Engineering Capstone Project

Turbulent Wind Flow in Motorized Vehicles

Mason Ciari, Trey Gohlmann, Shawn Laikupu, and Roman West

Introduction and Initial Plan

This project focuses on harnessing turbulent airflow generated by moving vehicles and reducing aerodynamic drag, both to improve efficiency and recover usable energy. Every vehicle experiences drag as it travels; the size, shape, and speed determine the amount of drag. Drag is an effect that acts in parallel to the direction of travel. This all gets considered for the vehicle's drag coefficient; the average modern car has a drag coefficient of 0.40 [6]. This coefficient has been decreasing steadily as car manufacturers spend more attention on the aerodynamics of the vehicles. This can be seen as some of the best designs have a coefficient around 0.26 [6].

Aerodynamics plays a large part in the amount of drag that the vehicle will experience. Aerodynamics refers to how the air travels around the vehicle as it travels at a velocity [15]. Depending on how the vehicle is designed, this factor can impact things such as grip and stability of the vehicle. On most cars, there is an increased amount of drag located at the rear of the vehicle. With this in mind, our group wants to test different methods to utilize this energy [6]. This could come in many forms, but for this project, the harnessed energy would be repurposed into the vehicle to increase the range in electric vehicles. That would be the active harness method, while the passive method would be a redesign of the vehicle to break up the drag towards the rear. The reason for the vehicles to be electric is that the problem of storing the extra energy is easily redirected to the internal battery packs.

For the active method, there were a few methods that were considered before deciding on the use of a turbine design that would be fixed to a generator. There are two types of turbines: horizontal-axis and vertical-axis [16]. The difference between these two is how the turbine is oriented, and can be distinguished by which way the rotating axis acts. Each design had its benefits considered. The horizontal-axis turbine will generate more electricity, but comes with the downside that it tends to be impacted more by the direction of ambient wind [17]. Vertical axis turbines work in more turbulent conditions and work in 360 degrees. The drawbacks of these fans are that they generally have to be larger to generate similar power to horizontal-axis turbines [17].

Project Specifications

Below is the group's objective tree. This highlights the problems and objectives that were decided early on in the project. These objectives were updated as the project went on due to different ideas and issues. One example is after the week two update presentation, the idea of drag became more prominent. This was brought on by a series of questions that were asked that had not been previously considered. Thanks to this, we were able to narrow down our goal and broaden our understanding of the project. This better understanding allowed the team to continue with the project with a set goal in mind. For the outcomes of this project, we are taking a consumer report approach. The data that we will be looking at will be the energy of the vehicle without the turbine assembly and then with the assembly. The ideal outcome will be that the energy produced by the assembly will be enough to justify the addition of the turbine. Results will be discussed later in the report.

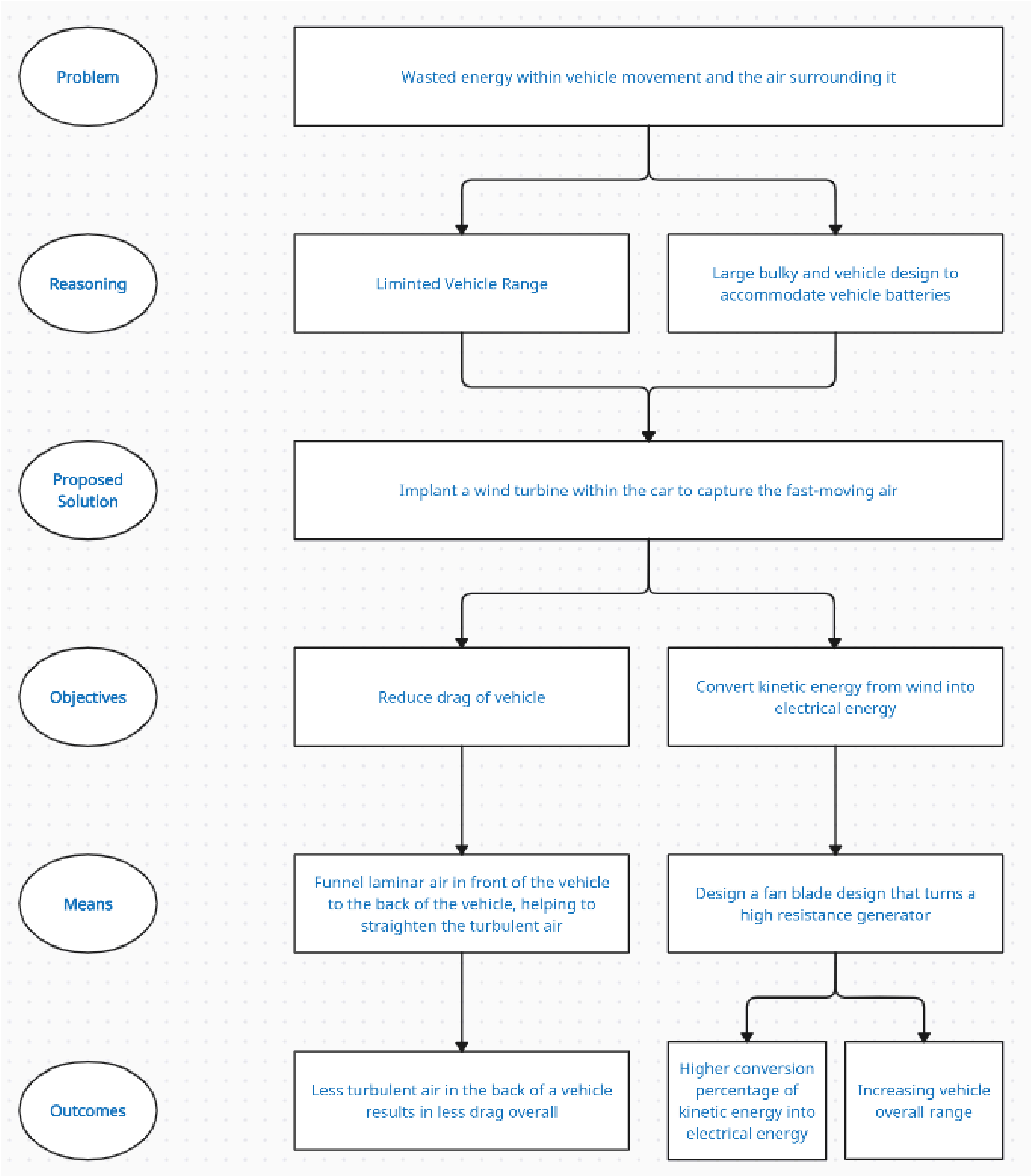


Figure 1: Objective Tree

Testing Methods

To be impartial, we tested both types of fans at the beginning of the project. The initial round of testing used three horizontal-axis turbines (two multi-blade designs and one toroidal) and two vertical-axis turbines (Darrieus and Savonius). The top two most efficient fans were then used for the practical design moving forward. This was done to help save time on designing the body of the vehicle to accommodate the two turbines.

After this initial testing, it was found that the toroidal had the best efficiency of 26%. This was found by taking the turbine's ideal output according to (1) and then comparing it to the actual energy output. Using a voltmeter hooked up to the generator to give the actual power output of the turbine. This was done using a box fan as the conditions would be repeatable and wouldn't vary like using outside wind.

$$P = \frac{1}{2}(\rho \bullet A \bullet v^3) \quad (1)$$

The next stage of testing used a baseplate design that will incorporate all the parts (Raspberry Pi, voltage sensors, generator, power supply). All of these components would be necessary for monitoring the voltage and would be kept for both types of testing. They were then mounted to the 1:14 scale RC Car. After the car was set up, multiple trials were run using either the fan setup or the base car setup. The base car set would be the control, where we would only be measuring the voltage drain to run the vehicle on both high and low speed settings. The other trials would be run with the turbine assembly and would measure the power drain from the battery. At the same time, the second INA219 sensor will be reading the energy generated by the turbine. For the speed of the tests, we used a speedometer alongside the RC car as it was being tested.

