NORTHERN ARIZONA UNIVERSITY
School of Informatics, Computing, and Cyber Systems

To: Robert Severinghaus

From: Team 5

Date: March 26, 2021

Subject: Testing Results Report

Project Overview:

The purpose of this project is to design a functional small scale wind energy conversion system (WECS). The role of the team is to represent Northern Arizona University (NAU) at the Collegiate Wind Energy Competition (CWC). This competition occurs every year at the American Wind Energy Association (AWEA) CLEANPOWER convention sponsored by the Department of Energy (DOE). The problem is, as more wind energy is incorporated into the U.S. power generation mix, qualified workers are needed to fill related jobs at all levels. The goal of the CWC is to prepare students from multiple disciplines to enter the wind energy workforce by providing real-world technology experience. The NAU CWC team consists of one electrical engineering subteam as well as two mechanical engineering subteams. These three independent capstone teams must work together to design, build and test a small scale wind turbine. The electrical engineering subteam is responsible for all the components of the electrical power conversion system. This includes the selection of a three-phase AC generator, a AC to DC converter, a boost converter, the electrical load supplied by the system, and the housing for all of these components. The system architecture (Figure 1) shows how each of these subsystems is integrated into the system as a whole. The electrical team consists of five team members, each of which were assigned to one of these subsystems.

Executive Summary:

Each of the subsystems mentioned above, except the housing, requires testing to be done. Therefore the team performed four distinct sets of tests. The first test was a matrix unit test. This testing was used to determine which three-phase generator the team should select as the source for the WECS. There were four different motors with four different power output ratings from which to select. After performing cogging torque and power output testing at various rpm values, the team selected a motor, the MAD 5010 110 Kv. The testing for the rectifier was similar to the testing for the turbine. It was also a matrix unit test in which the team collected the output voltage and current at various rpm values which would be expected at the collegiate wind energy competition. The boost converter circuit is regulated by an arduino mega which provides a switching frequency to a MOSFET in the boost converter circuit as well as the duty cycle for the boost converter. The boost converter testing is a step by step unit test. For a specific input, the system should output a specific voltage. Finally, the integration test for our system involves each of these components functioning together as a single unit. In this test, the three-phase motor provides power to the rectifier which converters the power from AC to DC. The rectifier sends a DC current through the boost converter which should raise the voltage level.

The boost converter feeds the power through the point of common coupling (PCC) where all the data from the circuit is read and then into the load which draws current through the system. Most the time spent in the lab this semester was used for testing of the various subsystems of the project. The team spent anywhere from four to eight hours in the lab each week, comprised mostly of testing. The results of the turbine testing was selecting the MAD 5010 110 Kv motor. This selection was based on the motor's low cogging torque and high voltage output at low rpm.

Introduction to the system:

The client for this project is a professor and researcher at NAU, Dr. Venkata Yaramasu. Dr. Yaramasu's research specializes in renewable energy, high-power converters, variable speed drives, electric vehicles, energy storage, and smart grid. Dr. Yaramasu is the client for the electrical engineering team while David Willy is the client for the mechanical engineering teams. Because of the integrated nature of the project as a whole, the electrical team also meets with and is responsible to David Willy, since he is the liaison between NAU and the CWC. The problem presented to the team is how to design, build, and test a wind energy conversion system that operates at a range of wind speeds, from 0 m/s all the way up to 22 m/s. The system architecture shown in Figure 1 displays each of the major components implemented in the system and how these subsystems are connected. This single-line diagram shows a three-phase AC generator (the turbine block), an AC to DC converter (the rectifier block), a boost converter (DC/DC converter block), the electrical load supplied by the system (consisting of both the optional diversion load and the competition setup with load). As shown in Figure 1, the boost converter is controlled by an arduino mega, which is used to specify a duty cycle and the switching frequency for the PWM sent to the transistor in the boost converter circuit. The turbine is powered by a rotational force supplied by the turbine blades. These blades and the rotating shaft as well as the turbine tower are designed by a mechanical engineering subteam. The WECS has many technical constraints described in the Rules and Regulations document supplied by the CWC. One major constraint is that the voltage measured at the PCC must be below 49V DC. This limits the boost converter output, which might need to be restricted by a simple off-the-shelf buck converter, that only lets a set voltage pass through.

System Architecture:

Figure 1 shows the system architecture including the four specific tests the team ran on the system. These four tests were the rectifier test, the boost converter test, the microcontroller test, and the system integration test. The first three tests mentioned are targeted at the most critical components of the system architecture. Without the rectifier, the boost converter, ro the MCU the system would not function properly.

