Micro-Scale Wind Turbines: Potential in Energy Deprived Regions

James Madison University

Fall 201   – Summer 2014

National Collegiate Wind Competition

Table of Contents

INTRODUCTION

In order to properly design a product, mathematical analysis and computational modeling for each component and system are necessary to obtain parameters and confirm feasibility of proposed designs. The following sections document designs correlated to and derived customer needs. Each design was modeled analytically using SolidWorks and mathematically analyzed using known mechanical methods.

DESIGN OBJECTIVE

The team selected and the business plan reflects target market of developing, energy-impoverished countries. This was narrowed down to Sub-Saharan Africa, with Kenya being the initial location for deployment. Manufacturing would occur in India due to optimal manufacturing and transportation strategies. Distribution centers would be located in the countries where sales would occur. These distribution centers would employ trained professionals to engage in social marketing and to assist in installation and maintenance.

There were number of design requirements that were necessary in order address to the objective of the business plan. The requirements: highly durable construction with affordable materials, universal mounting system to expedite installation, removable and replaceable parts, scalable rotor diameter, and general system simplicity in order to encourage trainable maintenance were all included.

DESIGN OVERVIEW

Micro-scale wind turbines show promise for third-world development in energy-impoverished countries by lessening the demand for fossil fuel dependent systems. Further, the proposed design targets developing regions that require less expensive alternatives to satisfy the needs of communities.

Materials access, maintenance, and education in developing countries are limited. Therefore, the design must be robust, easily reparable, and able produce the required output to meet the needs of the end-user. The turbine is designed with safety features to quickly shut down when disconnected from the load. This is done both manually through a fail-safe switch and passively to prevent freewheeling if the electrical components of the turbine were to fail. The rotor is made from RDG525, very durable polymer, and a safety factor of 20% has been added to the thickness of blades to ensure it does not fail in the field. If one or more of the blades were to become nonfunctional, the turbine has a high level of modularity, allowing the user to replace a single blade for much less cost than replacing the entire rotor. This modularity also allows for easy access to the electrical components of the turbine for repairs and replacements. There is a high degree for interchangeability of mounting components, such as screws, that optimize for easy assembly and disassembly. These features increase the overall lifespan of the product by designing to prevent malfunctions and allowing for easy fixes if malfunctions d occur.

Kenya was chosen because of the community's need for a small charging station. Most individuals in Kenya need to charge small electronic device, especially cell phones, which are the primary means of communication for most people in the region. Thus, the control system is designed to provide power regulated at a chosen voltage pertinent to the needs of the individual. In this design, the voltage is regulated at 5 volts and is intended to achieve a minimum power output of 10 watts. A gearbox with a high gearing ratio is necessary to produce the required power. To account for the torque required to spin up such a gearbox, the turbine features a five-bladed rotor design. For these reasons, this micro-scale wind turbine is a viable option for deployment in Sub-Saharan Africa.

DESIGN TEAM

The overall engineering team comprised two sub-teams: the *Electronics an Controls (E&C) Team* and the *Rotor Team* group members are introduced below:

The *E&C Team* included *Mick Blackwell* (Team Manager) who coordinated each step of the design process and served as the liaison between the Business Team, and Rotor Team. *Kyle Kingsborough*

(Circuit Designer) specified the components necessary for circuit completion and fabricated the circuit prototype. *Ben Condro* (Machinist) was one of the two main fabricators for the team; he helped to fabricate most of the structural components for the turbine. *Joey Abla* (Machinist) contributed to the fabrication of the turbine and components for the testing rigs. *David Harootyan* (Fund Allocator) was in charge of managing finances and also helped to conceptualize early prototypes.

The *Rotor Team* included *Dixon Drumheller* (Team Manager) who kept the team on track and oversaw each step of the design process. *Genevieve D'Antonio* (Testing coordinator/Secretary) handled many aspects of the prototyping and testing process. *Corey Allison Blake Chapman,* and *Alex Donley* assisted in a variety of ways.

MODELING AND TESTING

The top four challenges identified in terms of designing our wind turbine were:

- Select a gearbox appropriate to the torque and RPM capabilities of the blade design.
- Refine the blade design to optimize RPM while maintaining required torque.
- Construct a circuit in order to maintain proper power outputs and operating conditions.
- Fabricate a structure to hold all components effectively and safely.

Estimates of generator characteristics were produced early in the design process. Shown below in Figure is graph demonstrating the performance characteristics of the supplied generator relative to three rectifying systems.

