NAU NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing, and Cyber Systems

cyclone CWC23

Final Documentation

CWC23 - Team Cyclone
Ian Luff - inl23@nau.edu
Danielle Dockins - dld288@nau.edu

Sponsors: U.S. Department of Energy, National Renewable Energy Laboratory, and Energy Club

Mentors: Dr. Venkata Yaramasu, Dr. David Willy, and Alexander Dahlmann (GTA)

Revised: May 5, 2023

Table of Contents

List of Figures

List of Tables

Table 6: Input and Output Voltages of the Voltage Regulator

Appendix

Appendix A: Functional Decomposition

Appendix B: Arduino Code

Introduction

Our capstone project is a part of the Collegiate Wind Competition (CWC), which is an annual event that is hosted by the Department of Energy and the National Renewable Energy Laboratory. This competition was created to promote innovation and inspire future generations to learn about the applications of wind turbines while applying real life experience. Renewable energy has become popular due to the worldwide goal of reducing fossil fuel usage and in turn, global warming. This year the CWC is turning their focus to offshore wind turbine applications. Due to the temperamental offshore environment, it is expected to have rapidly changing wind speeds. These changes and inconsistencies make wind turbine operations difficult when compared to their mainland counterparts. One major mechanic that is implemented is the presence of an emergency stop system to protect the turbine during extreme wind speeds that are unsafe for operation. With this, our goal is to replicate a small-scale offshore wind turbine.

The main objective of our project is to be able to design and build an electrical system that compliments the mechanical engineering team's turbine design to efficiently convert wind power to electrical power, with the main goal of having a stable output. Well designed wind turbines need to be able to operate at various wind speeds, while also taking into account safety measures. The general guidelines and rules can be found in the CWC23 handbook on the competition website. Furthermore, our clients have added on extra conditions that needed to be met. Overall, these requirements and constraints have to be applied in order to meet the competition' sand our client's needs.

Problem statement

Our mission is to create an electrical system that outputs electrical power, while taking into account stability, efficiency, and safety. This system needs to be versatile enough to withstand variable wind speeds and have durability that is acceptable for the voltage, current, and power that is running through the various components.

Engineering Requirements

Majority of our requirements and constraints are taken from the competition rules and requirement documentation [1], although we were given a few from our clients as well. For the requirements given by the competition, they have been put into two categories: technical and testing, which are the following:

CWC Technical Requirements:

- Cable from electrical and load to turbine must be at least 2.5 meters in length
- Cable from button of turbine to base of turbine must be at least 1 meter in length
- Connectors are to be kept dry at all times and support weight of the cables
- Electrical components must be contained in a fireproof and waterproof container
- Enclosure and electrical components must meet the National Electrical Manufacture's Association Type 1 Rating
- Connections must be in cable form, no bare wires or tapped connections
- No broken wires within the vicinity of the water tank
- Voltage at PCC must be less than 48V
- Base plate must be tied to ground and all electrical connections must be tied to the base
- Load and turbine must be in two separate enclosures
- No batteries allowed on turbine side
- Capacitors must not have ratings that correspond to greater than 10J of energy
- Wires to PCC must have PP15-45 Anderson Powerpole connectors
- Must use appropriate powerpole connectors for the equivalent amperes
- Emergency-stop connector and wiring must be able to withstand 3A and carry a small signal
- Cable to the PCC must be no smaller than 28 gauge and have at least two wires and end with standard JST RCY female receptacle housing connector

CWC Testing Requirements:

- Turbine will have zero state of charge at testing start
- Emergency shut off open circuit voltage should not exceed 48V

- Turbine must restart for any wind speed above 5m/s
- Turbine must hit shutdown levels (<10% rpm of maximum) within 10s after initiated and last indefinitely until restart

Client Requirements:

- Use a permanent magnet synchronous generator
- Use a boost converter
- Preference of passive rectification
    - Try MOSFET active rectification if time permitted

Proposed Solution

Functional Decomposition:

Our team's solution to the problem given to us by our client was to build a system that contains the following major components: V602 KV180 Motor, AC/DC rectifier, DC/DC boost converter, Arduino for controls (pitch/braking, emergency stop), voltage sensor, and voltage regulator. This system, as shown in Figure 1, was designed to take a three-phase AC voltage that is generated by the motor, transform it to a DC voltage, and step up the voltage to keep it at a constant maximum voltage for the various wind speeds. The voltage sensor is used to measure the input and output voltages to and from the boost converter to calculate the duty cycle needed to step up the voltage to a value of 40V. The voltage regulator is used to power the Arduino when load is connected and also when disconnected.

