Wind Turbine Design Report

Group C4
Austa Jiang 40383168
Richard Chang 39989165
Jodi Gunawan 10980150
Jarvin Rivera 21685169

0.1 Table of Contents

0.2 List of Figures

0.3 List of Tables

1.0 Project description:

- Wind operated turbine
- Minimum of 90 degree winding tracking direction
- Delivers 12V DC power
- Sensor and actuator to track wind direction
- Maximum power point tracking
- Two independent control loops
- Maintain maximum output power
- 50x50x60 cm max

1.1 Requirements:

Each role is comprised of different requirements.

General	1. Minimum 90 degree angle sweep 2. Contained within a 50cm x 50cm x 50cm
Generator	● Minimum power output of between 3-5W ● Generate Minimum 6V Output (Power Electronics will step up voltage)
Power Electronic	● Generate 12V DC output (10V to 15V) ● Control blade speed to generate maximum power (through boost converter current control) ● Maximum Power point tracking
Control	● Create angle control for stepper motor movement using wind sensor input ● Ability to control blade speed ● Implements Maximum Power Point Tracking
Sensing/ Modelling	● Wind direction sensing (90 degree range) ● Voltage Sensor ● Current Sensor ● Maximum Power Point Tracking (change resistance for maximum power generation) ● Wattmeter displaying wattage

Table 1.1A: Table of requirements

1.2 Constraints:

In order to achieve success for this project, these constraints are considered in order to predict the best action to take:

Role	**Constraints**
General	• Budget constraints of maximum $500
Generator	• Will use PMSG • Material will be made out of metal • Will maximize magnetic flux and minimize eddy current losses
Power Electronic	• Optimizes power output efficiency • Will depend on generator early in the timeline
Control	• Utilizes hardware for PWM generation • Will not use a single board computer as the microcontroller • Will depend highly on the output of sensor components as controller input
Sensing/ Modelling	• All sensors have to be built, not purchased • Will require knowledge of Simulink

Table 1.2A: Contraints Table

1.3 Goals:

These are some goals we aim to achieve by the end of the project. These goals are prioritized by highest impact on the project functionality and labelled whether it has been completed by the end of the project:

1. 10W final power output
2. Backup battery power storage
3. Power all components such as microcontrollers using generated power
4. Function as a powered electric fan
5. Wireless communication
6. Email notification of specific wind turbine status
7. Display logged data (Voltage, Current, Power, Angle)

2.0 Power Electronics

The power electronics includes all the analog electronics/circuitry involved in the regulation, and flow of power in the wind-turbine system. This includes functions such as converting AC voltage to DC voltage, voltage regulation, and providing power for digital electronics such as microcontrollers.

2.1 Requirements, Constraints, and Goals

The initial requirements, constraints, and goals were defined, and carried till the end as follows:

Requirements:

- Convert 3-phase AC to DC
- Maintain 10-15V output voltage while experiencing input voltage variations
- Use a Pulse-Width-Modulation controlled boost converter regulate blade speed

Constraints

- Power losses through diodes used in rectification
- Power losses through inductor and switch resistance

Goals

- Additional functionality: Battery for powering electronics and for energy storage
- Additional functionality: Inverter circuitry to drive generator as a 3-phase motor using a DC power supply such as a battery
- Support up to 10W power output

2.2 Design iterations

Preliminary research:

In order to gain a clear understanding of what kinds of prototypes needed to be developed, preliminary research was done on what types of circuits may achieve the the requirements. (Note that research done during this stage was primarily targeted at meeting the requirements for the power electronics, and not the goals). This lead to a decision on using the following 3-phase full wave rectifier, and boost converter circuits as a basis for the power electronics design.

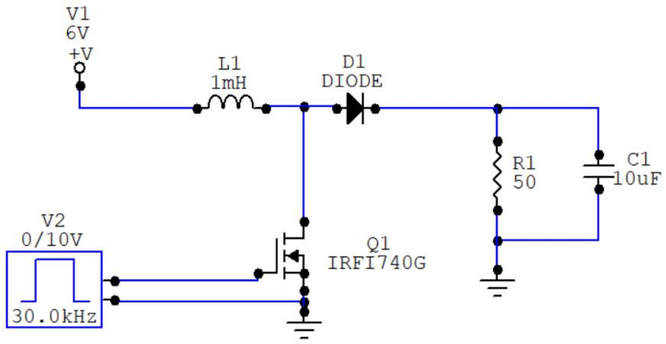


Figure 2.2A: Boost converter circuit

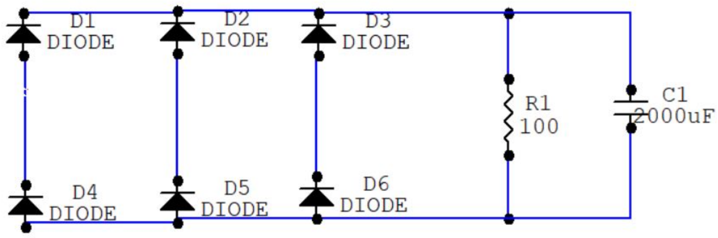


Figure 2.2B: 3-Phase full wave rectifier

The 3-phase full wave rectifier was chosen to fulfill the 3-phase AC to DC conversion requirement. The main performance metrics of such as circuit are the output ripple voltage, and the voltage drop through the diodes.

The boost converter was chosen to fulfill the output voltage regulation, and propeller speed regulation via pulse-width modulation requirements. To determine the component values used in this circuit, we use the following equations:

$Duty\ Cycle\ =\ V\,in/V\,out \qquad Imax = V\,out^2/(Rload * V\,in)$

$L = (V\,in * Duty\ Cycle)/(Iripple * freq)$

$C = (V\,in * Duty\ Cycle)/(V\,ripple * (1 - Duty\ Cycle) * Rload * freq)$

In order to obtain our circuit component values, we choose the following specifications:

$V\,ripple = 4\%$ $\quad V\,in = 6V$ $\quad V\,out = 12V$ $\quad I\,ripple = 20\%$ $\quad L = 1mH$

The Vin specification was chosen based on the expected output voltage of the generator. The Vout specification was chosen based on the requirement for the output voltage. Finally the inductance was preemptively chosen instead of obtained through calculations due to the accessibility and real world trade-offs of inductors available on the market. 1mH tends to be the highest inductance available while maintaining a low DC resistance (<1ohm), while maintaining a reasonable price that minimizes the monetary risk of each inductor purchase. Furthermore, we desire a large inductance because it increases the amount of energy stored between each cycle of the pulse-width-modulation signal, which allows for lower switching frequencies. This is important because higher switching frequencies are harder to implement efficiently on the controller side, and high switching frequencies also increases overall impedance of the circuit components.

Using these specifications, we obtain the following circuit component values:

$L = 1mH$ $\quad C = 8uF$ $\quad freq = 30KHZ$

Simulation:

Using the obtained component values, we move on the simulate the boost-converter circuit in circuitmaker to confirm that we have the correct theory of operation.

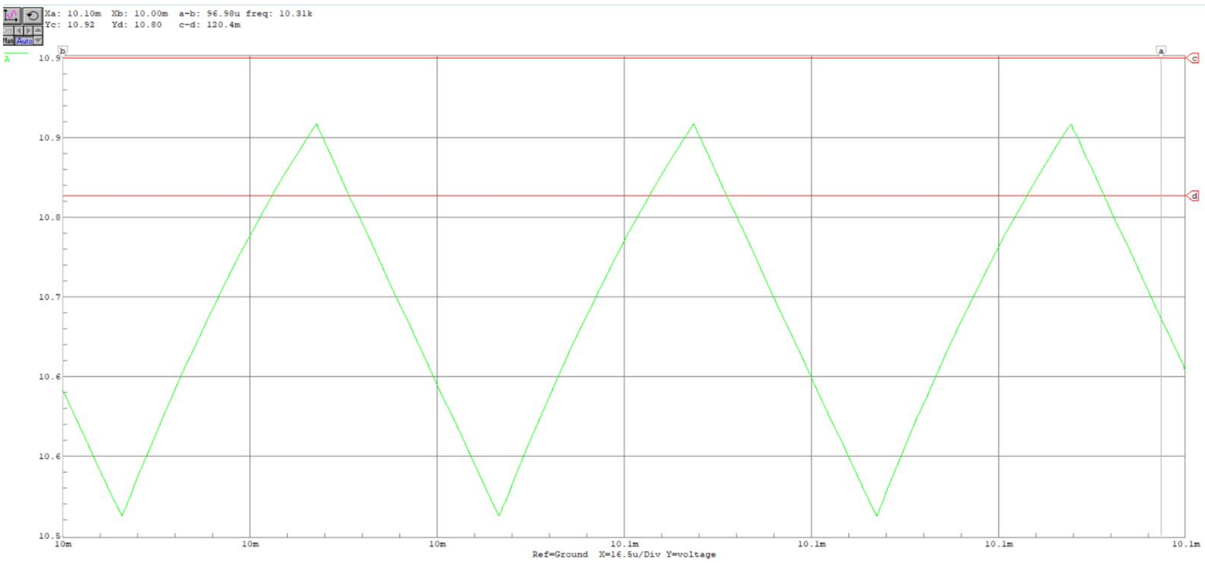


Fig 2.2C: Boost converter simulation

This simulation result shows that in theory, this circuit meets our requirements. With an acceptable ripple voltage, and output voltage mean at almost 11V, we see that the equations closely model the circuit (with the drop in output voltage due to the diode used in the simulation circuit)

Breadboard Prototype:

