator can be achieved by removing the caps from the nacelle.

The final design uses a 3D printed ABS nosecone in front of the blades, based on a power series to improve air flow to the blades at low speeds. The 3D printed blades slot into the hub of the nosecone and are fixed in place with small thumbscrews through the nosecone. The blades were 3D printed, in order to precisely create the complex geometry from the blade modelling script. In designing the blade profile, many NACA 4-digit profiles were tested, looking for the maximum lift/drag ratio possible. NACA profile 2415 was eventually chosen over 4415, due to the thicker blade profile throughout for a higher strength blade. This decreases the risk of failure of the blade, improving its life.

Analysis of the blade showed it can generate 0.102 Nm torque at 12 m/s and 0.44 Nm at 8 m/s. As the generator starting torque is only 0.0016 Nm, the blade profile is sufficient to start charging the device at the cut-in speed and the overall power generated from the blades is 18.54 W

The other half of the design is the electrical power generation and electrical systems. The generator is an axial flux generator, with a minimum 15W design specification providing a net output of 11.9W. This is sufficient to power a 2A USB port (or two 1A USB ports), the Arduino controller and OLED screen, without the need for an external battery, although a 9V battery is used for this prototype.

A control circuit was also designed for the electrical power generated. This included conversion of a 3 phase AC input to a steady DC output, as well as the ability to only supply current to the USB once the 8 m/s wind speed is achieved and the slow down the turbine after 12m/s is exceeded via electrical braking. A total system shutdown can also be achieved with a 30W dump load, for safe assembly and dismantling of the turbine.

The design includes an aforementioned OLED screen, for displaying useful information to the user. Given that the target consumer market is hiker's and outdoor enthusiasts, a temperature and humidity sensor was included to provide useful information. The OLED screen also outputs data on the wind-speed, RPM and power output.

The turbine shaft and tower have been extensively analysed with an inbuilt safety factor, to ensure the choice of material and sizes is appropriate for safe long-term operation, with minimal risk of failure. One of the greatest strengths of our design is that the location of the Nacelle's CoM lies almost directly over the centre of the tower, resulting in a low torque at the tower for increased dynamic stability.

To fully adhere to the customer design specification, the turbine was also created to be as water-proof and dust-proof as possible to give it an IP67 rating. This was achieved through careful design of parts that fit together to form a tight seal, and epoxy was used for any open ports.

References

3-D FEM Analysis, Prototyping and Tests of an Axial Flux Permanent-Magnet Wind Generator. (2017). 9th ed. [ebook] pp.1-14. Available at: http://www.mdpi.com/1996-1073/10/9/1269 [Accessed 5 Apr. 2018].

Airfoil Tools. (2018). *NACA 4 Digit Airfoil Specification.* [Online] Available at: http://airfoiltools.com/airfoil/naca4digit
[Accessed: 01 Mar. 2018]

Ashby, M.F., 2017. *Materials selection in mechanical design* Fifth., Burlington, MA: Butterworth-Heinemann.

Axial Flux Permanent Magnet Generator Design for Low Cost Manufacturing of Small Wind Turbines. (2012). *Wind Engineering*, 36(4), pp.411-442.

Centurion Energy. (2013). Types of Wind Turbines. [online] Available at: http://centurionenergy.net/types-of-wind-turbines [Accessed 28 Mar. 2018].

DESIGN, PROTOTYPING AND ANALYSIS OF A LOW-COST DISK PERMANENT MAGNET GENERATOR WITH RECTANGULAR FLAT-SHAPED MAGNETS. (2008). *Iranian Journal of Science & Technology*, 32(B3), pp.191-203.

Dunlop BTL. (2018). *Dunlop BTL – Single Row Ball Bearings*. [PDF], p.163-164, 185-186. Available at: http://www.dunlopbtl.com/downloads/Dunlop%20BTL%20-%20Single%20Row%20Ball%20Bearings.pdf
[Accessed: 01 Mar. 2018]

FIRST4MAGNETS. (n.d.). *10 x 10 x 3mm thick N42 Neodymium Magnets - 2.2kg Pull*. [online] Available at: https://www.first4magnets.com/rectangular-magnets-c35/10-x-10-x-3mm-thick-n42-neodymium-magnets-2-2kg-pull-p2441#ps_1-667 [Accessed 5 Apr. 2018].

FIRST4MAGNETS. (n.d.). *10 x 10 x 5mm thick N42 Neodymium Magnet - 3.5kg Pull*. [online] Available at: https://www.first4magnets.com/rectangular-magnets-c35/10-x-10-x-5mm-thick-n42-neodymium-magnet-3-5kg-pull-p2553#ps_1-738 [Accessed 5 Apr. 2018].

Gere, J.M. & Goodno, B.J., 2009. Mechanics of materials Seventh edition, SI version.., Stamford, Conn.: Cengage Learning.

Ingram, G. 2011. *Wind Turbine Blade Analysis using the Blade Element Momentum Method*. [Online]. Version 1.1. Durham: Durham University. [Accessed 18 March 2018]. Available from: https://community.dur.ac.uk/g.l.ingram/download/wind_turbine_design.pdf

Liu, X., Wang, L. & Tang, X. 2013. Optimized linearization of chord and twist angle profiles for fixed-pitch fixed-speed wind turbine blades. *Renewable Energy*, 57, pp.111–119.

Manwell, J., McGowan, J.G. & Rogers, A.L., 2009. *Wind energy explained : theory, design and application* Second., Chichester: Wiley.

N/A. (2016). Implementation Within Wind Turbine. Available: https://www.liebherr.com/en/deu/products/wind-energy/wind-energy.html#!/content=cp-wind-energy-tab-components. Last accessed 30th March 2018.

N/A. (2016). Wind Turbine Technologies. Available: http://www.greenrhinoenergy.com/renewable/wind/wind_technology.php. Last accessed 30th March 2018.

N/A. (2018). Upwind Turbine Vs. Downwind. Available: http://www.power-talk.net/upwind-turbine.html. Last accessed 30th March 2018.

N/A. (2018). Wind Shade. Available: http://mstudioblackboard.tudelft.nl/duwind/Wind%20energy%20online%20reader/Static_pages/calculator_wind_shade.htm. Last accessed 30th March 2018.

N/A. (2018). Yaw Systems. Available: http://www.wikiwand.com/en/Yaw_system. Last accessed 30th March 2018.

Plasticker. (2018). *Real Time Price List.* [Online] Available at: http://plasticker.de/preise/pms_en.php?show=ok&make=ok&aog=A&kat=Mahlgut [Accessed: 01 Mar. 2018]

Rehman, M. (2017). Fan Blade. [online] GrabCAD. Available at: https://grabcad.com/library/fan-blade-35 [Accessed 28 Mar. 2018].

REUK. (2018). Giromill Darrieus Wind Turbines. [online] Available at: http://www.reuk.co.uk/wordpress/wind/giromill-darrieus-wind-turbines/ [Accessed 28 Mar. 2018].

Shigley, J.E., 1986. *Mechanical engineering design* First metric., New York ; London: McGraw-Hill.

Siemens, 2016. *How to conduct a failure modes and effects analysis*. [Online]. [Accessed 1 April 2018]. Available from: https://polarion.plm.automation.siemens.com/hubfs/Docs/Guides_and_Manuals/Siemens-PLM-Polarion-How-to-conduct-a-failure-modes-and-effects-analysis-FMEA-wp-60071-A3.pdf

Stein, U. 2018. Micro Wind Turbine. MECE09035 Mechanical Engineering Design 3. University of Edinburgh.

The Descriptive Geometry of Nose Cones. (2011). 1st ed. [ebook] Gary A. Crowell Sr., p.2. Available at: https://en.wikipedia.org/wiki/Nose_cone_design [Accessed 27 Mar. 2018].

TPS63036 High-Efficiency Single Inductor Buck-Boost Converter in Tiny WCSP. (2015). 2nd ed. [ebook] Texas Instruments, p.1. Available at: http://www.ti.com/lit/ds/symlink/tps63036.pdf [Accessed 5 Apr. 2018].

Wikidot. (2007). Types of Wind Turbines and Associated Advantages. [online] Available at: http://me1065.wikidot.com/types-of-wind-turbines-and-associated-advantages [Accessed 28 Mar. 2018].

Windy Nation. (2010). Sizing Your Wind Turbine Tail. [online] Available at: https://www.windynation.com/jzv/inf/wind-turbine-tail-fin-sizing-your-wind-turbine-tail [Accessed 27 Mar. 2018].

Zhi Wu and Hengsheng Wang. (2012). Research on Active Yaw Mechanism of Small Wind Turbines. Energy Procedia. 16 (1), 53-57.