Electrical and Technological Design

Raspberry Pi and Voltage Sensors

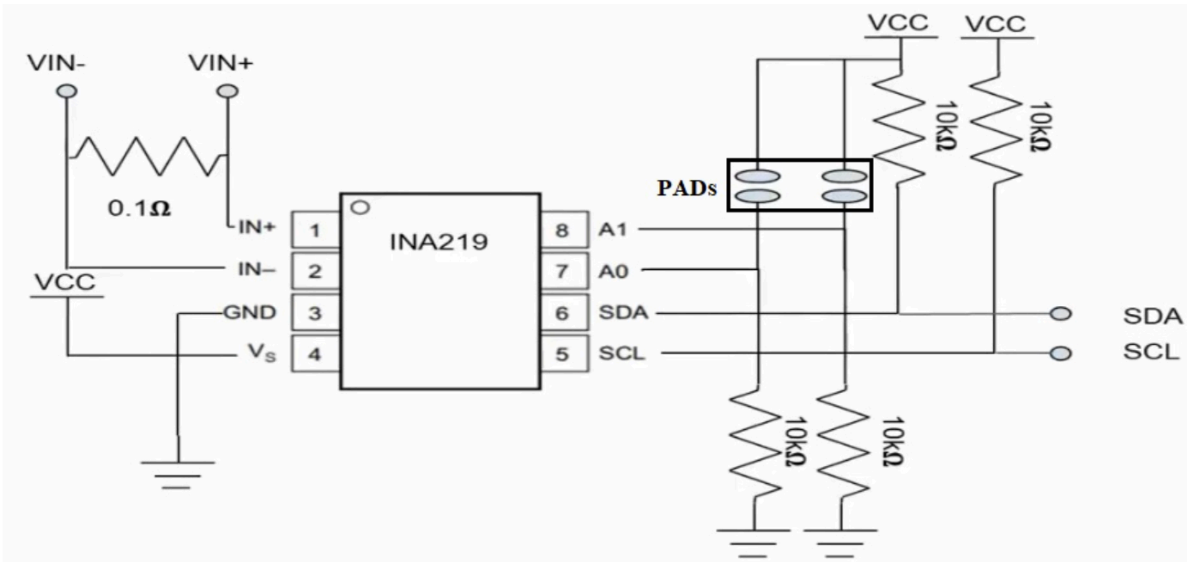


Figure 2: INA219 Voltage Sensor Schematic

The electrical components of this project are very important as they will be the main way that the data is collected. Our group has come up with a way that we believe will be best given the space and task at hand. Utilizing a Raspberry Pi, the I2C bus will be used to receive data from two INA219 direct current (DC) monitors. The INA219 sensor works by measuring the voltage drop over a shunt resistor (0.1 ohm). The first sensor will be placed upstream of the RC car motor, allowing us to get a baseline energy consumption from the motor. The second sensor will be downstream of the generator, which will give the group the amount of energy recovered from the motion of the car due to the rotating turbine.

To make the Raspberry Pi portable, a PiSugar S Plus [9] is fastened to the underside of the general-purpose input/output (GPIO) pins. The PiSugar connects to the Raspberry Pi by using pogo pins, which maintain a firm connection to the GPIO pins. By using pogo pins, all of the Raspberry Pi's GPIO pins remain available for use, while also providing the Raspberry Pi with a constant five-volt, three-amp power supply. The PiSugar is rated for 5000 mAh, which will provide the Raspberry Pi with around 6 hours of run time on a single charge. If you want to utilize the Raspberry Pi's I2C bus one, ensure the automatic power restore on boot toggle switch is positioned to the off position. Otherwise, the PiSugar will disrupt all I2C communication on bus one. The PiSugar also comes with an external on-off switch, located under the Raspberry Pi's HDMI port, which will cut the power supply to the Raspberry Pi.

Python Programming

The Python coding language was used to create programs for the Raspberry Pi to interpret and record data from the INA219 sensors. A DC power supply was used to calibrate the INA219 sensor by supplying a constant 6 volts and 0.5 amps DC power. From there, the code within the Raspberry Pi was altered to account for sensor drift and power calibration. Sensor drift was calculated by supplying 0W of power to the sensors and recording the output of the sensors, then modifying the code to nullify the power recorded by the sensors.

To import the Python libraries, a virtual environment (venv) must be set up in the directory you choose to contain all of the code. A venv is needed due to some of the following libraries being downloaded from the internet. The first library used is the adafruit_ina219 Python library, so the Pi can decode and calibrate the voltage sensors. Using this library, the bus voltage, current, and power will be harvested from each of the sensors. To improve measurement accuracy, the ADC resolution is configured to take a sample size of 32 and average those values to be recorded each iteration of the script. Without setting the ADC resolution, only a sample size of 1 will be taken, which could occur at a peak or valley, leading to distortion in the data collection. The libraries Busio and Board will also be imported to tell the Raspberry Pi to look at the I2C bus, and which INA219 address to read.

The openpyxl [8] library will be used to log each bit of data to an Excel sheet. This Excel sheet will store all of the data from the voltage sensors along with the number of iterations that occurred while running the data collection program. A new sheet will be created within the workbook each time the program is initialized with a unique name by using the DateTime Python library. To store the Excel sheet, the OS [11] library is used to check to see if there is already an xlsx file with the same name. If there is, then that workbook will be loaded and a new sheet will be created. Otherwise, a new workbook will be made with the specified name and a new sheet will be created. After each iteration of the Python script, the data collected will be appended to the worksheet and saved. The worksheet is saved after each interaction to reduce the risk of data corruption, data loss due to power loss, and space complexity. Finally, the workbook is closed to ensure nothing else is being read or written to the file, and the USB drive can be removed to analyze the data.

The Time [12] Python library will be imported to control the speed of the program so there is a pause between each interaction, so an iteration will roughly occur every second. Iteration speed can vary due to the performance speed of the Raspberry Pi's processor and communication time through the I2C.

There will need to be a way for a user to remotely access the Pi to be able to start and stop the Python script; to do this, the Raspberry Pi's built-in SSH feature will be utilized. This will need to be enabled within the configuration menu. The Raspberry Pi can also act as its server and send out a Wi-Fi hotspot signal. This will be used to connect to the device using a software called PuTTY [10], which allows users to connect to servers by logging in using the server's static IP address. When using PuTTY, only the terminal command line will be able to be used. To run the Python program, the user must first change the current directory to point to where the Python file is located, and then tell the computer to use the venv where the Python libraries are located. When running the Python programs, a live sensor readout will be displayed within the terminal. If the script needs to be stopped, the user will use a keyboard interrupt (Ctrl+C) which will immediately stop the iteration of the script, then run the save and close function for the workbook.

Refer to Appendix A for a detailed reference of the Python programs used.

Figures 3 and 4: Raspberry Pi and Pi Sugar S Plus (Portable Battery Pack)

RC Car Motor and Generator

The energy capture system uses two DC motors. The first motor is used to propel the system forward. The speed of this motor is controlled using pulse-width modulation (PWM). PWM is an electrical process of rapidly turning a DC energy source off and on. When graphed, the power flow will look like rectangular waves ranging from the working power to zero. PWM is useful in conserving energy by reducing the energy a motor consumes.

The second motor is used as a generator, which is attached to a turbine. The turbine rotates the shaft of the motor, converting mechanical energy into electrical energy. This energy can then be stored in a battery or fed back into the system.