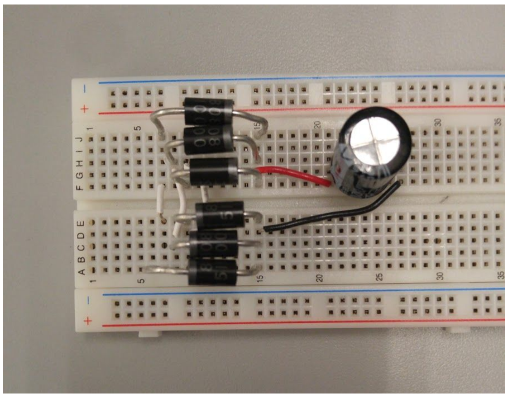


Fig 2.2D: Rectifier breadboard prototype

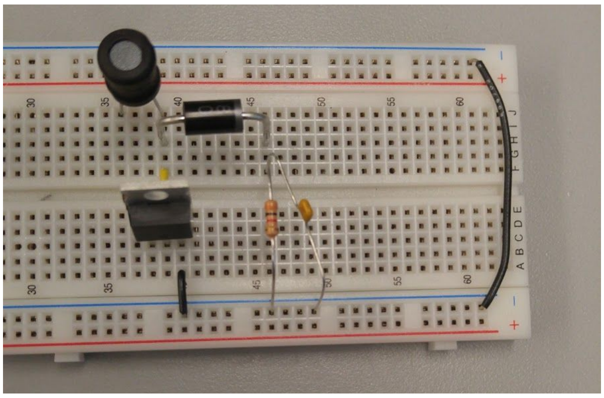


Fig 2.2E: Boost-converter breadboard prototype

Moving from the circuit schematics and simulations to the first breadboard prototype, required two additional component types to be determined: diodes and MOSFET. The diode was chosen as a high power schottky diode because of the potential 10W rating goal, and the low voltage drop which would increase the efficiency of the power electronics. The MOSFET was also chosen based on the power rating, as well as having a low drain-source resistance. The performance of the rectifier prototype was sufficient and fulfilled the 3-phase AC to DC conversion requirement.

Component values for the boost-converter were not chosen as those determined in the calculations because of the hasty nature of this prototyping method, and the availability of components. The inductor used in this prototype had 50mH inductance, and very high 100ohm resistance. These values resulted in very poor performance of this boost converter prototype, but demonstrated the basic principles of operation as DC input voltages were able to be "stepped-up" to much higher voltages.

Aside from the improper boost-converter component values chosen, the main limitation of the breadboard prototype was the inability to perform comprehensive tests due to the power rating of the breadboard itself. Tests could not be safely conducted at full voltage and load.

Perfboard Prototype:

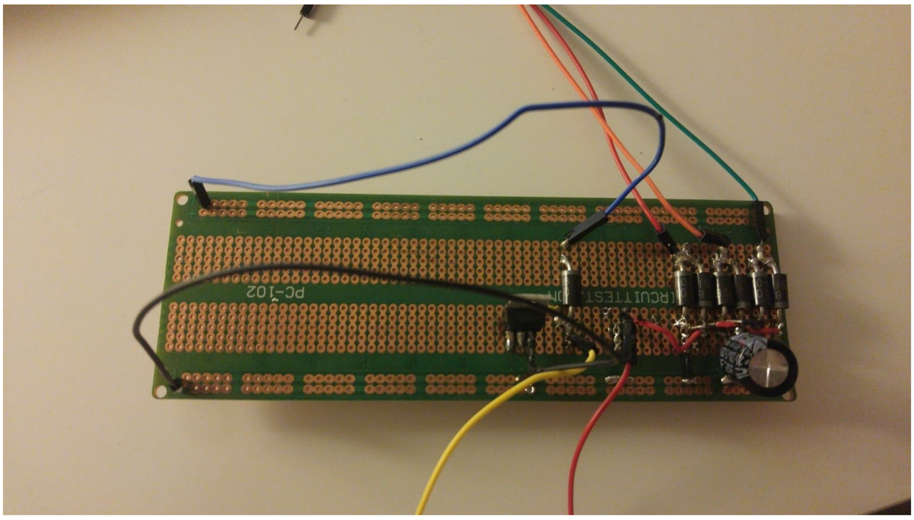


Fig 2.2F: Perfboard prototype

When moving to the perfboard prototype, the primary design objective was to create a testing platform that could support comprehensive testing. This meant having sufficient power ratings to perform tests at full voltage and load. Additionally, jumper cable leads extending from the perfboard in place of components such as the load and inductor allow for easy switching of components for testing. This prototype also features the rectifier integrated with the boost converter to allow for testing the full electronics system from generator to output.

Using this perfboard prototype, the component values determined above, during preliminary research, were confirmed to work properly. Additionally, this prototype was used along with the starter-generator to prove that the entire system works properly.

PCB Prototype:

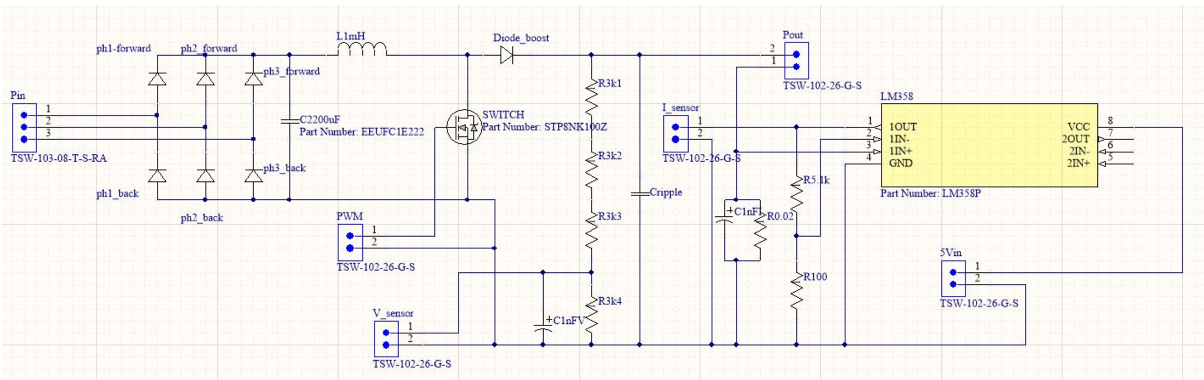


Fig 2.2G: Schematic for PCB Prototype

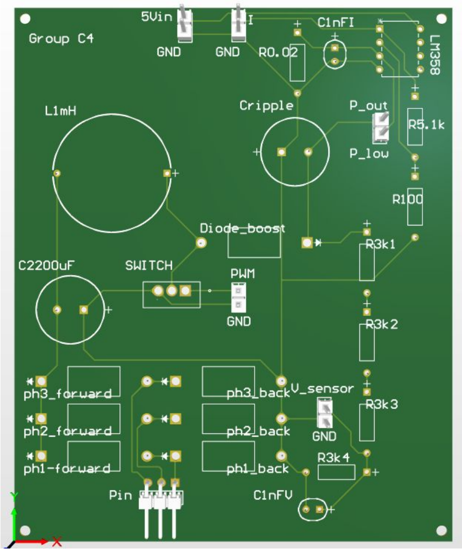


Fig 2.2H: PCB Prototype 3D Render

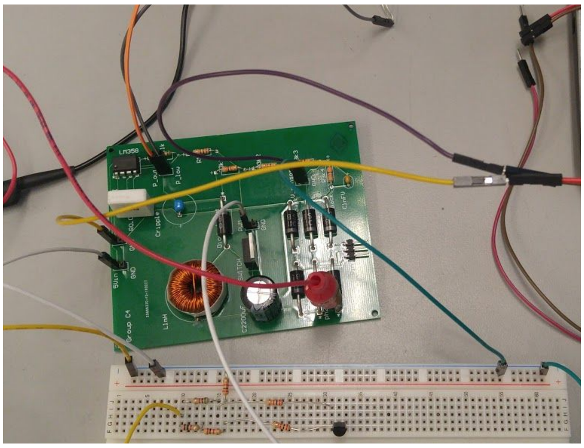


Fig 2.2I: PCB Prototype

The PCB prototype features the voltage and current sensors integrated into the rest of the power electronics. This prototype features input pins for the 3-phase output from the generator, an input pin for the pulse-width-modulation signal from the controller, and output pins from the voltage and current sensors.

This prototype was tested to work properly and meets all requirements, as well meeting the goal of 10W capability due to the trace width and component ratings chosen.

2.3 Final Design Description

The final design for the power electronics is comprised of two separate printed circuit boards, and meets all the requirements and goals that were initially set. The first PCB is similar to the last PCB prototype, and fulfills all the base requirements but also features some improvements that allow it to more easily integrate with the rest of electronics, and improves the ease of testing. The second PCB fulfills the goal of being able to drive the generator as a motor.