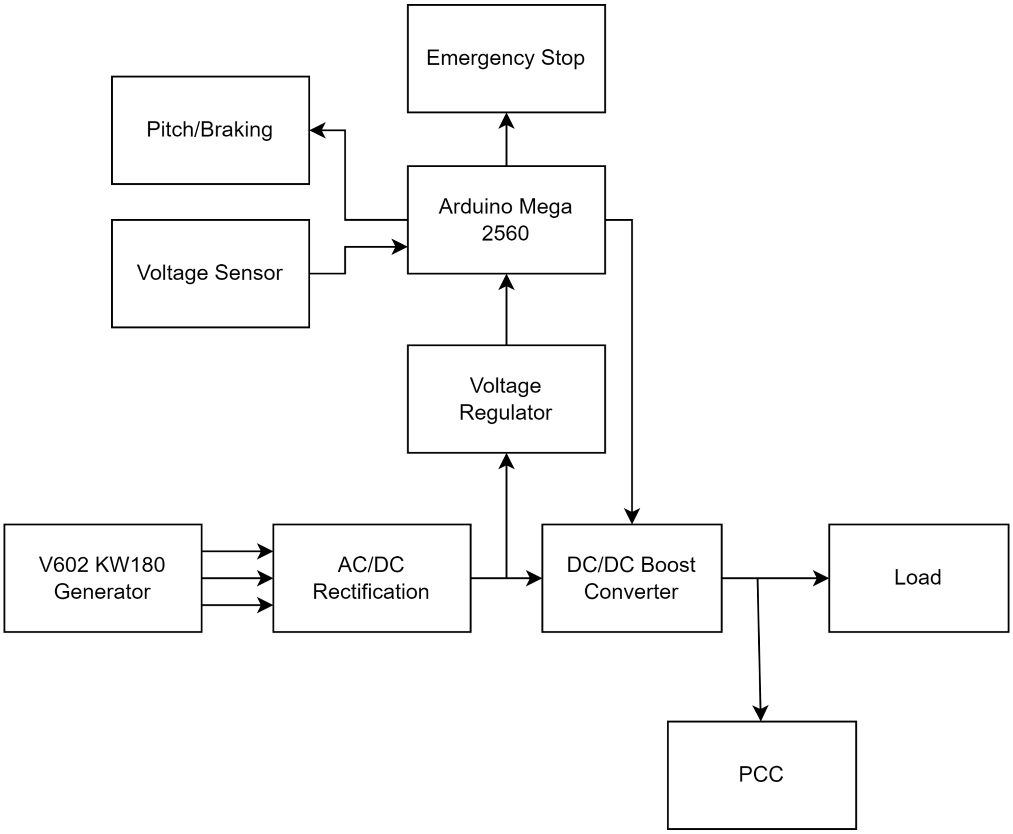


Figure 1: Functional Decomposition

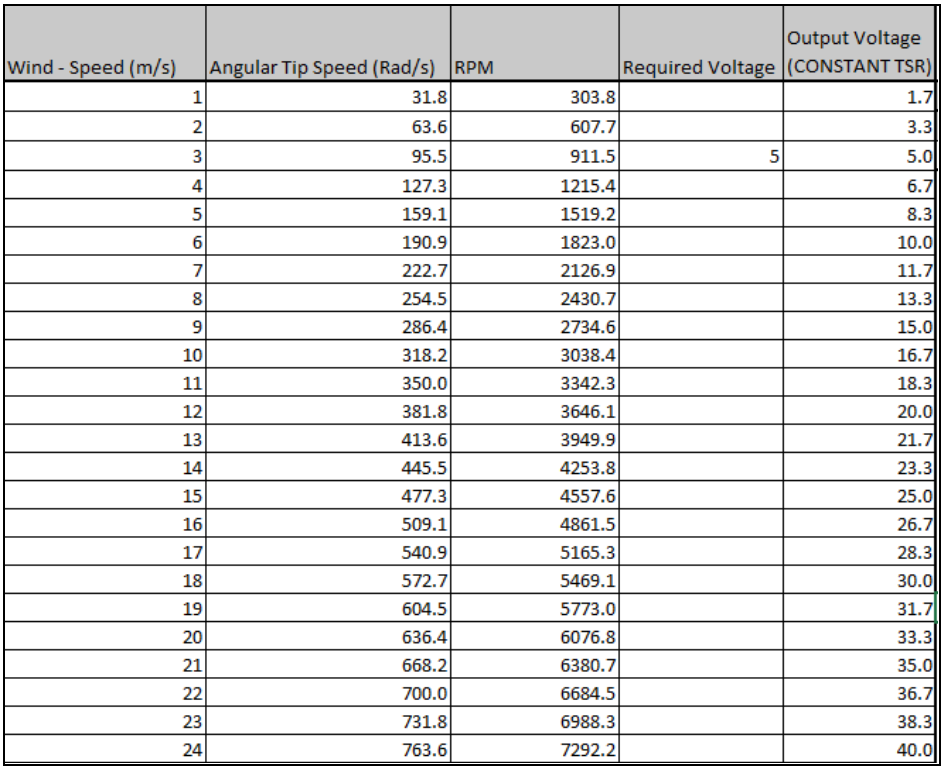
The team divided the project into two sections: power electronics and control systems. The power electronics are the rectifier, boost converter, voltage sensors, and voltage regulator. The control system encompasses emergency shut off, braking system, duty cycle calculation for boost converter, and pitch angle control. Both of these subsystems are necessary for proper operation of the entire electrical side of the wind turbine. Further details about each components are as follows:

<u>Generator:</u>

The generator was selected by the mechanical engineering team, which was the V602 KV180 Motor. This is a three-phase AC brushless motor that was chosen based on their needs and constraints, while taking in consideration that our team wanted an output voltage of 5V with a cut-in wind speed of 3 m/s.

This was done because the Arduino needs an input of 5V to operate and as per our requirements the wind turbine must be able to start when wind speeds are above 5 m/s. Other calculations, such as the specifications of the diodes, inductors, resistors, and capacitors needed for the power electronics were based on Table 1, which shows the output open circuit voltages of the motor at various wind speeds. This data was collected by the mechanical team.

Table 1: Open Circuit Voltages and RPM at Various Wind Speeds

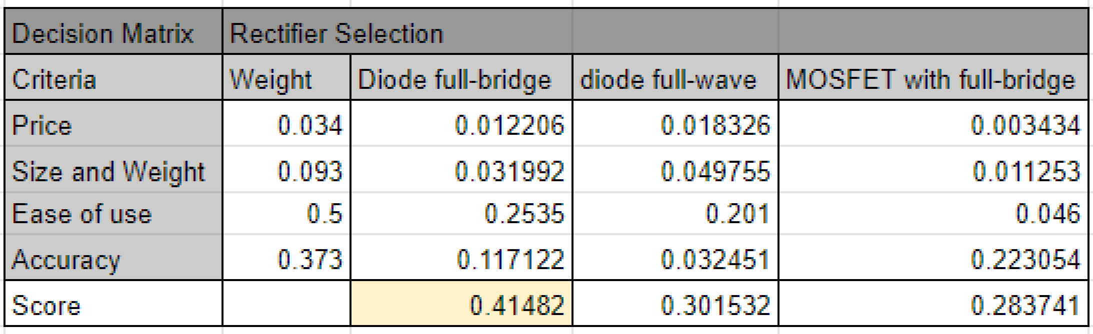


Wind - Speed (m/s)	Angular Tip Speed (Rad/s)	RPM	Required Voltage	Output Voltage (CONSTANT TSR)
1	31.8	303.8		1.7
2	63.6	607.7		3.3
3	95.5	911.5	5	5.0
4	127.3	1215.4		6.7
5	159.1	1519.2		8.3
6	190.9	1823.0		10.0
7	222.7	2126.9		11.7
8	254.5	2430.7		13.3
9	286.4	2734.6		15.0
10	318.2	3038.4		16.7
11	350.0	3342.3		18.3
12	381.8	3646.1		20.0
13	413.6	3949.9		21.7
14	445.5	4253.8		23.3
15	477.3	4557.6		25.0
16	509.1	4861.5		26.7
17	540.9	5165.3		28.3
18	572.7	5469.1		30.0
19	604.5	5773.0		31.7
20	636.4	6076.8		33.3
21	668.2	6380.7		35.0
22	700.0	6684.5		36.7
23	731.8	6988.3		38.3
24	763.6	7292.2		40.0

<u>AC/DC Converter:</u>

The rectifier that was chosen is a passive full-wave bridge rectifier. This selection was made by comparing various designs with our requirements and constraints using a decision matrix, as shown in Table 2. Some of the criteria that were reviewed were accuracy, power efficiency, and ease of use.

Table 2: AC/DC Rectifier Decision Matrix



Decision Matrix	Rectifier Selection			
Criteria	Weight	Diode full-bridge	diode full-wave	MOSFET with full-bridge
Price	0.034	0.012206	0.018326	0.003434
Size and Weight	0.093	0.031992	0.049755	0.011253
Ease of use	0.5	0.2535	0.201	0.046
Accuracy	0.373	0.117122	0.032451	0.223054
Score		0.41482	0.301532	0.283741

Initially the team was going to build the circuit based on Figure 2, but due to time constraints and making sure the rectifier was efficient with minimal voltages losses, the team made the decision to purchase a pre-made rectifier, as shown in Figure 3.