Design Specifications and Changes

Our project contained many renditions for all of the parts. Our major design alternatives happened within the fan blade design. Through research and testing, we deduced that a larger area and a sweet spot of about 5 blades was the best design [2]. Therefore, we decided to go with the toroidal design, which met our needs and the desired power output. This design helps to maintain smoother airflow and potentially improve efficiency in low-speed wind conditions [16]. The Toroidal blade designs significantly reduce aeroacoustic noise [17]. The toroidal fan has a ring-shaped design that removes the swirling air at the blade tips that normally wastes energy and creates noise. This helps the turbine run more efficiently and generate power more smoothly. Regular wind turbine blades create little air spirals at their tips, which waste energy and make noise. The toroidal (ring-shaped) design stops those spirals from forming, so the turbine can spin more efficiently and produce more electricity with less noise [16]. Also, the continuous loop structure might allow for increased structural strength with less material. This could make manufacturing more efficient or reduce weight in certain applications. Furthermore, a toroidal design outperformed the standard blade sweep, or any other design for that matter. It was able to achieve an efficiency of 26%, compared to other blades not even able to reach 10%. This efficiency was calculated through the power output from the generator divided by the power of the wind.

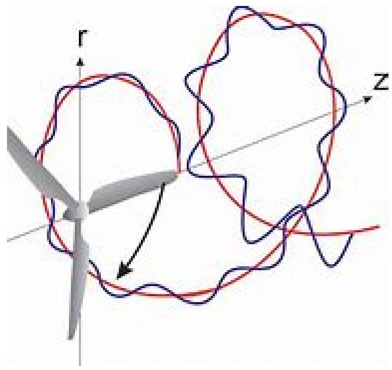


Figure 5: Tip Vortices

Figure 6: Toroidal Fan Design

Just as important as fan blades, our decision for motor size mattered. We could go with a smaller motor that turned easily with the slightest of breezes, but we would reduce our power output. On the other hand, a larger motor would produce more power, but require more torque to rotate and generate it. Again, through testing in a moving vehicle, we concluded that our confirmed toroidal design was capable of rotating the larger motor with ease.

To mount the fan and motor to the platform system, we started with a tripod mount. The mount accommodated the smaller motor, but was able to be adjusted using a conversion kit for the big motor. The mount worked originally for preliminary testing, but problems arose when we tried to connect it to the platform. Our first issue was screw holes to attach the mount to the platform, which was added to our final design. Our second issue was print time. It takes the Prusa Mk4 around 5 hours to print the tripod design, using about 123 grams of filament to do so. Going back to the drawing board, Mason Ciari created a bipod mount with screw holes for attachment. The bipod cut printing time down to 2.5 hours, and used 68 grams of filament. The lower print time was ideal in case anything broke during testing, and less filament would be wasted.

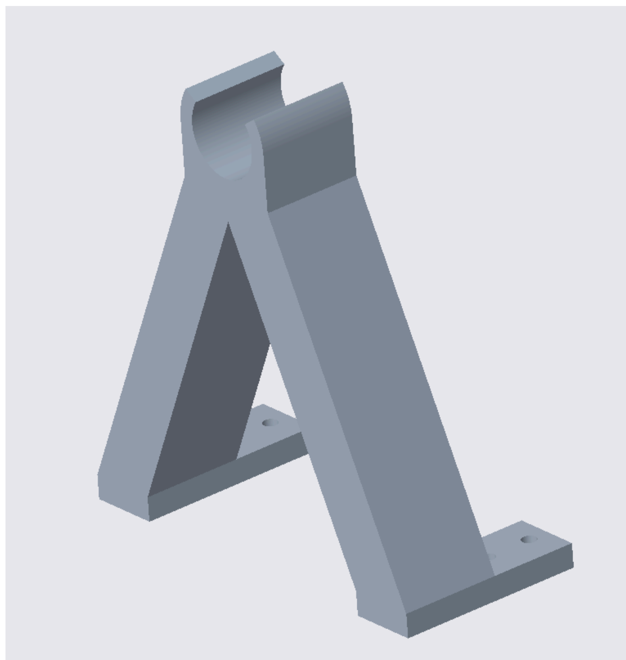


Figure 7: Fan Mounting System

Figure 8: Motor Mount Conversion

One necessary part for the RC car modifications was a platform that could accommodate all the components that were used. This began by finding a way to mount the platform to the undercarriage, either by using the existing structure or adding to it. Adding to it wasn't very viable as there was little working room due to the existing components (motor, battery pack, drive shaft). This left using the existing structure. For this method, the first plan was to add zip ties, which worked for a short period. The next iteration was to keep the back two zip ties on while the front zip ties were replaced with a wire that could be tightened or removed. This would provide easier access to the battery. The downside was that this process was very difficult as the round wire didn't interact well with the slots. The following iteration was an elastic compression that wrapped around the frame of the RC car.

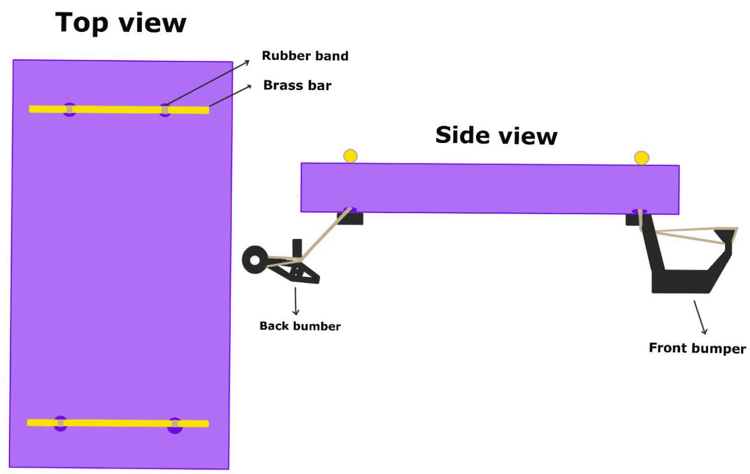


Figure 9: Rubber Band System

The rubber bands that were used broke during some initial testing and were removed for a pin design. This design worked on the first board, after redesigning the board to accommodate the Raspberry Pi holder, these pins did not align in the rear of the platform. This led to the final design for securing the platform, which was two pins by the front and two magnets on the rear. The purpose of this was to provide support at the front due to the weight of components, and an added force from the magnets where there was less weight.

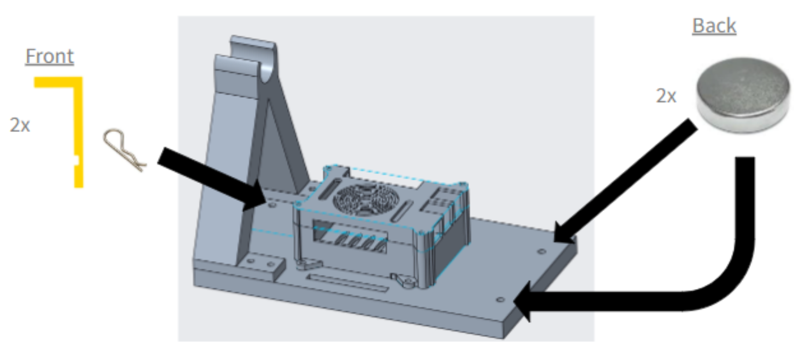


Figure 10: Pin and magnet release system

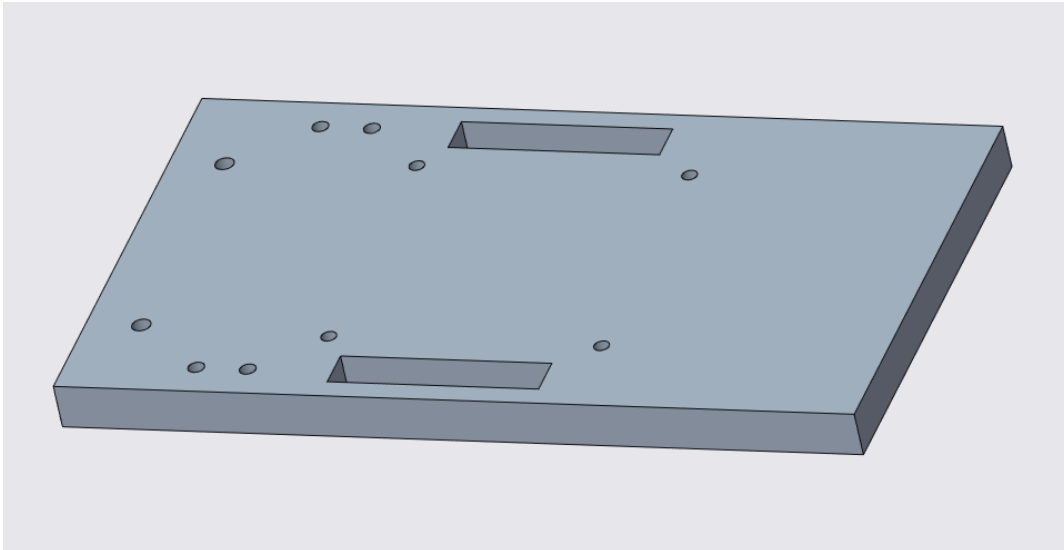


Figure 11: Final Platform Design

The final platform design also included two added slots through which the wires below the board could be fed. These slots were located on either side of the modified Raspberry Pi holder. This holder went through multiple iterations of design.