Base PCB:

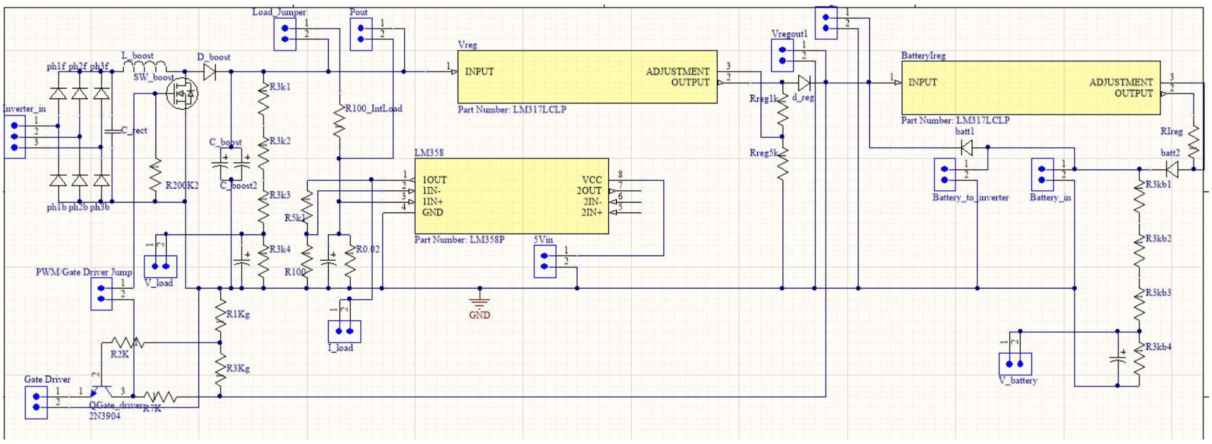


Fig 2.3A: Power electronics base PCB schematic

This PCB features the same rectifier, boost converter, and sensors as the last PCB prototype, but also features:

- **Optional gate driver** to allow the controller to drive the boost-converter MOSFET. This option can be used depending on the presence of a pin jumper, and choice of the specific pin that the controller attaches to. This optional feature allows for more versatility depending on whether the specific microcontroller used is capable of 5V or 3.3V output.
- **Optional integrated load** of a fixed resistance that can be enabled or disabled easily by connecting a jumper across two pins. This option improves convenience during testing.
- **Regulated 7.8V** output to power the microcontroller(s) when generator output power is sufficient (fulfills one of the goals)
- **Regulated 100mA, 7.8V battery charging port** (fulfills one of the goals)
- **Microcontrollers powered by battery** when generator output power is low

Note that when this circuit behaves such that when generator output voltage is low(<8V), microcontroller port is powered by batteries. When generator output voltage is high, the microcontrollers are powered by the generator, and the battery is charged by the generator.

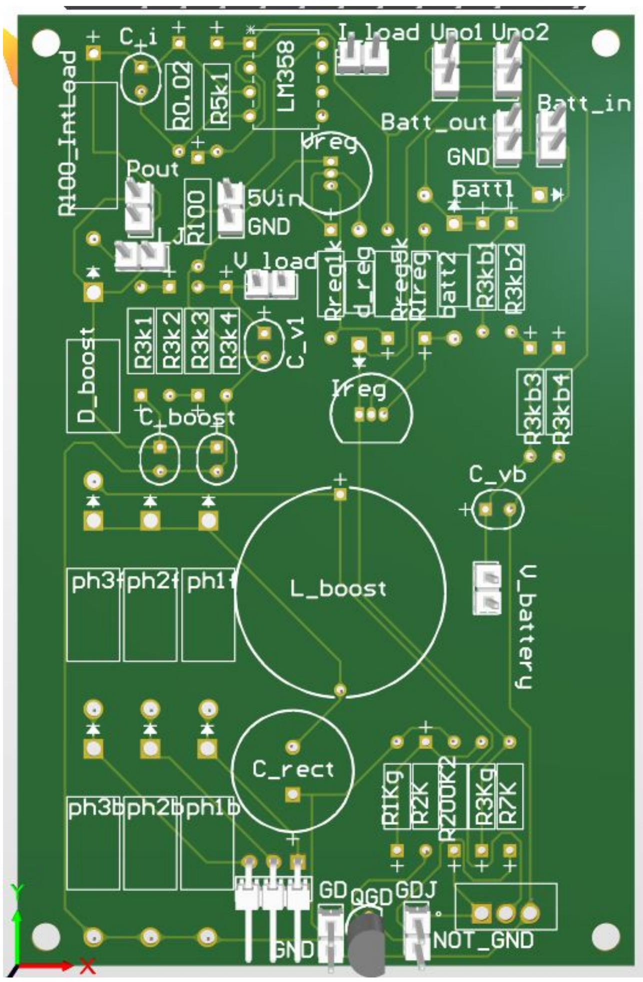


Fig 2.3B: 3D render of base PCB

Fig 2.3C: base PCB

3-Phase Inverter PCB:

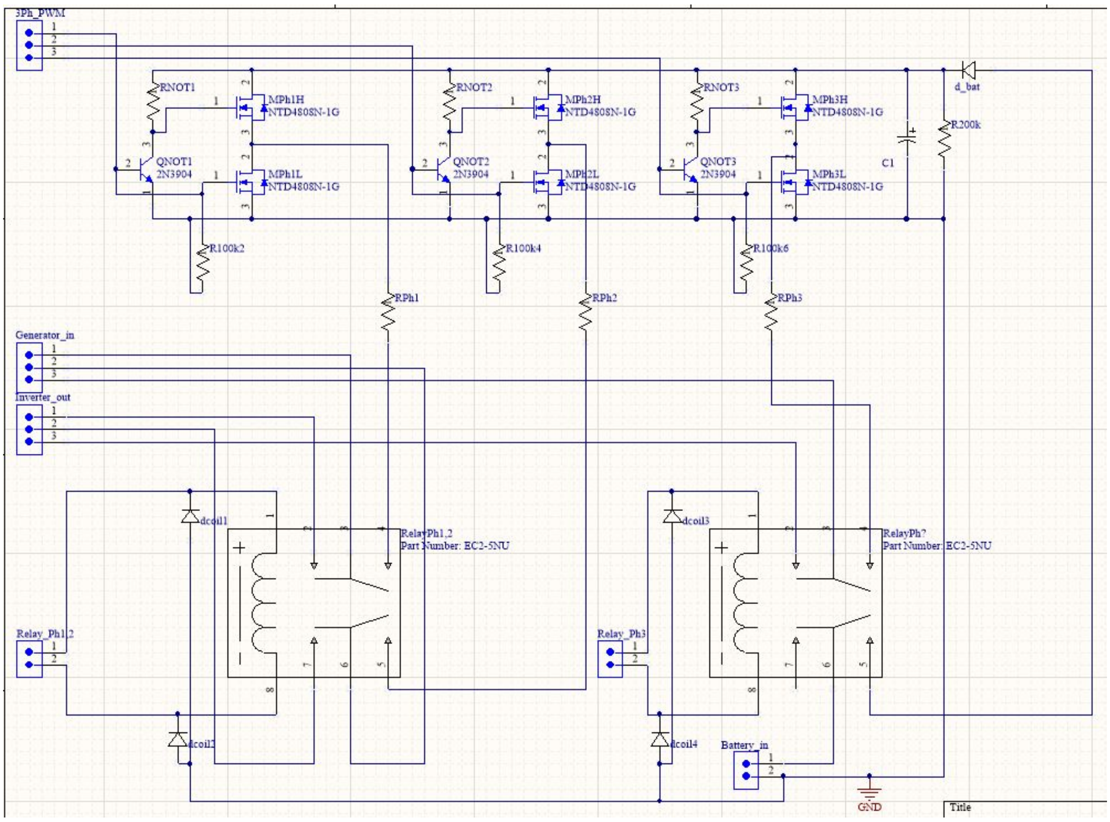


Fig 2.3D: 3-Phase Inverter PCB schematic

This circuit operates by taking a 3-phase square wave signal from the microcontroller, and using it to drive three separate complementary MOSFET pairs, which pulls each output phase to the DC source voltage or to ground. In other words, this circuit amplifies 3-phase square wave reference signals from the controller to drive the generator. Additionally, this circuit features:

- **Latched relays** to switch the generator between connecting to the base PCB (rectifier and boost-converter) or connecting to the inverter circuitry. Latched relays were chosen to decrease the power expenditure through the coils of the relays.
- **BJT NOT gate** to create the complementary MOSFET pairs using only N-type MOSFETs.
- **Battery port** which is used as the DC source to be inverted into 3-phase.

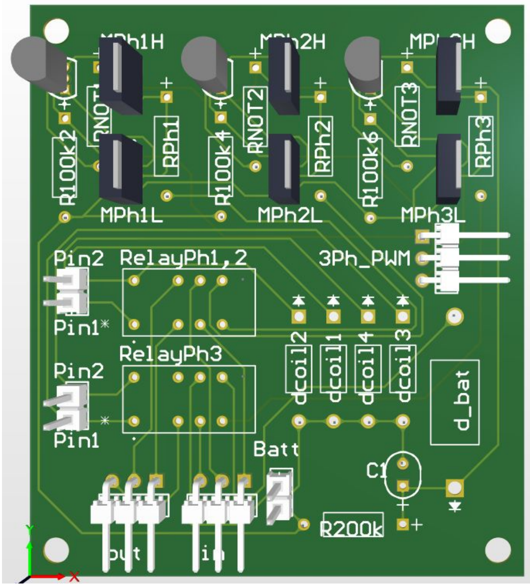


Fig 2.3E: 3D render of inverter PCB

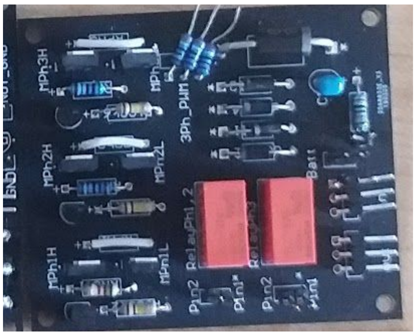


Fig 2.3F: Inverter PCB

2.4 Performance and Validation

Base PCB:

The performance of the base PCB was evaluated against the following metrics:

Requirements:

- Convert 3-phase AC to DC
- Maintain 10-15V output voltage while experiencing input voltage variations
- Use a Pulse-Width-Modulation controlled boost converter regulate blade speed

Goals

- Additional functionality: Battery for powering electronics and for energy storage
- Support up to 10W power output

Specifications

- 4% voltage ripple

Because the rectification stage of the PCB was proven to work many times before, the rest of the functionality of the circuit was tested by using a power supply as the input voltage to the boost-converter. The results showed that all requirements were met when the boost-converter was given a 30KHZ, approximately 50% duty cycle pulse-width-modulation input signal.