Appendix

Mechanical Engineering Design 3 - Group Project

Project Plan	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Max Benson - Management	CONCEPT											
Maurice Rahme - Electronics				DESIGN								
Yidan Xue - Analysis								MANUFACTURING				
James Yates - Analysis											ASSEMBLY AND TEST	
Murdo Lunn - Design												
Tom Wilson - Design												

Roles

Analysis	Create a full set of calculations for use in report - Given the required power output, dimensions our turbine need
	Gearbox analysis and design
	Produce MATLAB code (discuss with design team)
	Perform fluid mechanics analysis on the design
Design	Come up with useful additional design ideas (how / when / where will it be used)
	Create a detailed CAD design in accordance with the customer requirements and our analysis, label clearly
	Create full set of Engineering drawings, Bill of Materials and Parts list
	Perform Finite Element Analysis on main parts
Management	Create project plan and distribute roles
	Manage the budget and timescale
	Contact suppliers and procurement of parts
Electronics and Controls	Plan the electronics to be used in the turbine
	Create a control system (to include the regulation of voltage), develop code
	Design the charging system (convert from AC to DC)
	Design the generator

Project Plan

Project Manager: Max Benson Start Date: 15/01/2018

Section	Tasks	Start	End	Duration (Days)	% Complete	1 15 - Jan - 18	2 22 - Jan - 18	3 29 - Jan - 18	4 05 - Feb - 18	5 12 - Feb - 18	6 19 - Feb - 18	7 26 - Feb - 18	8 05 - Mar - 18	9 12 - Mar - 18	10 19 - Mar - 18	11 26 - Mar - 18	12 02 - Apr - 18	13 09 - Apr - 18
0	**Overall Project**	15/01/18	09/04/18	84	98%							98						
1	**Concept and Planning**	15/01/18	22/01/18	7	95%	95												
1.1	Choose turbine type	15/01/18	16/01/18	1	100%	###												
1.2	Basic design ideas	15/01/18	22/01/18	7	100%	###												
1.3	Research	15/01/18	16/01/18	1	100%	###												
1.4	Initial materials ideas	15/01/18	23/01/18	8	100%	###												
1.5	Initial costings	15/01/18	31/01/18	16	100%	###												
1.6	Project plan	15/01/18	30/01/18	15	100%	###												
1.7	Concept Design	18/01/18	22/01/18	4	100%	###												
2	**Design**	22/01/18	05/03/18	42	100%				100									
2.1	Rotor and Nose Cone / Hub	16/01/18	19/02/18	34	100%			10000%										
2.2	Tower, rudder and connection to stand	16/01/18	30/01/18	14	100%		10000%											
2.3	Electronics and Circuitry	16/01/18	30/01/18	14	100%	########												
2.4	Generator	16/01/18	30/01/18	14	100%		###											
2.5	Bill of Materials	16/01/18	31/01/18	15	95%		9500%											
3	**Manufacturing**	05/03/18	19/03/18	14	100%							100						
3.1	Order parts	01/02/18	05/03/18	32	100%					10000%								
3.2	Generator	16/01/18	31/01/18	15	100%					10000%								
3.3	3D print rotor blades	16/01/18	31/01/18	15	100%					10000%								
3.4	3D print nose cone / hub	16/01/18	31/01/18	15	100%					10000%								
3.5	Laser Cut rudder sails	16/01/18	31/01/18	15	100%					10000%								
3.6	3D print rudder support pieces	16/01/18	31/01/18	15	100%					10000%								
3.7	Steel rods - aquire and thread	16/01/18	31/01/18	15	100%					10000%								
3.8	Nacelle	16/01/18	31/01/18	15	100%					10000%								
3.9	Tower / electronics housing	16/01/18	31/01/18	15	100%					10000%								
4	**Assembly and Testing**	19/03/18	09/04/18	21	66%											66		
4.1	Quality control of parts	19/03/18	09/04/18	21	100%					10000%								

4.2	Assembly of turbine	19/03/18	09/04/18	21	98%	9800%
4.3	Testing of turbine	19/03/18	09/04/18	21	0%	0%
5	**Report**	01/03/18	06/04/18	36	90%	90
5.1	Introduction	01/03/18	06/04/18	36	80%	8000%
5.2	Approach to the design	01/03/18	06/04/18	36	100%	10000%
5.3	Each person's section	01/03/18	06/04/18	36	85%	8500%
6	**Oral Examination**			0	0%	0
				0	0%	
7	**Group Presentation**	07/03/18	27/03/18	20	100%	###

BOM

Part	Q	Price	Manufacture	Ordered?	Dependent	Notes		Received
10x10x3mm N42 Magnet	16	£ 7.92	Order	15/02/2018	N/A	Price is for the 16 used, not the 20 ordered		22/02/2018
Arduino Uno / Nano	1	£ 8.49	Given	N/A	N/A		£8.49	27/02/2018
Buck-boost converter	1	£ 8.00	Given	N/A	N/A	Price unconfirmed (not conservative)		15/02/2018
Turbine blade	3	£ -	Manufacture	N/A	N/A			15/03/2018
Nosecone / Hub	1	£ -	Manufacture	N/A	N/A	Awaiting quality 3D print		
Rudder Sails	3	£ -	Manufacture	N/A	N/A			05/03/2018
Tail Fin front connection	1	£ -	Manufacture	N/A	N/A			06/03/2018
Tail Fin back connection	1	£ -	Manufacture	N/A	N/A			06/03/2018
Nacelle 3D parts	3	£ -	Manufacture	N/A	N/A	Awaiting quality 3D print		
Nacelle Laser parts	10	£ -	Manufacture	N/A	N/A			09/03/2018
Nacelle / Tower Connection parts	2	£ -	Manufacture	N/A	N/A			
Steel rod - shaft	1	£ -	Given	N/A	N/A			15/03/2018
Steel rod - rudder arm	1	£ -	Given / Machir	N/A	N/A			15/03/2018
Nacelle	1	£ -	Manufacture	N/A	Generator	Awaiting generator assembly and nacelle 3D parts		
Arduino / LCD / Electronics Housing	1	£ -	Manufacture	N/A	Tom designing	Awaiting finalised design		
Steel rotors	2	£ 8.00	Order	05/03/2018	N/A	Acquire actual costing for these from Uwe		15/03/2018
Tower	1	£ -	Manufacture		Tom designing	Awaiting finalised design		
OLED screen	1	£ 6.50	Order	05/03/2018	N/A			12/03/2018
USB sockets	2	£ 4.20	Order?	05/03/2018	N/A			15/03/2018
SLIP RING - Paid by MAX	1	£ 7.49	Order	05/03/2018	N/A			07/03/2018
Travel Case?!	1	£ -	Manufacture		Budget			
Super capacitor	1	£ 2.00	Order	05/03/2018	N/A	Maybe not used? £2.00		15/03/2018
Load dump	1	£ -	Given	N/A	N/A	Resistors from Uni		15/03/2018
626-ZZ Bearing (6mm)	2	£ 4.40	Order	05/03/2018	N/A			15/03/2018
61802-ZZ Bearing (15mm)	1	£ 5.62	Order	05/03/2018	N/A	Might not have been ordered		15/03/2018
Black acrylic sheet - Paid by MAX	1	£ 2.85	Order	05/03/2018	N/A			
Copper wire (0.28mm)	1	£ -	Given	15/03/2018	Uni suppliers	Hopefully arrives by 19/03/18		
Epoxy resin (casting)	1	£ -	Given			Hopefully arrives by 19/03/18		
EPOXY - Paid by MAX	1	£ -	Order	05/03/2018	N/A		£4.15	
5V Relay	1	£ -	Order	15/03/2018				
Schottky Diodes	6	£ 1.62	Order	15/03/2018				
Another super capacitor	1	£ 6.64	Order	15/03/2018				
Logic Level MOSFETs	2	£ 2.00	E Lab Order					
Power Resistors (5W each)	6	£ 2.18	E Lab Order					
Schmitt Trigger	1	£ 0.20	E Lab Order					

Fuse + Holder		£ 1.00	E Lab Order
Zener Diode		£ 0.29	E Lab Order
INA219 Current Sensor		£ 1.42	E Lab Order
DHT11 Temp/Humidity Sensor		£ -	
10k resistors	4	£ -	Lab
680uF Capacitor	1	£ -	Lab
Heat Sinks	2	£ -	Lab
Insulated wire	16	£ -	Lab
Mounting Pins	6	£ -	Lab
Soldering Wire Roll	1	£ -	Lab
Budget		£ 100.00	
Total Cost		£ 80.82	
Budget Remaining		£ 19.18	
£10 Budget		£ 10.34	

£ 65.95 --> Electrical

The Iterative Design Process

In designing the wind turbine, every constituent part and sub-assembly went through the 'Iterative Design Process', to create the best possible design. Furthermore, as every part in the turbine is related to another one, the whole team was usually involved in the development of a part. Initial concepts and designs would be created by the design team and passed onto the analysis team for analysing. Recommendations for further part changes would be made and the designing would implement these into a new design, or iteration. This ensured that every part was the best it could possibly be.

Throughout the report, examples of the iterative design process are can be seen for each part, as every part underwent at least 3 major design changes. Because of this, our design is the most refined.