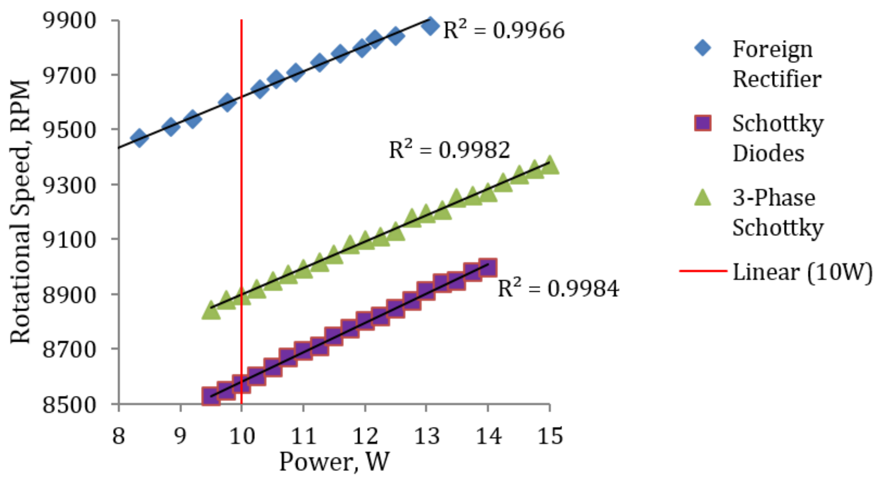

Figure 1. RPM vs. Power for three rectifying systems.

The timeline for this project provided for unique experience relative to the standard design methodology approach. Testing was conducted early on to parameterize the generator required for turbine construction. The generator was required to produce 10 of power when regulated by a 5 V voltage regulator. Thus, as shown in the figure above, a RPM vs. Power curve was produced relative to three separate rectifying systems. The 3-Phase Schottky rectifying system was chosen because of the minimal housing space required for the rectifier.

In order to procure the results obtained in Figure 1 above and torque requirements for select gearboxes, testing rigs were fabricated. Shown in Figure 2 below are examples of two testing rigs used throughout experimentation.

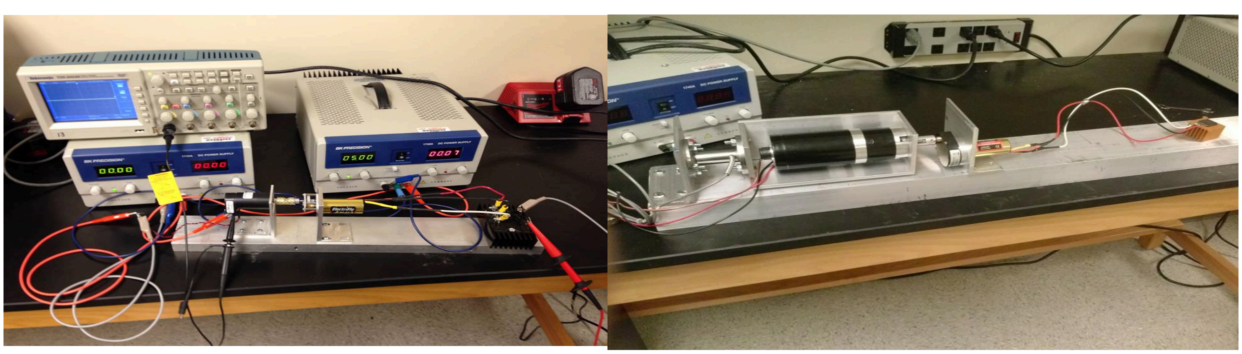

Figure 2. The earliest testing rig (left) and the torque cell rig (right).

Early testing rigs, similar to the rig shown above (left), were utilized often to conduct feasibility studies at each step during the design process. This particular rig was used to determine power as a function of rpm. Later rigs incorporated high-end components, such as torque cells, to achieve metrics used in determining the optimum RPM vs. Torque ratio. An example of this rig is shown above (right).

Results from testing multiple gearboxes yielded the torque required to activate the turbine. NI LabVIEW was utilized for data acquirement from the OMEGA Torque Cell. For instance, when testing the fabricated gearbox shown below in Figure 3, the torque required to spin the gearbox and generator was approximately $0.5\ ft{\cdot}lb$.

Figure 4. An example of gearbox tested with the torque cell rig. The gearbox ratio was 4.5:1.

Analytical and physical modeling was highly valued aspects of this design process. Shown below in Figure is the first conceptualized prototype. The model depicted was a streamline turbine utilizing a universal gearbox. The model below failed to meet required torque ratings and, thus, was not considered feasible alternative.