With the power supply used in testing, the goal of battery power storage and “generator” powered electronics was also proven to be fulfilled. However, it was found that in order for the output of the boost converter to power both the microcontrollers, and charge the battery at the same time, **the output power from the generator would have to be above 1W.** This would later prove to be too high of a requirement for the generated procured in this project. The ability to power the microcontrollers using the battery was also tested to be working.

3-phase inverter PCB:

The performance of the 3-phase inverter PCB was tested based on the following metric:

Goals

- Additional functionality: Inverter circuitry to drive generator as a 3-phase motor using a DC power supply such as a battery

This metric would be more specifically evaluated based on the circuit’s ability to provide a powerful 3-phase signal when connected to a Y-connected inductive load. This was first tested by connecting a Y-connected inductive load on a breadboard and using a microcontroller to provide a 3-phase reference signal. Each phase of the load was then probed with an oscilloscope and proved to be operating properly with each phase being a distorted square wave that were phase shifted from each other.

This test revealed that there was a lack of resistance between the base of the BJT to the 3-phase microcontroller input, causing a short circuit. This issue was fixed by soldering resistors directly to the input pins. Another issue that was revealed was that the neutral node of the Y-load needed to be connected to the ground of the circuit.

The second phase of the test involved testing with the actual generator. With a small amount of power being applied (approximately 3V, 0.3mA) the shaft of the generator was observed to vibrate which proved that electrical power was being successfully converted into mechanical power, however, due to the limited current ratings of the generator windings, more power could not be safely put into the generator without causing potential damage. In order to create enough torque on the shaft to cause rotation, more power would have been required. Thus, this part of the testing is inconclusive.

Overall, the final version of the power-electronics was confirmed to fulfill every requirement and goal except for the ability to drive the generator as a 3-phase motor. That capability is not completely tested to be operational, however its capability of inverting DC power into 3-phase power is confirmed through testing.

3.0 Controls

3.1 Requirements, Constraints, and Goals

The requirements, constraints, and goals for controls design throughout the project are as follow:

Requirements:

- Utilize boost converter voltage and current sensor input to generate MPPT algorithm which varies PWM duty cycle for maximizing power extraction
- Utilize wind direction sensor input to drive stepper motor and move the turbine

Constraints:

- Hardware generated PWM
- No external computer or Raspberry Pi utilized in the design of main turbine functionality

Goals:

- Non blocking code
- Running on the same microcontroller
- Microcontroller board powered by battery
- Proportional Control design
- 50kHz frequency PWM

Additional Functionality:

- Display parameters on LCD
- Bluetooth Communication between 2 Arduinos and Manual Control
- Fan Mode Control

3.2 Design Iterations

After analyzing the requirements, constraints and goals for the design, Arduino Uno is chosen as the main microcontroller. The Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. The main significant aspect of this microcontroller is the internal timer that is required to generate non-software/hardware PWM.

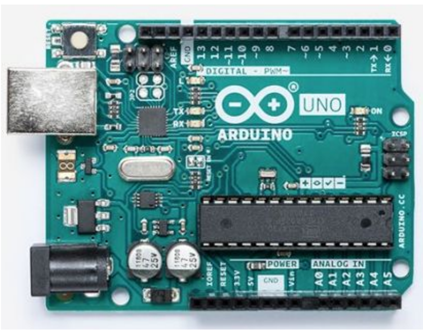
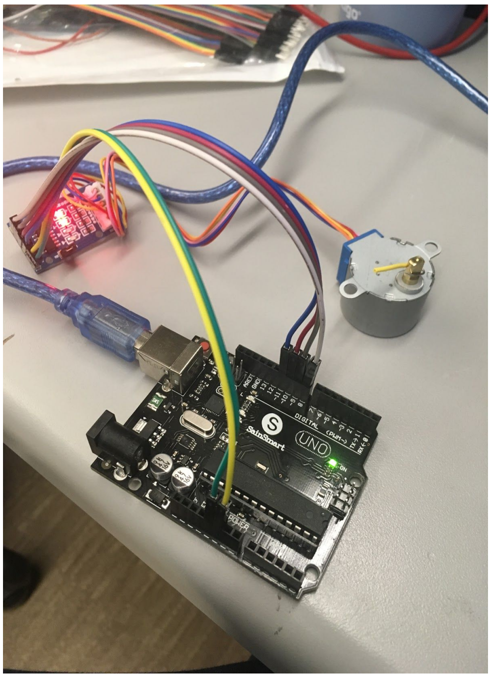


Figure 3.2A: Arduino Uno

Controlling the stepper motor:



Iteration #1:

The output of the position sensor as the input for controlling the stepper motor provided was initially within a range of -345 degree to 345 degree which allows reading movements from both clockwise and counter-clockwise. In order to create an effective algorithm, a dynamic module which calculates the angle difference between first state and next state of position sensor is required. This value is then used to move the stepper motor.

Iteration #2:

After changing the position sensor to potentiometer, the angle is restricted to between 0 and 180 degree and therefore will provide an absolute value. Similar logic was used as iteration #1. Algorithm for these iterations are as follow:

```
#include <Stepper.h>
Stepper stepper(STEPS, 8, 11, 10, 12);
void move_stepper(double angle_curr){
	if (angle_curr != angle_prev){
		angle_diff = angle_curr - angle_prev;
		stepper.step(angle_diff*CALIBRATION_CONST);
		angle_prev = angle_curr;
	}
}
```

Maximum Power Point Tracking:

Iteration #1:

Initially, it was not possible to allow testing without circuits and generator. A method that allows controlling a non-blocking, hardware PWM duty cycle is created using Voltage from the Voltage Generator. The algorithm is as follows:

```
#include <TimerOne.h>
void display_pwm(){
	float volt = analogRead(A0) * (5.0 / 1023.0); // voltage ratio for Arduino
	dutyc = 1024/(volt); // Max duty cycle is 1024/1024
	Timer1.pwm(9, dutyc); // Utilize Timer1 library to use Arduino hardware clk
}
```

Iteration #2:

The perturb and observe algorithm is used as a foundation for the maximum power point tracking. With the input of voltage and current from the sensors, it is possible to calculate the power. It is then important to store the previous state values before 'perturbing'. The next state will require reading the voltage and current again and referring to the previous stored value. As a result, duty cycle is adjusted based on where the maximum power is. It will increase if voltage is higher and power is higher and vice versa. The graph showing this algorithm is as follow:

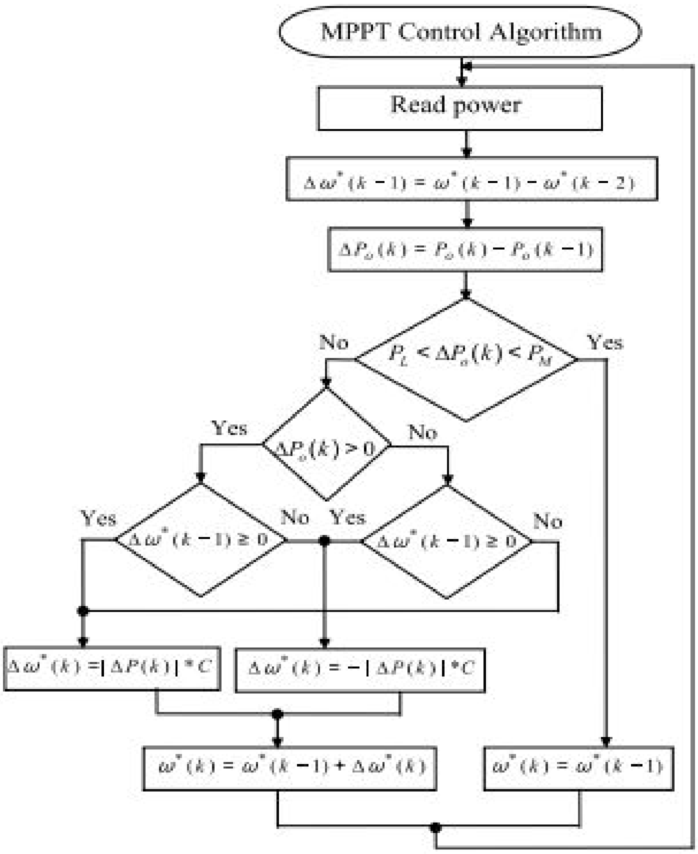


Figure 3.2B: Maximum Power Point Tracking Algorithm

Fan Mode Control:

Creating a fan mode using the design setup requires 2 parts in the control aspect. First, the microcontroller requires sending a signal to 2 relay switches that changes the power electronics to fan mode. Having to use Arduino, with a lot of pins, allows sending 2 pairs of HIGH and LOW signals to create this attitude. Second, in order to power the fan, a 3 phase PWM is required.

Bluetooth Control:

The bluetooth module used for this feature is the HC-06 Bluetooth Module. This module allows communication between a mobile application and the arduino. The main feature was to create a priority logic which reads the input of the mobile application and send signal to Arduino based on the input. These inputs includes: changing from generator/fan mode (sends a flag to another arduino with the status on the mode) and manual control of stepper motor (sends an angle value to move the stepper motor).