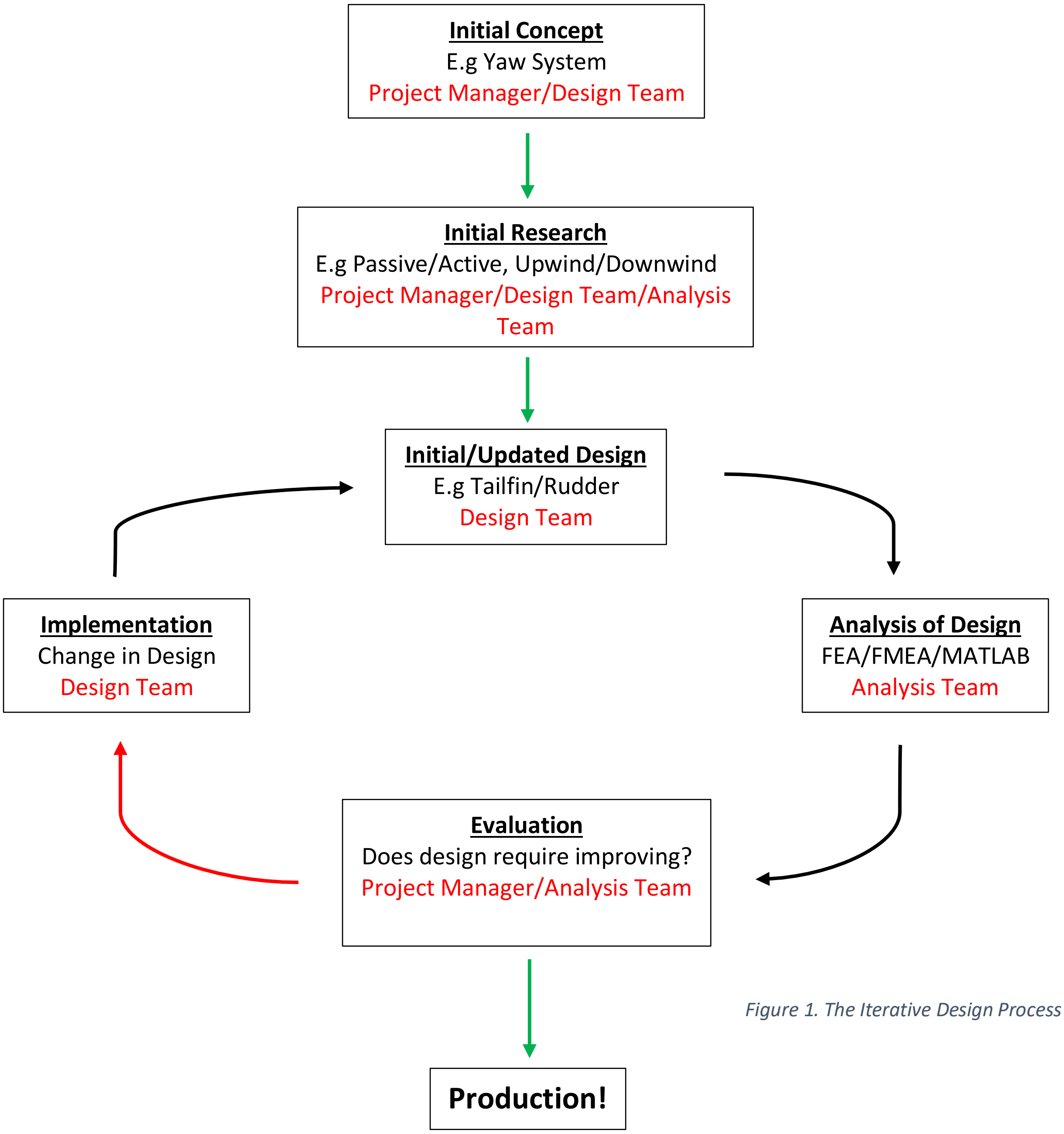
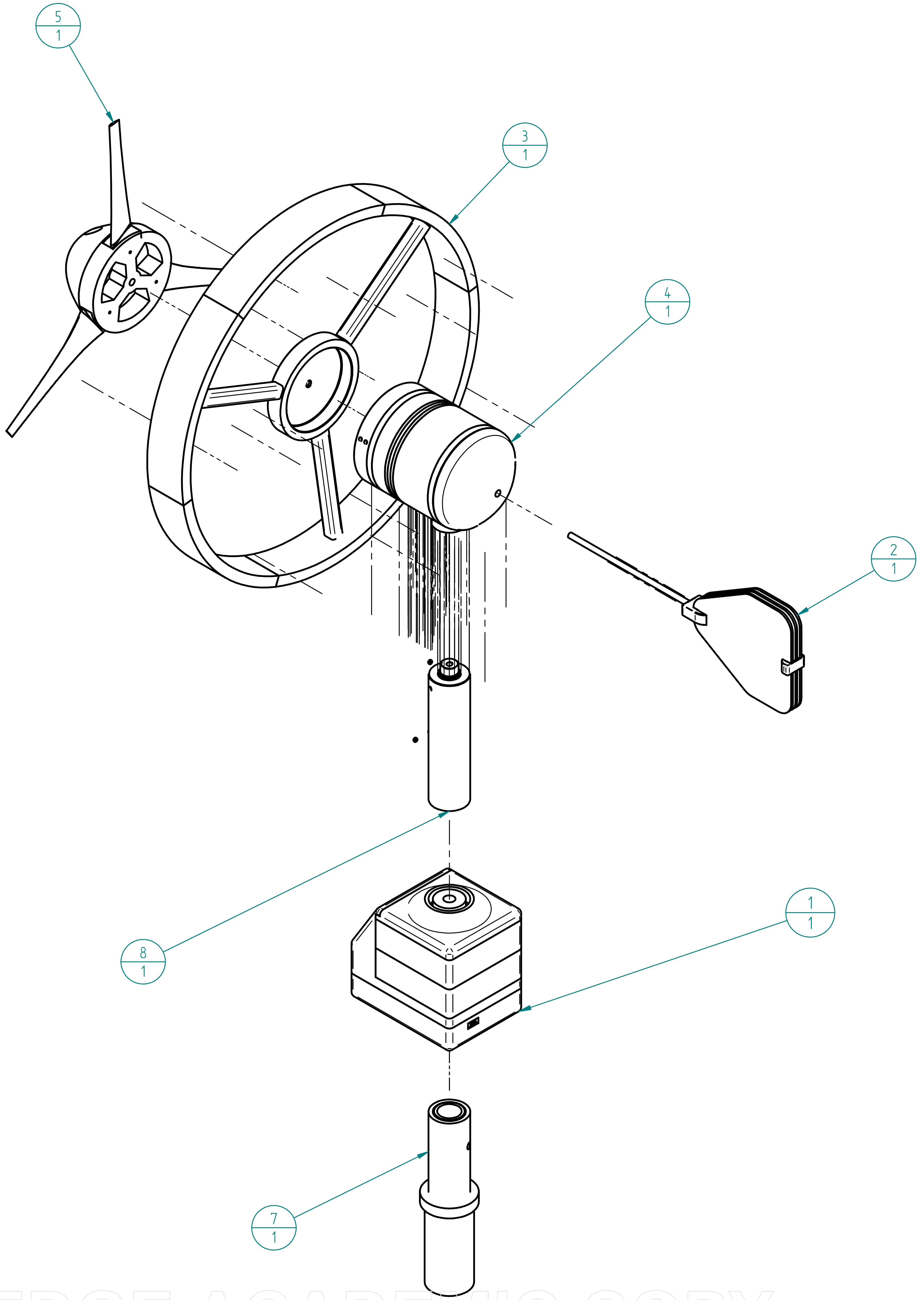


Figure 1. The Iterative Design Process

REVISION HISTORY			
REV	DESCRIPTION	DATE	APPROVED



Item Number	File Name (no extension)	Author	Quantity
1	Electrics Box	s1428622	1
2	Rudder Assembly	maxbe	1
3	Shroud Assembly	s1521312	1
4	Nacelle Assembly	James Yates	1
5	Nosecone Assembly	Tom	1
6*	Rotor_Assy	s1428622	2
7	Bottom Tower and Stand	Tom	1
8	Upper Tower	Tom	1

	NAME	DATE
DRAWN	Tom Wilson	04/05/18
CHECKED	Murdo Lunn	
ENG APPR		
MGR APPR		

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

Solid Edge

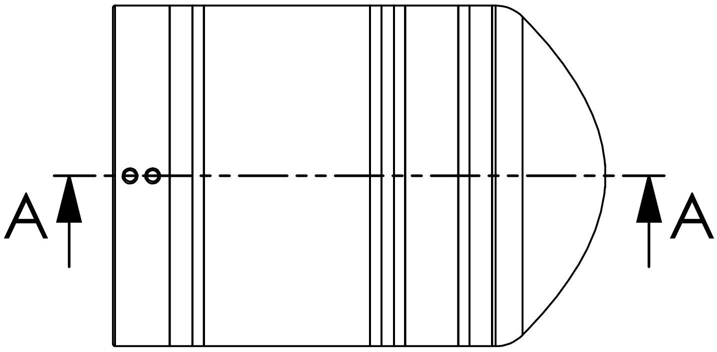
TITLE
Full Exploded Assembly

SIZE	DWG NO	REV
A2	1	1

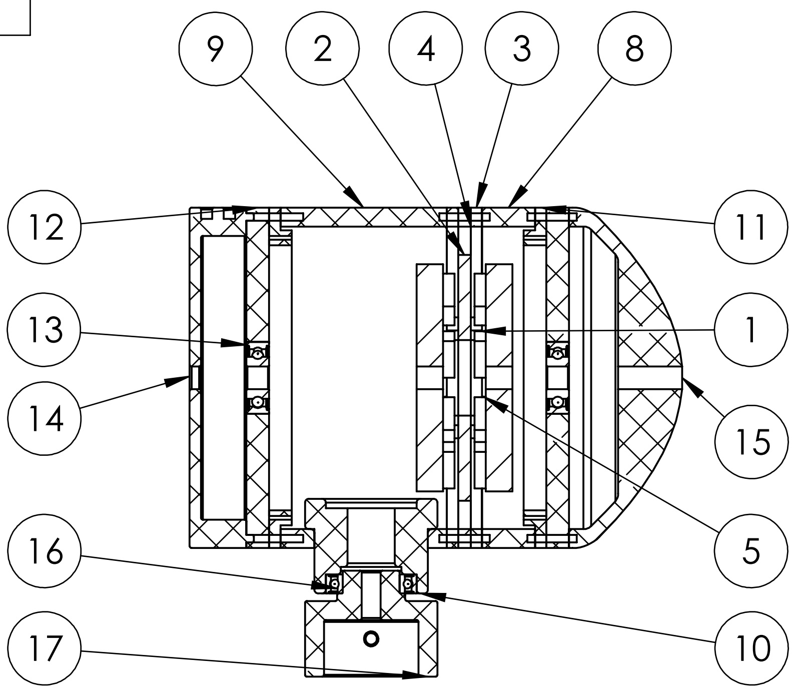
FILE NAME: SE3_Full_Assembly_3 (for introduction video).dft

SCALE: 0.35	WEIGHT:	SHEET 1 OF 1

ITEM NO.	PART NUMBER	QTY.
1	Rotor	2
2	Stator	1
3	Stator Mount	2
4	Centre	1
5	M3 Washer	8
6	M3 Nut	4
7	M3x14mm	4
8	End Casing	1
9	Start Casing	1
10	Tower Mount	1
11	Connection	2
12	Bearing Mount	2
13	626-2Z Bearing	2
14	Hub Cap	1
15	Rudder Cap	1
16	61802-2Z Bearing	1
17	Tower Attachment	1



Top View



Section A-A

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES:
LINEAR: 0.01mm
ANGULAR: 0.1 Degrees

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION: 1

	NAME	SIGNATURE	DATE
DRAWN	James Yates		02/04/18
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