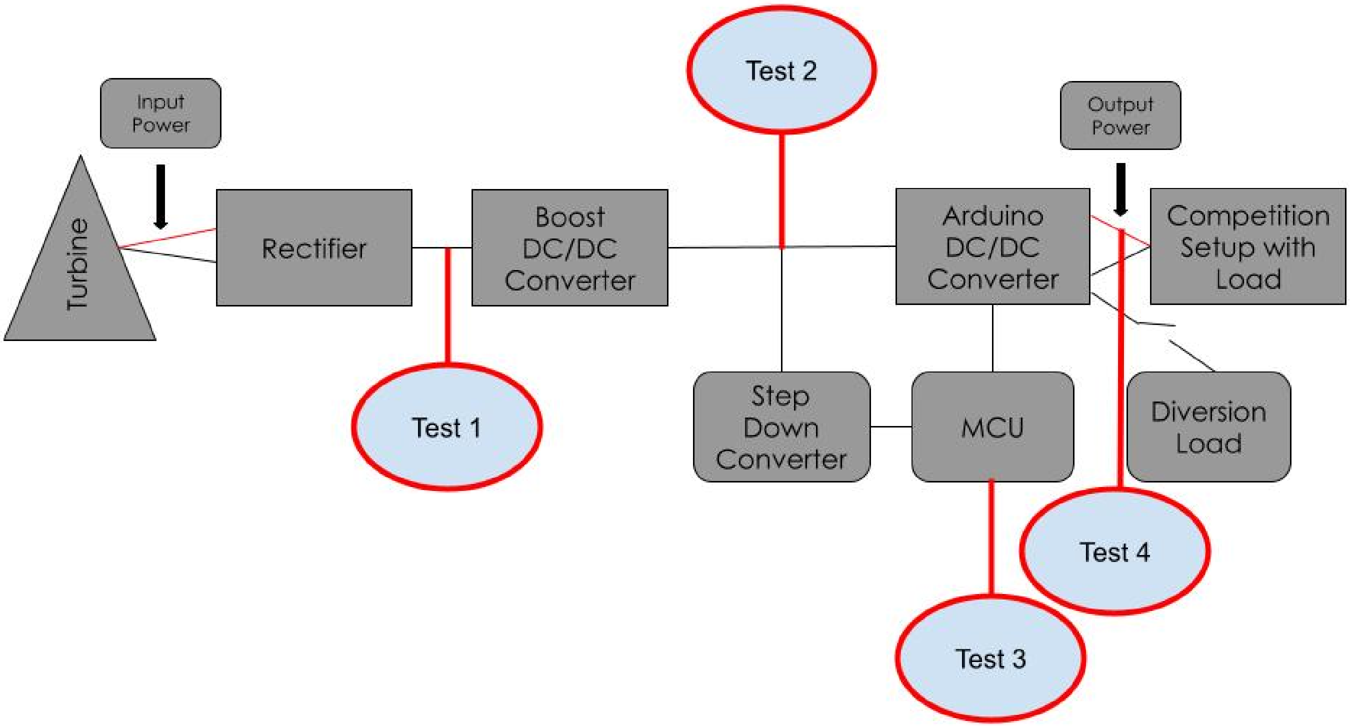


Figure 1: System Architecture

Requirements, status, type of test:

The testing workbook includes every major requirement or constraint on the project. This includes the constraints given by the client, the team mentor, and the CWC competition. The team closely followed each requirement and finished the excel sheet shown in figure 2 which describes each requirement in detail. Some requirements changed throughout the course of the project, and these are also noted in the testing workbook.