3.3 Final Design Description

The final design of the control algorithm for MPPT and position sensor is then finalized when integrated together with power electronics and sensors. In terms of controlling the stepper motor, additional change is required because a bigger gear is used to move the turbine direction. Measured dimension is approximately 4x bigger than the angle of the smaller, initial gear. Therefore a higher calibration value is required:

```
#include <Stepper.h>
Stepper stepper(STEPS, 8, 11, 10, 12);
void move_stepper(double angle_curr){
	if (angle_curr != angle_prev){
		angle_diff = angle_curr - angle_prev;
		stepper.step(angle_diff* 4 *CALIBRATION_CONST);
		angle_prev = angle_curr;
	}
}
```

For MPPT, an additional function to sweep the entire duty cycle is created. This function checks what is the maximum voltage and power value can be achieved from a specific duty cycle value. This allows for better testing and validation of the MPPT logic. The algorithm for this function is shown below:

```
void check_max(){
	double current = current_sensor(A3,global_currents);
	double voltage = voltage_sensor(A2,global_voltages);
	curr_power = current * voltage;
	Timer1.pwm(9, tdutyc);
	if(curr_power > prev_power){
		if(curr_power > maxp){
			maxp = curr_power;
		}
	}
	prev_power = curr_power;
	if(tdutyc > 1024){
		Serial.println(maxp);
		tdutyc = 1024;
	}
	tdutyc = tdutyc + 7;
}
```

A function which increases the perturbation of the MPPT is also added. The non-linear power graph has many local maxima and minima, which will make the MPPT logic stuck on a local maxima if it is found.

3.4 Performance and Validation

By using the proportional design, it allows less calculation especially since the stepper motor is very accurate. The design was also able to perform with 25 samples / second which is pretty fast and it compensates for the time it takes to find maximas and perturbation. In the end, we were able to use MPPT to find the maximum power and voltage which is around 0.6W and 10V. The time it takes to get to the maximum power is less than 15 seconds. The movement of the position sensor and the stepper motor is very accurate with a range of 180 degree and controllable through bluetooth. The movement of the stepper motor is uninterrupted and almost instant. The design of the algorithm itself did not change very much throughout the project however, a lot of testing was done to ensure that everything is working together.

The performance and validation method of each performance from each aspect of controls is described below:

1. Performance of stepper motor movement:
    a. Fast movement high accuracy, 180 degree range, controllable manually through mobile application, movement almost instant, no oscillation
    b. Tested by manual testing and Serial printing on Arduino IDE
2. Performance of MPPT:
    a. 25 samples / second, able to reach maximum within at least 15 seconds
    b. Tested by comparing the value of maximum voltage and power with the value found from sweeping the model
3. Performance of Fan Control:
    a. Able to generate 3 PWMs with different phase and changes relay switch
    b. Tested by manual testing and probing the pins of the Arduino. Although did not fully functional, a small vibration can be observed on the turbine from the 3 phase PWMs.

4.0 Generator

4.1 Requirements, Constraints, and Goals

The initial requirements, constraints, and goals were defined, and carried till the end as follows:

Requirements:

- Convert rotational energy into electricity
- Permanent Magnet Synchronous Generator
- Generate 6V AC

Constraints

- Compact, cost efficient
- Minimize losses (i.e. frictional losses, eddy current losses)
- Size restraint

Goals

- Maximize flux
- Optimize poles to be able to operate at lower speeds
- Optimize synchronous speed

4.2 Design iterations

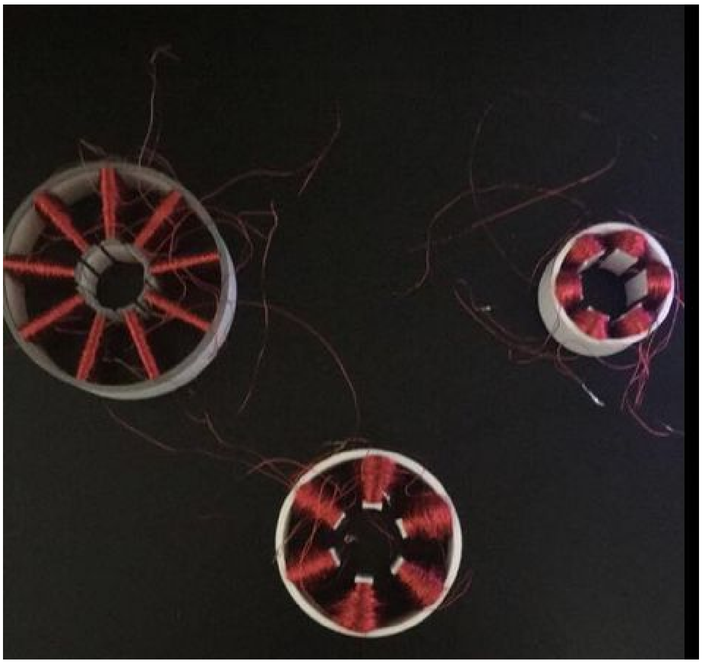


Figure 4.2A: Stator iterations

Multiple different stator design configurations were tested each varying with size, winding ratio, poles, etc. The goal was to optimize the peak voltage and current while still meeting the requirements.

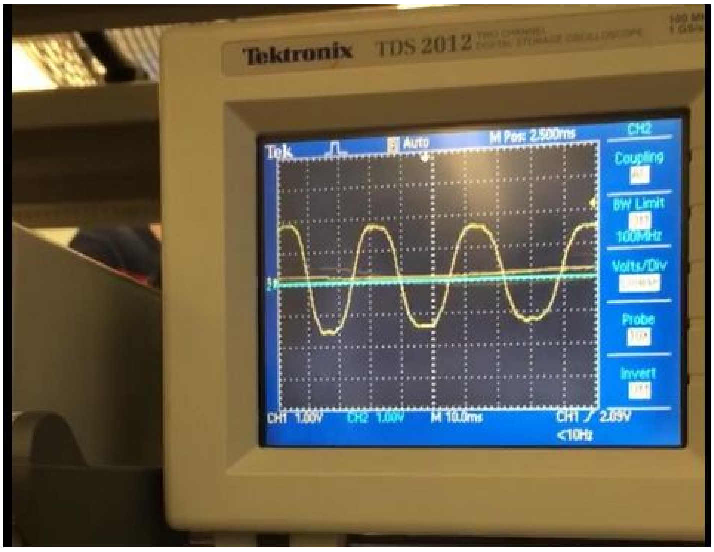


Figure 4.2B: Prototype-1 waveform

The follow waveform corresponds to the performance of the right prototype in the image above which was prototype #1. As the waveform shows, we obtained a peak voltage of 2V at a relatively low rpm with 30mA of current.

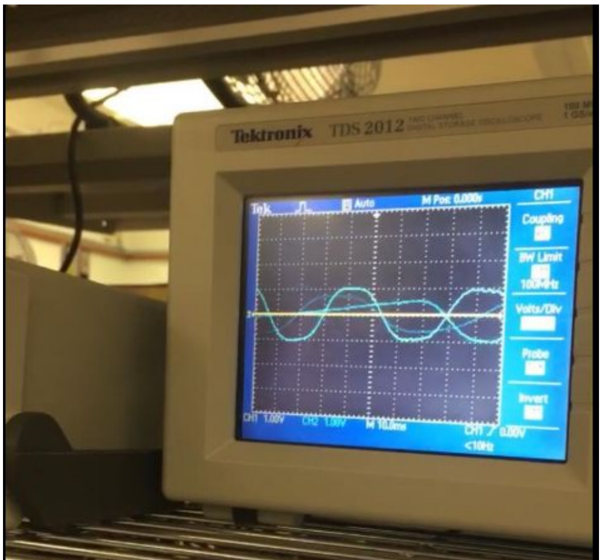


Figure 4.2C: Prototype-2 waveform

For the next iteration (middle), we increased the size of stator in order to allow more turns. As was done as well with the left iteration with more poles. However, in doing so, this increased the distance of the windings from the magnetic field of the rotor, since the prototypes were all made of plastic, this was essentially increasing the air-gap and thus the performance of the prototypes both suffered with slightly less peak voltages and current at a faster rpm than the first prototype. For the final design, we optimized the turn ratio using the equation below and having the stator made out of steel increased the permeability by orders of magnitude which in turn increased flux thus increasing voltage as they are proportional.

$$E_{1,rms} = 4.44 f N_1 \Phi_{peak}$$

Figure 4.2D

For the equation above we consider the voltage induced by two coils on one another, but in our case, our flux comes from a rotating magnetic field as opposed to another coil. Knowing that we want a peak Voltage of 6 to be boosted, we can solve for the optimum number of turns, poles, etc.

4.3 Final Design Description

Figure 4.3A: Final stator

This generator above was encased in the final design with the following specifications:

Stator:

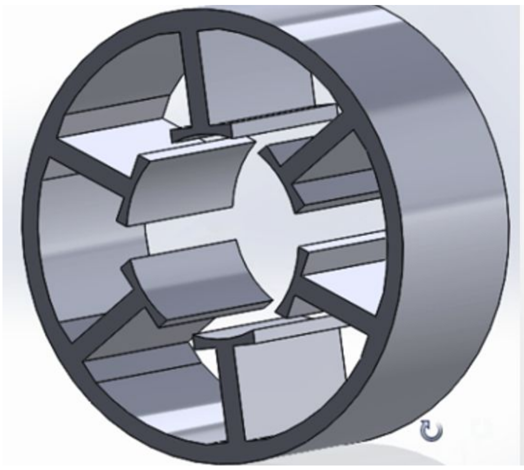


Figure 4.3B: Final stator design

- 6 stator windings (200 turns)
- 8 copies ~ 27 mm
- 26 AWG wire

Rotor:

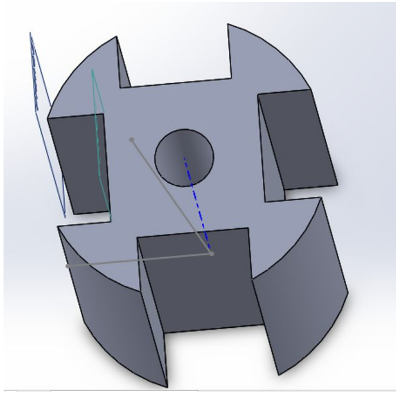


Figure 4.3C: Final rotor design

- 4 magnetic poles
- Large surface area (250mm squared)
- 1.5 mm air gap between magnets

Different rotor options were not considered during iterations because this rotor design optimized the surface area which in turn would optimize the flux while also allowing the generator to operate and at lower speeds. The goal was to keep the generator as compact as possible in order to reduce the cost as much as possible while maintaining functionality.