SE3 Nacelle Assembly

MATERIAL: AS PER COMPONENT

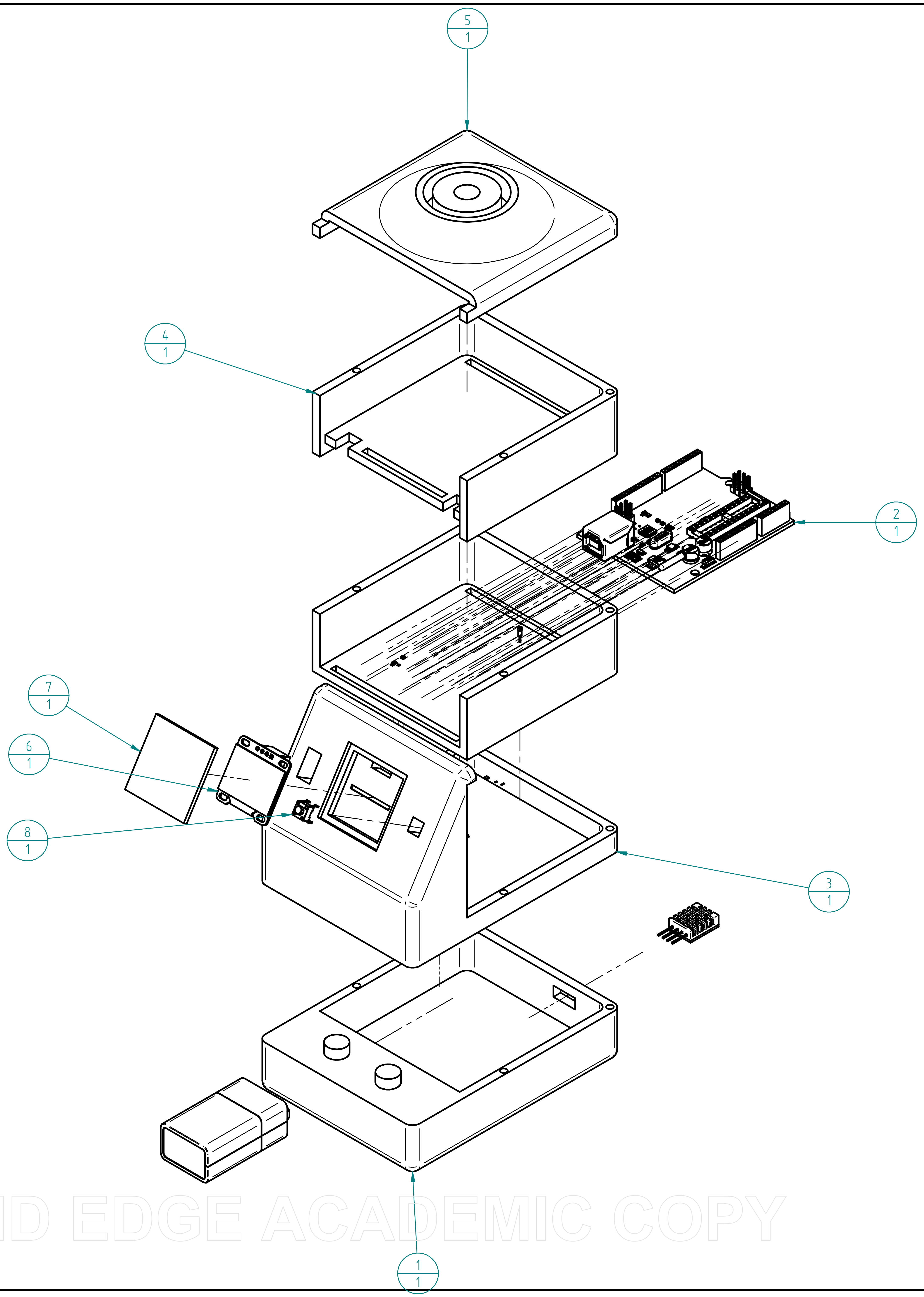
DWG NO. 1

A4

WEIGHT:

Scale: 1:2

SHEET 1 OF 1



REVISION HISTORY			
REV	DESCRIPTION	DATE	APPROVED

Item Number	File Name (no extension)	Author	Quantity
1	Bottom Electrics Box with Batteries and Sensor	Tom Wilson	1
2	Arduino Shelf + Arduino	Tom Wilson	1
3	Screen Housing	Tom Wilson	1
4	Circuitry Shelf	Tom Wilson	1
5	Cover of Box	Tom Wilson	1
6	OLED Screen	Tom Wilson	1
7	OLED Window	Tom Wilson	1
8	Button	Tom Wilson	1

	NAME	DATE
DRAWN	Tom Wilson	04/05/18
CHECKED	Murdo Lunn	
ENG APPR		
MGR APPR		

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

Solid Edge

TITLE
Electronics Box

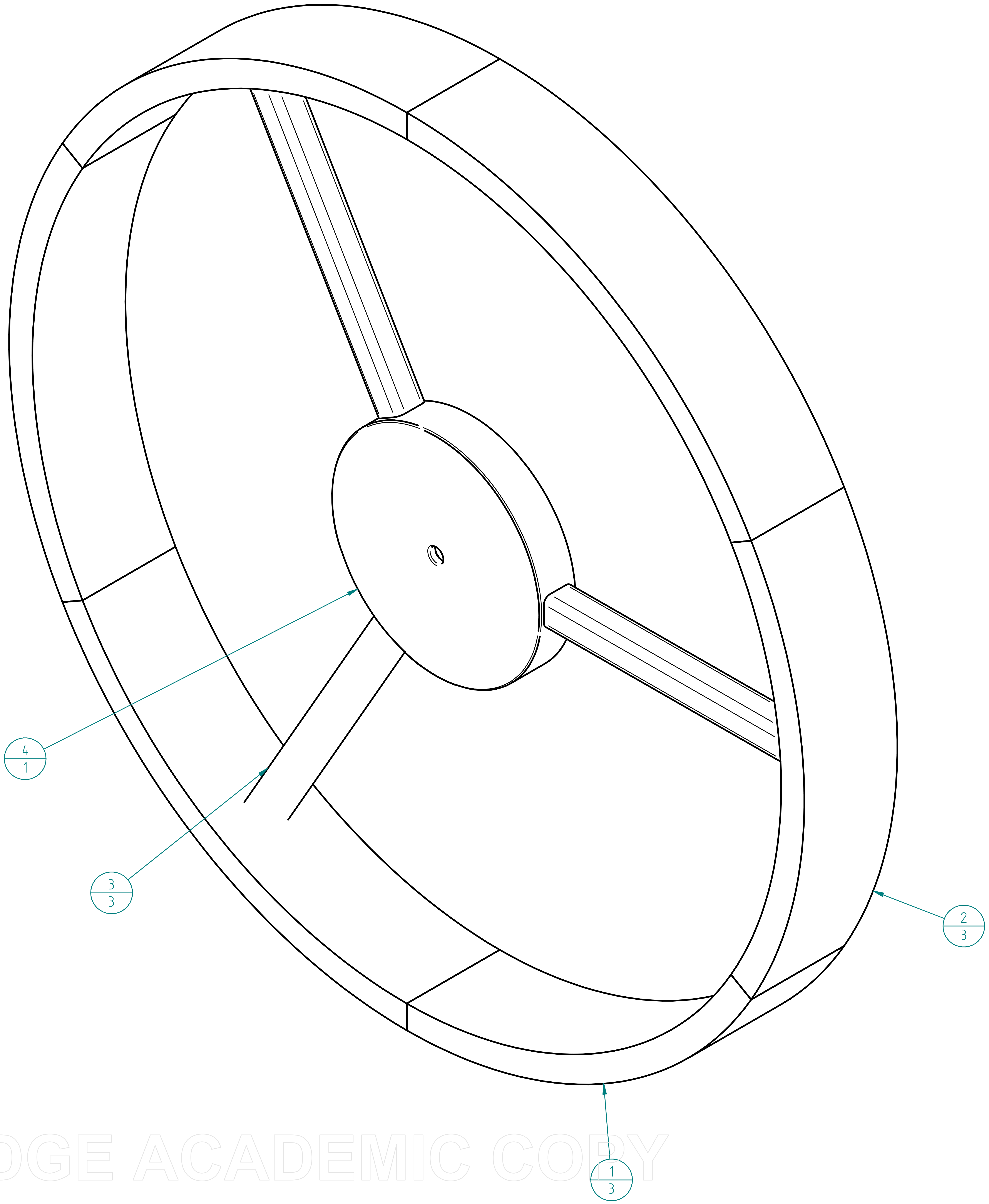
SIZE A2 | DWG NO | REV

FILE NAME: SE3_ElectricsBoxAssembly_4_C.dft

SCALE: 1.25:1 | WEIGHT: | SHEET 1 OF 1

REVISION HISTORY			
REV	DESCRIPTION	DATE	APPROVED

Item Number	File Name (no extension)	Author	Quantity
1	Shroud part 1	Murdo Lunn	3
2	Shroud part 2	Murdo Lunn	3
3	Shroud Connection	Murdo Lunn	3
4	Nacelle Cap	James Yates	1



	NAME	DATE
DRAWN	Tom Wilson	04/05/18
CHECKED	Murdo Lunn	
ENG APPR		
MGR APPR		

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX

Solid Edge

TITLE
Shroud Assembly

SIZE	DWG NO	REV
A2	1	1

FILE NAME: SE3_Shroud_1_a.dft

SCALE: 1:1 | WEIGHT: | SHEET 1 OF 1

FMEA for Micro Wind Turbine									
Sustainable Energy Design 3					Group 3		28/03/2018		
Item	Function	Potential Failure Mode	Potential Effects of	v r t	Potential Causes of	c r n	Current Controls for	t c t	R P N
1	Blade	Broken in the middle	The product cannot continue working. It may also cause injuries	9	The strength of material (3D printing) is not enough	2	Epoxy	3	54
2	Blade	Broken at the connection with nosecone	The product continue working. It may also cause injuries.	9	The connection is thin and the moment there is large	3	Epoxy	4	108
3	Nosecone	Loose at the connection with the shaft	Cannot transport torque to the main shaft	5	The friction is not enough at the connection between nosecone and main shaft	2	Interference fit	2	20
4	Nosecone	Broken	The product cannot continue working	9	The strength of material (3D printing) is not enough	3	Epoxy it and increase the thickness of surface when 3D printing	3	81
5	Nosecone	Screw falling out	The blade will fall out from the connection hole	8	Screw is not tightened or the hole is too big	3		2	48
6	Main shaft	Broken	The torque cannot be transported	7	The stress at some point is too high	1	High safety factor	2	14
7	Main shaft	Falling out of the bearing	There will be vibrations and the product cannot work well	5	The friction is not enough between the bearing and the shaft	1	Interference fit	1	5
8	Bearings	Broken(supporting the shaft)	The shaft will vibrate and it may damage other parts	6	Cannot stand the stress	1		2	12
9	Bearings	Broken(between nacelle and tower)	The whole product will fall down	7	The bearing is not strong enough	1		2	14
10	Tail	Back tail link falling out	The two tail will no longer be parallel with each other	2	The friction is not enough	2	Interference fit	3	12
11	Tail	Front tail link falling out	The tail will fall to the ground	5	The friction is not enough	2	Interference fit	3	30
12	Tail	Tail broken	The product will no longer face the wind automatically	6	The material is not strong enough	1		2	12
13	Tail	Rod broken	The product will no longer face the wind automatically	6	The stress is too high	1		2	12
14	Tail	Rod fall out from the nacelle hole	The tail will fall to the ground	5	The friction is not enough	1	Using thread to connect	2	10