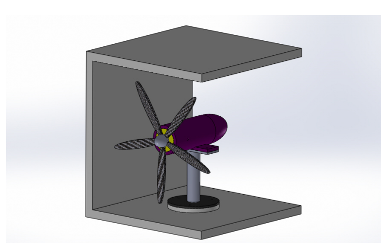

Figure 5. The streamline turbine model.

The *Rotor Team* developed and analyzed six different turbine blade designs from scratch. It became evident from electronics-related testing and calculations that the gearbox would require significant torque to spin up, and the rotor would need to be designed accordingly. The team initiated airfoil research by examining the NACA airfoil database. The database contains thousands of different airfoil shapes categorized by series according to the attributes of each airfoil shape.

Table 1: Lift to Drag ratio Based on Airfoil Family.

Series	Airfoil	Lc/Dc
Joukowski	(joukowsk0009-jf) Joukovsky f=0% t=9%	**23.7**
1 series	NACA 16-01 (naca16018-il)	**22.6**
Series	NACA 64A410 (naca64a410-il)	**38.1**
Digit	(Naca2408-il) NACA 2408	**37.4**
Digit	NACA 25112 (naca25112-jf)	**17.1**
Series	NACA 747A315 (naca747a315-il)	**20.2**

One airfoil was chosen from each of the six series shown in Table 1. The Joukowski, 6 Series, and 4 Digit series blades were considered. The three blades selected were chosen based on their lift to drag ratios. However, the Series airfoil, with the most favorable lift-to- drag coefficient, was selected to progress into the modeling and testing phase. The next step was to dimension the blades.

A hub was designed in Rhinoceros 5.0 to fit the driveshaft chosen by the *E&C Team*.

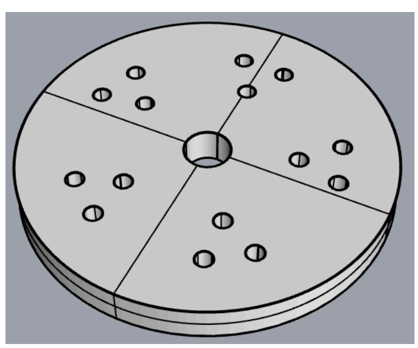

Figure 6. The initial hub designed for early testing.

The rotor diameter was restricted by the campus wind tunnel testing protocol that requires 1.5 cm of clearance between the blades and the wall of the tunnel. Including the 6.35 cm diameter of the hub, the team was limited to maximum blade length of 15.325 cm.

The hub is configured to allow operation with either three or five blades and maintain balanced turbine; however, five-blade configuration was selected in an effort to generate greater torque

The ideal thickness of the blades was provided by the NACA based on a fraction of the chord length of the blade. The chord length was determined by examining existing small-wind turbines and averaging the ratio of blade length to chord length along the span of the blade.

The NACA database also provided graphs of the lift-to-drag ratios of the airfoils based o the angle of attack as shown in Figure 7 below.

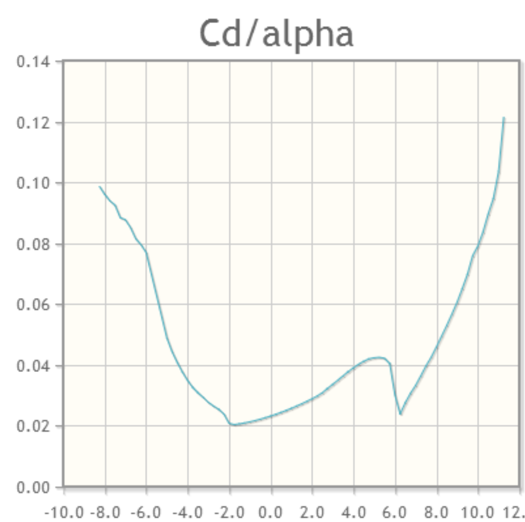

Figure 7. Example of drag coefficient vs angle of attack graph for the 6 Series airfoil.

The pitch angle was determined using the angle of attack data provided. Blades were tested using two different pitch configurations. The first iteration had the blades angled to maximize the liftto-drag in an attempt to mimic aircraft. The second iteration simply maximized both the lift and drag coefficients as turbine start-up is largely based on drag.

The blades were then created in CAD along with the hub and the design was printed using an Objet Pro 3D printer. The assembled blade and hub concept is shown in Figure 8.