Figure 12: Initial Design found online

The main reason was the addition of a portable power bank on the underside of the Pi. This meant that the design that was found online would not work and needed to be heightened. In addition two of the original screw holds were lowered and served as support for the battery that hung down slightly. The two remaining screw holds were also modified into the walls for extra support as they broke numerous times before the change. One possible reason for this break was due to the increased height, but the initial thickness. This could have led to weakness when the torque of screwing the screws in was applied.

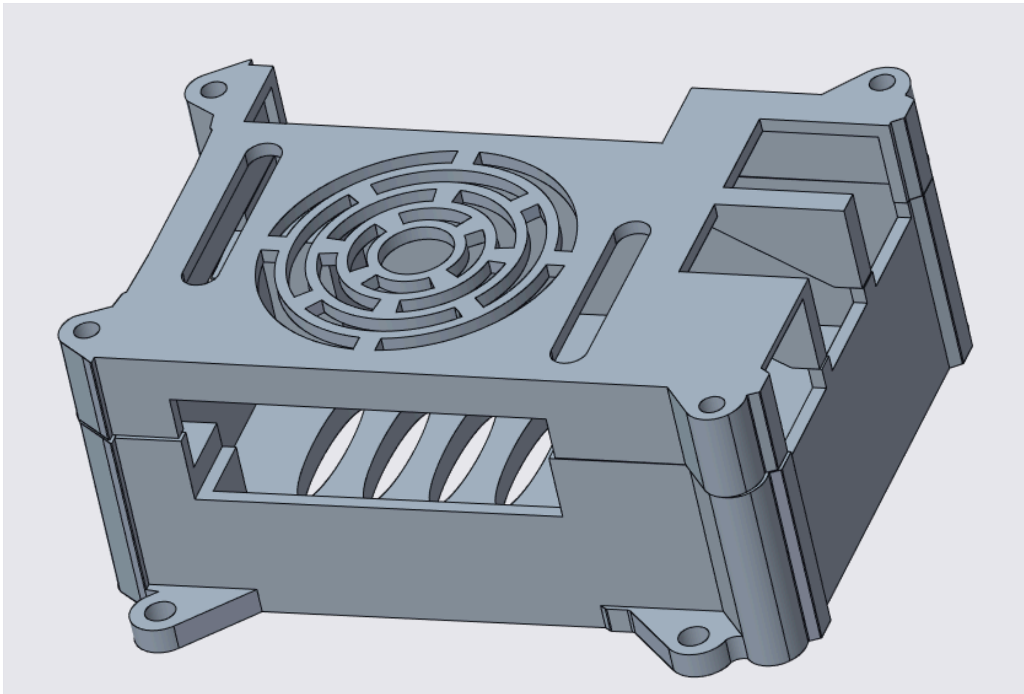
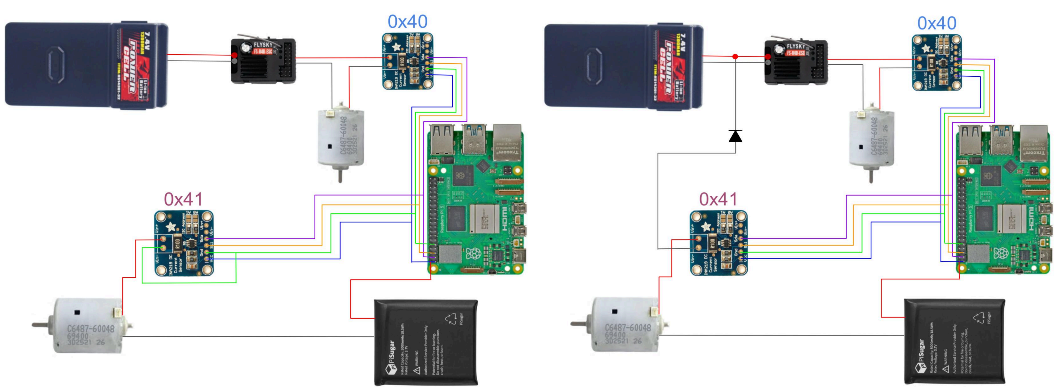


Figure 13: Final Assembly Raspberry Pi Holder



Figures 14 and 15: Inner electrical components

Figure 14 shows the circuit diagram illustrating the internal components that will be used in the actual implementation. In Figure 15 the second INA219 sensor (monitors the power input) is wired to ground for testing purposes.

Design Analysis

Overall, the device that was designed does a good job at comparing with and without the turbine, but has a few downfalls. First is that the smaller scale is difficult to scale up to what would be more realistic for a vehicle. The second is that the design would have to be implemented in a real vehicle design. Most vehicles have turbulent flow at the front and rear of the vehicle. Pictured below is the Tesla Model 3 Aerodynamic testing (this vehicle was used as it is electric and already has a battery array in place). The Turbine could only go to locations where turbulent flow is occurring. Laminar flow is already fairly efficient and would generate less energy.

Figure 16: Tesla Model 3 Aerodynamic Test

This problem had a few solutions that, given additional time, could be further studied. The first would be to print a scaled-down model of this vehicle and test the results of the turbine. For this solution, the turbine would be placed near the rear of the vehicle where the laminar flow begins to turn into turbulent. That solution could apply the results of the best turbines that were found from our group's testing.

The alternative solution to this would be to design a system into the frame of the vehicle. That design would funnel the laminar flow at the front of the vehicle to the back. In doing so, the flow of new air would interrupt the laminar flow at the rear. This alternative design would be something that would need further research to see the extent of the impact and the best way to move the air to the rear. This could also be something that could be modeled using Creo modeling software. The downside to this comes both with the scale and the application side of the idea.

If this were applicable, the funnel would interfere with the internal design of the vehicle. Depending on the automotive manufacturer, they might not be willing to do this. This problem would be present in our solution as well as the alternative. A good example of this would be the Mythbusters experiment on divots in golf balls applied to vehicles. While their testing found that it made the vehicles fractionally more efficient, it wasn't as practical on a wide production scale. It also wasn't repeatable according to a major car manufacturer.