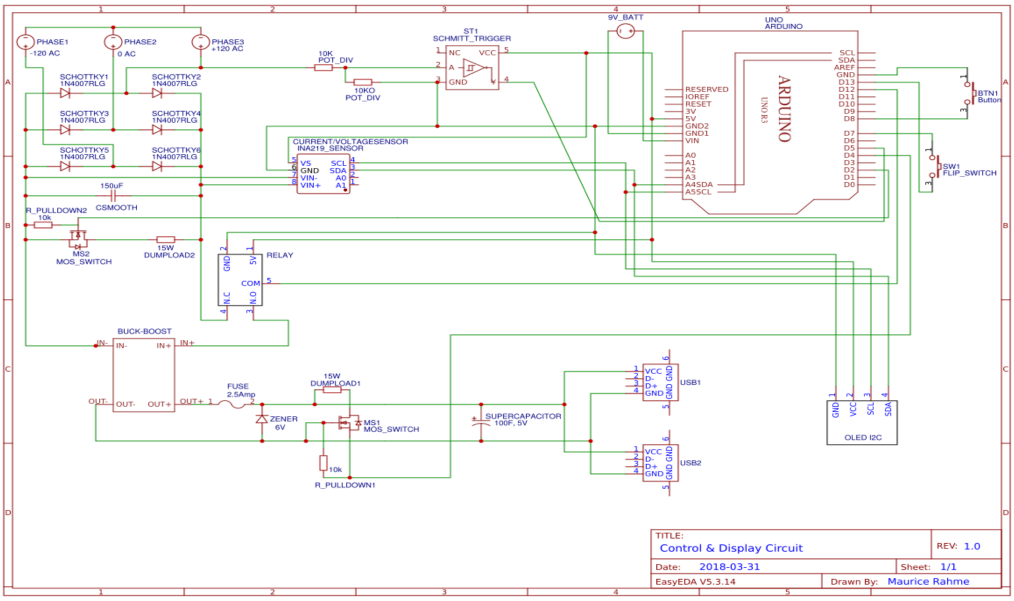
15	Nacelle	Front cover touch the shaft	Noise and wear at front cover	3	The hole is not big enough or there is a vibration of shaft	1		1	3
16	Nacelle	Back cover broken (connection with the tail rod)	The tail will fall to the ground	6	The strength of material (3D printing) is not enough	2	Increasing moment of inertia for the cross section	2	24
17	Nacelle	Back cover broken (connection with main nacelle body)	The tail will fall to the ground	6	The strength of material (3D printing) is not enough	2	Increasing moment of inertia for the cross section	2	24
18	Nacelle	Broken at the layer connection	The nacelle will break	7	The connection is not strong enough	1	Epoxy	2	14
19	Nacelle	Generator stator falls	Power cannot be generated	7	The connection is not strong enough	1	Using four screws to fasten it	1	7
20	Nacelle	Shaft supporting layer broken	The main body will break	7	The strength of material is not enough	1		1	7
21	Nacelle	Falls from the tower adaptor	The product will break	8	The connection is not strong enough	1		2	16
22	Tower	Broken at the connection with stand	The product will fall down or rotate	6	The strength of material is not enough	1		2	12
23	Tower	Screen falls out from the slot	Customers need to put it back	2	The connection is not strong enough	1		1	2
24	Tower	Connection breaks	The product will fall	8	The material is not strong enough/The screws fall out from the holes	3	Epoxy	2	48
25	Shroud	Connection broken at the shroud	Shroud will fall	5	The strength of material (3D printing) is not enough	2	Epoxy	3	30
26	Shroud	Connection broken at the cap	Shroud will fall	5	The strength of material (3D printing) is not enough	4	Epoxy	3	60
27	Generator rotor	cannot keep pace with the main shaft	The power generated will not be sufficient	5	Cannot connect tightly with the shaft	3	Interference fit	2	30
28	Generator rotor	Magnetics falls out from the slots	The power generated will not be sufficient	5	The glue is not strong enough	1		1	5
29	Generator stator	The coils melt	The generator will dramatically lose efficiency	9	Excess current through the windings	1		1	9

30	Generator stator	The generator phases touch	The generator will stop supplying power until the phases are uncoupled	9	Insufficient/deteriorating insulation or improper contacting	3	Electrical tape	2	54
31	Generator (general)	Uncontrolled Back-EMF	The generator will spin too fast and potentially break	7	A combination of excessively high wind speed and low loading	1		1	7
32	Circuit	water infiltration	Damage to circuit/short circuit	8	USB ports, or push button/toggle switch having poor elemental protection (water going through epoxy)	2	Epoxy to seal each possible holes	5	80
33	Circuit	Control doesn't actuate	Will not be able to read and display generator RPM/ assume below *test value*	4	Input phase for output square wave generation into display and control below 0.4V threshold	4		1	16
34	Circuit	Connections are broken/weakened	Some or all aspects of the now open circuit will not function	6	A combination of bad soldering and knocking the components around	1	Test the circuit before using	7	42
35	Software/Digital	Arduino fails due to RAM limitation	Code will glitch/not work	6	Small amount of SRAM in arduino (32KB)	1		1	6
36	Software/Digital	Generator start-up function fails	Generator will struggle to produce torque at high initial load	4	Improper initial setup function (solution: set SSR1 OFF as part of sleep function)	1		4	16

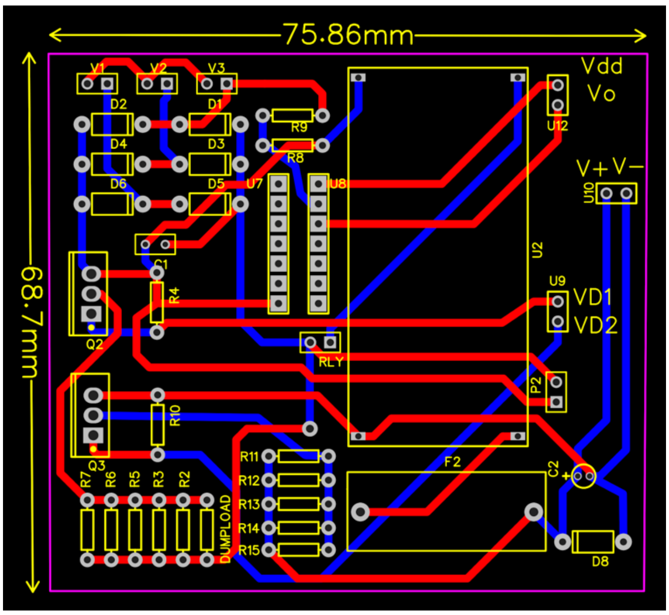
Turbine Design Matrix for Passive Yaw System

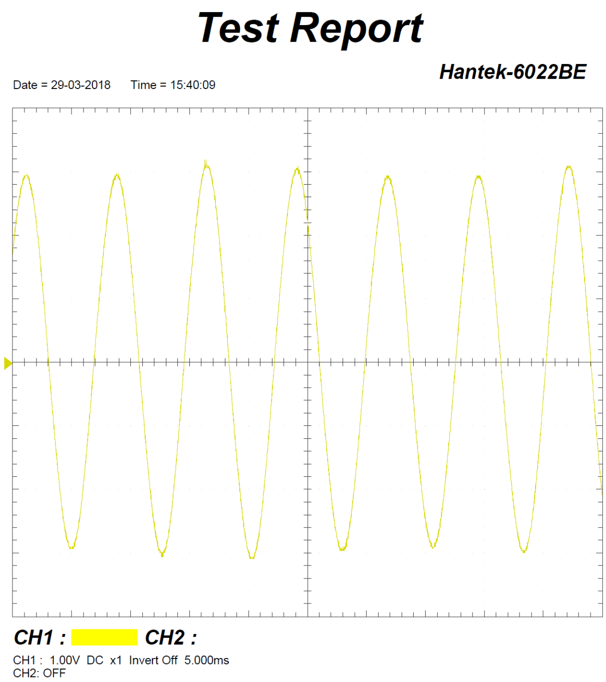
Factor	Weighting	Upwind	Downwind
Performance	6	1	0
Resistance to 'wind shade'	5	1	0
Ease of Design	4	0	1
Ease of manufacture	3	1	0
Cost	2	0	1
Compactness	1	1	0
Total	21	15	6