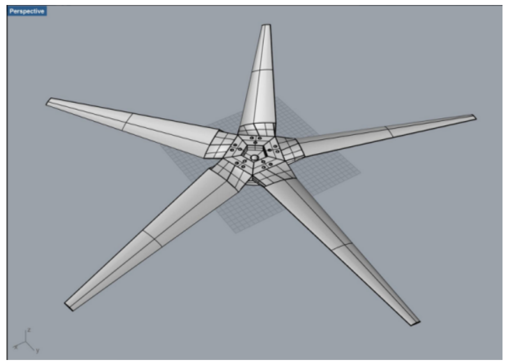

Figure 8. Blade and hub iteration assembled in CAD.

The hub and blades were assembled and connected to makeshift tower to fit in the campus wind tunnel. The tunnel test section has dimensions comparable to the tunnel that will be utilized at the competition.

Figure 9. Testing apparatus in wind tunnel.

The two iterations of the Series blade design were both tested in the wind tunnel, and the more aggressive pitch angle was found to be more effective than the blades that had greater lift-to-drag ratio. In order to maximize torque from the rotor, another iteration of the blade was created. This design widened the blades to the extent possible using the existing hub, and included a slightly more aggressive pitch angle to induce greater drag in order to further reduce cut-in speed.

When tested in the wind tunnel, the most recent iteration produced a cut-in speed of 0.9 m/s unloaded. With the 12:1 gearbox attached, the cut-in speed was raised significantly. The RPM and tip speed were not tested, although theoretical values are 705 RPM with a 12.57 m/s tip speed at 5 m/s wind speed, and 2,117 RPM with 37.7 m/s tip speed at 1 m/s wind speed.

Further trial-and-error methodology was applied and feasible design alternatives conceived. A new rotor design was developed using a blade profile based off the patented S818 airfoil, with non-uniform pitch angle ranging from 0° to 10° along the length of the blade, in accordance with standard blade design practice.

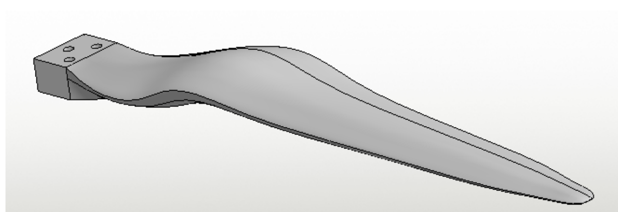

Figure 10. The S818 style blade used in the most recent prototype.

The S81 blade was fabricated via the 3D printers available. The Objet Pro was utilized to print the scale models. Within the time constraints for the final deliverables of the competition, this blade design was unable to be tested. In the remaining time period, the S818 blade will be tested extensively, along with further iterations of the 6 series blade. Due to the high torque requirements, an additional, and final, blade design that significantly increases drag to further address start-up concerns is, at present, in 3D printing[1] and will be tested next week.

[1] Our 3D printer has been out of commission for more than one week.

The structural performance of the turbine was established using known column loading scenarios. The maximum drag forces the wind turbine would sustain were miniscule relative to the tower strength chosen to support all components.

Note that the base dimensions for the turbine were determined by the Department of Energy. The wind tunnel made available was only usable if the base was specified appropriately. Further, the entirety of the turbine is required to fit in a fit in a 45 x 45 x 45 cm housing. Naturally, precise fabrication and measuring techniques were applied throughout the design process.

The most recent prototype shown below in Figure 11 utilizes the blade design described in Figure 6. Each component displayed in Figure 7 is described in the Engineering Diagrams section.

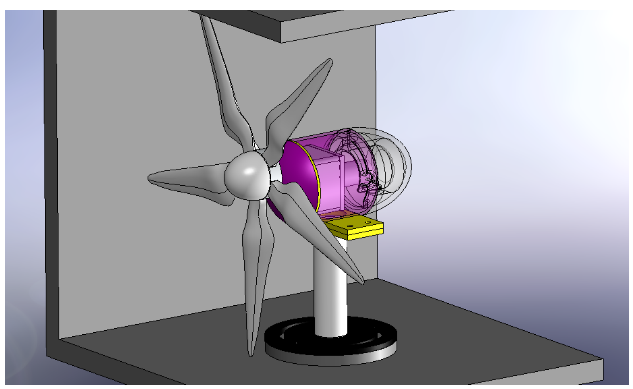

Figure 11. A SoildWorks rendering of the final prototype.

The wind turbine input shaft was centered in the testing cubicle in order to maximize blade length, swept area, and power output. The gearbox used for demonstration was a 25:1, solid metal multipurpose gearbox (Model: V9264). This model is designed to sustain a running torque of 4 Nm, which is expected to be at least   times the predicted operating torque. Also, the design was completely modular; in this case, design components can be easily separated from one another.