Results

Category	RC Voltage	RC Amp	RC Power	Fan Voltage	Fan Amp	Fan Power
Average	3.4108	0.5021	1.8136	1.0705	0.1508	0.1619
Standard Dev	1.8833	0.3799	1.7416	0.3925	0.0079	0.0606
95% CI (-)	-0.3558	-0.2577	-1.6696	0.2855	0.1350	0.0406
95% CI (+)	7.1775	1.2619	5.2968	1.8555	0.1666	0.2831
Efficiency (Fan Input/RC Output)=	8.9253					

Table 1: Low speed with turbine mount

Category	RC Voltage	RC Amp	RC Power	Fan Voltage	Fan Amp	Fan Power
Average	4.1274	0.8851	3.6504	1.9055	0.2252	0.4262
Standard Dev	0.6207	0.0552	0.5847	0.6423	0.0149	0.1334
95% CI (-)	2.8861	0.7746	2.4810	0.6209	0.1953	0.1593
95% CI (+)	5.3688	0.9956	4.8199	3.1901	0.2551	0.6931
Efficiency (Fan Input/RC Output)=	11.6752					

Table 2: Full speed with turbine mount

Category	Voltage	Amp	Power
Average	3.1440	0.4564	1.5775
Standard Deviation	1.8547	0.3843	1.6853
95% CI (-)	-0.5653	-0.3122	-1.7930
95% CI (+)	6.8533	1.2249	4.9481

Table 3: Low speed without turbine mount

Category	Voltage	Amp	Power
Average	3.0987	0.4920	1.6021
Standard Deviation	1.7024	0.3842	1.6174
95% CI (-)	-0.3061	-0.2764	-1.6326
95% CI (+)	6.5036	1.2603	4.8369

Table 4: Full speed without turbine mount

Our goal for efficiency, determined by dividing the input power from the fan by the output of the RC car motor, was above 10 percent. Based on our testing methods, we saw an efficiency of 8.93 with the turbine mount and turbine on the RC car at low speeds, almost reaching the goal, but not able to. However, with the full assembly and at the higher speed setting, the efficiency increased to 11.66, which exceeded the goal percent. To calculate the efficiency, the average voltage generated by the turbine was divided by the average output of the RC car's motor.

Tables 3 and 4 can be referred to for the baseline testing of the RC car without the assembly. These tables show the impact of the full assembly acting on the car. For example through increased power output to overcome the extra friction from extra weight.

RC Without Full assembly			**Difference (V0 - V10)**	**Percentage Change**
Low Speed	V(0)=	8.085	0.303	
	V(10)=	7.782		
High Speed	V(0)=	7.79	1.56	
	V(10)=	6.23		
Full Assembly				
Low Speed	V(0)=	8.156	0.275	110.18
	V(10)=	7.881		
High Speed	V(0)=	7.881	0.589	264.86
	V(10)=	7.292		

Table 5: Voltage difference across 10-minute intervals

Table 5 is the voltage drop across 10-minute intervals. The main goal of Table 5 is to determine the voltage drop caused by drag from the full assembly compared to without. From baseline, there was a voltage drop of 0.303 volts at low speed, and 1.56 volts at high speeds. By adding the full assembly, we saw a decrease of 110.18% in voltage drop at low speeds, and a decrease of 264.86% at high speeds. Having a decrease in voltage drop like this affirms that the fan leads to a lower drag because the RC car needs to use less power to keep a consistent speed.

Refer to Appendix B for a detailed reference of the collected power graphs.

Future Improvements

While this project achieved what our group wanted to do, several improvements could be made in the future. The first recommended improvement would be further testing with actual vehicle designs. This would give conclusive results of how the drag of an active capture system impacts real vehicles. As discussed earlier in the paper, the Tesla Model 3 would be a good base vehicle for this type of testing. Based on the data that is present for that vehicle model, there would be a large base for comparison to the printed version. This would be important because the expected outcome of the 1/14 scale would be similar to the real-world counterpart.

Another improvement that could be made would be an increase in scale after the previously mentioned testing. Due to the smaller scale, there may be additional factors that we may not have been able to account for. One such example would be the weight of electric vehicles' battery packs, which is a significant downward force on models like the Tesla 3. The increase in scale could also allow for realistic testing conditions that we were not able to achieve.

The last recommendation that we have as a group would be to test using realistic conditions. The testing done with the RC car was done at speeds of 14.6 mph to 16.1 mph on the lowest speed and 19.4 mph to 20.6 mph on the highest setting. Modern vehicles, especially electric can reach and maintain much higher speeds, especially with standard speed limits around the United States. Testing accounting for this could allow for more accurate results for the power input to the vehicle, as well as the output of the turbine. Another condition that would impact the fan would be ambient wind. Due to testing in a covered arena, we would not have the added factor of external wind. This wind could impact the overall input of the vehicle and output of the turbine, depending on the direction. For this project, this was not as necessary because we aimed to see if this idea could be applied without the external factors. An additional factor that could be more realistic was the testing surface. The surface of the track is a synthetic material, such as rubber does not have similar properties to asphalt. This difference could see a greater or lesser traction factor when interacting with the wheels. This overall could impact the drag on the vehicle [15].

Conclusion

This project set out to evaluate the potential of capturing turbulent airflow generated by a moving vehicle as a means to recover energy and reduce aerodynamic drag, specifically within the context of electric vehicles. By designing and testing both active and passive aerodynamic systems, we aimed to determine whether such methods could contribute meaningfully to vehicle efficiency.

Through constant testing, the toroidal turbine emerged as the most effective design, achieving a peak energy recovery efficiency of 11.66% at higher speeds and nearly 9% at lower speeds. These results and data were gathered using a Raspberry Pi system equipped with INA219 sensors, which allowed for more accurate measurements of both power consumption and energy generation on a 1:14 scale RC vehicle. Also, the presence of the fan assembly was shown to reduce overall power draw from the motor, indicating a potential decrease in drag.

While the small scale model and controlled testing environment limits real world applications, the team's findings suggest that integrating such systems into electric vehicle designs could enhance overall performance and energy efficiency. Lastly, any future improvements should focus on scaling the system, incorporating modern vehicle models, and conducting tests under varied environmental conditions to better assess its viability in full-scale automotive applications.

Bill of Materials

EGR Capstone: Scale RC Car Model				
Item	**Qty.**	**PP Qty.**	**Vendor**	**Total Cost (USD)**
Hatchbox 3D Filament 1kg	345g	$25.99	Amazon	$8.97
Auoshi RC Car	1	$89.99	Walmart	$89.99
Raspberry Pi Gen 3	1	$47.85	Amazon	$47.85
Hp Deskjet 5550 Carriage Assembly Motor	1	$9.99	Missingcord	$9.99
PiSugar S Plus Portable Battery	1	$29.99	Amazon	$29.99
USB Thumb Drive	1	$6.99	Amazon	$6.99
Notebook M2*8 Screws	8	$0.02	Amazon	$0.16
HiLetgo INA219 Sensor Module x2	1	$8.19	Amazon	$8.19
Socket Cap 8-32 Screw	4	$0.23	Bolt Depot	$0.92
Hex Nuts 8-32	4	$0.07	Bolt Depot	$0.28
Washer	8	$0.07	Bolt Depot	$0.56
Machine Screws 10-32 ⅞”	8	$0.12	Bolt Depot	$0.96
½” Magnets	4	$15.99	Amazon	$1.28
24-gauge wire	4ft	$8.99	Amazon	$0.36
Total Cost Per Unit				$206.49

Background on Group Members

Mason Ciari Background (Email: m.ciari.ason@gmail.com)

In the summer of 2024, I designed and built a functional wind tunnel for the Cornell College Engineering department. The goal of the wind tunnel was to test the lift and drag of any object placed within the testing chamber. With a tight budget, it was difficult to accomplish the goal of a larger wind tunnel compared to the previous years. However, due to well-made connections within the community, I was able to build it for less than the allotted amount. This project helped to spark my interest in wind power, especially harnessing wind energy on freeways. This idea eventually snowballed into what we are currently doing, with a wind turbine becoming a part of the vehicle.

Trey Gohlmann Background (Email: treygohlmann@gmail.com)

For this project, I am using a culmination of my computer science minor education and work experience down in the Cornell Information Technology office to integrate technical components into the project. My extended experience with the Python coding language is also helping me to be able to put together a program that integrates multiple different packages and deals with the data processes of reading, writing, and manipulating. Being able to efficiently run these tasks allows our CPU to run more cycles of data collection, which in turn will give the group a more precise image of whether our designs are worthwhile.

During the summer of 2024, I had an electrical engineering internship with Design Engineers PC, which also gave me a strong background in the circuitry for the project. This will be very useful when analyzing the data collected to ensure that the numbers we are getting look correct; otherwise, the data collection sensors will need to be calibrated to get in the correct range for the data we are expecting.