Circuits

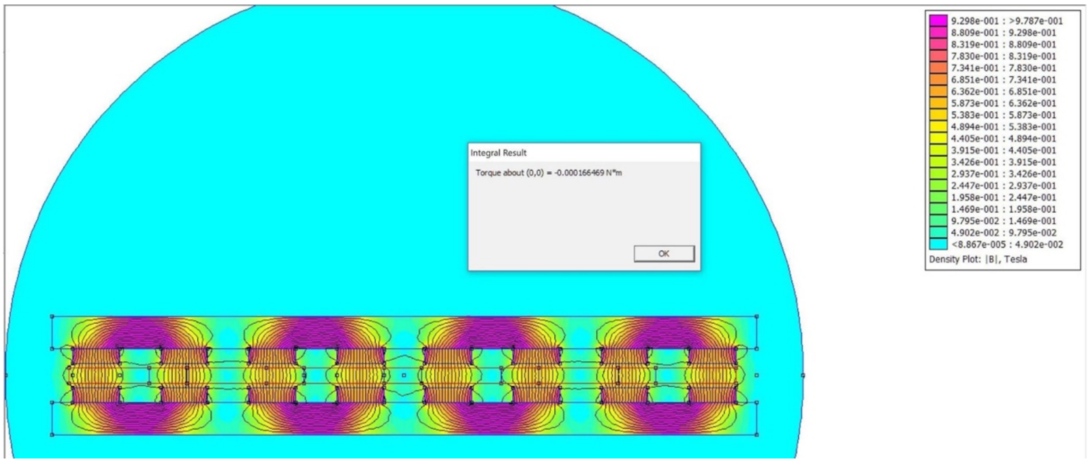


Circuits 2

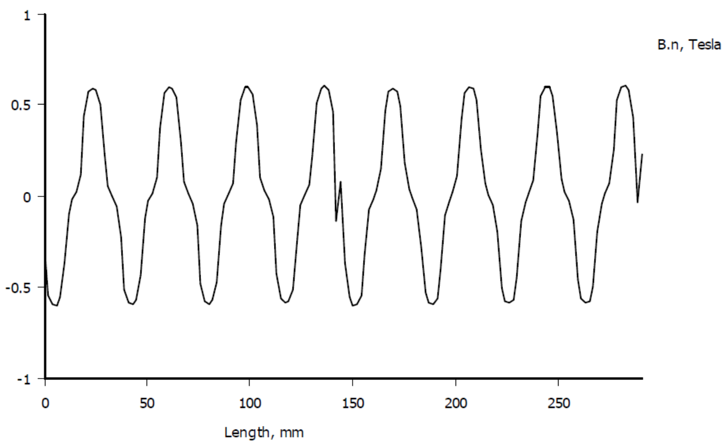




Flux Density with Torque



Flux Curve



```
% Generator_Calc
% Generator Design Calculations
% Maurice Rahme - 28/01/2018
clc; clear all;

%% Input Variables -------------------------------------------------------
%Basic
P_turb = 15; % Power generated at Turbine [W]
n_eff = 0.9; % Generator Efficiency

% Generator Voltage Range (2.1 Battery)
V_batt = 30; % Battery Voltage - according to v.regulator (4V-36V)

% Generator Frequency, Pole Pairs and Coil Number Per Phase (3)
f_nom = 400; % Generator Nominal Frequency [Hz]
n_nom = 6000; % Nominal RPM

% Generator Axial Dimensions
g = 2/1000; % mechanical clearance gap [m] (1mm for resin over mag, 1mm for resin over coil, 1 mm mechanical gap)
B_r = (1280+1320)/2000; % remanent magnetic flux density [T] - N42
B_mg = B_r/2; % Magnetic flux density on magnet surface [T] half of B_r
mew_0 = 1.257*10^(-6); % vacuum permeability [Wb/(A*m)]
H_c = 915*10^(15); % corrective field strength [pA/m]
h_m = 5/1000; % permanent magnet thickness/height [m] - see Figure 3 page 8 PDF
rk_sat = 1; % saturation factor =1 since coreless stator

% Generator Winding Dimensions
w_m = 10/1000; % permanent magnet radial width [m]
l_a = 10/1000; % active/magnet length [m] - see Figure 4 page 8 PDF
rk_w = 0.95; % winding coefficient
n_cutin = 4000; % Cut-in RPM
C_q = 0.3; % heat coefficient [W/cm^2] - page 9 PDF
rho_cu = 1.72*10^(-8); % Electrical resistivity of Copper [Ohm m]
rk_f = 0.65; % Coil fill factor - between 0.55-0.78 - see page 9 PDF!!!1 IMPORTANT

% Generator Terminal Voltage
L_s = 0; % Generator inductance [mH]
R_c = 0; % Coil Resistance [Ohm]

% Generator Outer Radius (already included)

%% Generator Voltage Range (2.1 Battery) ----------------------------------
P_nom = P_turb*n_eff; % Nominal Generator Power [W]
V_ac = (V_batt+1.4)/(sqrt(3)*1.35);
E_fcutin = V_ac; % Cut-in EMF [V]
E_fnom = E_fcutin*(n_nom/n_cutin); % Nominal EMF [V]

%% Generator Frequency, Pole Pairs and Coil Number Per Phase (3) ----------
p = 120*(f_nom/n_nom); % Number of pole pairs
Q = (3/4)*p; % Total number of coils
q = Q/3; % number of coils per phase
n_m = (4/3)*Q; % Number of magnets per rotor

%% Generator Axial Dimensions (4) -----------------------------------------
mew_rrec = B_r/(mew_0*H_c); % Recoil Permeability
%%mew_rrec = 1.05*10^(-6);
%%t_w = 2*(((B_r-B_mg)*h_m)/((mew_rrec*B_mg*rk_sat))-g); % Stator Axial Thickness [m]
%%t_w = 0.1;
t_w = 2*((h_m/(mew_rrec*rk_sat))-g);
%%t_w = ((2*h_m)/(rk_sat*mew_rrec))*((B_r/B_mg)-1)-(2*g);
h_r = h_m;% page 7 of PDF - READ - back iron thickness [m]

%% Generator Winding Dimensions (5) ---------------------------------------
I_acmax = (1.1*P_nom)/(3*E_fnom*n_eff); % Maximum Generator AC current [A]
phi_max = B_mg*w_m*l_a; % Max Magnetic Flux per pole (Wb)
N_c = (sqrt(2)*E_fcutin)/(q*2*pi*rk_w*phi_max*n_cutin*(p/120)); % Number of turns per coil
w_c = (I_acmax*N_c)/sqrt((2*C_q*rk_f*t_w)/rho_cu); % coil leg/side width [m]
s_c = (rk_f*w_c*t_w)/N_c; % Copper cross section [mm^2]
J_max = I_acmax/s_c; % Maximum current density [A/mm^2] - around 6A/mm^2?
d_c = sqrt((4*s_c)/pi); % copper diameter [mm]

%% Generator Terminal Voltage (5) -----------------------------------------
I_rms = I_acmax/sqrt(2); % Generator RMS Current [A]
delta = asin((I_rms*2*pi*f_nom*L_s)/E_fnom);
V_t = E_fnom*cos(delta)-I_rms*q*R_c; % Terminal Voltage [V]

%% Generator Outer Radius (6) ---------------------------------------------
R_in = (((2*Q*w_c)+(p*w_m))/(2*pi))*1000; % Generator Inner Diameter [mm]
R_out = R_in + l_a*1000; % Generator Outer Diameter [mm]

%% Display Results -------------------------------------------------------
Array_Basics = [R_in,R_out,V_t,d_c,s_c,N_c,w_c,n_m,Q,q,h_r,t_w,P_nom,...
        J_max,I_acmax,E_fnom];
Basics = array2table(Array_Basics,'VariableNames',{'Inner_R_mm','Outer_R_mm',...
            'Terminal_V','Copper_D_mm','Copper_X_Sect_mm2',...
            'Turns_per_coil','Coil_Leg_Side_Width_m',...
            'Magnets_per_Rotor','Number_of_Coils',...
            'Coils_Per_Phase','Back_Iron_Thickness_m',...
            'Stator_thickness_m','Nominal_Generator_Power',...
            'Max_Current_Density','Max_AC_Current',...
            'Nominal_Generator_Voltage'});
disp(Basics)
```

```
% Nosecone Design using Power series
% Tom Wilson and Murdo Lunn 29/1/18

clear all;
close all;
clc;

% Variable dictionary
%
% R         hub radius [m]
% L         length of nose cone [m]
% n         power serirs coefficients
% x         coordinate that describes the position along the centre line
% y         vertical distance from the centre line

% Values of parameters

R = 0.045;
L = 0.045;
n1 = 0.3;
n2 = 0.4;
n3 = 0.5;
n4 = 0.6;
n5 = 0.7;

% Set up x vector
x = linspace(0,L,1000);

% Power series calculations

y1 = R.*((x./L).^n1);
y2 = R.*((x./L).^n2);
y3 = R.*((x./L).^n3);
y4 = R.*((x./L).^n4);
y5 = R.*((x./L).^n5);

% Plot of nosecone shapes

plot(x,y1,'b',x,y2,'r',x,y3,'g',x,y4,'black',x,y5,'y'), xlabel('length of nosecone (m)'), ylabel('Radius of Hub'), title('Nose cone designs')

% Transpose row vectors for x and y into column vectors so that tables can
% be created

X=x';
Y1=y1';
Y2=y2';
Y3=y3';
Y4=y4';
Y5=y5';

% Create tables for x and y coordinates in nose cone profile

T1=table(X,Y1);
T2=table(X,Y2);
T3=table(X,Y3);
T4=table(X,Y4);
T5=table(X,Y5);

% Create excel sheet for each nose cone profile

filename = 'Nose_cone_designs.xlsx';
writetable(T1,filename,'Sheet',1,'Range','B2')
writetable(T2,filename,'Sheet',2,'Range','B2')
writetable(T3,filename,'Sheet',3,'Range','B2')
writetable(T4,filename,'Sheet',4,'Range','B2')
writetable(T5,filename,'Sheet',5,'Range','B2')
```

```
% Profile_Selection
% Determining Optimum Profiles
% James Yates - 25/01/2018
clear all; close all; clc;

%% Input Value -----------------------------------------------------------
Nu = 50; % Number of Profiles
voltage = 5; % Volatge [V]
current = 2.1; % Current [A]
Arduino = 1.5; % Arduino Power [W]
v_wind = 12; % Wind Speed [m/s]
rho = 1.22; % Air Density [kg/m^3]
mu = 1.983E-5; % Dynamic Viscosity of Air [Ns/m^2]
Eff = 0.6*0.95*0.85; % Efficiency of Gear / Generator
CP = 0.3; % Coefficient of Performance
TSR = 55; % Tip Speed Ratio
R_Hub = 3; % Hub Radius [cm]
Theta = linspace(-10,10,200); % Range of Attack Angles [Degrees]

%% Power Calculation -----------------------------------------------------
P = current*voltage + Arduino; % Power [W]

%% Radius Calculation ----------------------------------------------------
A = 2*P/(Eff*CP*rho*v_wind^3); % Blade Area [m^2]
R_1 = sqrt(A/pi); % Tip Radius [m]

%% Round Turbine Radius --------------------------------------------------
Round = round(100*R_1);
if (100*R_1 - Round) < 0
   R = Round/100;
else
   R = (Round + 0.5)/100;
end

%% Torque Calculation ----------------------------------------------------
v_tip = TSR*v_wind; % Tip Speed [m/s]
Omega = v_tip/R; % Angular Velocity [rad/s]
N = 60*Omega/(2*pi); % Rotation Rate [RPM]
torque = P/Omega; % Net Torque [Nm]

%% Reynold's Number ------------------------------------------------------
r = linspace(R_Hub/100,R,Nu);
D = 2.*r;
Re = rho*v_wind*D/mu;

%% Display Results -------------------------------------------------------