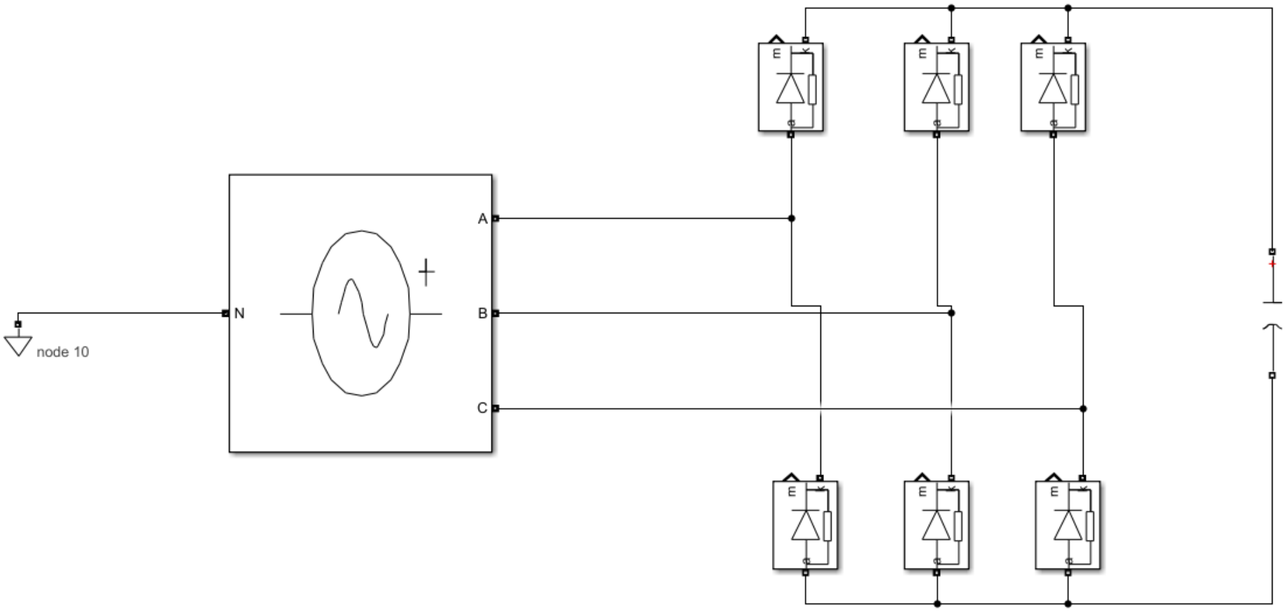


Figure 2: AC/DC Rectifier Circuit Design

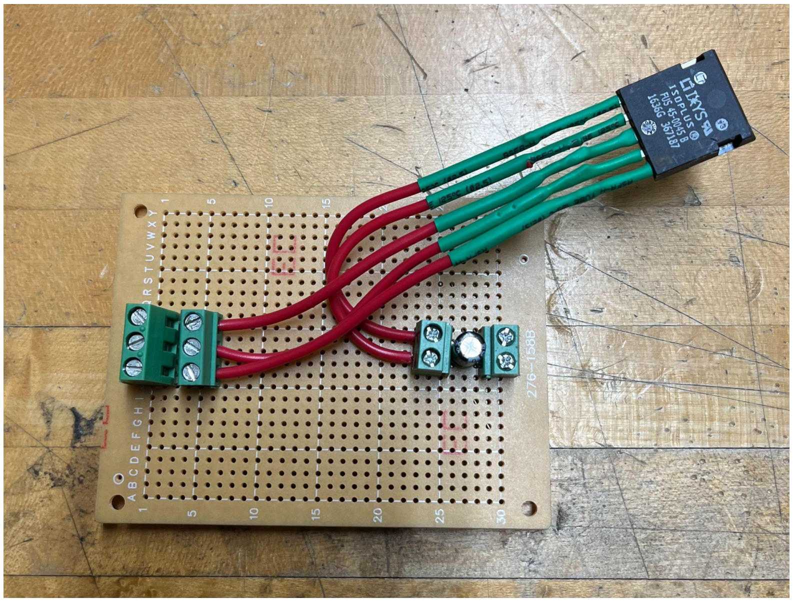


Figure 3: Final Rectifier Prototype

To smooth out the output voltage, a 100µF capacitor was included at the output to lower the voltage ripple to under 1V. The value for the capacitor was initially calculated using equation 1, but during testing the final value was found to be 100µF.

$$C = I \times \frac{\Delta U}{\Delta t} \quad (1)$$

Equation 1 shows that C is the capacitance of the smoothing capacitor, I is the current, ΔU is the ripple voltage, and Δt is half the period. The general rule that our team wanted to follow was to keep all ripple voltages within 5% of the nominal value. As shown in Table 1, the possible voltages ranged up to 40V. This large range made the selection process difficult, but once we had a general range of possible values it was trial and error in order to figure out the final capacitor value.

<u>DC/DC Boost Converter:</u>

The final design for the boost converter was built as a single stage with the following components: 100µH inductor, Schottky diode, MOSFET, and a 47µF capacitor. The original circuit design is shown in Figure 4, while our prototype is shown in Figure 5. Heat sinks were added to the MOSFET and diode to prevent overheating. The boost converter was the most difficult subsystem, this is due to the fact that we are taking in a range of input voltages (1V to 20V) and attempting to boost to a constant output voltage of 40V. Generally boost converters are designed to take in a single input voltage and a single output voltage. We noticed at lower input voltages, the in-rush current tends to spike to unreasonable values for our system. To combat this, we limited the duty cycle at approximately 78%. This was also done because we noticed a drop off in output voltages and it was inefficient to surpass this value. This will be explained in greater detail within the testing section.

