Wind Turbine Data Analysis

By:
<Name Redacted>
<Name Redacted>
<Name Redacted>
Matthew Zaksek

May 5, 2019

Introduction

"A wind turbine is a device that converts kinetic energy from the wind into electricity" [1]. Considering how in the present day, electricity is so greatly relied upon and utilized on a daily basis in countless ways by the entirety of society, intuitive and environmentally-friendly inventions that make use of renewable resources (air), which can then be used to generate mass amounts of electricity, like wind turbines, are as a result vital breakthroughs for the betterment of our modern, technology-crazed society. In fact, "the first electricity-generating wind turbine was invented in 1888 in Cleveland, Ohio by Charles F. Brush" [2]. Since Bush's major discovery, many different individuals have analyzed, implemented, and improved upon the idea to further assist in the sheer amount of electricity wind turbines are capable of generating nowadays. For example, to illustrate how far wind turbines have come since Bush's breakthrough in 1888, Bush's turbine only "generated about 12 kilowatts (kW) of power" [2]. While this seemed like an unreal amount at the time, with the advancements of modern society the "average onshore wind turbine [has] a capacity of 2.5–3 MW" [3] which is astonishingly about 200 times greater than Bush's initial wind turbine. As a result, when our group was assigned with the task to analyze the dataset which we had been provided in MATLAB, which were results that had been gathered from differing numerical simulation regarding an incompressible flow around a wind turbine within a specific domain (showcased within figure 1 [4] below), we realized just how crucial it was to utilize this data and correctly yield and/or verify the tasks which we had been given, especially considering how important it is to understand the inner workings of wind turbines and their related calculations, especially for students within the engineering field.

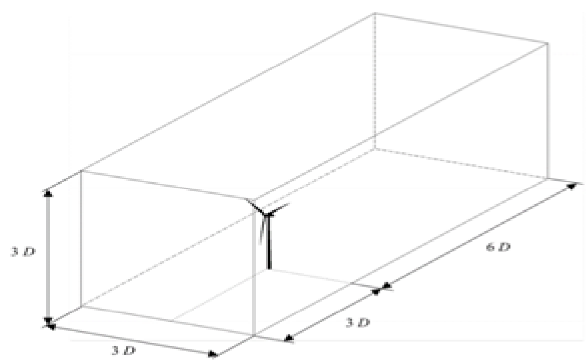

Figure 1 - Diagram of wind turbine in wind tunnel

Thus, in this report we will be commenting upon and of course discussing how we were able to implement, utilize, and combine various fluid mechanics concepts learned throughout the semester with handy functions within MATLAB in order to accomplish the five outlined project tasks, which include verifying the conservation of mass within the domain we were assigned, applying the theory of conservation of momentum in order to calculate the thrust force upon the wind turbine, utilizing the theory of conservation of energy to allow for the computation of the power which had been extracted by the turbine, taking the values for the pressure of a horizontal plane at the height of the hub and plotting them, and finally of course graphing requisite particle lines. In addition, we will also describe and comment upon the numerical code which was input into MATLAB in order to properly achieve the previously mentioned project tasks.

Results

Conservation of Mass

The purpose behind the law of the conservation of mass is to essentially sum two different quantities in order to get a total value of zero. These two different quantities are displayed within equation 1 [4] below which essentially outlines the basic application of the law of conservation of mass to a system.

$$0 = \frac{d}{dt}\int_{cv} \rho \, dV + \int_{cs} \rho\left(\vec{U} \cdot \vec{n}\right) dA$$

As one is able to see the first quantity describes “the rate of change of mass inside the control volume” [5], whereas the second quantity to the right of the addition sign is essentially “the net rate of mass flux out through the control surface” [5]. Furthermore, if one is provided the information that the system which will be analyzed consists of an incompressible flow and is steady state, as described within this project, then one is able to reduce equation 1 above to the simpler equation 2 [4] described below in which the first quantity from equation 1 essentially is canceled out as steady state implies no change in rate over time and additionally density is removed from the second quantity in equation 1 in which mass fluxes are summed within the control surface due to the flow within the system being incompressible.

$$0 = \int_{cs} \left(\vec{U} \cdot \vec{n}\right) dA$$

Thus, if one is able to take this showcased quantity within equation 2 and sum it to equal zero after being applied to each of the faces of a system, then the conservation of mass will have been verified within the entirety of the system. So, when applying this theory to the system which we had been provided, consisting of the wind turbine and aforementioned incompressible flow, we essentially utilized coding capability within MATLAB in order to go one-by-one computing the flux at each different face of the domain through the use of numerical integration. Then, by taking these values of the flux at each face which were computed we determined whether the law of conservation of mass had been upheld based upon whether this summing of each of the flux totals at each face combined to equate to zero, which as has been discussed above is ultimately the determinant upon whether the law has been verified by the system or not.