4.4 Performance and Validation

Parameters:

- Maximum Peak Voltage: 9V peak at 1500 rpm
- Peak Current: 0.2 A
- Synchronous Speed (60 Hz): 1800 rpm
- Winding resistance: 2 ohms

At the lowest wind speed, (around 900 rpm), the peak Voltage was approximately 5.5 Volts with a 0.1 A peak which satisfied the boost converter specifications and minimized the loss as the path of reluctance was reduced.

5.0 Sensing and Modeling

5.1 Requirements, Constraints, and Goals

The main requirements, constraints, and goals for sensing and modeling are briefly outlined below.

Requirements:
- Accurately sense wind direction
- Sense voltages and currents and convert them to a readable signal
- Model control loops for position control and power tracking
- Map out the perturb and observe algorithm

Constraints:
- Percentage error of resistors
- Power losses in ICs and resistors
- Accuracy of readings limited by arduino (reads with an accuracy of 5mV)
- Performance of IC is voltage dependent

Goals:
- Minimize power losses in resistors and ICs
- Minimize effects of noise
- Sense current to at least 0.01 A of accuracy
- Sense voltage to at least 0.1 V of accuracy
- Have sensors operate at least a maximum of 12Vdc and 0.5A
- Have the position sensor detect at least 10° angle changes

5.2 Design Justification and Calculations

Below each design is individually explained; design choices are justified by calculations and other specific reasons.

Voltage Sensor

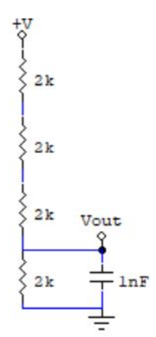


Figure 5.2A: Voltage sensor schematic

Justification:

The arduino's max readable voltage is 5Vdc, hence the output of the load (Max 12Vdc) must be stepped down. We choose to use a voltage divider because it is the cheapest option, easily replicable (if we need multiple voltage sensors), and has enough accuracy for our purposes. We use a voltage divider with a one fourth ratio so that at max voltage, this will map to 3Vdc. We choose a one fourth ratio over a one third ratio since voltage at a one third ratio the max voltage will be 4Vdc meaning that if our load reaches voltages above 12V this may cause issues. We do not choose a ratio lower than one fourth since the arduino can only read to an accuracy of 5mV meaning that the lower the voltage divider ratio, the lower the accuracy. For resistances, we must choose a low value because the reading of the arduino becomes less accurate as the output resistance seen by the arduino increases. However, we must also take into account that as the resistance decreases the power losses will increase, hence we choose a middle ground of 2k ohms. A capacitor is used to filter out any noise as the voltage of the load fluctuates.

Calculations:

Max power loss (assuming 12Vdc at load)

$$\frac{V^2}{R} = \frac{12^2}{8k} = 0.018\ \mathrm{W}$$

Precision

$$\frac{V_{in}}{V_{out}} = 4 \rightarrow V_{in,min} = 0.02\ \mathrm{V}$$

Current Sensor

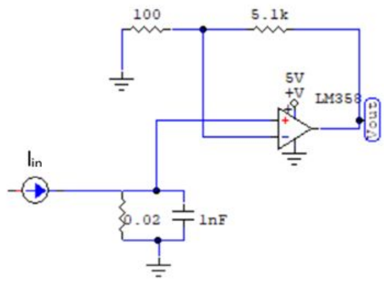


Figure 5.2B: Schematic of Current Sensor

Justification:

We must take into account the max current the load receives and the output voltage to be read by the arduino. We choose to use the LM358 because it is able to run on low voltage (hence drains less power), is a cheap option, and outputs consistent values. To use the op-amp, we send the load current directly through a resistor and step up the voltage across the resistor; the voltage is then read by the arduino. We choose a low resistance for the resistor in series with the load current to reduce power losses. A low resistance also has minimal effect on the load current. Since the voltage across the 0.02 ohm resistor is so low, we need a reasonable step-up ratio. We choose the 5.1k ohm resistor to reduce power losses and a 100 ohm resistor so we have a decent gain of 52 V/V. A capacitor is placed in parallel to the 0.02 ohm resistor to reduce noise effects.

Calculations:

Max power loss (assuming 1A of load current)

$$V_{out,max} = (1)(0.02)(1 + \frac{5100}{100}) = 1.04 \text{ V}$$

$$P_{max\ resistor,losses} = (0.02)(1) + \frac{0.02^2}{100} + \frac{(V_{out,max} - 0.02)^2}{5.1k} = 0.02 \text{ W}$$

$$P_{op\text{-}amp} = (1.04)(\frac{V_{out,max} - 0.02}{5100}) = 0.000208 \text{ W}$$

$$P_{net} = P_{op\text{-}amp} + P_{max\ resistor,losses} = 0.02 \text{ W}$$

Precision

$$I_{in,min} = \frac{V_{out}}{(0.02)(1 + \frac{5100}{100})} = 0.005 \text{ A}$$

Position Sensor

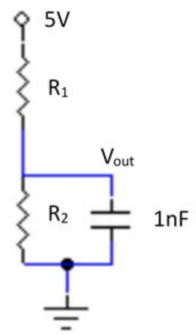


Figure 5.2C: Schematic of Position Sensor

Justification:

There are two main inexpensive options for position sensors: the rotary encoder and the potentiometer. The potentiometer is chosen over the rotary encoder because the rotary encoder is sensitive to undesirable small mechanical oscillations in the wind vane. These oscillations are difficult to accomodate in code with the rotary encoder, but with the potentiometer this issue can be easily fixed in software. We choose a 10k ohm potentiometer to reduce power losses, and to improve the angle precision we use the 5Vdc pin of the arduino over the 3.3Vdc pin. A capacitor is placed in parallel with Vout to reduce noise effects.

Calculations:

Power losses

$$\frac{5^2}{10k} = 0.0025\ \mathrm{W}$$

Precision

$$\frac{V_{in}}{V_{out}} = \frac{5}{0.005} = 1000 \rightarrow \frac{V_{out}}{V_{in}} = \frac{1}{1000} = \frac{10}{10k}$$

We can see above that the position sensor is precise enough to read 10 ohm changes in the resistor ratio leading to an accurate angle reading.

5.3 Control Loop Simulations

Below we will discuss the perturb & observe algorithm, position control loop, and power tracking loop.

Perturb & Observe Algorithm

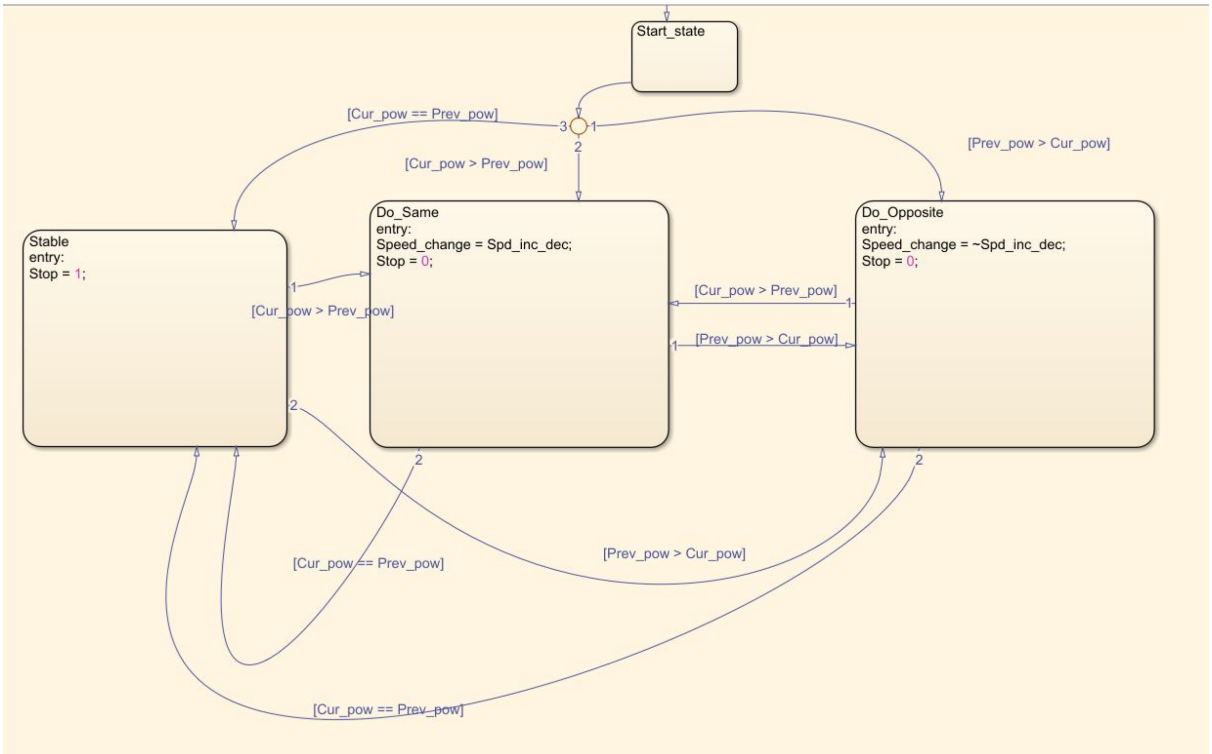


Figure 5.3A: Perturb & Observe

We choose the perturb and observe algorithm because it is a consistent algorithm implementable with the arduino. As shown above, the arduino sends a PWM and the change in power is observed. If the power increases, we continue sending the same PWM until it decreases. If the power decreases, we then send a different PWM.