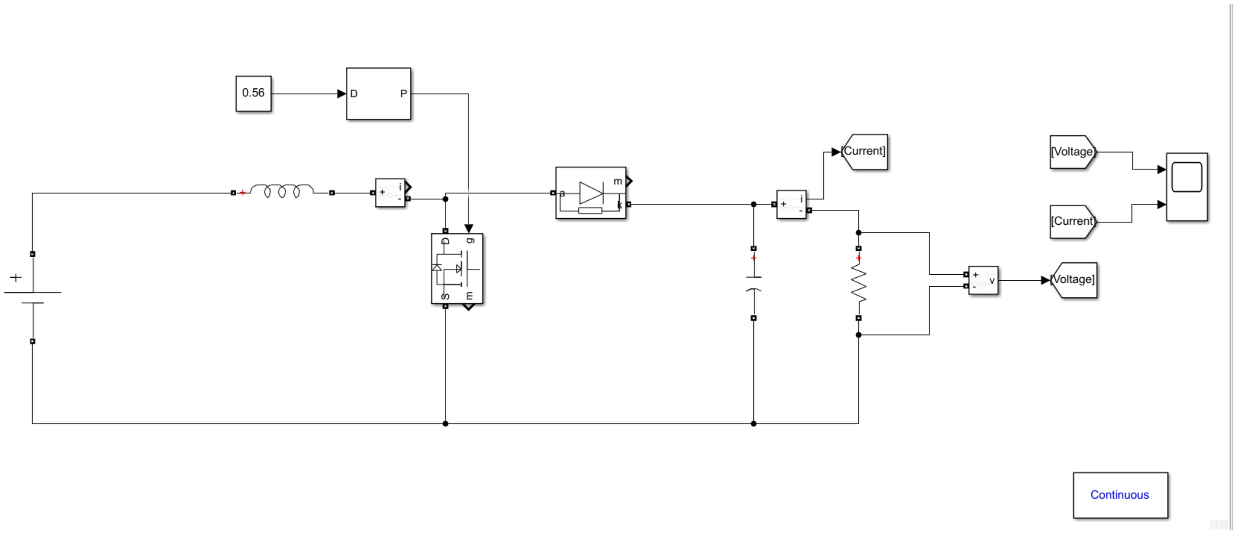


Figure 4: Boost Converter Circuit Design

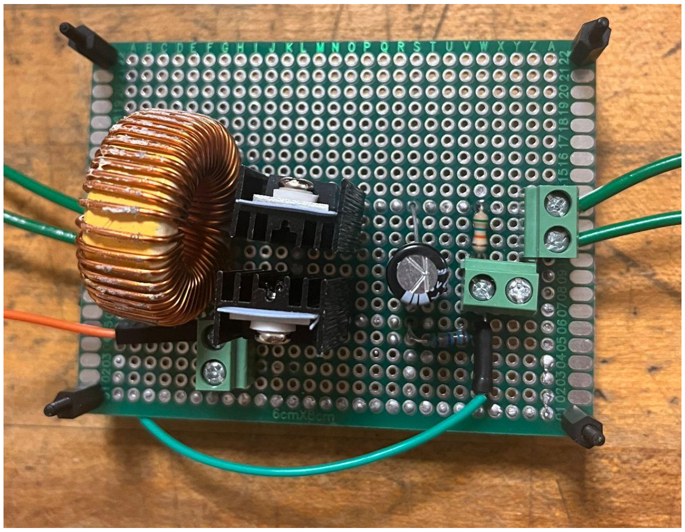


Figure 5: Boost Converter Prototype

The switching frequency that was used was 31kHz and was within the range of the Arduino's PWM signals. To get the values for the duty cycle, inductor, and capacitor, our team has been using equations 2, 3, 4, and 5. These equations were taken from [2], which gave a step by step instruction on designing a boost converter with given input values.

$$D = 1 - \frac{V_{MIN} \times \eta}{V_{OUT}} \quad (2)$$

Equation 2 is how to calculate the duty cycle (D) based on minimum input voltage (V_{MIN}), desired output voltage (V_{OUT}), and efficiency of device (η). In our design, we had our minimum voltage set to 1.7V and the desired set to 40V. Equations 3 and 4 were used to determine the range of values to use for our inductor based on maximum output current, input and output voltages, change in current, and frequency. While equation 5 was used to calculate the range of values to use for the capacitor based on the maximum output current, duty cycle, frequency, and change in desired output voltage. The reason why our team chose to solve for a range of values was because when doing parts selection there are specific values that are available for purchase. Not only did we have to calculate the equivalent values needed for our system, we also had to match it with the marketed values.

$$\Delta I = 0.3 \times I_{OUT\ MAX} \times \frac{V_{OUT}}{V_{IN}} \quad (3)$$

$$L = \frac{V_{IN} \times (V_{OUT} - V_{IN})}{\Delta I \times f \times V_{OUT}} \quad (4)$$

$$C = \frac{I_{OUT\ MAX} \times D}{f \times \Delta V_{OUT}} \quad (5)$$

Arduino:

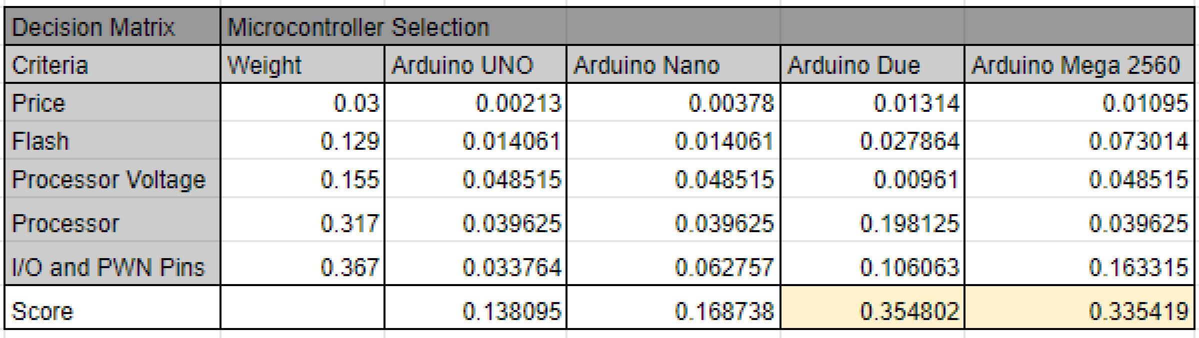
The microcontroller that was selected was the Arduino Mega 2560, shown in Figure 6, this was due to the applications needed to be completed. Selection for the microcontroller was done by a decision matrix, Table 3, while taking into consideration various criteria such as price, i/o pins, and PWM pins. The Raspberry Pi was considered, but due to their cost, compatibility, and features our team decided to choose the Arduino Mega. The final use was to be able to control turbine pitching and braking, duty cycle of boost converter, and emergency stop system.

Pitching/Braking: There are two linear actuators that can be extended with a brake pad, one for braking and one for pitching. The pitch actuator is connected to a disk that can extend and push rods that are connected to the blades. There are five pins for each linear actuator: potentiometer wiper, potentiometer reference x2, and power reference x2. The potentiometer inside of the actuator is used to tell the user where the extruder arm is in relation to the full range of motion. If the potentiometer has a voltage of 2.5V that means it is halfway through in its range of motion. To retract and extend, 5V and ground need to connect to one of the power references. To change directions (reverse), the references have to be switched around. With this, the code for the Arduino (shown in Appendix B) allows for two buttons to be utilized to extend and retract the actuators.

Duty cycle: As mentioned before, our team is using the general formula for a boost converter duty cycle as shown in equation 2. The Arduino is using the voltage input and voltage output values from the voltage sensors to calculate and limit the voltage output boosting to 40V. Furthermore, the Arduino is monitoring the output voltage in order to make the appropriate adjustments to the duty cycle or if there is a case of an over voltage, which triggers the brakes to turn on.

Emergency Stop: The emergency stop procedure has two possible events: a load disconnect and an emergency stop button. In either event the response is the same, pitch the blades to maximum drag position and apply the brakes. To brake and pitch to maximum drag, both linear actuators will have to fully extend.