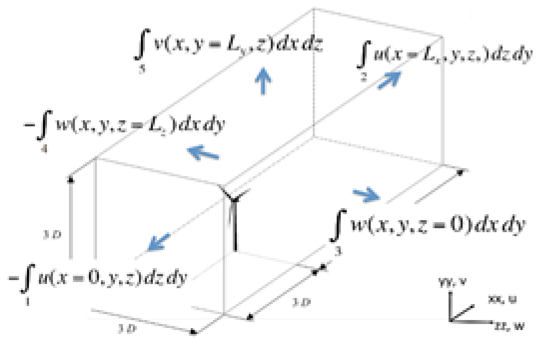

Figure 2 - Conservation of mass through each face of control volume

The six faces which we applied equation 2 (above) regarding the law of conservation of mass to, in order to deem whether their summed flux values derived from numerical integration totaled to zero are showcased within figure 2 [4] above, and as one can see only five are specifically detailed upon as the sixth in the negative y direction consists of a face at the ground when y=0, thus meaning it can be assumed to have a flux value of zero, and no calculation must be applied to that specific face. Aside from that sixth face, one can see the other 5 integrals at each face comprise of two in the x-direction, one when x=0 and one when $x=L_X$ (the distance covered within the domain in the x direction), two in the z-direction, one when z=0 and one when $z=L_z$ (the distance covered within the domain in the z direction), and one in the y-direction when $y=L_y$ (the distance covered within the domain in the y direction). Now, specifically, when using MATLAB to calculate flux value at each of the six faces of the domain provided, we made sure to make use of three different for loops to account for each plane xy, xz, and yz. The piece of our code which shows the for loops we utilized to simulate numerical integration in each axial direction is provided below.

```
%flux  going through the faces
f1=0;
f2=0;
for j=1: n2-1
    for k=1:n3-1
        da=(yy(j+1)-yy(j))*(zz(k+1)-zz(k));
        f1=f1+u(1,j,k)*da;
        f2=f2+u(n1,j,k)*da;
    end
end
f3=0;
f4=0;
for i=1:n1-1
    for j=1:n2-1
        da=(yy(j+1)-yy(j))*(xx(i+1)-xx(i));
        f3=f3+w(i,j,1)*da;
        f4=f4+w(i,j,n3)*da;
    end
end
f5=0;
f6=0;
for i=1:n1-1
    for k=1:n3-1
        da=(xx(i+1)-xx(i))*(zz(k+1)-zz(k));
        f5=f5+v(i,n2,k)*da;
        f6=f6+v(i,1,k)*da;
    end
end
total_flux=-f1+f2+f3-f4+f5-f6
```

After each of these for loops had been written into MATLAB, we summed the values of flux for each face which had been stored in the variables f1, f2, f3, f4, f5, and f6 and store the sum of these values within the variable called “total_flux” while also making sure to make f1, f4, and f6 negative to account for the negative direction of the normal line at these faces in comparison to the coordinate system we utilized, which was shown in Figure 2 above. This summing of the 6 flux integral values at each face is shown in the above final line of code. Finally, we ran our script within MATLAB and analyzed whether our calculated value for “total_flux” totaled to a value near zero to determine whether we had succeeded in utilizing the dataset which had been provided to us and had successfully verified the law of conservation of mass. Our value for “total_flux” ended up equaling 0.0258, which we deemed was close enough to zero to reasonably deem that we had successfully verified the law of conservation of mass using the data which we had been given within the provided system holding the wind turbine.

Thrust Force on Wind Turbine

Theoretically speaking, the thrust force is equal in magnitude but opposite in direction of the drag force of a body. This force can be computed by applying the conservation of momentum within a given control volume. Assuming the flow is incompressible and the control volume is provided, the principle of conservation of momentum states that within that control volume, the total momentum remains constant. In other words, the net amount of momentum entering the control volume, in this case through face 1, equals the net amount of momentum exiting the control volume, or face 2, at any given time.