In regard to circuitry and electrical components, the generator is connected to the circuitry outside of the wind tunnel. The wires from the generator are fed through the turbine tower and out the base of the wind tunnel to the circuit prototyping board shown below in Figure 12.

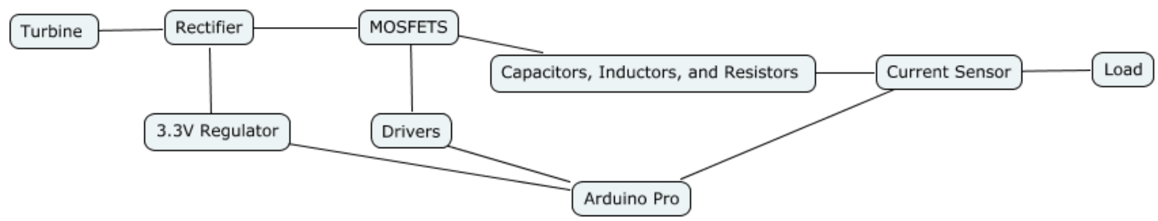

Figure 12. The Control System Prototype

The wires from the generator enter into the 3-Phase Schottky rectifying system, which converts the 3 phases from the generator into DC signal. This DC signal is sent to two places: 3.3-V regulator and a buck convertor system. The purpose of the regulator is to provide the drivers and the Arduino with the correct voltage. The buck circuit system involves the use of N-channel MOSFETs and high/low-side drivers, capacitors, inductors, and resistors. The MOSFETS and drivers act as switches to allow voltage to pass through the circuit for an allotted amount of time. When the MOSFETS and drivers are not providing any current, the capacitors and inductors provide the current needed to provide power. The drivers are switched using a pulse width modulation technique, made possible with the Arduino Pro. The Arduino Pro serves as a proportional integral derivative controller. The Arduino input can be adjusted by a desired variable in order to maintain a certain output. This implies that the Arduino change the “duty cycle” or amount of time that the signal was active. current sensor is used to read the current across the circuit, and to send this current as a voltage to the Arduino. The voltage is used in the Arduino, so that the Arduino can change the duty cycle. Arduinos cannot read current, so the current is seen as voltage; this was why the maximum voltage was desired from the circuit.

In addition, the voltage is read at multiple points across the circuit in order to obtain the greatest accuracy in the readings. This circuitry is based off small wind turbine systems. Testing using DC power supplies, oscilloscopes, and function generator were applied when testing the circuit to determine the correct operating range. The overall intention of the circuit is to provide the greatest current possible from the wind turbine to the load.

ENGINEERING DIAGRAMS

Each component used in the final wind turbine design was modeled using SolidWorks 2012. In addition, technical documents were produced for each component and are shown below. Due to the massive amount of technical documentation produced, only the most pertinent documents are included.

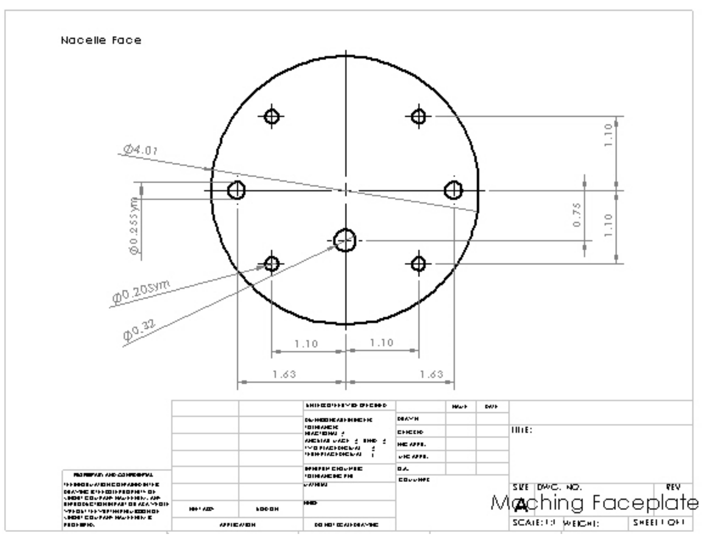

Figure 12. The front of the Nacelle.