Table 3: Microcontroller Decision Matrix



Decision Matrix	Microcontroller Selection				
Criteria	Weight	Arduino UNO	Arduino Nano	Arduino Due	Arduino Mega 2560
Price	0.03	0.00213	0.00378	0.01314	0.01095
Flash	0.129	0.014061	0.014061	0.027864	0.073014
Processor Voltage	0.155	0.048515	0.048515	0.00961	0.048515
Processor	0.317	0.039625	0.039625	0.198125	0.039625
I/O and PWN Pins	0.367	0.033764	0.062757	0.106063	0.163315
Score		0.138095	0.168738	0.354802	0.335419

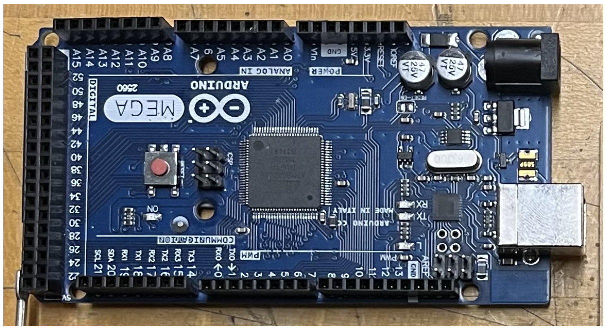


Figure 6: Arduino Mega 2560

<u>Voltage Sensor:</u>

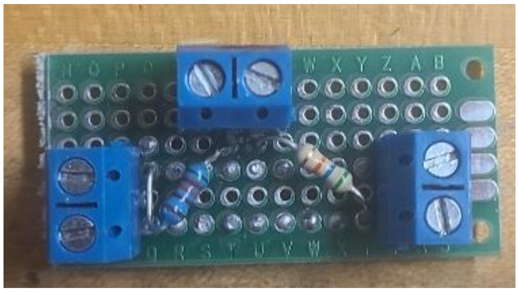


Figure 7: Voltage Sensor Prototype

The voltage sensor (Figure 7) is a voltage divider designed by us to fit within our range of voltages and currents. Our team used the voltage divider formula to calculate the needed resistances. The final values for the resistors used were ? Ω and ? Ω. Our team chose to build our own rather than purchase due to the fact that the ones on the market did not meet our specifications with maximum voltages and currents.

<u>Voltage Regulator:</u>

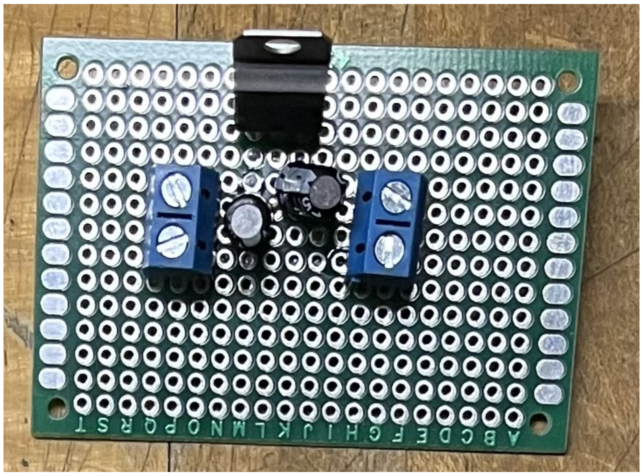


Figure 8: Voltage Regulator Prototype

The voltage regulator (Figure 8) is a pre-build regulator that is used to supply 5V and 1.2A to the arduino. We also added two 10µF capacitors to smooth the input and output voltages.

Testing

There were three key tests that were carried out to test the major components of our electrical system. These three tests were the rectifier, the boost converter, and the voltage regulator. The instrumentation used to conduct these tests were a function generator, DC power supply, digital multimeter, oscilloscope, and a programmable load. Further details are explained below for each test.

<u>Rectifier Testing:</u>

The rectifier testing was done by connecting the function generator to our rectifier circuit and testing various AC voltage inputs based on the data given to us by the ME team. An oscilloscope is used to measure the output DC voltage wave and its ripple voltage. The data collected for this portion is shown in Table 4.

Table 4: Rectifier Input, Output, and Ripple Voltages

RPM	Input Voltage (V)	Output Voltage (V)	Ripple (mV)
300	1.7	1.4	4.08
608	3.3	2.96	40
900	5	4.51	48
1200	6.7	6.24	49
1500	8.3	7.78	56
1800	10	9.43	32

Boost Converter Testing:

For boost converter testing, this was done by connecting the boost converter to our DC power supply and our programmable load, which was set to constant current mode. The team varied the input voltage between 5V and 20V. The duty cycle was capped at 78%. The equivalent output voltages and current were recorded to get the output power, as shown in Table 5. The expected result was that the output voltage was never to exceed 40V, while keeping an efficiency of over 80%. This was proven correct through our testing. The maximum output voltage was at 39.195V, and the lowest efficiency was at 80.16%. Figure 9 shows our estimated power curve based on wind speeds and power output. At approximately 6 m/s the power output tends to level off and stay constant at a little under 4W. There is a drop in power from input to output but that is expected due to the fact that our system cannot fit the ideal conditions of zero power loss (Pin = Pout). Some power losses would be due to heat and inefficiencies with any of the electrical components.

Table 5: Boost Converter Input and Output Values

Input DC Voltage	Duty Cycle	Input Current	Input Power	Output Voltage	Output Current	Output Power	Efficiency
5	78%	0.51	2.555	21.825	0.1	2.19	85.71%
6	78%	0.53	3.179	26.431	0.099	2.62	82.42%
7	78%	0.54	3.78	31.095	0.099	3.08	81.48%
8	78%	0.55	4.394	35.683	0.099	3.53	80.34%
9	77%	0.54	4.865	39.387	0.099	3.9	80.16%
10	71%	0.48	4.8	39.136	0.099	3.87	80.63%
11	63%	0.43	4.725	39.128	0.099	3.87	81.90%
12	56%	0.39	4.676	39.147	0.099	3.87	82.76%
13	51%	0.36	4.68	39.107	0.099	3.87	82.69%
14	46%	0.33	4.616	39.175	0.099	3.88	84.06%
15	42%	0.3	4.5	39.174	0.099	3.88	86.22%
16	39%	0.28	4.48	39.195	0.099	3.88	86.61%
17	36%	0.26	4.42	39.166	0.099	3.88	87.78%
18	32%	0.24	4.32	39.189	0.099	3.88	89.81%
19	29%	0.23	4.37	39.247	0.098	3.85	88.10%
20	27%	0.22	4.4	39.193	0.099	3.88	88.18%