Shawn Laikupu Background (Email: laikupus@gmail.com):

My background in this project reflects the knowledge and skills I've developed through previous projects and coursework at Cornell College. Courses such as Introduction to Engineering, Circuits, Electronic Instrumentation, and Manufacturing Processes have been especially influential. These classes provided both theoretical understanding and hands-on practice, allowing me to build a solid foundation in core engineering concepts like circuit analysis, sensor integration, signal processing, and fabrication techniques. This technical base proved essential in completing various parts of the project, from designing and testing components to troubleshooting system-level issues.

Beyond the technical aspects, my experiences working in diverse project teams have been equally valuable. Collaborating with peers on multi-phase assignments taught me the importance of clear communication, effective time management, and shared accountability. I learned how to navigate differing opinions, delegate tasks based on individual strengths, and adapt when plans inevitably changed. These interpersonal and project management skills were crucial in ensuring our group remained aligned and productive throughout this project.

This project has also deepened my appreciation for the engineering design process as a whole. It reinforced the importance of testing ideas, learning from failure, and refining solutions over time. It has been rewarding to see how the knowledge I once viewed in isolation across separate courses came together to solve real, complex problems. Altogether, this experience has not only strengthened my technical competence but has also helped me grow into a more thoughtful, collaborative, and resilient engineer.

Roman West Background (Email: Rwest00011@gmail.com):

My background for this project is primarily in the form of in-class skills I have learned at Cornell College. The two main classes that pertain to this project are fluid mechanics and various classes working on the design and redesign process using PTC Creo. The fluid mechanics class first introduced me to the idea of turbulent and laminar flow, which increased my interest in how this can be applied to vehicles. Coupling this knowledge along with my understanding of the efficiency of generators allowed me to help think of this project idea. With that idea in mind, the idea posed several challenges that could be solved by creating in-house parts. The use of PTC Creo and the slicing software Prusa would be a lucrative part of the project. My experience using both would help me to contribute to the overall group.

Additionally, my previous roles throughout college and in other team environments allow me to understand the importance of teamwork, communication, and other important skills to work efficiently. This will also allow me to be a better teammate and contributor to the group.

References

[1] "Adafruit INA219 Current Sensor Breakout." Available:

https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ina219-current-sensor-breakout.pdf

Date Last Accessed April 8th, 2025

[2] "Blade Types for Wind Turbine Users | The Complete Guide," *Missouri Wind and Solar*,

2024. https://windandsolar.com/blog/blade-types-for-wind-turbine-users-the-complete-guide/

Date Last Accessed March 27th, 2025

[3] B. Schaffer, "Independent Aerodynamic Study of Tesla Model 3 by Unplugged

Performance," *Unplugged Performance*, Oct. 18, 2020.

https://unpluggedperformance.com/aerodynamic-study-of-tesla-model-3/

Date Last Accessed April 8th, 2025

[4]"busio – Hardware accelerated external bus access — Adafruit CircuitPython 9.2.1

documentation," *Circuitpython.org*, 2019.

https://docs.circuitpython.org/en/latest/shared-bindings/busio/

Date Last Accessed April 8th, 2025

[5] ÇengelY. A. and J. M. Cimbala, *Fluid mechanics : Fundamentals and applications*, 3rd ed.

New York: Mcgraw Hill, 2014.

Date Last Accessed March 28th, 2025

[6] F. Gray, "The Science Behind Car Design: How Aerodynamics Affects Performance,"

Nationwide Vehicle Contracts, Feb. 16, 2023.

https://www.nationwidevehiclecontracts.co.uk/blog/the-science-behind-car-design-how-aerodynamics-affects-performance

Date Last Accessed March 26th, 2025

[7] “INA 219,” *Google.com*, 2025.

https://www.google.com/url?q=https://www.electroniclinic.com/ina219-current-sensor-with-arduino-circuit-and-code-explained/&sa=D&source=docs&ust=1744140438392802&usg=AOvVaw1LfiCdtN1ULv1w9ZcpmB_O

Date Last Accessed April 8th, 2025

[8]“openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.0.3 documentation,” *openpyxl.readthedocs.io*. https://openpyxl.readthedocs.io/en/stable/

Date Last Accessed April 8th, 2025

[9] “Pi Sugar Portable Raspberry Accessories,” *Google.com*, 2025.

https://www.google.com/url?q=https://www.amazon.com/Pisugar-Portable-Raspberry-Accessories-handhold/dp/B09J3WS695&sa=D&source=docs&ust=1744140567567137&usg=AOvVaw1GNCt7kIbMZ-rLMfVCExR1

Date Last Accessed April 8th, 2025

[10] “PuTTY User Manual,” *Earth.li*, 2025. https://the.earth.li/~sgtatham/putty/0.83/htmldoc/

Date Last Accessed April 8th, 2025

[11] Python Software Foundation, “os — Miscellaneous operating system interfaces — Python 3.8.0 documentation,” *Python.org*, 2019. https://docs.python.org/3/library/os.html

[12] Python Software Foundation, “time — Time access and conversions — Python 3.7.2 documentation,” *Python.org*, 2000. https://docs.python.org/3/library/time.html

Date Last Accessed April 8th, 2025

[13] “Raspberry Pi 3B,” *Google.com*, 2025.

https://www.google.com/url?q=https://magazine.raspberrypi.com/articles/raspberry-pi-4-vs-raspb erry-pi-3b-plus//&sa=D&source=docs&ust=1744140776278426&usg=AOvVaw157JY6hJ4s9s7zvCBnJuDK

Date Last Accessed April 8th, 2025

[14] S. Ivanell, R. Mikkelsen, J. N. Sørensen, and D. Henningson, “Stability analysis of the tip vortices of a wind turbine,” *Wind Energy*, vol. 13, no. 8, pp. 705–715, Mar. 2010, doi: https://doi.org/10.1002/we.391.

Date Last Accessed April 8th, 2025

[15] T. Benson, “The Drag Coefficient,” *www.grc.nasa.gov*. https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/dragco.html

Date Last Accessed March 26th, 2025

[16] “Vertical Axis Wind Turbines vs Horizontal Axis Wind Turbines,” *Windpower Engineering & Development*, 2009. https://www.windpowerengineering.com/vertical-axis-wind-turbines-vs-horizontal-axis-wind-turbines/

Date Last Accessed March 28th, 2025

[17] X.-D. Vu, A.-T. Nguyen, T.-L. Nha, H.-Q. Chu, and C.-T. Dinh, “Numerical Aerodynamic and Aeroacoustic Analysis of Toroidal Propeller Designs,” *International Journal of Aviation Science and Technology*, vol. vm05, no. is01, pp. 20–33, Jun. 2024, doi: https://doi.org/10.23890/ijast.vm05is01.0102.

Date Last Accessed April 8th, 2025

Appendix A: Python Code

This is the Python program used on the Raspberry Pi when running tests. VoltageSensors.py is used to record data from both sensors, while VoltageRC.py is used to only record data from the RC car during the no A-Frame testing. There is no program for just recording the data from the generator because for there to be data from the generator, the RCcar must be running.