Type of Test	Status	Req #	Requirement
	🟨	1	General Rules:The turbine must be designed, and loads analyzed to withstand continuous winds of up to 22 meters per second (m/s) but no testing will be done beyond 13 m/s.
	⬛	1.1	Size Requirements: Within practical limits, there is no size restriction for components located outside the tunnel.
	⬛	1.2	Rule Changes: An important change in the rules for the 2020-2021 collegiate wind competition (CWC) is that teams are required to clearly describe what materials and designs they have referenced and used for this year’s design
	⬛	2	Project Breakdown
	⬛	2.1	Auxillary connections
	⬛	2.2	Purchasing
	⬛	2.3	PCB Design
	⬛	2.4	Arduino Code
	⬛	2.5	DC/DC Converter
	⬛	3	Auxillary Connections
	⬛	3.1	I/O
Inspect	🟨	3.1.1	All I/O must use anderson power pole connectors
	🟩	3.1.2	Output voltage must be <=48V at any given time
	🟥	3.1.3	~~Output current must be between 15-45 amps~~
Inspect	⬛	3.1.4	All I/O wires should be 10-20 AWG
Matrix	🟩 *	3.1.5	Rectifier must output DC with as little drop as possible
Inspect	🟨 *	3.1.6	Load with emergency stop button
		3.1.6.1	Verify current cuts to diversion load
		3.1.6.2	Verify brakes get power
Inspect	⬛	3.2	Turbine Connection
	⬛	3.2.1	All the wires that connect to the turbine should exit at the turbine base.
	⬛	3.2.2	All cable pass throughs in enclosures must use cable glands or other similar devices that provide both strain and chafe protection.
	⬛	3.2.3	Each cable connection from the turbine to the enclosure should employ a quick-attach connector
	⬛	3.2.4	The turbine base plate shall be tied to earth ground. The turbine electrical system ground(s) must be electrically tied to this base plate with a 100 kΩ or lower resistance connection.
	⬛	3.2.5	All electrical cables leading from the turbine to the electronic components located outside the tunnel must be in cable form (no individual strands) and have connectors
Inspect	⬛	3.3	Point of common coupling(PCC)
	⬛	3.3.1	Wires exiting the base of the turbine must be at least 1.5 meters in order to reach the competition testing point of common coupling (PCC).
	⬛	3.3.2	The turbine electronics must be in a separate enclosure from the load in order to clearly differentiate load and the control during inspection by judges
	⬛	3.3.3	Teams can use the load to power the turbine but the load (a capacitor bank) must not be charged at the beginning of the competition.
	⬛	3.3.4	The enclosure should be outside the turbine and the wires should be long enough to reach the electrical enclosure and terminated with a single red and single black anderson power pole connector
Inspect	⬛	3.4	Electrical enclosure
	⬛	3.4.1	Enclosures are constructed for indoor use to provide a degree of protection for personnel against access to hazardous parts and to provide a degree of protection for the equipment inside the enclosure against ingress of solid foreign obj
	⬛	3.4.2	All electrical components must be incorporated into closed enclosures that are fire safe and meet or exceed a National Electrical Manufacturers Association (NEMA) Type 1 rating
	⬛	3.4.3	All components must be electrically insulated from the enclosures.
	⬛	3.4.4	All electrical components shall be mechanically secured to the enclosure.
Inspect	⬛	4	Safety Requirements
	⬛	4.1	Teams must follow Occupational Safety and Health Administration rules for safety equipment based on expected activities (see NREL/university subcontract, Appendix B Clause 8: Worker Safety and Health Requirements, for more inform
Inspect	⬛	5	PCB Design
	⬛	5.1.1	The PCB integration will be designed through the software, Altium
	⬛	5.1.2	All components and connections on the PCB layout will be labeled
	⬛	5.1.3	In order to define widths of power lines and communication lines, net classes will be made to avoid error within the PCB board
Inspect	⬛	6	Arduino Coding
	⬛	6.1	Client Constraints
Step by Step	🟨 *	6.1.1	Arduino Mega must be used to automatically adjust DC/DC converter
	🟩	6.1.2	MATLAB Simulink should be used
	🟨	6.1.3	A smaller DC/DC Converter should be used to power MCU
Inspect	⬛	7	DC/DC Converter
	⬛	7.1	Client requirements
	⬛	7.1.1	Team must use previous team's layout
Inspect	⬛	7.2	Competition requirements
	🟩	7.2.1	Capacitor cannot store >10J
Matrix	*	7.2.2	Max power variance must only vary +/-10% of total power at PCC
		7.2.3	Submit a one page write up of detailed testing with instrumentation used
Integration	*	7.2.4	Whole system test
		7.2.4.1	Perform a time series of power measured at a frequency of at least 200Hz, then compare to section 3.5.2 of competition guidelines

🟩	Passed
🟨	test partially passed
🟥	Failed
⬛	not testable
*	vital to system

Figure 2: Requirements with test progress

Most important requirements:

The most important requirements of this project were marked with an asterisk in the testing document. These signify tests that simply have to pass to either meet client requirements, or may be vital to the system’s overall function. The first of which is 3.1.5: The rectifier must output DC with little to no voltage drop. This is vital to our system because the DC/DC converter of course requires DC input, or else it cannot boost the voltage. Therefore, outside of ME turbine requirements, the rectifier working properly is the start of the whole system, and therefore if it does not function properly, neither will the system. The next is a client required safety feature, 3.1.6: The emergency stop button. This requirement is quite simple, and yet, extremely important. The necessity comes from not only the competition requiring it, but for the safety of our engineers as well as our turbine in the event of a runaway event. The stop button engages the brakes on the turbine, thus slowing them down and allowing for the turbine to remain intact. Should this system not work, the turbine blades may eject in the wind tunnel, as seen in past projects. The next test is 6.1.1: Arduino Mega must be used in the project. This is for the sake of simplicity, as well as the ability to easily convert simulink models into arduino code. Should the arduino not function properly, practically the whole system fails. This is because the arduino logic controls the boost PWM, the diversion load, the emergency stop button, and the brakes. The next requirement marked as important is within the DC/DC converter, 7.2.2: Max power variance within +/- 10% of total power. This is a test of the boost converter's ability to stabilize and properly boost to our predetermined voltage. This is a large section of the competition, and where the whole team gets to see if the system works well, or if there are inefficiencies. Should this not function properly, the rest of the system can still function, but the brakes cannot, which can be very dangerous. The brakes use a buck converter tied to the PCC to get 6V out, and if the boost does not supply exactly what we expect, the buck cannot properly supply the brakes. The final requirement is a whole system test, 7.2.4. This is the sum of the parts, and if everything prior works, this test should work perfectly. There are quite a few pieces within this test, so the importance to the client is of course the system doing what it is supposed to. Should it not work, any of the above consequences may apply.

Types of tests:

In order to accurately describe and see results from each subsystem we used a series of 4 different types of tests. Each different test would have a greater significance when applied to a specific subsystem in terms of displaying and illustrating the results that would ultimately help us the most in the long run to diagnose and fix problems that would otherwise provide us with better results.

The Unit Test Matrix (UTM) was mainly implemented to showcase direct results of outputs of a certain subsystem. Outputs consisting of values such as, voltage, current and power could be easily compared to input values. We found that the UTM ended up being perfect for trying to find a correlation between input and output values which was very useful while testing subsystems that perform functions specifically made for producing a certain output that needed to meet a certain quota. We implemented the UTM on the Boost subsystem as well as the Rectifier subsystem.

We have ultimately found that the UTM is most effective when testing hardware compared to software. In other words, a different testing process is required when testing software that follows a similar concept to debugging lines of code. When testing the arduino, we used the unit step by step test (UTS). We concluded that this test would be more effective in this situation for a few different reasons. The UTS is designed to be a flow-chart like test that will display each output following the given condition that is being implemented. This allows us to run through each step of the code that will provide an idea of what is happening when a command is executed. From here, we can change conditional statements within the code to match our expectations if we do not like the results being yielded.

After testing our individual subsystems through the UTS and UTM methods we needed to find a process that would be well suited for testing the whole system. We found that the user integration test would be best for this scenario. This test will mainly simulate results that will be judged in the competition such as values like voltage and power. The actual testing process for this consists of a series of actions where each action will have an expected result or yield that will either pass or fail. From this point, depending on whether the action passed or failed, we will revert back to testing each subsystem through the UTM or UTS methods. This method of testing will allow us to see a bigger picture of how all of the subsystems are working together and how well they are working together.

Major tests:

Rectifier:

The testing for the rectifier was successful, and yielded useful data for the overall circuit and specifically for the design of the boost converter. The rectifier is a critical component in the circuit. This is what converts the AC waveform, generated by the three-phase motor, into a DC signal. The rectifier chip that the team purchased consists of a six-diode full bridge rectifier. The chip has five pins that are threw-hole mounts for a pcb. The first two pins are the DC output pins, positive (+) and negative (-). The other three pins are the input pins for the three-phases. The team performed a loaded test with the rectifier in which the current was held constant at ~1A and the rpm was varied. This provided the data shown in Figure 3. The team also performed an open circuit test with the rectifier. The results of this test are shown in Figure 4. As described in the Testing workbook, the team originally expected the output of the rectifier to be in the range of 10-15 Amps. However, after performing some testing and having difficulty finding inductors with a high enough current rating, the team decided to settle for an output current of approximately 1 Amp. This decreased the expected output power (W) by a factor of ten. The team made this change to the expected values in the testing workbook and the rectifier output values passed for each test. the lowest rpm at which the motor is able to overcome the startup (or cogging) torque is approximately 360 rpm. For the rectifier test, the motor actually started at 300 rpm, which is better than expected. This was probably a result of drawing fewer amps with the DC load.