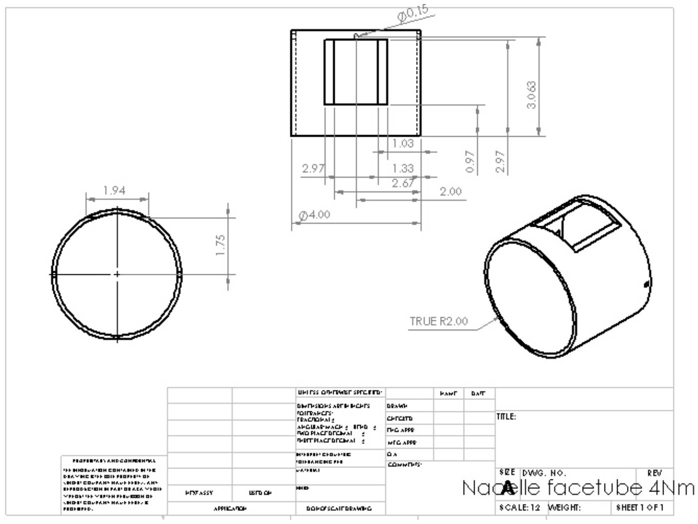

Figure 13. The tube of the nacelle.

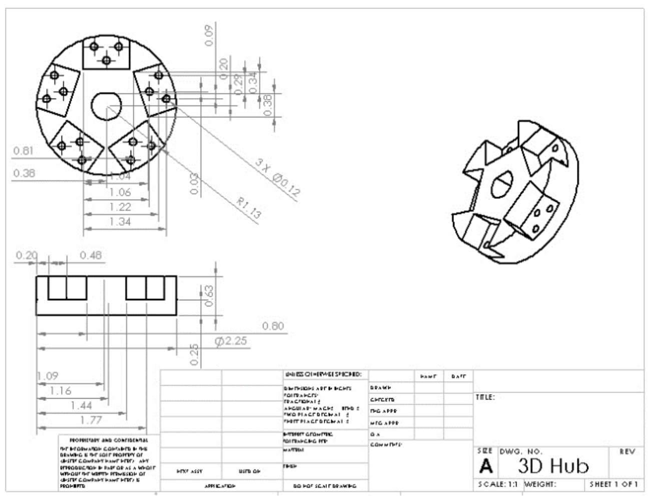

Figure 14. The hub for the blades.

The figures directly above were manufactured in-house. Each component is critical in order for the final design to function appropriately.

The circuit diagram for the control circuit prototype is shown below. The diagram has been broken down to show the exact detail of the Figure that was described earlier.

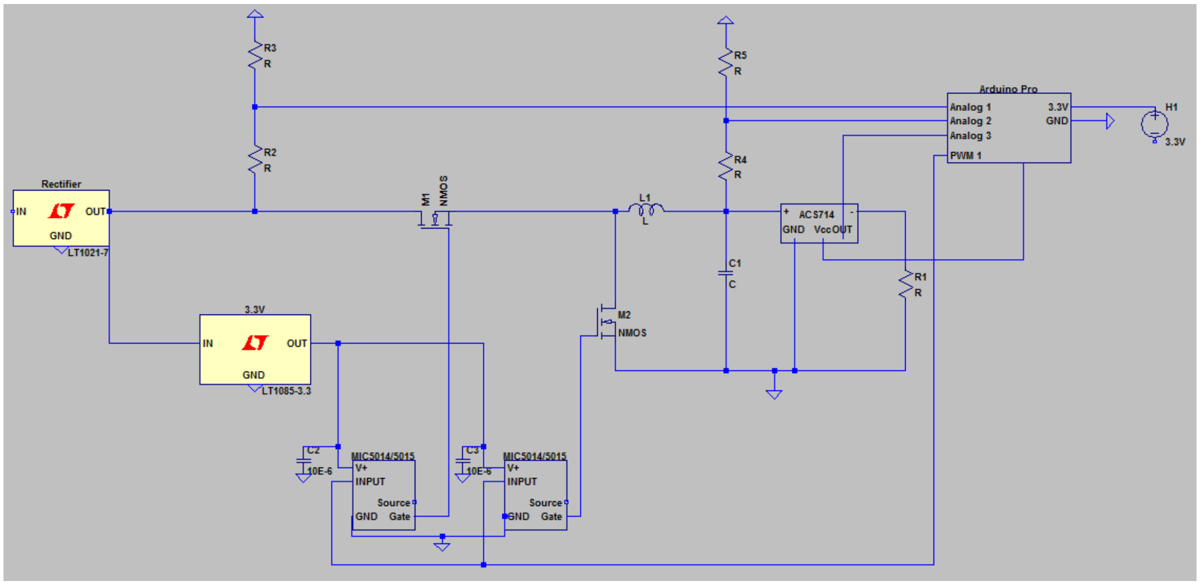

Figure 15: Circuit Diagram for Circuit Prototype.