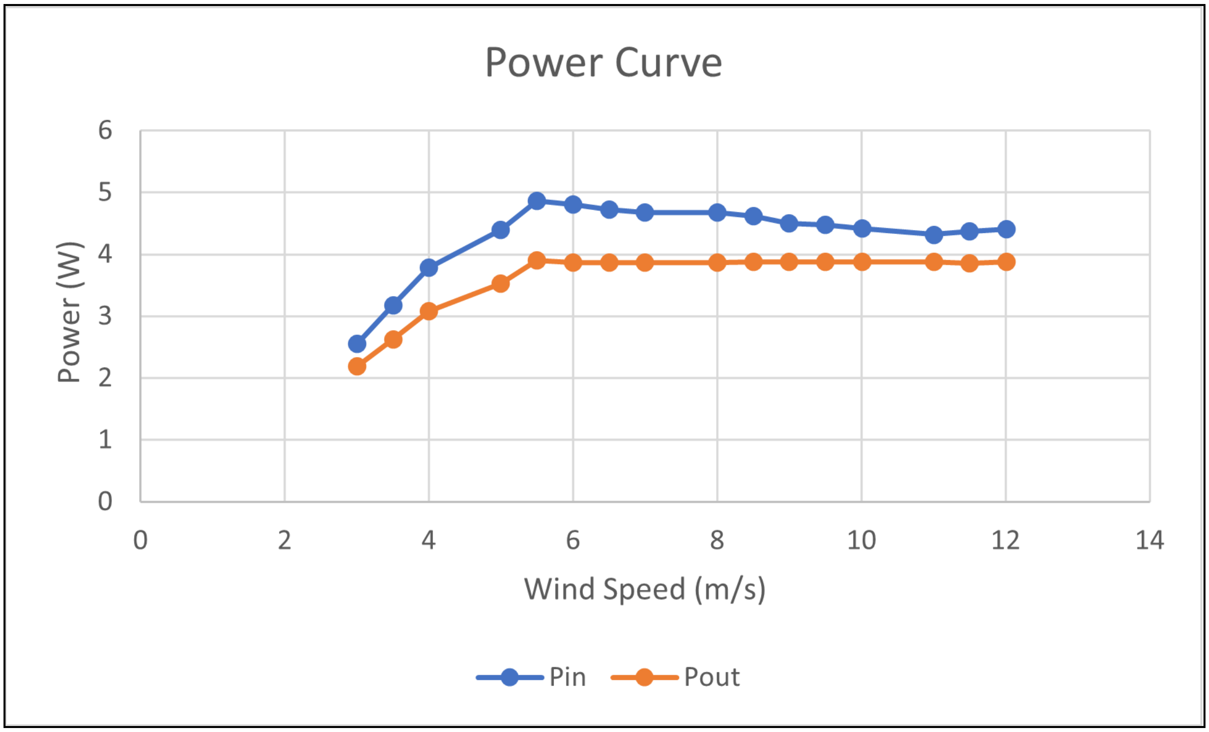
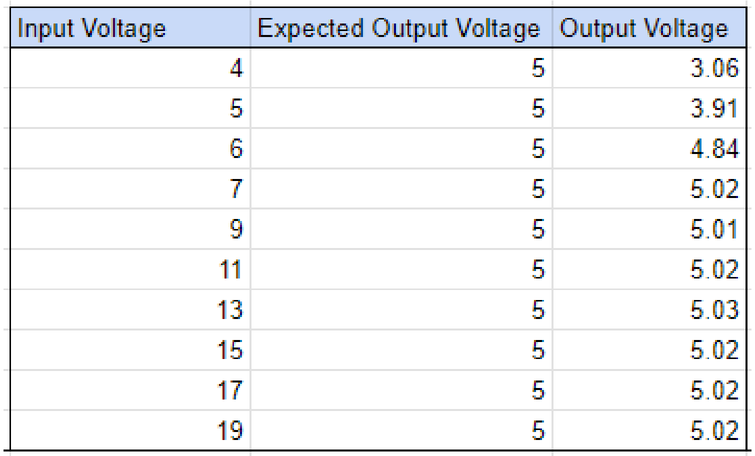


Figure 9: Power Curve of the Boost Converter

<u>Voltage Regulator Testing:</u>

The voltage regulator circuit was tested to make sure that it would be able to operate the Arduino at various input voltages. The tested input voltages were between 4V and 19V. As shown in Table 6, the ideal output voltage would be around 5V. This does not occur for input voltages lower than 6V. This still allows for the Arduino to operate at cut-in wind speed of 5 m/s because at this speed the output voltage from the generator is around 8V.

Table 6: Input and Output Voltages of the Voltage Regulator



Input Voltage	Expected Output Voltage	Output Voltage
4	5	3.06
5	5	3.91
6	5	4.84
7	5	5.02
9	5	5.01
11	5	5.02
13	5	5.03
15	5	5.02
17	5	5.02
19	5	5.02

<u>Analysis of Results:</u>

Based on the testing we conducted, our team is confident that our design works as expected. Our rectifier circuit was able to rectify a 3-phase signal to a DC signal with a minimal ripple voltage on the output. Our voltage sensor circuits were able to accurately read output voltages within an acceptable margin. The voltage regulator circuit was able to supply a constant 5V to our Arduino after receiving 7V. The boost converter circuit was able to maintain a constant 40V after receiving 9V at its input, while still boosting voltages below 9V substantially. The power curve in Figure 9 shows that our device was able to create a stable power curve, especially at higher voltages. When doing a full system integration test, our team was able to power our Arduino with the voltage regulator after supplying 7V at the output of the rectifier circuit and we were able to obtain similar boost converter output voltages to the isolated test.

Conclusion

The most significant aspects of our electrical system are the following:

- DC/DC Boost Converter
- Arduino controls
- Voltage Regulator

The boost converter was a requirement set by our client to use. We also had to take into consideration the competition rules of not boosting over 48V, our requirements of having greater than 80% efficiency, and limit the output to 40V. As shown in our initial testing of the boost converter, we had met the goals for the boost converter circuit design. For Arduino controls, this entails duty cycle calculation and pitching/braking. Our Arduino code is able to control the duty cycle for the MOSFET of the boost converter, while also taking in a safety measure to make sure the output voltage from the boost exceeds that 40V. As for pitching and braking, we are able to pitch the actuators to their full range and utilize the actuators to help with braking as well. Lastly, the voltage regulator is important because without it, the Arduino will not operate or has the potential to break due to high voltage inputs. If the Arduino is not functioning, the boost converter will not be able to boost and the pitching and braking systems will not operate as well. Generally, at this moment the majority of our electrical system is based on theory and calculations of how it should operate. This is due to the fact that integration testing was not completed at this time with the ME team due to delays on both sides, but it is still in the works to be completed before the competition.

This project has taught our team some significant lessons about teamwork, time management, component selection, theoretical versus practical, and learning how to figure out a solution when problems arise. Our team consisted of two members with two different engineering backgrounds. We split the work up between power electronics and programming, along with having one member in charge of presentations and reports, while the other was responsible for communication with clients, mentors, and ME team. We learned that despite having our set roles and responsibilities, we had to be flexible to complete tasks outside of our roles for the success of the project. As for time management, we learned that in the future we need to apply more time to testing and building, compared to calculating and running simulations. Calculations and simulating did give us a better understanding of how the system should work, but after a certain point we learned that implementing the electrical system was more time consuming and complex than building a simulation. This was because of faulty components, incorrect soldering, or even the selection of parts due to us having to find the closest values for the inductors, capacitors, or resistors. The small differences in values for the components changed our outputs and the difference between an ideal simulation and an actual build had some minor output differences as well. Whenever our team ran into problems, our process that we followed was first to do some research, then try out various solutions, and if that did not work we would ask questions to our mentors to help guide us. From there we would come up with new solutions and if needed do additional research. Being able to solve problems and debug issues was one of the most important and satisfying parts of this project.

Lastly, our initial thoughts on the project was that it was simple, but we realized quickly that it was way more complicated and complex, which helped expand our knowledge on the engineering design process and work on our communication skills, which were initially lacking in the beginning. Overall, as we learned new skills and processes, we implemented them as we moved forward.