VoltageSensors.py

```
import time
import os
from openpyxl import Workbook, load_workbook
from adafruit_ina219 import INA219, ADCResolution, BusVoltageRange
import busio
import board
from datetime import datetime

#initialize the i2c bus and addresses for the ina219 sensors
i2c= busio.I2C(board.SCL, board.SDA)
rcSensor = INA219(i2c, 0x40)
genSensor = INA219(i2c, 0x41)


rcSensor.bus_adc_resoltion = ADCResolution.ADCRES_12BIT_32S
rcSensor.shunt_adc_resoltion = ADCResolution.ADCRES_12BIT_32S
rcSensor.bus_voltage_range = BusVoltageRange.RANGE_16V

genSensor.bus_adc_resoltion = ADCResolution.ADCRES_12BIT_32S
genSensor.shunt_adc_resoltion = ADCResolution.ADCRES_12BIT_32S
genSensor.bus_voltage_range = BusVoltageRange.RANGE_16V

#sets the location to be stored to be on a USB
filepath = "/media/egr385/ESD-USB/Sensors_Data.xlsx"
```

```
try:
   if os.path.exists(filepath):
      #look to see if there is a workbook already created, if yes then load the workbook
      wb = load_workbook(filepath)
      print("loaded workbook")
   else:
      raise FileNotFoundError
except(FileNotFoundError, KeyError):
   #If filepath is not found, create a new workbook with that file path.
   wb = Workbook()
   wb.save(filepath)
   print("created new wb")

#creates a unique sheet name for each trial
sheet_name = datetime.now().strftime("Track_Data_%d_%H%M%S")
ws = wb.create_sheet(title=sheet_name)

#set the headers in the excel sheet
ws.append(["Timestamp", "rcVoltage (V)", "rcCurrent (A)", "rcPower (W)","","genVoltage
(V)", "genCurrent (A)", "genPower (W)"])
wb.save(filepath)

try:
   for timestamp in range(600):    #10 min
      #get readings from the RC motor, and calibration
      rcVoltage = abs(rcSensor.bus_voltage - 0.872)*10
      rcCurrent = abs(rcSensor.current/3500)
      rcPower = rcVoltage * rcCurrent

      #get readings from the fan generator, and calibration
      genVoltage = abs(genSensor.bus_voltage - 0.868)*10
      genCurrent = abs(genSensor.current/1000)
      genPower = genVoltage*genCurrent

      print("Timestamp: ",timestamp)
      print("RC")
      print("Voltage: {:6.3f}V".format(rcVoltage), "Current: {:7.4f}A".format(rcCurrent),
"Power: {:6.3f}W".format(rcPower))
      print("Gen")
```

```
        print("Voltage: {:6.3f}V".format(genVoltage), "Current: {:7.4f}A".format(genCurrent),
"Power: {:6.3f}W".format(genPower))

        #save data between every literation to reduce the risk of the data being lost

ws.append([timestamp,rcVoltage,rcCurrent,rcPower,"",genVoltage,genCurrent,genPower])
        wb.save(filepath)
        time.sleep(0.25)

except KeyboardInterrupt:
    """This will only occur when False"""
    #keyboard interrupt is ctrl + c
    print("Data logging stopped")
    #saves the data at the very end to ensure all data is saved

finally:
    wb.save(filepath)
    wb.close()
    print("Data saved to sensor_data.xlsx")
```

VoltageRC.py

```
import time
import os
from openpyxl import Workbook, load_workbook
from adafruit_ina219 import INA219, ADCResolution, BusVoltageRange
import busio
import board
from datetime import datetime

#initialize the i2c bus and addresses for the ina219 sensors
i2c= busio.I2C(board.SCL, board.SDA)
rcSensor = INA219(i2c, 0x40)

#set the adc resolution and voltage range for more accurate data collection
rcSensor.bus_adc_resoltion = ADCResolution.ADCRES_12BIT_32S
rcSensor.shunt_adc_resoltion = ADCResolution.ADCRES_12BIT_32S
rcSensor.bus_voltage_range = BusVoltageRange.RANGE_16V
```

```
#sets the location to be stored to be on a USB
filepath = "/media/egr385/ESD-USB/Sensors_Data.xlsx"

try:
    if os.path.exists(filepath):
        #look to see if there is a workbook already created, if yes then load the workbook
        wb = load_workbook(filepath)
        print("loaded workbook")
    else:
        raise FileNotFoundError

except(FileNotFoundError, KeyError):
    #If filepath is not found, create a new workbook with that file path.
    wb = Workbook()
    wb.save(filepath)
    print("created new wb")

#creates a unique sheet name for each trial
sheet_name = datetime.now().strftime("Track_Data_%d_%H%M%S")
ws = wb.create_sheet(title=sheet_name)

#set the headers in the excel sheet
ws.append(["Timestamp", "rcVoltage (V)", "rcCurrent (A)", "rcPower (W)"])
wb.save(filepath)

try:
    for timestamp in range(600):    #10 min
        #get readings from the RC motor, and calibration
        rcVoltage = abs(rcSensor.bus_voltage - 0.872)*10
        rcCurrent = abs(rcSensor.current/3500)
        rcPower = rcVoltage * rcCurrent

        print("Timestamp: ",timestamp)
        print("RC")
        print("Voltage: {:6.3f}V".format(rcVoltage), "Current: {:7.4f}A".format(rcCurrent),
"Power: {:6.3f}W".format(rcPower))
        #save data between every literation to reduce the risk of the data being lost
        ws.append([timestamp,rcVoltage,rcCurrent,rcPower])
        wb.save(filepath)
        time.sleep(0.25)
```

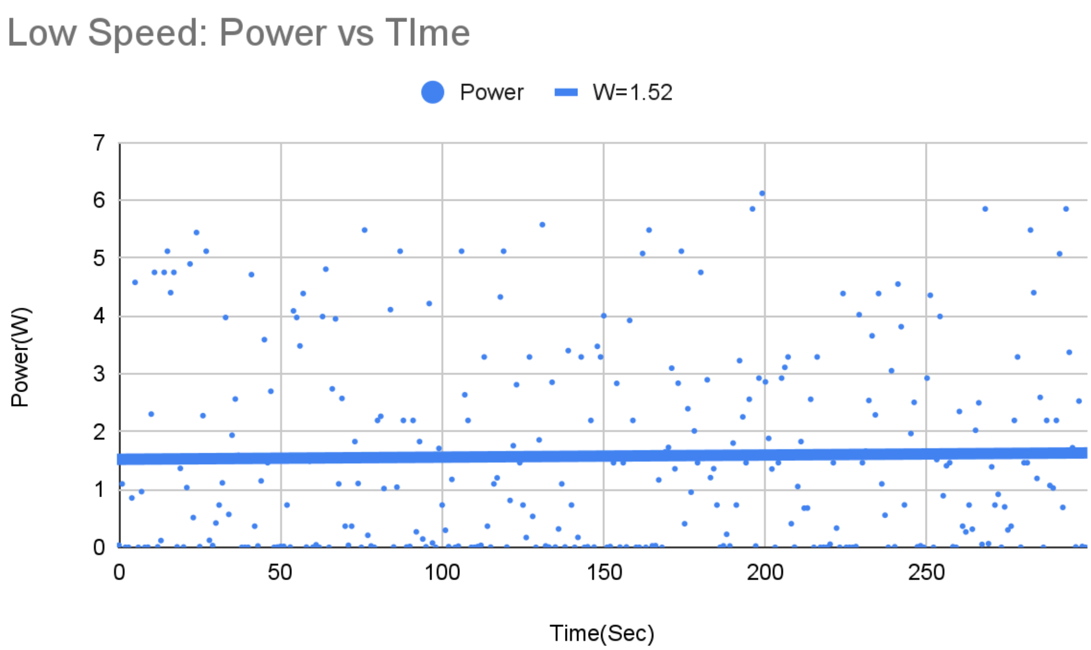
```
except KeyboardInterrupt:
    """This will only occur when False"""
    #keyboard interrupt is ctrl + c
    print("Data logging stopped")
    #saves the data at the very end to ensure all data is saved

finally:
    wb.save(filepath)
    wb.close()
    print("Data saved to sensor_data.xlsx")
```