Array_Basics = [voltage,current,P,v_wind,Eff,CP,TSR,100*R,v_tip,Omega,...
          N,torque];
Basics = array2table(Array_Basics,'VariableNames',{'Voltage','Current',...
              'Power','Wind','Eff','CP','TSR','Radius','Tip',...
              'Omega','RPM','Torque'});
disp(Basics)

%% Determining Profile ---------------------------------------------------
% Importing existing Ratio Data
Ra1 = xlsread('Profile Comparison_2.xlsx','Re - 50,000');
Ra2 = xlsread('Profile Comparison_2.xlsx','Re - 100,000');
Ra3 = xlsread('Profile Comparison_2.xlsx','Re - 200,000');
Ra4 = xlsread('Profile Comparison_2.xlsx','Re - 500,000');
Profiles = xlsread('Profile Comparison_2.xlsx','Profile Names');

% Setting Initial Vector Values
Location = zeros(size(r));
Ra = zeros(size(r));
A = zeros(size(r));

% Interpolation and Extrapolation to determine optimum NACA Profiles
for t = 1:length(r)
   if Re(t) <= 100000
      X = (Re(t) - 100000)*((Ra2(:,1) - Ra1(:,1))/(100000 - 50000)) +...
         Ra2(:,1);
   elseif Re(t) <= 200000
      X = (Re(t) - 100000)*((Ra3(:,1) - Ra2(:,1))/(200000 - 100000)) +...
         Ra2(:,1);
   else
      X = (Re(t) - 100000)*((Ra4(:,1) - Ra3(:,1))/(200000 - 100000)) +...
         Ra3(:,1);
   end
   [Ra(t),Location(t)] = max(X);
end

Profiles = Profiles(:,1);
Selection = [Profiles(Location)';Ra];

selections = array2table(Selection','VariableNames',{'Profile','Ratio'});
disp(selections)
```

```
%Turbine Shape Optimisation

%This code can provide calculation for twist angle, chord length and
%thickness for each sections. It can also provide relative flow angle,
%axial induction factor and angular induction factor.
%Then it will check the power it can provide automatically.
%This code needs Coefficient Data Table to run.
%This code has limitations of just dividing the turbine into 100 sections
%and starting optimisation with an initial guess of inlet angle and chord
%length.

%Developed by MED3 group 3
%General calculation, curve fitting and coefficients calculation written
%by James Yates, 26 Jan 2018
%Initial guess of variables, iteration section and power checking after optimisation
%written by Yidan Xue, 27 Jan 2018

%% Variable Dictionary

%i   Incidence angle
%cl   Lift coefficient
%cd   Drag coefficient
%beta   Relative flow angle onto blades
%gamma   Twist angle
%c   Chord length
%sigma   Local solidity
%a   Axial induction factor
%a1   Angular induction factor
%t   Torque
%power   %Final power

%% Clearing stuffs
clc; clear all; close all;

%% Parameters

Nu = 100; % Number of Sections

v_wind = 12;   % Wind Speed [m/s]
v_cut = 8; % Cut-In Speed [m/s]
Torque_Req = 0.089*16/1000; % Required Generator Torque [Nm]
P = 12;   % Power [W]
R_Hub = 0.045; % Hub Radius [m]
Eff = 0.6*0.95*0.85; % Efficiency of Gear / Generator...
rho = 1.22; % Air Density [kg/m^3]
mu = 1.983E-5; % Dynamic Viscosity of Air [Ns/m^2]
TSR = 5;   % Tip Speed Ratio
CP = 0.3; % Coefficient of Performance


%% Importing Data and Fitting Curves

% Importing Data
Data_1 = xlsread('Coefficient Data.xlsx','2415, Re - 50,000');
Data_2 = xlsread('Coefficient Data.xlsx','2415, Re - 100,000');
Data_3 = xlsread('Coefficient Data.xlsx','2415, Re - 200,000');
Data_4 = xlsread('Coefficient Data.xlsx','2415, Re - 500,000');
% Fitting Lift Curves
L_1 = fit(Data_1(:,1),Data_1(:,2),'smoothingspline');
L_2 = fit(Data_2(:,1),Data_2(:,2),'smoothingspline');
L_3 = fit(Data_3(:,1),Data_3(:,2),'smoothingspline');
L_4 = fit(Data_4(:,1),Data_4(:,2),'smoothingspline');
% Fitting Drag Curves
D_1 = fit(Data_1(:,1),Data_1(:,3),'smoothingspline');
D_2 = fit(Data_2(:,1),Data_2(:,3),'smoothingspline');
D_3 = fit(Data_3(:,1),Data_3(:,3),'smoothingspline');
D_4 = fit(Data_4(:,1),Data_4(:,3),'smoothingspline');

%% Calculation of the Radius ---------------------------------------------

A = 2*P/(Eff*CP*rho*v_wind^3); % Blade Area [m^2]
R_1 = sqrt(A/pi); % Tip Radius [m]

%Round turbine radius
Round = round(100*R_1);
if (100*R_1 - Round) < 0
   Radius = Round/100;
else
   Radius = (Round + 0.5)/100;
end

Omega = TSR*v_wind/Radius; % Angular Velocity [rad/s]
N = Omega*60/(2*pi); % Angular Velocity [RPM]
Omega_s = TSR*v_cut/Radius; % Cut-In Angular Velocity [rad/s]
N_s = 60*Omega_s/(2*pi); % Cut-In Anglular Velocity [RPM]


%% Initialization --------------------------------------------------------

C = zeros(1,Nu);
Gamma = zeros(1,Nu);
Beta = zeros(1,Nu);
A = zeros(1,Nu);
A1 = zeros(1,Nu);
CL = zeros(1,Nu);
CD = zeros(1,Nu);
R = linspace(R_Hub,Radius,Nu);


%% Calculating Reynold Numbers -------------------------------------------

% Design Speed
Re = rho*v_wind*2*R/mu;
% Cut-In Speed
Re_s = rho*v_cut*2*R/mu;


%% Iteration -------------------------------------------------------------

for iter = 1:Nu

   r = R(iter);
   tsr = TSR*r/Radius;   % Local Speed Ratio

   % Find Max Lift/Drag Ratio and Optimal Angle of Attack
```

```
    I = linspace(0.1,10,100);
    k = 0;
    opt_ratio = 0;
    % Finding Optimal Ratio
    for m = 1:100
        % Get Efficiencies
        i = I(m);

            if Re(iter) <= 100000
                cl = (Re(iter) - 100000)*((L_2(i) - L_1(i))/...
                    (100000 - 50000)) + L_2(i);
                cd = (Re(iter) - 100000)*((D_2(i) - D_1(i))/...
                    (100000 - 50000)) + D_2(i);
            elseif Re(iter) <= 200000
                cl = (Re(iter) - 100000)*((L_3(i) - L_2(i))/...
                    (200000 - 100000)) + L_2(i);
                cd = (Re(iter) - 100000)*((D_3(i) - D_2(i))/...
                    (200000 - 100000)) + D_2(i);
            else
                 cl = (Re(iter) - 200000)*((L_4(i) - L_3(i))/...
                     (500000 - 200000)) + L_3(i);
                 cd = (Re(iter) - 200000)*((D_4(i) - D_3(i))/...
                     (500000 - 200000)) + D_3(i);
            end

        % Lift/Drag Ratio
        ratio = cl/cd;
        % Determining if Optimal
        if ratio > opt_ratio
            opt_ratio = ratio;
            k = m;
            cl_opt = cl;
            cd_opt = cd;
        end
    end

    % Optimal Coefficients
    cl = cl_opt;
    cd = cd_opt;

    % Initial Guesses
    beta = 90 - (2/3)*atand(1/tsr);
    c = (8*pi*r)/(3*cl)*(1-sind(beta));
    sigma = (3*c)/(2*pi*r);
    gamma = beta + I(k);
    a = (1 + (4*(cosd(beta))^2)/...
        (sigma*(cl*sind(beta)+cd*cosd(beta))))^(-1);
    a1 = (1 + (4*cosd(beta)*sind(beta))/...
         (sigma*(cl*sind(beta)-cd*cosd(beta))))^(-1);

    % Iteration
    for j = 1:1:50
        aold = a;
        a1old = a1;
        betaold = beta;
        beta = atand(tsr*(1 + a1)/(1 - a)); % New Beta
        i_angle = gamma - beta;   % New Angle of Attack

        % Correct if Outwith Accurate Range of Curve or Get Close to the Range
        % of Stall
        if i_angle > 10
            i_angle = 10;
            beta = gamma - 10;
        end
        if i_angle < 0
            i_angle = 0;
            beta = gamma;
        end


        % Calculating New Coefficients
        i = i_angle;
        % 2415 Profiles

            if Re(iter) <= 100000
                cl = (Re(iter) - 100000)*((L_2(i) - L_1(i))/...
                    (100000 - 50000)) + L_2(i);
                cd = (Re(iter) - 100000)*((D_2(i) - D_1(i))/...
                    (100000 - 50000)) + D_2(i);
            elseif Re(iter) <= 200000
