Car-Snow Clearing Drone

A Major Qualifying Project to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Science

06 May, 2020

PROJECT CONTINUED BY:

Razan Andigani
Mechanical + Robotics Engineering

Jeremy Branch
Mechanical Engineering

Ryan Keller
Mechanical Engineering

Chioma Onyenokwe
Robotics Engineering

PROJECT SUBMITTED TO:

Professor Mehul Bhatia
Mechanical Engineering
Worcester Polytechnic Institute

Professor Nicholas Bertozzi
Mechanical + Robotics Engineering
Worcester Polytechnic Institute

Abstract

Manually removing snow from car windshields exposes people to frostbite and can be especially challenging for those with physical disabilities. This Major Qualifying Project (MQP) aims to make improvements to a drone designed one year prior to solve such problems. Unfortunately, COVID regulations restricted our access to campus resources and team collaboration. Despite these hurdles, this iteration generated a MATLAB simulation of the drone, improved the reliability of the spray-bar, and began implementation of a Real-Time Kinematics (RTK) system for precise navigation.

Table of Contents

Table of Figures

Table of Tables

List of Accompanying Submissions

Submitted with this paper as a part of the eCDR to fulfill the Worcester Polytechnic Institute Major Qualifying Project for graduating students are a number of additional files that are of use to this project. They are:

- CAD files
- Final Presentation - Slides
- Project Presentation - Poster
- Project Presentation - Video
- RTK system demo - Video
- RTK system setup guide

Acknowledgments

We would like to thank our advisors, Mehul Bhatia and Nicholas Bertozzi, for guiding us through the majority of our project. Second, we would like to thank the ME and RBE department for providing our project with the required funding needed. Third, we would like to thank WPI's UAV club, who gave us some drone piloting experience that helped us safely perform test flights with our drone. Additionally, we also would like to thank the Demining Drone/Rover Team for assisting us with the setup and integration of the RTK system. Lastly, we would like to thank Tom Gravel at *Hydrocutter* for the donation of his waterjet services for our project.

Chapter 1. Introduction

The Snow Drone MQP team of 2019-20 designed and built an unmanned aerial vehicle (UAV) to clear off snow from the rooftop and windshields of a car. Inspired by existing agricultural spraying drones, they wanted to develop a solution to clearing snow off of a vehicle while protecting an individual from the harsh winter weather. The team had to overcome challenges such as designing a drone capable of withstanding cold temperatures and carrying an additional payload of deicing fluid during flight. However, due to on-campus restrictions from COVID-19, the team was unable to complete the implementation of the project. The goal of this year's team is to improve, implement, and test the mechanical, software, and electrical systems of the drone. This year's team is also facing the challenge of not being able to collaborate on-campus due to increasing COVID-19 regulations.

Some of the mechanical components of the drone we chose to redesign include the landing gear of the drone, which was not designed to absorb the shock resulting from landing post-flight. Some options available to us included adding shock absorbent material to the drone's landing gear so that it can land safely without damaging the landing gear or its other components. We also design a propeller guard for the drone that will protect other individuals from the high-speed rotating propellers, as well as protecting the drone from extensive damage during potential crashes or collisions with its surroundings. Some challenges included choosing a material for the propeller guard that will be lightweight enough to not interfere with the drone's flight, but also strong enough to ensure the drone's components remain protected from collisions.

The previous team's spray-bar mechanism was designed with one degree of freedom (DOF), rotational on one axis, to allow the drone to successfully clear snow from the target vehicle.

The spray-bar is friction fit with a set screw to avoid unwanted translation and eliminates the need for the drone to have extremely precise in-flight control. However, a lightweight and yet powerful motor to operate the spray-bar mechanism was not discovered by last year's team. Our plan was to improve the design of the existing spray-bar mechanism, so it is lightweight enough but also supplied with enough power to rotate.

For software, we planned to complement the QGroundControl software by adding a real-time kinematic (RTK) system. This will allow the drone to not only take-off but also autonomously navigate through an environment safely, while identifying the target car and snow/ice, successfully deicing the vehicle, and then returning to its starting point. Some challenges with the addition of these components included getting the drone to recognize its target and distinguish between snow and the car. Another issue would be creating a controlled environment and flight path that will allow the drone to identify the car, navigate through its environment while avoiding obstacles, and return to its starting point. One solution would be to place markers on the target car and the starting point that the drone can identify, allowing the drone to easily navigate through its flight path.