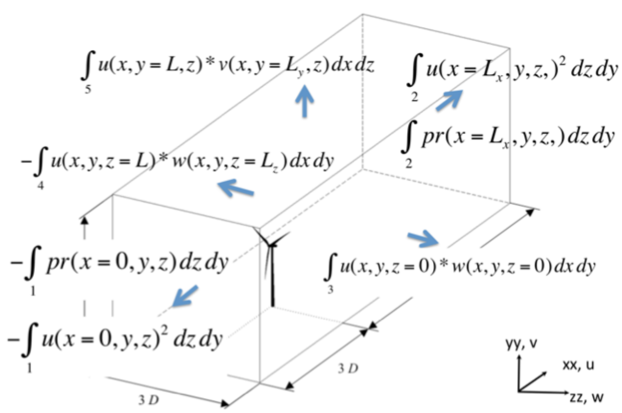

Figure 3 - Conservation of energy through each face of control volume

In the provided dataset, momentum is primarily gained or lost through faces 1 and 2, as the remaining faces are walls, so they function as streamlines. So, in theory, we should only need to consider the momentum along the xx,u axis. But experimentally, we will consider the momentum along all faces, as shown in our code. The sum of forces in the xx direction in relation to pressure and momentum is given below.

$$F_{thrust} + \int_{cs} Pn_x \, dA = \int_{cs} \rho U_x \left(\vec{U} \cdot \vec{n}\right) dA$$

In MATLAB, the momentum through each face was taken through numerical integration of the momentum through each face, as shown. Note the below figure contains a for loop to compute momentum only through faces 1 and 2. (Note: A more detailed explanation of the code used is provided in the MATLAB code)

```
% Flux at the faces of verification of cons. of momentum
f1=0;
f2=0;
p1=0;
p2=0;
for j=1:n2-1
    for k=1:n3-1
        da=(yy(j+1)-yy(j))*(zz(k+1)-zz(k));
        f1=f1+(u(1,j,k)^2.0)*da;
        f2=f2+(u(n1,j,k)^2.0)*da;
        p1=1+pr(1,j,k)*da;
        p2=p2+pr(n1,j,k)*da;
    end
end
```

The thrust coefficient can be computed using the equation below, derived from the conservation of momentum equation given above.

$$C_T = \frac{F}{\frac{1}{2}\rho U^2 A} = \frac{F}{\frac{1}{2}\rho U^2 \frac{D^2}{4}\pi}$$

$$= \frac{\int_{CS} \rho U(\vec{U} \cdot \vec{n})\, dA - \int_{CS} P\vec{n}\, dA}{\frac{1}{2}\rho U^2 \frac{D^2}{4}\pi}$$

$$= \frac{\int_{CS} U^2\, dA}{\frac{\pi}{8}} - \frac{\int_{CS} P\, dA}{\frac{\pi}{4}}$$

In MATLAB, the corresponding line of code below is used once the total net momentum is found among all faces and the pressure difference is known.

```
thrust_coefficient = (8/pi)*(-f1+f2+f3-f4+f5-f6)+(p1-p2)*(4/pi)
```

Once the thrust coefficient is known, then the thrust can be computed using the following relationship.

$$F_T = \frac{1}{2} C_T \rho U^2 A$$

In our computation, we found that the thrust coefficient is 0.6999. As density is known to be 1.225 kg/m^3, velocity is approximately 8.442 m/s (derived computationally), and the area is 12,469 m^2, then the computed thrust force is approximately 381 kN.

Power Extracted by the Turbine Applying Conservation of Energy

The first law of thermodynamics states that energy is conserved in any isolated system, which implies that the total amount of energy remains constant. For the purposes of the wind turbine experiment, we will use this principle of conservation of energy to compute the power extracted by the turbine. Below is the equation representing this physical principle.

$$\dot{Q} - \dot{W} = \oint \rho \left(\vec{V} \cdot \vec{n}\right) \left(\frac{V^2}{2} + gz + u + \frac{P}{\rho} \right) ds$$

Conceptually, this equation states that the rate of the energy being inputted into the system (Q) minus the rate of work being performed by the system (W) equals the total density multiplied by the velocity and the area, all distributed to the sum of the kinetic energy ($V^2/2$), potential energy (gz), the specific internal energy (u) and the flow work (P/ρ). Since we are dealing with an incompressible fluid, the density will remain constant. As there is no heat input, Q is equal to zero, and we assume we aren't dealing with substances that have given internal energies.