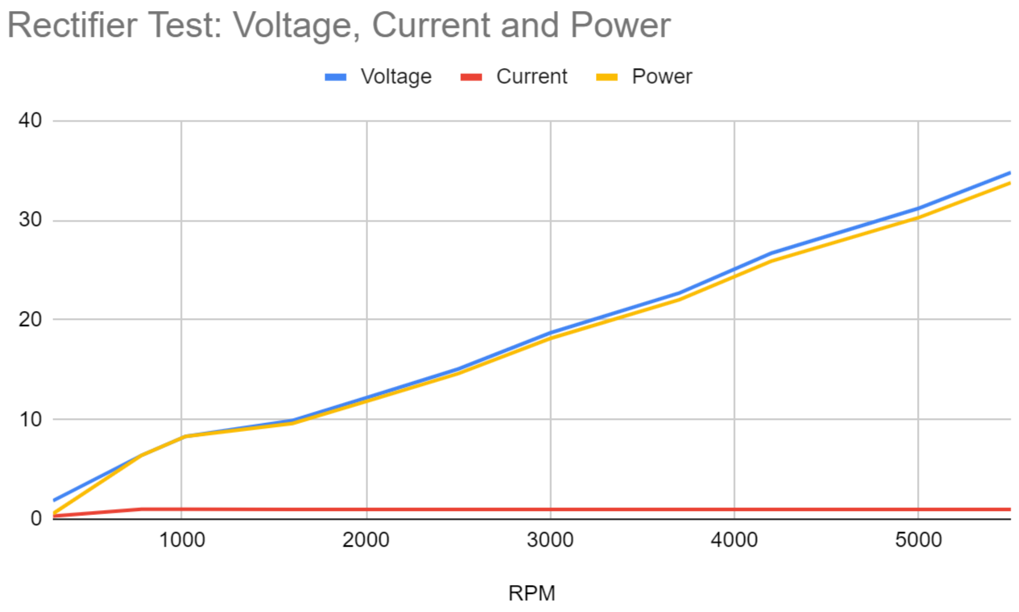


Figure 3: Rectifier CC Test Results

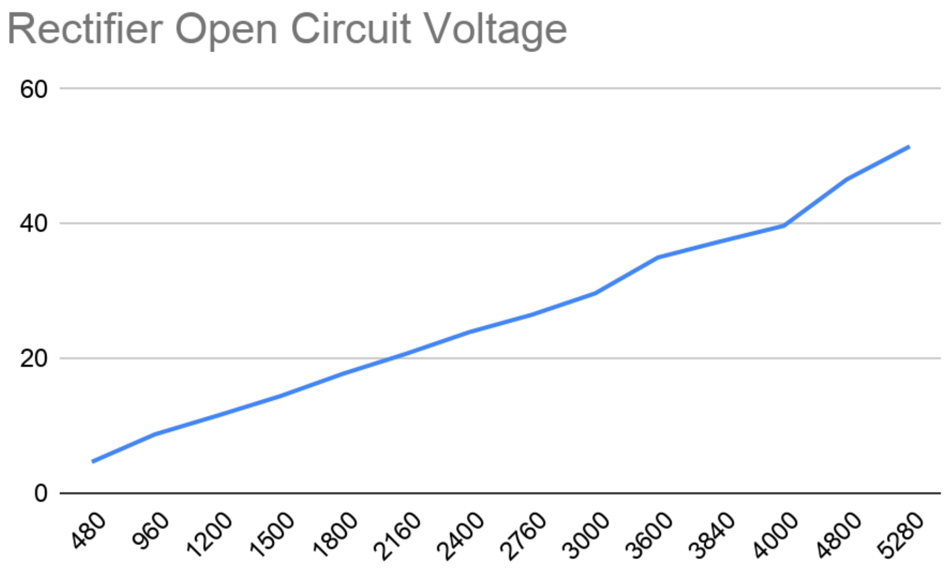


Figure 4: Rectifier Open Circuit Test

<u>MCU:</u>

The Testing of the control system was a success. The Control system is an Arduino Mega with a voltage sensor and emergency stop button. The Sensor and stop button control the amount the boost increases the voltage. To perform the MCU test we used the following items: DC Power supply, Arduino breadboard system, and an oscilloscope. The breadboard setup has power terminals that the power supply connected to. Channel one of the oscilloscope was connected to the PWM pin (Digital pin 45). A picture of the breadboard setup testing can be found in the appendix figure 1. The power supply was set at 4 volts for the start of the test. The DC voltage source was increased by one-volt increments. At each increase the PWM signal was analyzed by the oscilloscope, Duty cycle and frequency was recorded. After the voltage reached the max the power supply outputs (30 volts), We moved on to the Emergency system. The emergency system was tested by pushing the emergency button. The results should be no PWM signal and 6 volt power out of the relay.