Figure 15 presents the layout of the circuit was used in the turbine. The arrangement of MOSFETS, drivers, capacitors, inductors, and resistors were set up in such a way to allow the maximum current to the load. Two MOSFETs and two drivers were used. The use of the Arduino Pro was crucial to the circuit, as the Arduino Pro acted as the main voltage hub.

ENGINEERING SPECIFICATIONS

The first work package for the project was to obtain the general characteristics for the generator. As such, the testing rig shown in Figure 2 was produced. The equation used to determine the required RPM values shown in Figure 1 is as follows.

$$HertzCycle * Motor\ Ratio * 60s * G.R. = RPM \qquad \text{(Eq. 1)}$$

Where Hertz is the readout of the encoder, the cycle is $PulsesRev$ and G.R. is the gearing ratio used during testing. Appropriately, the power output was determined by multiplying the 5-V regulated voltage by the produced current at each RPM increment.

Shown below are derivations used to obtain turbine structure feasibility. Buckling and bending characteristics for the turbine tower were considered.

The bending derivation considered the turbine swept area as a circular disk. The circular disk would maximize the amount of drag experienced by the turbine. Reynolds number calculations were required to obtain the coefficient of drag used to determine drag force. Further, the drag force was assumed to be similar to a cantilever beam with a point load placed at the end of the beam. Thus,

$$Re = \frac{D_h * V}{v} \quad \text{(Eq. 2)}$$

where D_h is the characteristic length (Blade Radius*2 = 0.4m), V is the maximum expected wind velocity ($17 \frac{m}{s}$), and v is the viscosity of air at standard atmospheric conditions ($1.516*10^{-5} \frac{m^2}{s}$).

The Reynolds number was calculated to be greater than 10^4 thus the drag coefficient for a circular disk must be 1.1. The drag force was calculated as follows.

$$F_D = 0.5 * C_D * A * \rho * V^2 \quad \text{(Eq. 3)}$$

where C_D is the coefficient of drag, A is the swept area relative to the blade length ($0.126 m^2$), and ρ is fluid density (1.204 $\frac{kg}{m^3}$).

The cantilever beam equation used to calculate maximum deflection is presented below. Note that the drag force was converted into English units ($5.4 lbf$) and the assumed rule of thumb for appropriate structure safety is $(1/360)L$.

$$\delta_{max} = \frac{F_D * L^3}{3EI} = \frac{0.5 * C_D * A * \rho * V_w^2 * L^3}{3EI} \leq \frac{1}{360} L \quad \text{(Eq. 4)}$$

where L is the length of the tower (5.6”), E is the modulus of elasticity for 6061 aluminum ($10*106Psi)$ I is the moment of inertia for the circular cross sectioned beam ($0.2in4)$.

The max deflection was calculated to be 0.0002”. As a result, the structure safety deflection was 0.015”. Thus, the tower was appropriately specified to support the applied horizontal forces.

To account for possible column buckling under the vertical loads produced by nacelle weight, the following equations were used to determine if the chosen tower was effective. The turbine was assumed to be fixed-free structure, thus

$$S.R = kLr = kLIA \quad Cc \quad 2\pi 2E\sigma y, Cc > S.R. = Relatively\ Short\ Column \quad \text{(Eq. 5)}$$

where the S.R. is the slenderness ratio and Cc is the Column Constant. The Column Constant was greater than the slenderness ratio, thus the Johnson formula was used to calculate the critical load.

$$Pcr = A*\sigma y(1-\sigma yS.R.24\pi 2E) \quad \text{(Eq. 6)}$$

The critical load was calculated to be 17000 lbf. Since the nacelle weight is miniscule compared to the support capabilities of the tower, the tower is appropriate.

Running torque for the gearbox and generator couple was determined using the torque cell testing stand and NI LabVIEW. The start-up torque required for 25:1 gearbox is approximately 0.5 lbf. The running torque, when producing 10 W, is approximately 1.4 lbf.

Further, the operating conditions for the control circuitry is described sufficiently below. With the load being regulated at 5 volts, the amount of power generated relies solely on the current across the load. The power was determined using Ohms law where power equals voltage times current. When choosing the MOSFETS, drivers, capacitors, inductors, and resistors, a maximum of 15 amperes was selected as a requirement for the parts. This would ensure that the circuit would perform as intended without a short. With the constraint of an outside power being source limited to 3.3 volts, the controllers that could have been used were restricted. The Arduino Pro was chosen because it could operate off a 3.3-volt power source while also fulfilling all of the needs of the circuit.