This relationship is computed in MATLAB through numerical integration using code similar to the following.

```
% Power extracted by the wind turbine
power_1=0;
power_2=0;
istart=75
iend=127
for j=1:n2-1
    for k=1:n3-1
        da=(zz(k+1)-zz(k))*(yy(j+1)-yy(j));
        tke1=(u(istart,j,k)^2.0+v(istart,j,k)^2.0+w(istart,j,k)^2.0)/2;
        tke2=(u(iend,j,k)^2.0+v(iend,j,k)^2.0+w(iend,j,k)^2.0)/2;
        power_1=power_1+((-u(istart,j,k)))*((tke1)+pr(istart,j,k))*da;
        power_2=power_2+(u(iend,j,k))*((tke2)+pr(iend,j,k))*da;
    end
end
```

The code above computes the power through face 1 and face 2. The power through the remaining faces is computed in a very similar manner, which can be seen in the full MATLAB code.

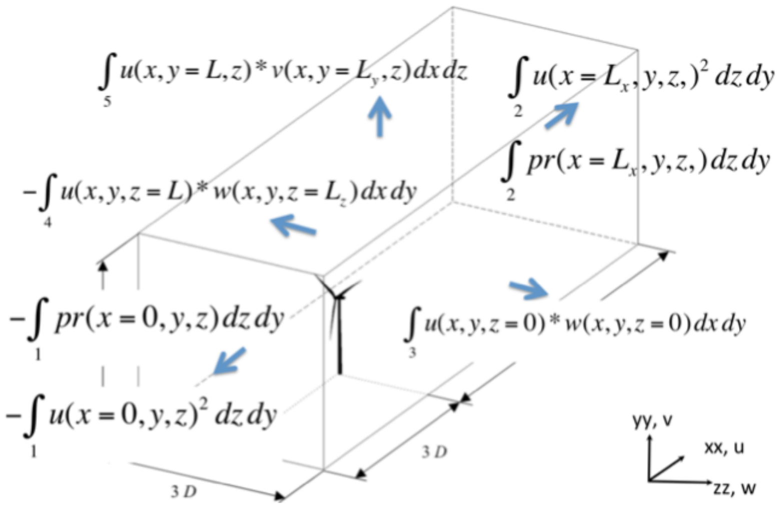

Figure 4 - Conservation of energy through each face of control volume

Once the power through each face of the given control volume is known, then these values can be summed for each dimension and the power coefficient C_p can be computed in MATLAB.

```
WT_power_1=4/pi*(power_1+power_2)*2;
WT_power_2=power_3+power_4;
WT_power_3=power_5+power_6;
c_p = -(WT_power_1+WT_power_2+WT_power_3);
```

For wind turbines, the power coefficient can be computed by taking the actual electrical power produced divided by wind power going into the turbine [6]. In our MATLAB code, we found that the power coefficient C_p equals 0.5608.

Once the power coefficient is known, it can be substituted into the following equation to find the total power output of the wind turbine [6].

$$P_{out} = \frac{1}{2} C_p \rho A V^3$$

$$= \frac{1}{2} C_p \rho \pi R^2 V^3$$

$$= \frac{1}{2}(0.5608)(1.225\ kg/m^3)\pi(63m)^2(8.4422\ m/s)^3$$

$$P_{out} \approx 2.577\ MW$$

This calculation was also performed in MATLAB using the below code.

```
power = 1/2*Rho*velocity^3*pi*R^2*c_p;
```

The MATLAB result was also approximately 2.577 MW, which confirms our expected result for extracted power using conservation of energy.

Color Contours of Pressure

The plot pressure can be described as the force that is exerted by the fluid on the motor and how the wind flow is translated into making the turbine blade spin.The plot shows a normalized pressure and velocity in front of the wind turbine. The contour also provides a good visualisation on how the pressure increases as the wind flow get closer to the wind turbine and then decreases after the wind turbine absorbs some of the energy from the airflow.

As observed in the conservation of energy equations, the energy within a closed system will remain constant, however it may be converted. In this case the kinetic energy from the winds are absorbed by the wind turbine in the form of rotational energy. As seen in the x-y pressure diagram at the hub the pressure is consistently higher before the wind turbine and peaking right before the wind turbine plane and decreasing after the energy transfer. This occurs because the air is pushing the turbine causing rotation. The blade absorbs energy in the form of velocity and pressure. As the air passes through the blade the net energy W of the fluid is transferred in the form of a reduction in pressure and velocity.