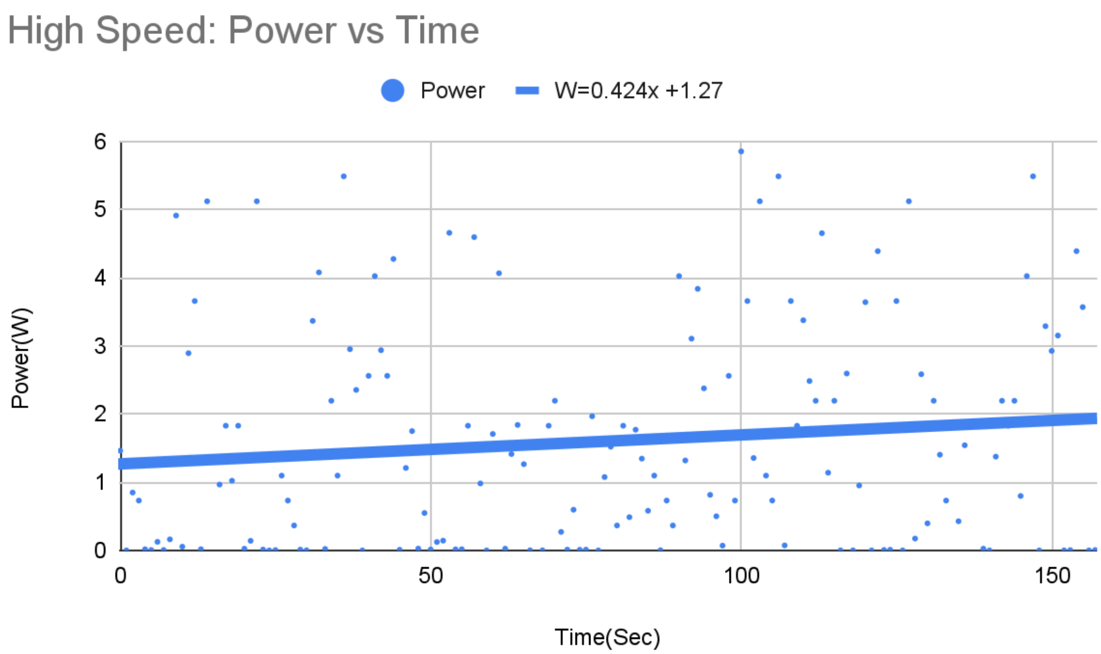
Appendix B: Power Graphs

These are the resulting graphs from each condition of testing the RC capture method.

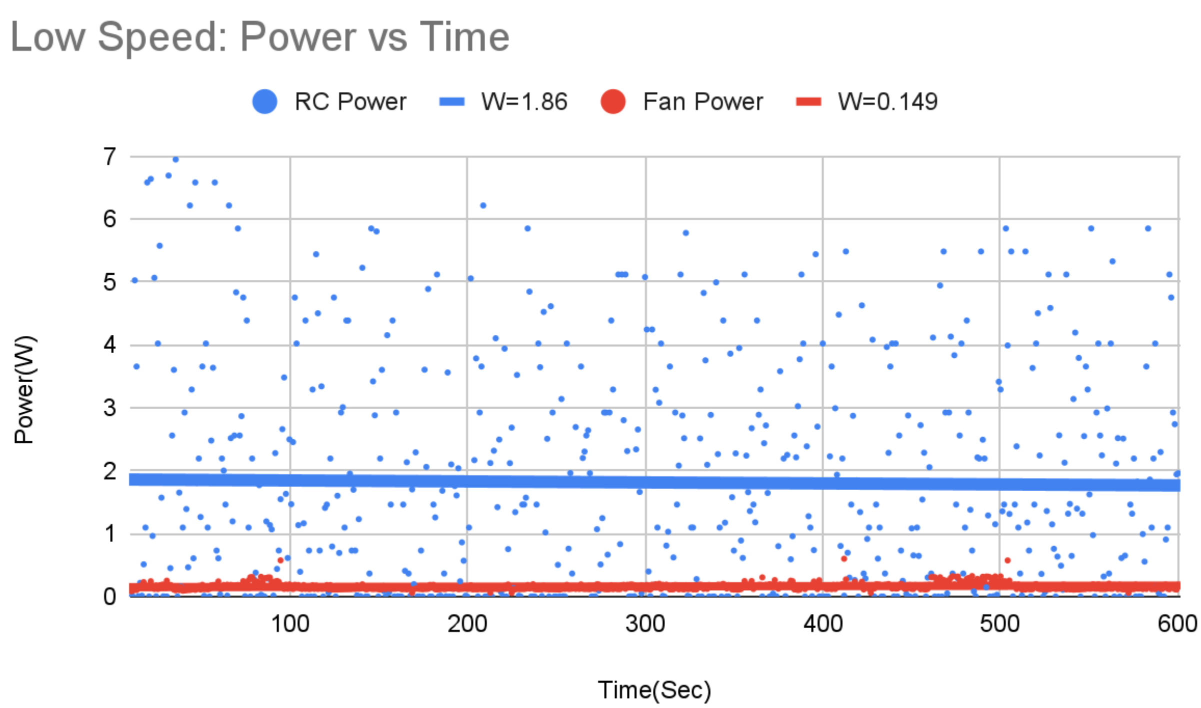
All collected data can be found: https://github.com/TreyGohlmann/RC-Energy-Capture

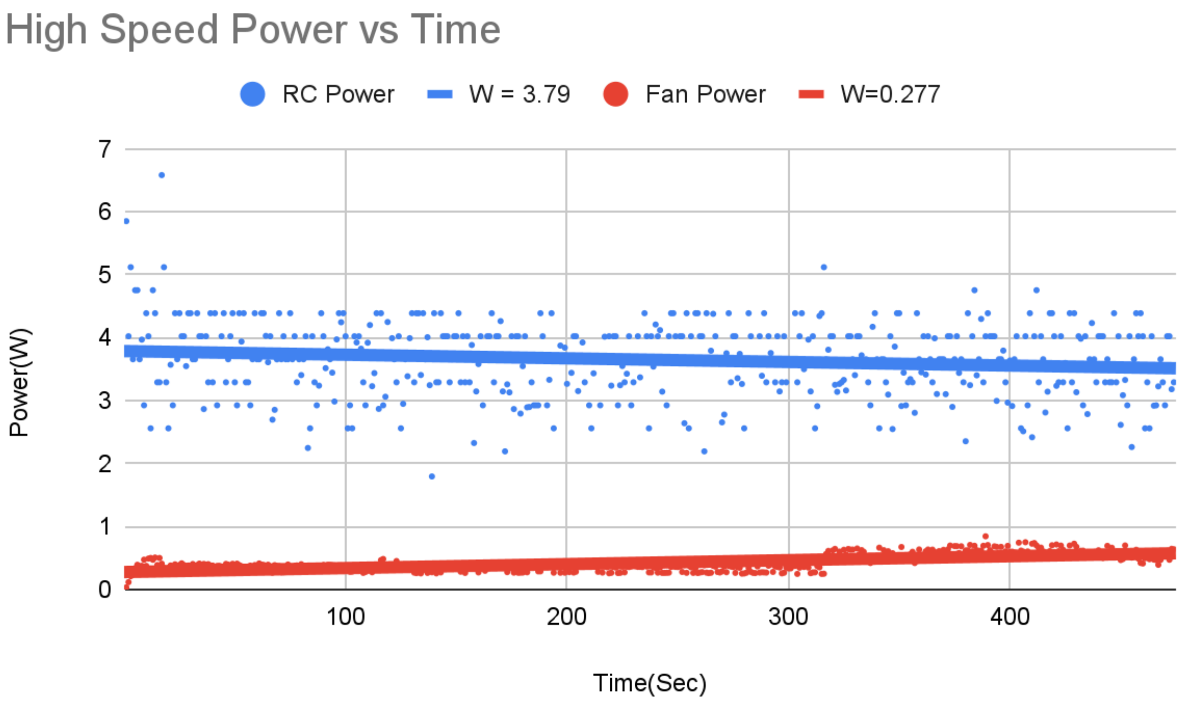
Control (No Assembly)





Energy Capture (Full assembly)




High Speed Power vs Time
RC Power
W = 3.79
Fan Power
W=0.277
Power(W)
7
6
5
4
3
2
1
0
100
200
300
400
Time(Sec)