                cl = (Re(iter) - 100000)*((L_3(i) - L_2(i))/...
                    (200000 - 100000)) + L_2(i);
                cd = (Re(iter) - 100000)*((D_3(i) - D_2(i))/...
                    (200000 - 100000)) + D_2(i);
            else
                 cl = (Re(iter) - 200000)*((L_4(i) - L_3(i))/...
                     (500000 - 200000)) + L_3(i);
                 cd = (Re(iter) - 200000)*((D_4(i) - D_3(i))/...
                     (500000 - 200000)) + D_3(i);
            end

        c = (8*pi*r)/(3*cl)*(1-sind(beta)); % New Chord Length
        sigma = (3*c)/(2*pi*r); % New Solidity
        a = (1 + (4*(cosd(beta))^2)/...
            (sigma*(cl*sind(beta)+cd*cosd(beta))))^(-1); % New axial induction factor
        a1 = (1 + (4*cosd(beta)*sind(beta))/...
             (sigma*(cl*sind(beta)-cd*cosd(beta))))^(-1); % New angular induction factor

        % Testing Approximate Error
        if abs((a1 - a1old)/a1old) <= 0.005 && abs((a - aold)/aold)...
                <= 0.005 && abs((beta - betaold)/betaold) <= 0.005
            Beta(iter) = beta;
            A(iter) = a;
            A1(iter) = a1;
            C(iter) = c;
            Gamma(iter) = gamma;
            CL(iter) = cl;
            CD(iter) = cd;
            break
        end
    end
end

% Angle of Attack
```

```
alpha = Gamma - Beta;


%% Power Calculation -------------------------------------------------------

% Torque Elements
dT = (3/2)*rho*C.*R.*(v_wind^2*(1-A).^2./(cosd(Beta)).^2).*...
    (CL.*cosd(Beta)-CD.*sind(Beta));

% Torque Integration
Elements_T = [R;dT];
T = fit(R',dT','smoothingspline');
A_i = Elements_T(:,1:end-1);
B_i = Elements_T(:,2:end);
K = (B_i(1,:)-A_i(1,:))/6;
M = (A_i(1,:)+B_i(1,:))/2;
Int_T = K.*(A_i(2,:)+4*T(M)'+B_i(2,:));

Torque = sum(Int_T);
Power = Omega*Torque;
Power_output = Power*Eff;
P_wind = 0.5*rho*pi*Radius^2*v_wind^3;
Power_coefficient = Power/P_wind;


%% Torque at 8m/s ----------------------------------------------------------

% Approximating Coefficients
CL_s = zeros(size(R));
CD_s = zeros(size(R));

for k = 1:length(R)
   r = R(k);
   if Re_s(k) <= 100000
      CL_s(k) = (Re_s(k) - 100000)*...
            ((L_2(alpha(k)) - L_1(alpha(k)))/...
            (100000 - 50000)) + L_2(alpha(k));
      CD_s(k) = (Re_s(k) - 100000)*...
            ((D_2(alpha(k)) - D_1(alpha(k)))/...
            (100000 - 50000)) + D_2(alpha(k));
   elseif Re_s(k) <= 200000
         CL_s(k) = (Re_s(k) - 100000)*...
               ((L_3(alpha(k)) - L_2(alpha(k)))/...
               (200000 - 100000)) + L_2(alpha(k));
         CD_s(k) = (Re_s(k) - 100000)*...
               ((D_3(alpha(k)) - D_2(alpha(k)))/...
               (200000 - 100000)) + D_2(alpha(k));
   else
         cl = (Re_s(k) - 200000)*((L_4(i) - L_3(i))/...
            (500000 - 200000)) + L_3(i);
         cd = (Re(k) - 200000)*((D_4(i) - D_3(i))/...
            (500000 - 200000)) + D_3(i);
   end

end

% Torque Elements
dT_s = (3/2)*rho*C.*R.*(v_cut^2*(1-A).^2./(cosd(Beta)).^2).*...
    (CL_s.*cosd(Beta)-CD_s.*sind(Beta));

% Torque Integration
Elements_T_s = [R;dT_s];
T_s = fit(R',dT_s','smoothingspline');
A_i = Elements_T_s(:,1:end-1);
B_i = Elements_T_s(:,2:end);
K = (B_i(1,:)-A_i(1,:))/6;
M = (A_i(1,:)+B_i(1,:))/2;
Int_T_s = K.*(A_i(2,:)+4*T_s(M)'+B_i(2,:));
Torque_s = sum(Int_T_s);


%% Axial Forces ------------------------------------------------------------

% Force Elements @ 12m/s
dF = (3/2)*rho*C.*(v_wind^2*(1-A).^2./(cosd(Beta)).^2).*...
    (CL.*cosd(Beta)+CD.*sind(Beta));

% Force Integration @ 12m/s
Elements_F = [R;dF];
F = fit(R',dF','smoothingspline');
A_i = Elements_F(:,1:end-1);
B_i = Elements_F(:,2:end);
K = (B_i(1,:)-A_i(1,:))/6;
M = (A_i(1,:)+B_i(1,:))/2;
Int_F = K.*(A_i(2,:)+4*F(M)'+B_i(2,:));
Force = sum(Int_F);

% Force Elements @ 8m/s
dF_s = (3/2)*rho*C.*(v_cut^2*(1-A).^2./(cosd(Beta)).^2).*...
    (CL_s.*cosd(Beta)+CD_s.*sind(Beta));

% Force Integration @ 8m/s
Elements_F_s = [R;dF_s];
F_s = fit(R',dF_s','smoothingspline');
A_i = Elements_F_s(:,1:end-1);
B_i = Elements_F_s(:,2:end);
K = (B_i(1,:)-A_i(1,:))/6;
M = (A_i(1,:)+B_i(1,:))/2;
Int_F_s = K.*(A_i(2,:)+4*F_s(M)'+B_i(2,:));
Force_s = sum(Int_F_s);

%% Stall point

%v_wind=12m/s
flag = 0;
for rpm_stall = 3580:-10:2000
   omega_stall = rpm_stall*2*pi/60;
   Beta_stall = 90 - (2/3)*atand(v_wind./(omega_stall.*R));
   for iter = 1:1:Nu
      i = Gamma(iter) - Beta_stall(iter);
      if Re(iter) <= 100000
         if i>=14
            flag = 1;
         end
```

```
        else
            if i>=17
                flag = 1;
            end
        end
    end
    if flag == 1
        break
    end
end

%v_wind=8m/s

flag = 0;
for rpm_stall_s = 2380:-10:1500
    omega_stall_s = rpm_stall_s*2*pi/60;
    Beta_stall_s = 90 - (2/3)*atand(v_cut./(omega_stall_s.*R));
    for iter = 1:1:Nu
        i = Gamma(iter) - Beta_stall_s(iter);
        if Re(iter) <= 100000
            if i>=14
                flag = 1;
            end
        else
            if i>=17
                flag = 1;
            end
        end
    end
    if flag == 1
        break
    end
end

%% Outputs ---------------------------------------------------------------

% Element Values
Outs_A = [1000*R;1000*C;Gamma];
Outs = array2table(Outs_A);
Outs.Properties.RowNames = {'Radius [mm]','Chord Lengths [mm]',...
                    'Angle of Twist [Degrees]'};
disp(Outs);

% Gear Values
Gears = [Force,Force_s,Ratio,N,Torque,...
        N_2,Torque_2,N_s,Torque_s,N_s_2,Torque_s_2,rpm_stall,rpm_stall_s];
Names = {'Axial_Force_12','Axial_Force_8',...
        'Ratio','RPM_12_1','Torque_12_1','RPM_12_2','Torque_12_2',...
        'RPM_8_1','Torque_8_1','RPM_8_2','Torque_8_2','Stall_RPM_12','Stall_RPM_8'};
Gears_Out = array2table(Gears,'VariableNames',Names);
disp(Gears_Out);

fprintf('The power by this bladet is ');
fprintf('%5.3f W',Power);
fprintf('\n');
fprintf('The power can be generated is about ');
fprintf('%5.3f W',Power_output);
fprintf('\n');
fprintf('The Rotor Efficiency is ');
fprintf('%5.3f',Power_coefficient);
fprintf('\n');

% Plots
subplot(2,2,1)
plot(R,T(R))
title('Torque Gradient at 12m/s - Design Velocity'); xlabel('Radius [m]');
ylabel('dT/dr [N]'); grid on
subplot(2,2,2)
plot(R,T_s(R))
title('Torque Gradient at 8m/s - Cut In Velocity'); xlabel('Radius [m]');
ylabel('dT/dr [N]'); grid on
subplot(2,2,3)
plot(R,F(R))
title('Force Gradient at 12m/s - Design Velocity'); xlabel('Radius [m]');
ylabel('dF/dr [N/m]'); grid on
subplot(2,2,4)
plot(R,F_s(R))
title('Force Gradient at 8m/s - Cut In Velocity'); xlabel('Radius [m]');
ylabel('dF/dr [N/m]'); grid on


%% Scaling Profile

Profile = xlsread('n2415.xlsx');

Profile(:,1) = Profile(:,1) - 0.5;

Coor_Profile = zeros(length(Profile),3,length(R));

for k = 1:length(R)
        Coor_Profile(:,1:2,k) = C(k)*Profile;
        [Test_t,Test_r] = cart2pol(Coor_Profile(:,1,k),Coor_Profile(:,2,k));
        Test_t = Test_t - deg2rad(Gamma(k));
        [X,Y] = pol2cart(Test_t,Test_r);
        Coor_Profile(:,1,k) = X;
        Coor_Profile(:,2,k) = Y;
        Coor_Profile(:,3,k) = R(k)*ones(1,length(Profile));
end

T_Profile = permute(Coor_Profile,[1 3 2]);
T_Profile = reshape(T_Profile,[],size(T_Profile,3),1);

Coor = 1000*T_Profile;

Thickness = 1000*C(1)*15/100;

Status = xlswrite('FINAL_COORDINATES.xlsx',Coor);
```