Boost Converter:

The boost converter test consists of a representation of the voltage going into the inductor compared to the voltage being discharged by the capacitor. It is a simple Input/Output test, but is one of the most vital to our system. The test begins by hooking up the positive end of the inductor as well as the ground of the circuit to either a DC power supply, or directly to the dynamometer and rectifier set up. A variable DC Load should be connected across the output capacitor, and set to 1A of constant current. From there, any form of DC Voltage measurement is sufficient, but an oscilloscope is preferred. When the circuit is powered, it should also be loaded. From there, the test may begin starting at 1V in, and increasing by 1 volt until we hit 36V. The current should be monitored to only be pulling 1A, and the voltage output should be measured and recorded. A power curve should also be generated using the results of this, as seen below. (Not included as the Boost Testing has not yet been completed.

Analysis of results:

Rectifier:

The conditions for the rectifier are a combined set of results for the rectifier's temperature increase and overall output power from the generator input power. Successfully, the rectifier performed close to the root means square output from the given input, and didn't increase in temperature in high rpms. The rectifier was a white box test, which created expected results from the parameters we choose.

MCU:

The test for the MCU was done and the results received were close to the expected values. We started at 4 VDC and got out a duty cycle of .83. The goal voltage out of the boost converter was set at 32 volts. This value can be changed in the code. The Arduino code prevents the PWM signal from outputting a duty cycle above .83 since the Mosfet doesn't work the best at zero or one duty cycle. The duty cycle decreased when the voltage was increased. The results can be found in the appendix below Figure 2. The values recorded for duty cycle were very close to the expected results. A graph of the expected vs actual duty cycle is found below Figure 5. When the emergency stop button was pressed the relay closed providing 6 volts to the braking system. The PWM signal was also turned off. This is exactly what should happen during an emergency situation.

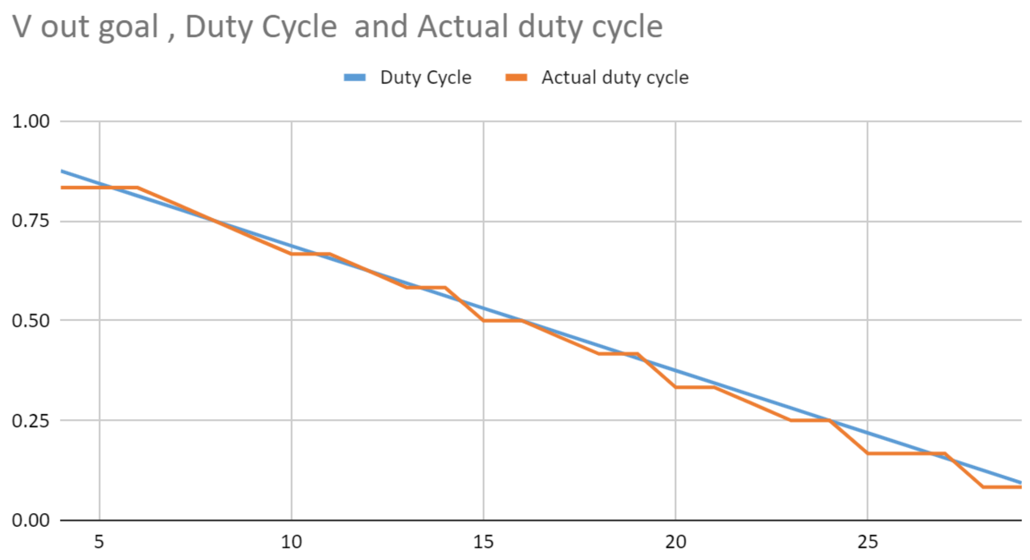


Figure 5: V_out vs expected and actual duty cycle

Boost:

The conditions for the boost are of the combined circuit elements collectively, including the MOSFET, the inductor, capacitor, and MOSFET with dead time. Integration testing between the circuit elements rendered initial testing issues at each node. The biggest issue was concerning powering the MOSFET. The boost was also a white box test, which created expected results from the parameters we choose.