The electrical components of the drone include electronic speed controllers (ESCs) which powers the drone's propulsion, flight control system, and spraying mechanism. A Pixhawk 4 flight controller and a global positioning system (GPS) allows the drone to hover in place and fly semi-autonomously. An infrared (IR) camera was used by last year's project team on the drone to detect the thermal differences between the snow, car, and spraying fluid. However, using an infrared camera would make it difficult to distinguish the accumulated snow and ice from a car that has been out in the cold weather for hours, since they both would have similar temperatures. To account for this problem, the target vehicle would have to be heated up before the drone begins to clear snow from the car. Last year's team suggested purchasing a stereoscopic camera to add to the drone. This camera will allow the drone to gain more accurate information when differentiating between the target vehicle and the surrounding environment through the differing contours of their surfaces.

Our plan for this year was to implement the RTK system into our drone design. The RTK system uses relative position data from two different satellites to pinpoint its exact location in longitude, latitude, and altitude. Both the base and rover stations of the RTK system will remove inconsistencies in the position data to determine its location up to the nearest centimeter, more accurate than the capabilities of a normal GPS. When paired with the drone's Pixhawk 4 flight controller, the RTK system can use QGroundControl to determine the drone's location, eliminating the need for a camera. However, the main focus for this year was to calibrate the base and rover station of the RTK system to receive satellite data and determine its location.

Through these changes to the mechanical, software, and electrical systems of the drone, we improved upon the work done by last year's team, as well as provided current and relevant testing data for future improvements to the drone's overall design.

Chapter 2. Background

2.1. Existing Work

Inspired by agricultural drones, last year’s MQP team started designing a drone meant for removing snow from vehicle rooftops and windshields. The project's previous iteration maintained a leak-free system in the drone's structure, essential to ensure the UAV’s safety and quality. However, due to the cancellation of on-campus projects, the team could not fully integrate and test the various components of their design, such as demonstrating the drone effectively taking off while carrying a one-gallon liquid payload and spraying a car with the de-icing fluid. This allowed our team to learn more about the drone and what improvements need to be made over the academic year.

2.1.1. Mechanical System

To ensure the drone works successfully, the goal was to design a mechanical system that produces maximum efficiency and runs seamlessly with the other systems. The previous team took several considerations into account when designing the drone, such as forces caused by lift, mechanical stress and strain, the operating temperature, vibrations, and many other considerations. This project's mechanical design focused on improving and designing the drone's frame, the spray system, and the rotor arms. The drone's central compartment has two main plates, each holding critical electronics for the drone’s function, such as the flight controller.

The spray system was modeled after similar agricultural drones used to spray fields. A lightweight high-density polyethylene tank with a 5-Liter capacity, mounted underneath the center compartment, stores the de-icing fluid. The fluid is gravity-fed to the tank's bottom, where it is then pumped into the spray line. The spray line consists of PVC tubing that runs to three separate nozzles mounted along a spray-bar. To allow maximum coverage of the car windshield without extreme precision in the drone's flight control, a rotating spray-bar, and multiple nozzle features were selected. The spray-bar was designed to be actuated by servo-driven poly-cord pulleys and have two degrees of freedom. The servo motor did not supply the torque required to rotate the spray-bar, so the team used a Pololu motor instead. This option was not lightweight and appeared to be challenging to control, and required more voltage than the Pixhawk 4 flight controller was able to give it. Due to the time restriction, the spray-bar actuation mechanism was not redesigned.

Finally, the rotor arms were designed using commercially available rectangular carbon fiber tubes. Each rotor arm is approximately 400mm long, with 50mm of this length constrained to the center frame assembly to prevent deflection from the thrust force. Carbon fiber seemed like the best choice as it is lightweight yet rigid and strong. The design of the collars fixture the arms to the center frame and the chosen materials performed as intended.