Future Work:

With this type of capstone project that is repeated yearly, we hope that our documentation will be helpful for future students to have a starting point in designing their electrical system for a wind turbine. Advice from our team is to start early and ask questions often. It is an in depth project that will require diligence and responsible teammates. Design and build your subsystems early because you will learn that there may be problems that will arise during testing and having the extra time to debug and fix those issues will be beneficial.

On a wider scale, we hope these kinds of projects will inspire future students to be more proactive in the renewable energy sector. Bring new solutions and innovations to an ever growing field that is in high demand by companies wanting to lower their carbon footprint.

User Manual

For the purposes of this user manual, there are 3 main systems: the generator/turbine, power electronics, and load. Starting with the generator/turbine, there are three cables coming from the turbine its self. One cable, which is AWG 12 with 3 wires, is for the generator with each wire signifying a single phase (Note: It does not matter which wire you call Phase 1, 2, or 3). The other two are for the linear actuators. To minimize the number of cables going to the turbine, the linear actuators share two wires, which are the positive and negative references for their internal linear actuators. For each linear actuator there is a wire for +5V, GND, Potentiometer wiper, 2x Potentiometer reference. These three cables are then fed into the main electrical box through the cable glands.

For the power electronics there are 3 main components: rectifier circuitry, boost converter, and the arduino. The rectifier circuitry involves the rectifier chip, voltage regulator, voltage sensor, and smoothing capacitor. At the input of the rectifier circuit is a terminal block with three terminals, one terminal for each phase. Those three terminals are connected to another three terminal block, which will have pins 3-5 of the rectifier chip. Pins 1 and 2 of the rectifier are then connected to the smoothing capacitor, voltage regulator, and voltage sensor. The voltage regulator has two output terminals, one for +5V and one for ground. These two wires need to be connected to the arduino Vin and GND terminals. The voltage sensor output needs to be sent to an analog read port on the arduino.

As for boost converter there are four main connections. The first connection is the input, which will come from the output of the rectifier circuit. The second connection is the output of the boost converter, which will go to the PCC via a 14 AWG wire. The third connection is the voltage sensor in the boost converter, which measures the output voltage of the boost converter. This wire needs to be connected to an analog read terminal on the arduino. The last connection is the gate of the MOSFET, which will need to be connected to a PWM output on the Arduino.

The last subsystem is the load. The load for our design is a variable DC load. To operate the load choose a channel, either 1 or 2, and connect the output of the PCC to the load with +5V going to the red terminal and GND going to the black terminal. Set the channel to CV (constant voltage). A green light will appear on the CV selection, when the selection has been made successfully. Once the CV mode has been selected, turn on the load by pressing the on button over the used channel. The load is now ready to use. *If the load has not been set to the proper settings follow the instructions below*

To set the load to the proper settings a DC power supply is required. Plug the DC power supply into the boost converter directly, with the output going straight to the DC load. Turn on the DC Load using the instructions above. Set the DC Power supply to 5V. The DC power supply should output a power value. The objective is to determine when that power value reaches a maximum, while changing the load settings. The load has two knobs on it, one for coarse and one for fine adjustment. Use the coarse adjustment knob. Rotate the knob clockwise and observe the power value on the DC power supply. The power value should rise as you turn the knob, but then start to drop. The moment the power supply goes from increasing to decreasing is the proper setting for the load. Once the knob is adjusted to the correct position, it is ready for use.

Appendix A: Functional Decomposition

The following figures show the decomposition of each subsystem that was included in our electrical system. The subsystems included were: generator, rectifier, boost converter, and arduino control with voltage regulator.