Lessons Learned:

Turbine:

Our team learned that motor selection would be a critical component of the project. The kv rating of the turbine has a large impact on what air speed at which the motor will start spinning. This determines when the boost converter turns on and at what wind speed our team can begin scoring points at the competition. The team learned that the kv rating is measured in units of rpm per volt. So, the rpm of the motor corresponds to the kv rating (a constant) multiplied by the output voltage.

$$kv = rpm / volt$$

MCU:

Before and during the test of the MCU a few things were learned. The Arduino code can be sped up when the serial output commands are removed. The Arduino did not change the duty cycle as fast as expected and took a few moments longer to change. The serial output lines were implemented for visual testing of individual components and hardware. During competition and further testing the computer will not be connected. The revised code for the Arduino can be found in the appendix.

Boost:

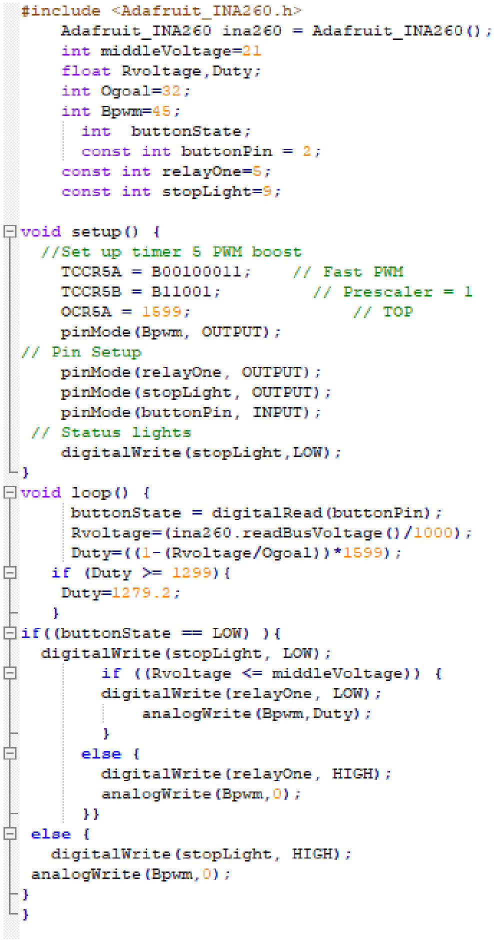
The main issue we ran into with the boost converter was switching the MOSFET. Originally we thought we may have had too low inductance value, or potentially the frequency was too low. It turns out that the MOSFET was likely not switching and was stuck in saturation. We believe that the solution to this is having a gate drive to control how much current is actually going into the MOSFET to be able to force it on and off. This is an issue because the boost relies on the transistor switching in order to switch the polarity of the inductor, and if it does not switch, the circuit will just run through the diode and discharge at the capacitor. We were unable to test further, as this issue cannot be resolved in time.

Team Coordination:

A problem faced early in the project was coordination with the the other subteams involved in the project. For instance, our team had been informed that one of the mechanical engineering subteams had already selected a motor and ordered it early on in the first semester. After waiting a few months with that information, the team still did not have a motor to perform testing. Our team was then informed that the selected motor was on back order and would not arrive in time for testing. Our team then had to make up for the lost time, researching, selecting and purchasing a motor. This sort of issue has been a recurring problem as communication continues to be a major issue when coordinating with multiple subteams.

Appendixes:

The arduino mega was coded using arduino ide. The code during the test is found below.



```
#include <Adafruit_INA260.h>
    Adafruit_INA260 ina260 = Adafruit_INA260();
    int middleVoltage=21
    float Rvoltage,Duty;
    int Ogoal=32;
    int Bpwm=45;
      int  buttonState;
      const int buttonPin = 2;
    const int relayOne=5;
    const int stopLight=9;

void setup() {
  //Set up timer 5 PWM boost
    TCCR5A = B00100011;    // Fast PWM
    TCCR5B = B11001;         // Prescaler = 1
    OCR5A = 1599;                // TOP
    pinMode(Bpwm, OUTPUT);
// Pin Setup
    pinMode(relayOne, OUTPUT);
    pinMode(stopLight, OUTPUT);
    pinMode(buttonPin, INPUT);
 // Status lights
    digitalWrite(stopLight,LOW);
}
void loop() {
     buttonState = digitalRead(buttonPin);
     Rvoltage=(ina260.readBusVoltage()/1000);
     Duty=((1-(Rvoltage/Ogoal))*1599);
   if (Duty >= 1299){
    Duty=1279.2;
   }
if((buttonState == LOW) ){
  digitalWrite(stopLight, LOW);
        if ((Rvoltage <= middleVoltage)) {
        digitalWrite(relayOne, LOW);
            analogWrite(Bpwm,Duty);
        }
      else {
        digitalWrite(relayOne, HIGH);
        analogWrite(Bpwm,0);
      }}
 else {
   digitalWrite(stopLight, HIGH);
 analogWrite(Bpwm,0);
}
}
```

Figure 6: Arduino Code

Setup of control unit and sensors during test. Dc voltage source powers the red and black cable. The oscilloscope is the gray cable on the bottom of the picture.