2.1.2. Electrical System

For the drone to autonomously navigate to and clear snow and ice from vehicles, a number of sensors and electrical parts were mounted on board. The drone is equipped with four HOBBYWING XRotor PRO 50A electronic speed controllers (ESCs) rated for six-cell batteries. It is powered by a 22.2V 25C 6S 22000mAh battery, which powers all of the drone's propulsion, flight control system, and spraying mechanism. The drone is also equipped with a HolyBro Pixhawk 4 flight controller, which acts as a waypoint coordinate navigation system enabling the drone to fly semi-autonomously and hover in place using GPS. GPS was used as the primary navigation sensor to determine the drone's latitude and longitude. It allows for location tracking and gives feedback to the control system. The flight controller is also equipped with some accelerometers and gyroscopes to determine the drone's orientation throughout the flight.

The drone used the MLX90640 infrared (IR) camera as a peripheral sensor to detect thermal differences between the snow, the car, and the spraying fluid. The thermal imaging unit was soldered to a Raspberry Pi Zero W and its pigtails were soldered to the Pi, where it will be powered from. To provide more in-depth information, a stereoscopic camera was planned to be mounted with servo control to rotate about the pitch. The depth information would help differentiate vehicles from the surrounding environment through their unique contours, offering a more precise drone location relative to its target car than a GPS provides.

2.1.3. Software System

The drone has many sophisticated electronics on board in order to control the motors and read the various instruments. Controlling all of these electronic components of the drone at once can be challenging. A drone is a flying machine possessing six degrees of freedom: forwards/backwards, left/right, up/down, roll, pitch, and yaw. However, this UAV only has four rotors; thus, the 6 degrees of freedom cannot be mapped to a single motor. By changing the speeds of individual rotors, or pairs of rotors, we can achieve roll, pitch, and yaw, and by changing the speeds of all four motors concurrently, we can change thrust. By combining pitch with thrust, we can create forward motion, while combining roll with thrust, we can create translational motion. This relies on careful balancing between the drone's thrust and rotation forces to prevent unwanted altitude changes or loss of control. This logic can be very difficult to generate; luckily, many open-source software packages exist to help streamline that process.

The main flight controller on the drone is a Pixhawk flight controller. Pixhawk controllers were developed with support for open-source flight control software. The team began the project using ArduPilot’s ArduCopter flight controller. This controller provided basic flight controls, mission planning, and simulations. However, due to this software’s limited feature set and functionality, the team elected to transition to QGroundControl. The drone was calibrated and test flown using QGroundControl. Unfortunately, QGroundControl takes some information about the drone and can only estimate the optimal control gains. In this case, the drone could perform a takeoff, but would become dangerously unstable at about six feet in the air. Further manual tuning would be required to ensure the UAV’s smooth flight.

QGroundControl, like all flight controllers, uses Proportional Integral Derivative (PID) control based-off the estimate of the drone’s position. The drone has a variety of sensors that help it determine its location, though some of the sensors have not yet been fully integrated. The drone has an internal inertial measurement unit (IMU), as well as a global positioning system (GPS), but these alone are not enough to accurately estimate the drone’s position. Both a real-time kinematics system (RTK), and a stereoscopic camera were planned to be added, but due to time restraints, these were not implemented.

2.2. Purpose of Work

More and more, drones are replacing dangerous jobs traditionally done by humans. In the winter, the effects of snow and ice in the environment can be a serious hazard. According to a study in the American Journal of Emergency Medicine, there are approximately 11500 hospitalizations a year alone related to clearing a driveway (Hilliard, 2020). Snow buildup on car windshields decreases visibility, and snow falling off cars can cause serious damage to other vehicles or passersby. Removing snow from cars is vital for road safety, but doing so can pose a challenge. Additionally, quickly and improperly cleaned cars can still have visibility and snow dropping problems.

Snow is not only dangerous on cars, but it can be damaging as well. Ice can freeze windshield wipers to the windshield. When a driver attempts to activate the windshield wipers, the wipers may fail, or the rubber blades can be torn, even if the driver brushed the snow off their windshield before driving. Replacing windshield wipers is expensive, and damaged wipers cannot clear rain, sleet, or snow as effectively from the windshield, reducing the diver's visibility.