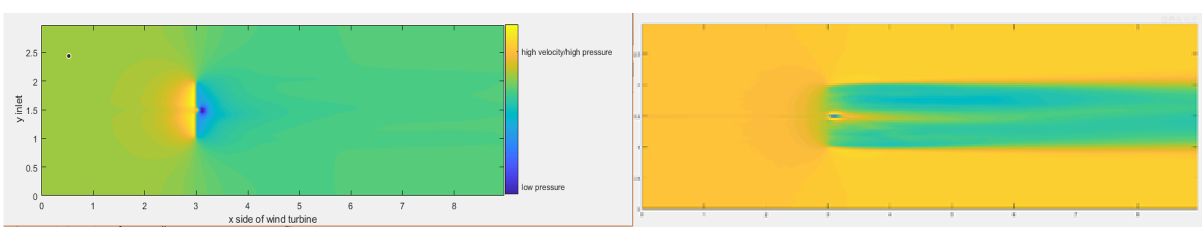

Figure 5 - In this figure the pressure is monitored on the horizontal plane at the center of the windmill nacelle.As observed by both the velocity and the pressure drop after the fluid encounters the blades

From the conservation of energy equation the energy, W, in a fluid comes from V, the velocity, z, the potential energy from gravity, and P, the pressure. Since there is a negative change on the total energy W the pressure and velocity must drop as well. We are not accounting for a change in elevation because we are representing one horizontal plane. We are also ignoring Q since the change in heat is not accounted for.

$$\dot{Q} - \dot{W} = \oint \rho\left(\vec{V} \cdot \vec{n}\right)\left(\frac{V^2}{2} + gz + u + \frac{P}{\rho}\right)ds$$

```
%Pressure plot
figure
pr_vsec(:,:)=pr(:,30,:);
contourf(xx,zz,pr_vsec',250,'linestyle',' none');
daspect([1,1,1]);
xlabel('x side of wind turbine');
ylabel(' y inlet');
colorbar('Ticks',[-.3, .1],'Ticklabels',{'low pressure','high velocity/high pressure'});
```

The plot is created using the contourf function which takes the pressure data at the plane and represents it as a 2D chart using the given coordinates. The pressure is given to us in the simulation data and was not altered to work in the plot.

<u>Particle Lines</u>

To observe the flow of the particles we can isolate data points and determine the flow of the particle and create a plot of the particle lines.By creating the plot of the particle lines it provides a graphical representation of the velocity of air flowing through the wind turbine and furthermore it allows for the ability to analyze the flow behavior before and after it goes through the wind turbine. The line is drawn so that the velocity vector of a particle is parallel to it at any given point so they visualized the path a particle will take. With this data we can observe how the airflow will interact with environmental factors such as the wind turbine. From the model we observe that before the rotors the wind is undisrupted and travels in a straight path at a constant velocity.This chart shows us the particle's velocity are changed by the movement of the blades and pick up some rotational velocity from the rotors. This change in velocity can be attributed to the drag from the blades; when the wind applies drag force to the blade is met with an equal and opposite force resulting in a change in velocity.