CONCLUSIONS AND RECOMENDATIONS

Overall, the project was a success. Devices used for defining appropriate values correlated with wind turbine design were fabricated. prototype was developed utilizing the obtained values. Further refinement will be conducted during the next two weeks to ensure optimal performance and safety factors are met.

Recommendations for future work are summarized under the term "design process." In order to select an appropriate gearbox for use in wind turbine, the blade performance should first be determined. The gearbox can then be selected and tested to ensure that the select range of operating conditions can be obtained. After initial testing, refinement of the blade system should begin. The tower selection process should occur only after the known forces from drag and nacelle weight have been appropriately determined. After all safety guidelines have been met, field testing of the turbine is possible. Thus, an appropriate design process was the key to achieving working prototype within the given timeframe.

Additional engineering analysis and optimization is necessary in order to confirm or deny the feasibility of micro-scale wind turbines. Blade length contributes directly to the power production capabilities for a turbine. Thus, slightly longer blades could provide the torque required to spin higher-geared systems, thus increasing power output.

Appendix A – Sample of the Code Used in Arduino Microcontroller

```
/********************************************************
 * PID Simple Example (Augmented with Processing.org Communication)
 * Version 0.3
 * by Brett Beauregard
 * License: Creative-Commons Attribution Share-Alike
 * April 2011
 ********************************************************/

#include <PID_v1.h>

//Define Variables we'll be connecting to
double Setpoint, Input, Output;
int inputPin=0, outputPin=3;

//Specify the links and initial tuning parameter
PID myPID(&Input, &Output, &Setpoint,2,5,1, DIRECT);


unsigned long serialTime; //this will help us know when to talk with processing

void setup()
{
```

```
  //initialize the serial link with processing
  Serial.begin(9600);

  //initialize the variables we're linked to
  Input = analogRead(inputPin);
  Setpoint = 100;

  //turn the PID on
  myPID.SetMode(AUTOMATIC);
}

void loop()
{
  //pid-related code
  Input = analogRead(inputPin);
  myPID.Compute();
  analogWrite(outputPin,Output);

#ifndef PID_v1_h
#define PID_v1_h
#define LIBRARY_VERSION	1.0.0

class PID
{
```

```
public:

//Constants used in some of the functions below
#define AUTOMATIC    1
#define MANUAL        0
#define DIRECT  0
#define REVERSE  1

//commonly used functions
**************************************************************************
  PID(double*, double*, double*,       // * constructor.  links the PID to the Input, Output, and
      double, double, double, int);    //   Setpoint.  Initial tuning parameters are also set here

  void SetMode(int Mode);              // * sets PID to either Manual (0) or Auto (non-0)

  bool Compute();                      // * performs the PID calculation.  it should be
                                       //   called every time loop() cycles. ON/OFF and
                                       //   calculation frequency can be set using SetMode
                                       //   SetSampleTime respectively

  void SetOutputLimits(double, double); //clamps the output to a specific range. 0-255 by default, but
                                                                          //it's likely the user
will want to change this depending on
                                                                          //the application
```

//available but not commonly used functions
****

void SetTunings(double, double, // * While most users will set the tunings once in the

double); // constructor, this function gives the user the option

// of changing tunings during runtime for Adaptive control

void SetControllerDirection(int); // * Sets the Direction, or "Action" of the controller. DIRECT

// means the output will increase when error is positive. REVERSE

// means the opposite. it's very unlikely that this will be needed

// once it is set in the constructor.

void SetSampleTime(int); // * sets the frequency, in Milliseconds, with which

// the PID calculation is performed. default is 100

//Display functions ***

double GetKp(); // These functions query the pid for interal values.

double GetKi(); // they were created mainly for the pid front-end,

double GetKd(); // where it's important to know what is actually

int GetMode(); // inside the PID.

int GetDirection(); //

```
  private:
	void Initialize();

	double dispKp;				// * we'll hold on to the tuning parameters in user-
entered
	double dispKi;				//   format for display purposes
	double dispKd;				//

	double kp;                  // * (P)roportional Tuning Parameter
    double ki;                  // * (I)ntegral Tuning Parameter
    double kd;                  // * (D)erivative Tuning Parameter
	int controllerDirection;

    double *myInput;              // * Pointers to the Input, Output, and Setpoint variables
    double *myOutput;             //   This creates a hard link between the variables and the
    double *mySetpoint;           //   PID, freeing the user from having to constantly tell us
                                  //   what these values are.  with pointers we'll just know.
	unsigned long lastTime;
	double ITerm, lastInput;
	unsigned long SampleTime;
	double outMin, outMax;
	bool inAuto;
};
#endif
```