Another common form of vehicle damage associated with snow is from salt. Roads are salted to keep them clear of ice, greatly reducing potential car accidents in the winter months. However, this road salt is often transported onto the car due to splashes of snow debris from the contact of the tires with the road. Without adequate cleaning, these salts can degrade and erode portions of the car, including the paint and some mechanical parts. Regular washing is needed to dissolve salt from all corners of the vehicle.

Finally, removing snow is a long and arduous process. It can frustrate people, who may rush to complete the task, endangering themselves. Additionally, frostbite poses a major issue. Depending on the ambient temperature and wind conditions, frostbite symptoms can develop in 10 – 30 minutes (National Oceanic and Atmospheric Administration, 2019). This process can be taxing on a healthy adult, but is much worse for the old or disabled. Reducing people's exposure to freezing conditions and keeping cars clean and free of flying snow is the ultimate goal of our project. To this end, a UAV is being developed in order to clear windshields using a de-icing spray, preparing a car for travel before the driver even has to step foot outside.

Chapter 3. System Overview

For a drone to efficiently clear snow off of cars, the hardware and software must interface ideally. Flying a drone requires a set of skills to control its flight, hover, and position. Rotors generate a force equal to or greater than the force of gravity by pushing down the air to lift the drone upwards. To descend, the exact opposite happens. Rotors thrust (speed) decreases; thus, the net power turns out to be downwards. For a drone to rotate, the set of rotors need each to spin in opposite directions to create a total of zero angular momentum, which means lower thrust in rotors. Now, to balance the drone's rotation and lift, the net upward force needs to exceed the gravitational force still while decreasing the rotors' thrust.

Maintaining and controlling the rotors' spin rate requires a controller to increment the voltage to each motor. To control each motor power level manually would be challenging and overwhelming; thus, having a computer control framework to handle all the mechanical and electrical systems operations will make flying a drone pretty easy. An electrical system with key sensors will further increase the ease. The accelerometer and gyroscope within a drone will increase the drone's flight stability, and the GPS will make it easier to track and maneuver the drone. Hence, flying a drone is a balancing act of forces and moments sensed and computed by a controller. This year's systems work together to make the drone navigate to clear the snow and ice off the vehicles' windshields. The mechanical system ensures a sufficient spray system, the electrical system improves on the drone's latitude and longitude, and the software system integrates a stereoscopic camera and an RTK system.

3.1. Mechanical Overview

When designing the drone, multiple mechanical systems were considered for the drone to perform tasks effectively. All of which are important and need to run smoothly together, or else the design would fail. For instance, the drone must be capable of lifting a heavy payload. The payload consists of the deicing liquid and all the peripherals required to fly, navigate, and spray the deicing fluid. Figure 1: Drone's Mechanical Structure shows that the basic frame design is composed of a top and bottom center plate and a third plate below the base plate. These plates hold the rotor arms, batteries, and all the drone’s electronics. Drones with a heavy payload adopt a landing gear mounted directly to the body to achieve higher ground clearance. Since we wanted the drone to take off and land while carrying the spraying gear safely, the landing gear was fixed to the center frame. To further ensure a safe flight and landing, the drone’s components were inspected thoroughly prior to every flight, and the drone was set to hover slightly above the ground at the beginning of every flight before takeoff.