Position Control Loop

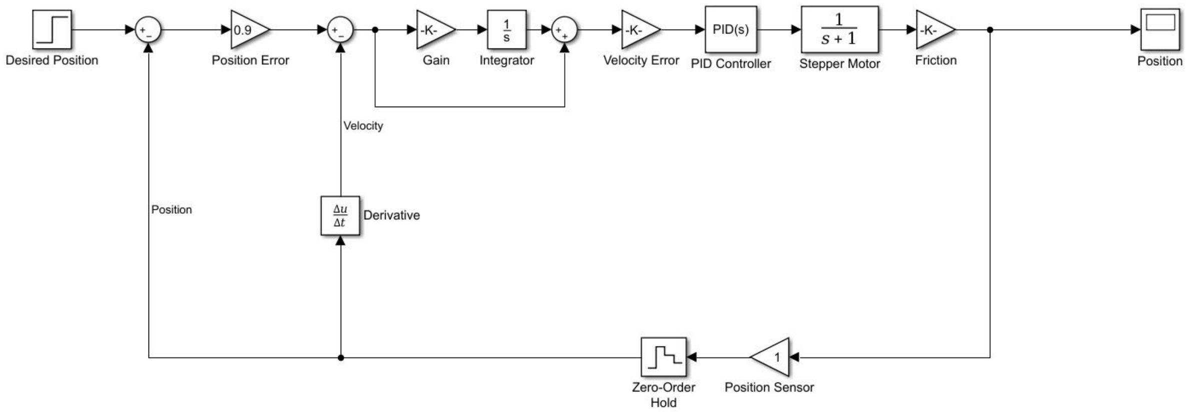


Figure 5.3B: Position Control

The above control loop outlines the control of the position of the turbine blades. If the wind is blowing in a direction away from the turbine blades, the desired position is input to the position sensor. The arduino then takes this desired position as an input and continually runs the stepper motor until it reaches its desired position.

Power Tracking Loop

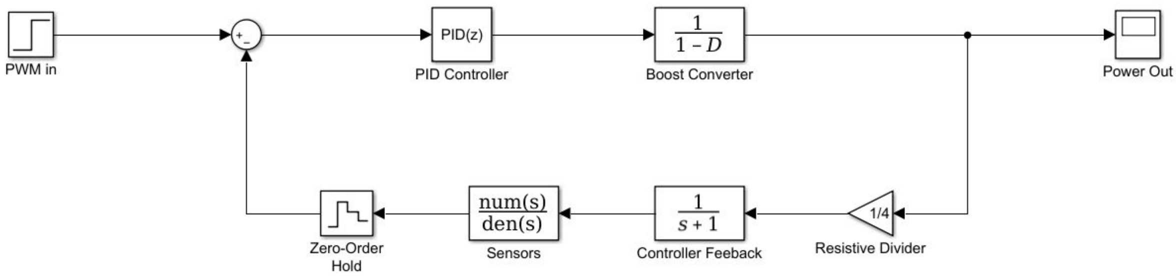


Figure 5.3C: Power Tracking

To track the power as PWM changes we use the voltage sensor together with the current sensor. As the PWM changes, we continually read values from the voltage and current sensor and display the power across the load.

5.4 Validation and Final Design Specifications

Current and Voltage Sensor Validation

To validate the current and voltage sensor, we first place a load resistance and power it with a DC power supply. We then measure its voltage and current using the voltage sensor, current sensor, and arduino board while altering the source power. The setup is shown below.

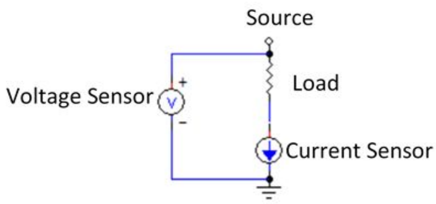


Figure 5.4A: Validation of Sensors Setup

For voltages ranging from 0 V - 5 V there is a 0.1 V maximum margin of error and for voltages ranging from 5 - 12 V there is a maximum margin of error of 0.04 V. For currents ranging from 0 A - 0.1 A there is a 0.007 A maximum margin of error, at higher currents maximum margin of error is 0.005 A. The accuracy of the sensors is tolerable for our purposes.

Position Sensor Validation

When rotated by hand, the position sensor has an accuracy of 1°. The fluctuation under low wind speed is 1°; under medium wind speed is 2°, and under high wind speed is 3°. The wind turbine blades are large enough that a small margin of error will not drastically affect power generation. Also, these small fluctuations in the angle can be fixed through software.

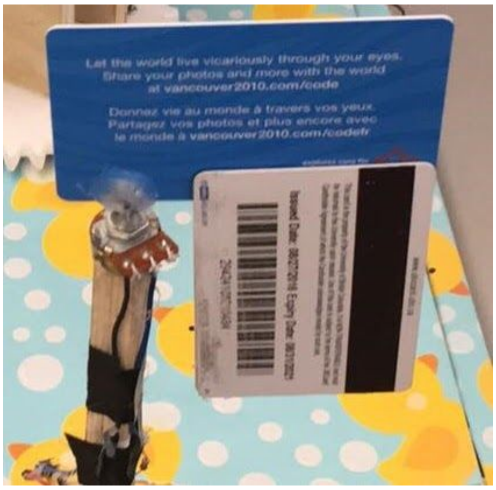


Figure 5.4B: Position Sensor

Final Design Specifications

The final design specifications of each sensor is outlined below.

1. Voltage Sensor:
    a. 0.018 W power loss
    b. 0.02 V precision
    c. Voltage sensing range: 0 V to 12 V

2. Current Sensor:
    a. 0.02 W power loss
    b. 0.005 A precision
    c. Runs on 5Vdc
    d. Current sensing range: 0 A to 1 A

3. Position Sensor:
    a. 0.0025 W power loss
    b. 1° of angle accuracy
    c. Runs on 5V dc
    d. Angle sensing range: 0° to 300°

6.0 Integration

6.1 Assembly

Physical Assembly of subsystems

Electronics are first measured in terms of height, width, and length. The height of the fan generating the wind is measured as well as the required clearance of the wind sensor and the turbine blades. For compactness, the dimensions of the system are minimized. For the required rotation of the turbine blades, a lazy susan is used in tandem with a 4:1 gear ratio to improve the torque of the stepper motor. A slip ring is used for the wires connected to the rotating structure to prevent the tangling of wires. Electronics are easily placed inside the box and joined together; holes are drilled for wires to reach structures outside the box.

Below is a brief sketch of the system holding all of the subsystems.

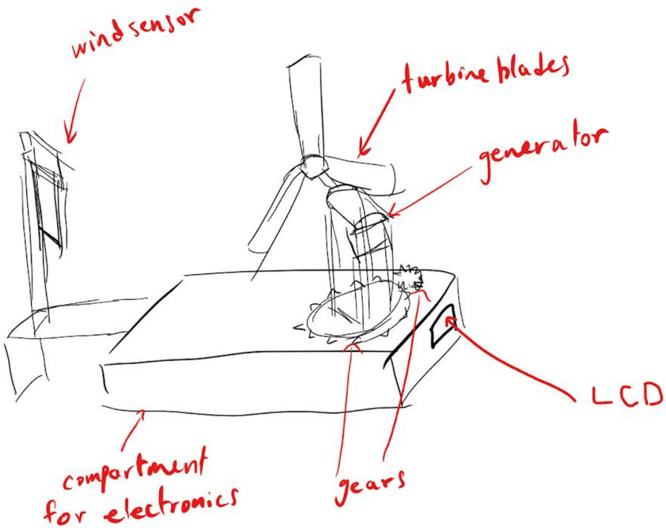


Figure 6.1A: Sketch of final system

System Block Diagram

Once placed inside the physical chassis, all the subsystems are connected as follows:

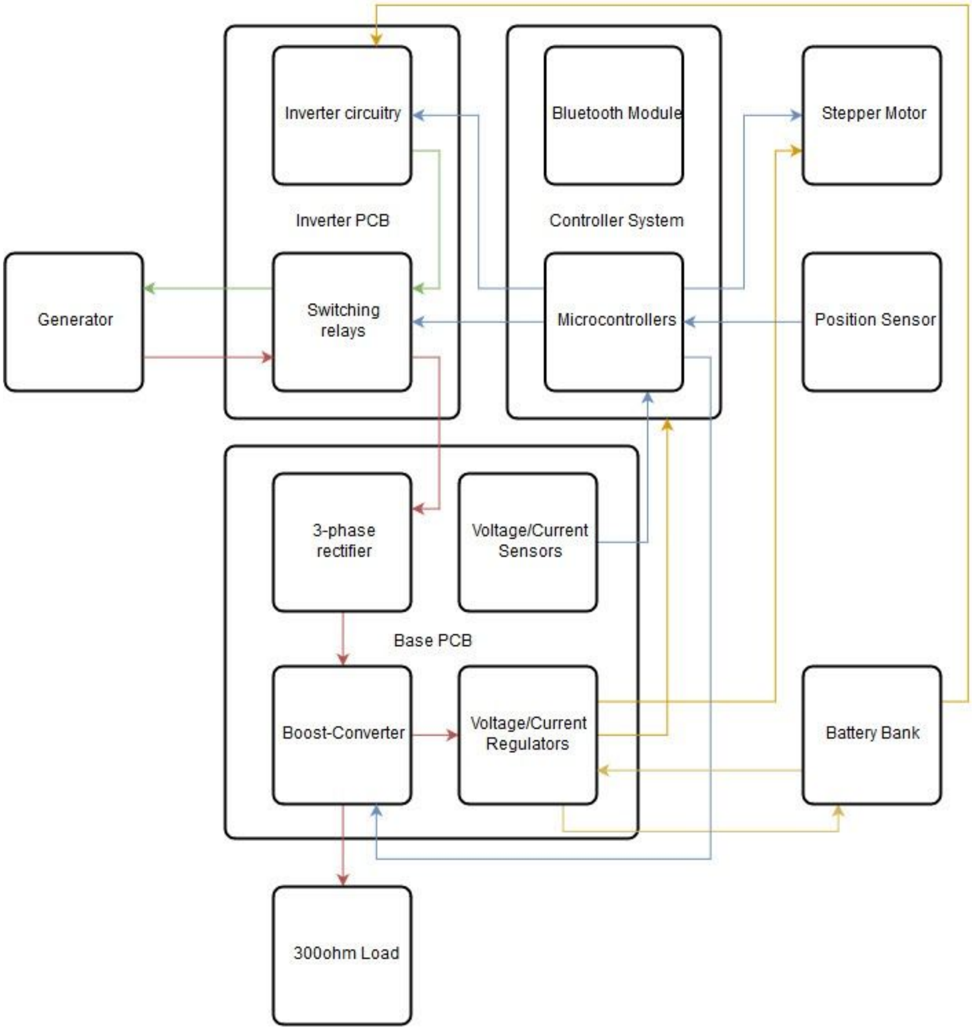


Fig 6.1B: Final system block diagram

Note that the different colors used for the subsystem connections represent the following:

- Red: Main generator output power connections
- Green: Inverter power connections
- Blue: Data and control connections
- Yellow: Electronics power connections

Extra Features Description

- LCD Display for voltage, current, and power readings
- 180° range of motion for wind direction tracking
- Wireless interfacing via bluetooth
- Batteries for powering electronics and for energy storage
- Inverter circuitry for inverting DC power into 3-phase AC power

6.2 Validation and Testing

Requirements:

- Minimum 90 degree wind direction tracking
- Contained within a 50cm x 50cm x 50cm
- Minimum power output of between 3-5W
- Generate 12V DC output (10V to 15V)
- Ability to control blade speed
- Maximum Power Point Tracking

Goals:

- 10W final power output
- Backup battery power storage
- Power all components such as microcontrollers using generated power
- Function as a powered electric fan
- Wireless communication
- Email notification of specific wind turbine status
- Display data (Voltage, Current, Power, Angle)

Validation was performed by running multiple tests to evaluate each requirement/goal. The nature of each test and their results for each requirement/goal are listed below:

Requirement	Test	Result
90° Wind direction tracking	Rotating system on moving platform and observing tracking behaviour	**PASSED** note: 180° range of motion achieved
Contained within a 50cm x 50cm x 50cm	Measure dimensions of physical system	**PASSED**
3-5W Power output	Measure output voltage, current with generator 1-foot away from fan at second speed setting	**FAILED** Note: 0.55W achieved
10V-15V output voltage	Measure output voltage with generator 1-foot away from fan at second speed setting	**PASSED**
Blade speed control through pulse-width-modulation	Blade speed measured while changing PWM input to boost-converter	**PASSED**
Maximum power point tracking	For constant wind speed, compare MPPT algorithm operating point to maximum power point found through sweep of entire PWM duty-cycle range	**PARTIALLY PASSED** Note: 75% of experiments show correct MPPT operation.

Table 6.2A: Requirements validation

Goals	Test	Result
10W Power output	Measure output voltage, current with generator 1-foot away from fan at second speed setting	**FAILED** Note: 0.55W achieved
Battery power storage / charging	Test battery charging circuitry with power supply and then with generator	**PARTIALLY PASSED** Note: tests with power supply show that circuitry operates properly. Generator does not supply enough power to charge batteries in conjunction with other electronics
Power all electronics using generated power	Test power regulation circuitry with power supply and then with generator	**PARTIALLY PASSED** Note: tests with power supply show that circuitry operates properly. Generator does not supply enough power to charge batteries in conjunction with other electronics
Function as a powered electric fan	Use power inverter circuitry to provide power to the generator	**PARTIALLY PASSED** Note: power inversion circuitry correctly outputs 3-phase power. Not enough power can be provided to the generator to cause rotation due to the low current rating of the generator windings
Wireless interfacing	Test all control functions through a wireless bluetooth interface (phone)	**PASSED**
Battery power for electronics	Test all electronics using only the battery power	**PASSED**
Email notification of specific wind turbine status	No test conducted due to this feature not being implemented at all	**FAILED**
Display data (Voltage, Current, Power, Angle)	Observe LCD screen contents during generator operation	**PASSED**

Table 6.2B: Goals validation

7.0 Appendix

7.1 Timeline

Proposed timeline

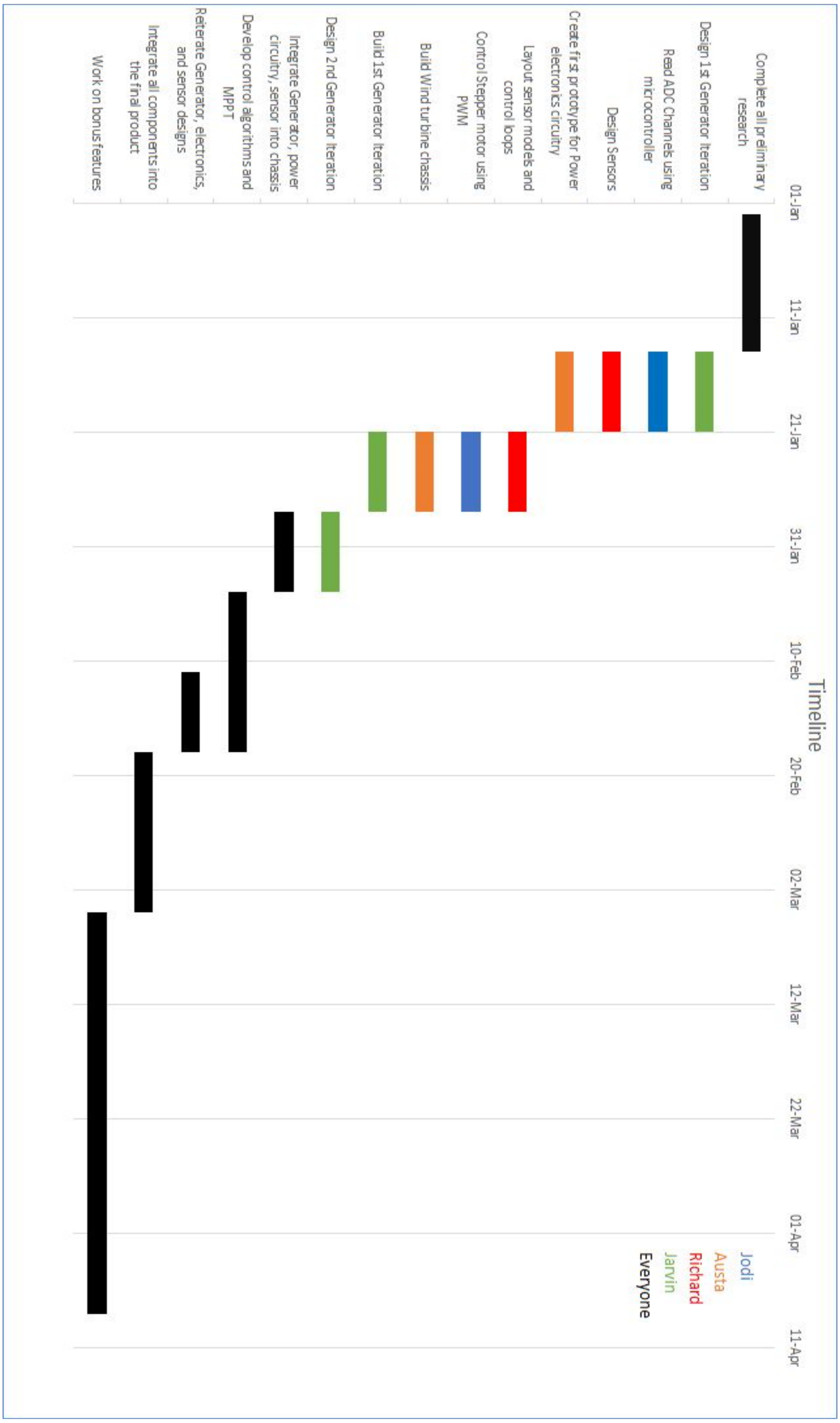


Figure 7.1A: Proposed timeline

Actual Timeline

Figure 7.1B: Actual Timeline

As we can see, the integration part took longer than expected. Unexpected error requires more allocated resources and time spent to debug and delaying the main goal of tasks to be accomplished.

7.2 Original Proposed Budget

Item	Quantity	Cost(each)
Lazy Susan	3	4.45
Permanent Magnets	6	1.85
Ball Bearings	3	1.25
Diodes	20	0.17
Perf Board	6	2.50
Arduino Board	1	30.00
Arduino PWM Shield	1	23.50
Stepper Motor	1	20.00
LCD	2	20.00
Opamp	3	1.00
Hall-Effect Sensor	2	10.00
Optical Encoder	1	20.00
Potentiometer	1	5
TOTAL		338.00

Table 7.2A: Proposed Budget

7.3 Final Budget

Components (Engineering services)	$181.85
Waterjet (Engineering services)	$58.00
PCB (3rd Party)	$64.27
TOTAL	$304.12

Table 7.3A: Final Budget