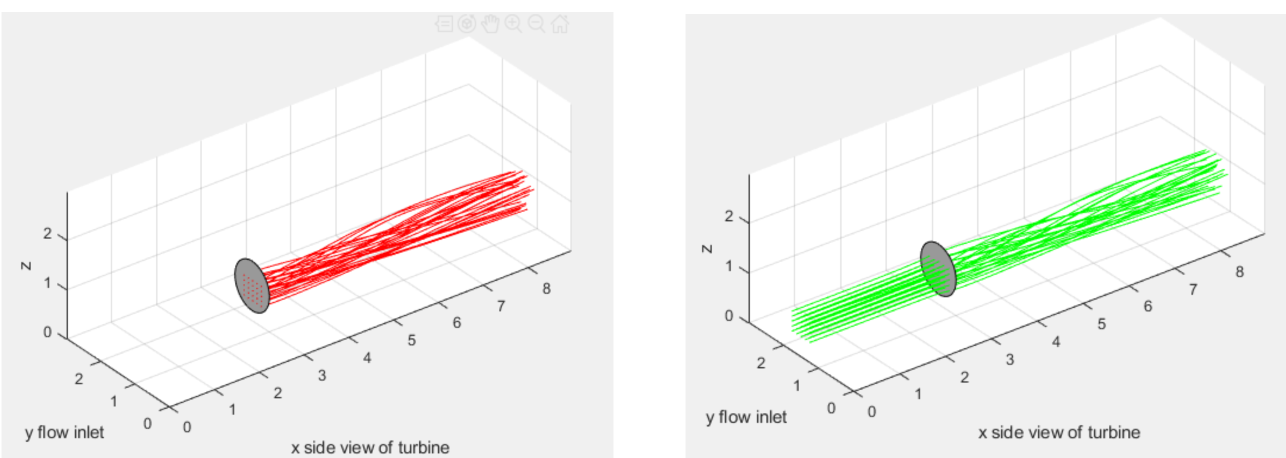

Figure 6 - The path the particles takes corroborates observations made regarding the previous sections

In Matlab, the particles are modeled by the streamline tool which takes coordinance, velocity, and starting position data. The coordinance are set up in a matrix using the meshgrid function which takes the given coordinates from the file and returns 3D grid coordinates. The velocity values are permuted switching the rows and columns to match orientation. The streamline takes the input data of velocity and position and outputs the particles traveled path.

Final Remarks

With the data obtained through the simulations within MATLAB that have been discussed at great length above, we can see how it is possible to both visually and mathematically represent the theories regarding the Conservation of Mass, Momentum and Energy. Some built in assumptions such as incompressibility and neglecting heat transfer are made to assist in simplifying the model. These assumptions can alter the results a bit, but for the most part the discrepancies are mostly accounted for in a sufficient manner. Air can be modeled as incompressible at low speeds under 100 m/s and the highest observable speed in the data is the wing tip speed at approximately 63 m/s therefore the model retains its value. As for heat transfer, most of the energy can be accounted for in the drag since the major source for the creation of heat is caused by friction resulting from drag. The model can accurately assess energy output and drag on the wind turbine which are useful metrics in the design and planning of wind turbines. The drag numbers help engineers understand the forces a wind turbine will experience and allows them to design accordingly. High wind speeds can cause wing tip deflection, tower bending and even catastrophic failure, so to prevent failures from happening, simulations like this can be used to predict potential faults in design. Another use for the data is in cost analysis. A wind turbine costs anywhere between $1-2 million per megawatt of capacity so it is important to figure out if a turbine is feasible based on cost. Too low of a wind speed could result in an obsolete wind turbine; a design could be efficient but if the correct wind speeds do not occur the turbine will fail financially. Thus in conclusion, when considering this importance of these discussed values within the report as they pertain to wind turbines it showcases the significance of gaining an understanding upon how to calculate the data figures both by hand and through software like MATLAB especially for students within the engineering field.

References:

[1] ACCIONA, "What is a wind turbine and how it works?," [Online]. Available: https://www.acciona.com/renewable-energy/wind-power/wind-turbines/. [Accessed 4 May 2019].

[2] Third Planet Windpower, "History," 2019. [Online]. Available: http://www.thirdplanetwind.com/energy/history.aspx. [Accessed 2019 April 2019].

[3] European Wind Energy Association, "Wind energy's frequently asked questions (FAQ)," 2016. [Online]. Available: http://www.ewea.org/wind-energy-basics/faq/. [Accessed 4 May 2019].

[4] E. J. García-Cartagena, "MATLAB tutorial," 2019. [Online]. Available: https://elearning.utdallas.edu/bbcswebdav/pid-2476180-dt-content-rid-64533538_1/courses/2192-UTDAL-MECH-3315-SEC001-23357/matlab_tutorial.pdf. [Accessed 4 May 2019].

[5] S. Leonardi, "Conservation of Mass," 30 January 2019. [Online]. Available: https://elearning.utdallas.edu/bbcswebdav/pid-2385505-dt-content-rid-59936862_1/courses/2192-UTDAL-MECH-3315-SEC001-23357/Chapter%204_18_01.pdf. [Accessed 4 May 2019].

[6] D. E. Watson, "Wind Turbine Power Coefficient," 2015. [Online]. Available: http://www.ftexploring.com/wind-energy/wind-power-coefficient.htm. [Accessed 4 May 2019].