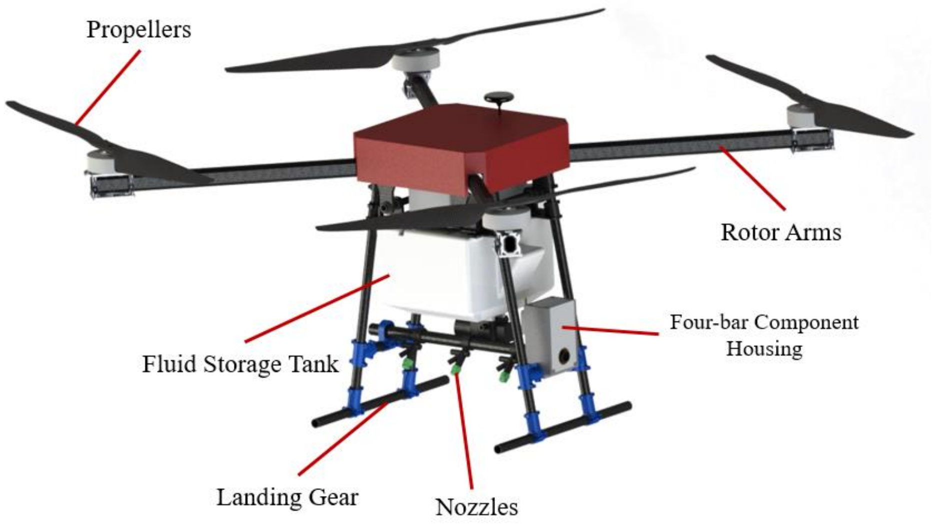


Figure 1: Drone's Mechanical Structure

The two major systems that needed improvement in this project's iteration were the safety and the spray-bar actuation mechanisms. To improve the drone's safety and protect the propellers, our team designed and implemented propeller guards. To improve on the spray-bar actuation mechanism, the four-bar linkage was redesigned for the nozzles to achieve a range of motion close to 60°. The actuation mechanism utilized a new servo (shown in Figure 2: JX 6208 Digital Servo) to rotate the four-bar linkage that rotates a collar-link epoxied to the spray-bar. The servo was easily controlled using PWM signals and eliminated significant weight on the drone. To know more about the servo specifications, please refer to Appendix A: Digital Servo Specifications.

Figure 2: JX 6208 Digital Servo

3.2. Electrical Overview

The drone is equipped with various electronic components to aid in its autonomous navigation when clearing snow and ice from its target vehicle. The four HOBBYWING ESCs direct power to the drone's propulsion system, flight control system, and spraying mechanism. The Pixhawk 4 flight controller, along with the GPS, allows the drone to hover in-place and fly semi-autonomously. The IR camera serves to detect differences between the temperature of the vehicle, snow, and deicing fluid. Due to the difficulty of an IR camera to differentiate from the temperature of a cold car and the accumulated snow, an RTK system was purchased instead.

3.3. Software Overview

The drone uses a Pixhawk flight controller, which comes with its own flight software. This software is connected to a ground station, which does the heavy calculations. We currently use QGroundControl, an open-source ground station package. QGroundControl has several basic options that can be selected to quickly generate flight code for the drone. However, this does not give us a stable flight due to the drone's unique structure and function. Luckily, QGroundControl allows the user to further tune the individual gains for the drone's flight control system.

Tuning the gains for a control system is challenging and generally involves many iterations. To avoid repeated crashes and reduce the risk of damaging the drone, which is made of several expensive and custom-made parts, we decided to use simulations to get initial values for the flight control system. While simulations will never perfectly accurately model the drone's response, it can help us narrow in on appropriate gains, making our test flights more stable and less likely to end in a crash. For this, we elected to use MATLAB's Simulink software, with the support for the parrot minidrones package. MATLAB is a powerful staple engineering tool in the industry.

Additionally, the package we are using is well documented, giving us an easier start. By using this base to model the environment, we can linearize the control system and use MATLAB's inherent functionality to automatically run repeated simulations to find ideal control values. Once those values are computed, we can use them as a baseline for manual tuning in QGroundControl (see Appendix B: MATLAB Simulation Models).

In addition to modifying the control scheme, we also aimed to practice drone flight in a simulator for similar reasons as described above. After consulting with WPI's UAV club, we purchased copies of the Drone Racing League Simulator. The Simulator is designed for recreational use but has an expansive library and in-depth mechanics and tutorials for flight practice. A controller can be linked with this program, and the user can then choose a drone. In this case, the ORCA drone was recommended due to its similarities with the actual project drone. Then this drone can be flown on a variety of maps with different obstacles and conditions. This should help us be more comfortable when flying the project drone.

Chapter 4. Methodology

4.1. Mechanical Design

This iteration of the project focused on improving the drone's spray-bar actuation and safety mechanisms. The following two sections discuss the work our team has done to design, analyze, and test the mechanical design of the drone.

4.1.1. Spray-bar

The spray-bar underwent several design iterations and tests, improving on the previous actuation mechanism. The previous project team used a Pololu 35T motor (shown in Figure 3: Pololu 35T Motor) controlled using PWM signals and tested using an Arduino UNO to supply the required torque to rotate the spray-bar. Their initial goal was to use a small 25g servo to rotate the four-bar linkage, but it did not supply enough torque to actuate the spray-bar.