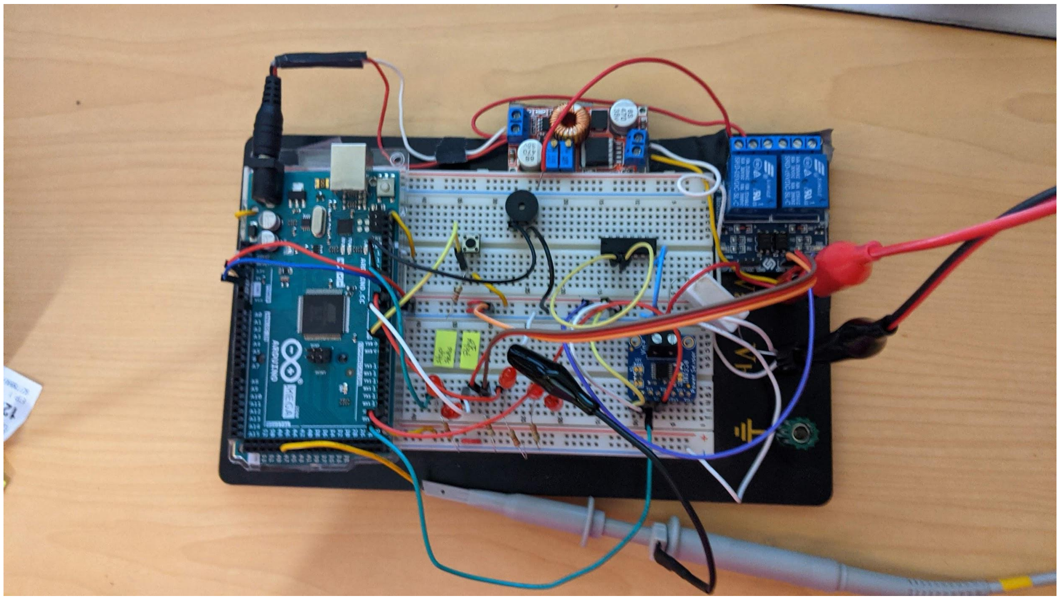


Figure 1: Breadboard setup of MCU Circuit

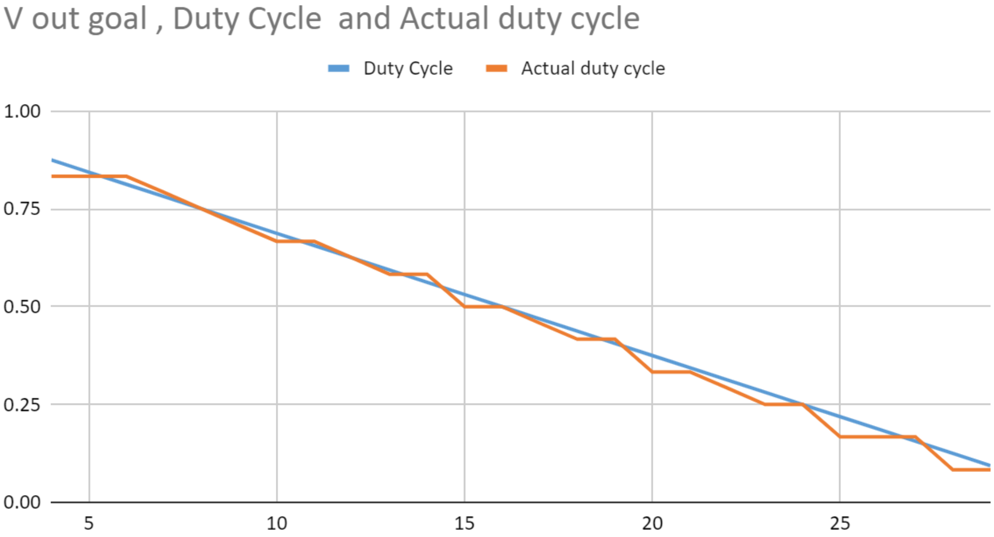


Figure 2: V_out vs expected and actual duty cycle