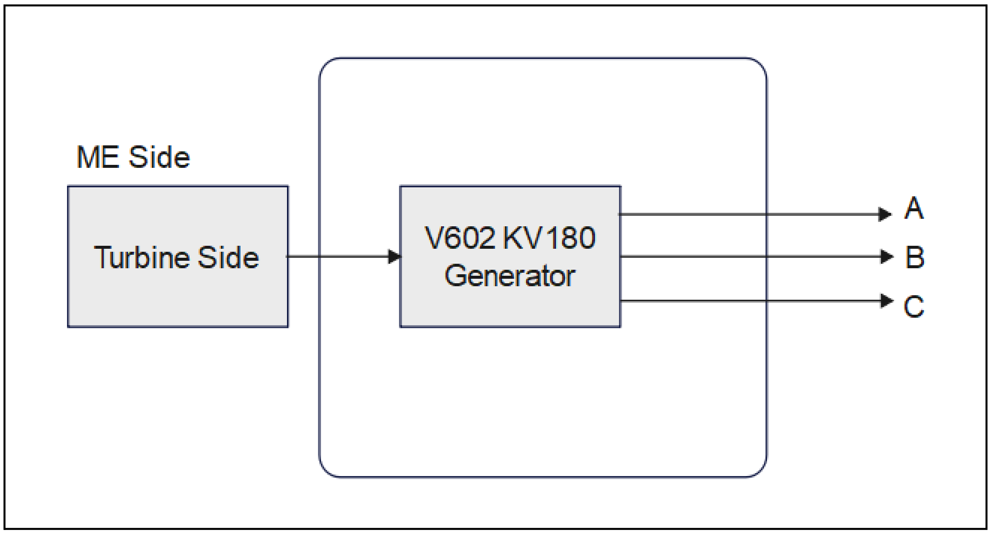


Figure 10: Generator Decomposition

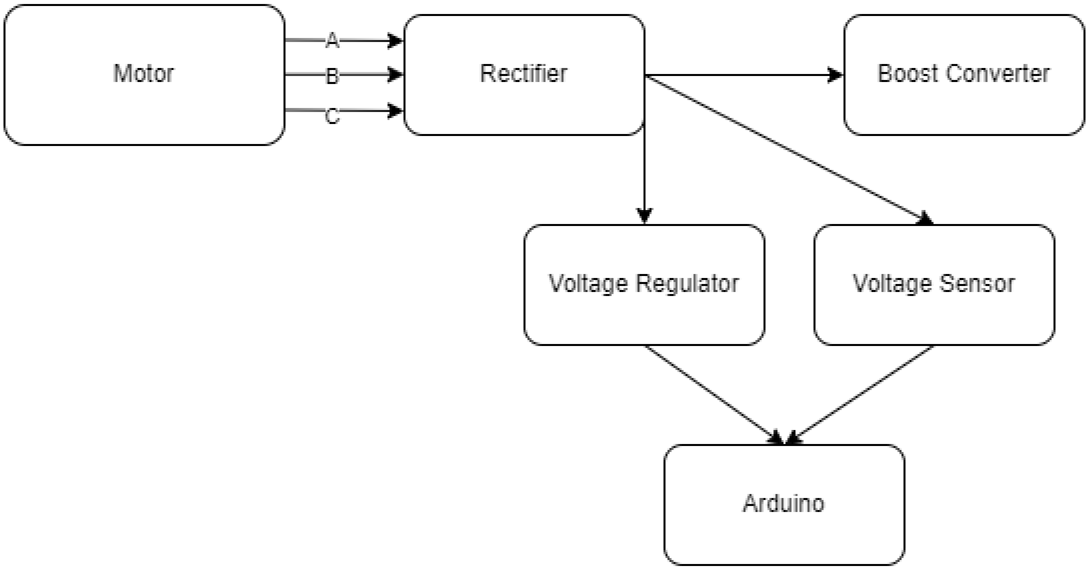


Figure 11: AC/DC Rectification Decomposition

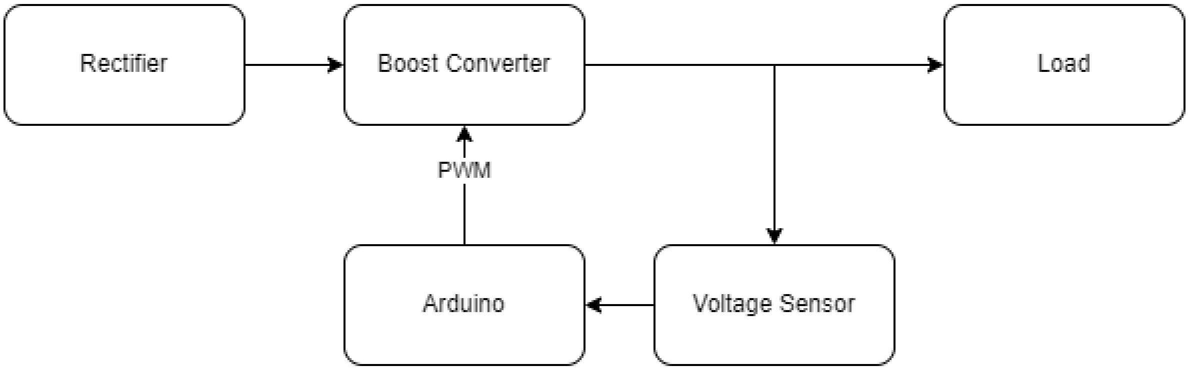
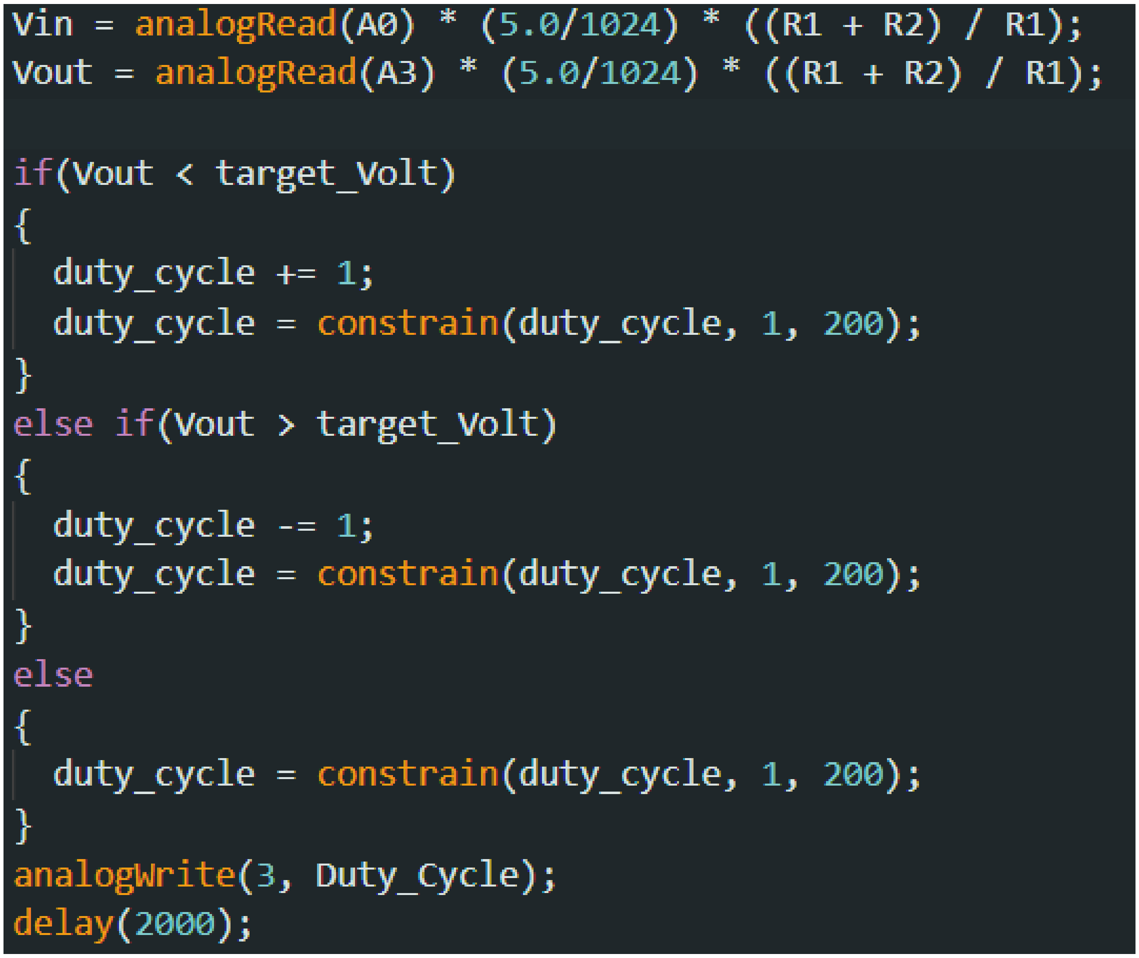


Figure 12: DC/DC Boost Converter Decomposition

Appendix B: Arduino Code

The following shows the Arduino codes for calculating the duty cycle, braking, and emergency stop.



```
Vin = analogRead(A0) * (5.0/1024) * ((R1 + R2) / R1);
Vout = analogRead(A3) * (5.0/1024) * ((R1 + R2) / R1);

if(Vout < target_Volt)
{
  duty_cycle += 1;
  duty_cycle = constrain(duty_cycle, 1, 200);
}
else if(Vout > target_Volt)
{
  duty_cycle -= 1;
  duty_cycle = constrain(duty_cycle, 1, 200);
}
else
{
  duty_cycle = constrain(duty_cycle, 1, 200);
}
analogWrite(3, Duty_Cycle);
delay(2000);
```

Figure 13: Arduino code for Duty Cycle

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(13, INPUT_PULLUP);
  pinMode(12, INPUT_PULLUP);
}

void loop() {
  // put your main code here, to run repeatedly:
  double linVal = analogRead(A1);
  Serial.println(linVal);

  if(digitalRead(13) == HIGH && digitalRead(12) == LOW)
  {
    digitalWrite(8, HIGH);
    digitalWrite(7, LOW);
  }
  else if(digitalRead(12) == HIGH && digitalRead(13) == LOW)
  {
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
  }
  else
  {
    digitalWrite(8, LOW);
    digitalWrite(7, LOW);
  }
}
```

Figure 14: Arduino Code for Actuators

References

[1] US Department of Energy, “Collegiate Wind Competition 2023 - Rules and Requirements,” 2022, pp. 1- 26

[2] “Basic calculation of a boost converter's power stage.” [Online]. Available: https://www.ti.com/lit/pdf/slva372. [Accessed: 11-Mar-2023].