Figure 3: Pololu 35T Motor

Unfortunately, the spray-bar was never tested. When tested by our team, the Pololu motor actuated the four-bar linkage; however, due to the motor's 360° motion, the linkage follower toggled with the carbon fiber tube that holds the spray nozzles. This made it impossible to rotate the linkage without manually fixing the linkage and getting it back to its initial starting position. Additionally, the Pololu motor was not lightweight, and it required more voltage than the Pixhawk 4 flight controller could supply. The only way to power the spray-bar through the Pololu motor was to hook it to a power supplier in one of the RBE labs. Therefore, our team decided to replace the motor with a new servo motor and modified the four-bar linkage dimensions to allow the carbon fiber tube a full 60° range of motion, which, as referred to us by the previous team, is an acceptable range of motion since the spray-bar will never need to move outside of a 90° range to spray a vehicle element. We edited the coupler and crank lengths to allow for a 60 degrees range of motion to achieve this. Figure 4: SolidWorks sketch of the linkage mechanism rotating through 60° shows the final four-bar linkage design dimensions shown in Figure 5: CAD model of the final spray-bar mechanism iteration.

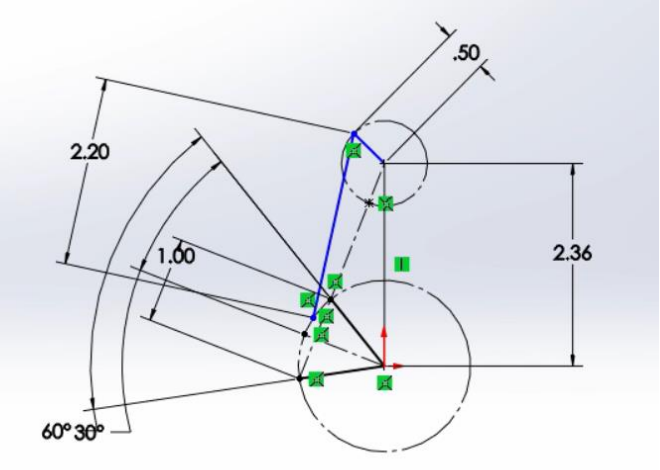


Figure 4: SolidWorks sketch of the linkage mechanism rotating through 60°

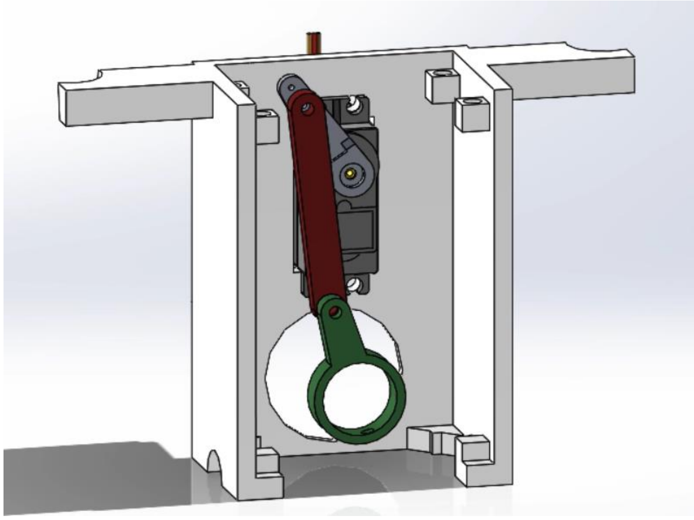


Figure 5: CAD model of the final spray-bar mechanism iteration

The drone's landing gear allowed for a fair amount of flexibility in the frame. This helped cushion the landings, and prevent the carbon fiber tubes or 3D printed joints from snapping, but it also introduced additional torques on the spray-bar, making it more difficult for the servo to turn. These difficulties are part of what resulted in the use of the pololu motor in the previous adaptation. To reduce pinching from the landing gear, the bronze bushings were replaced with customized bearings that allowed for multiple axes of rotation (see Figure 6: Customized spray-bar bearing design). These bearings were printed on the MarkForged 3D printer for its finer level of accuracy.

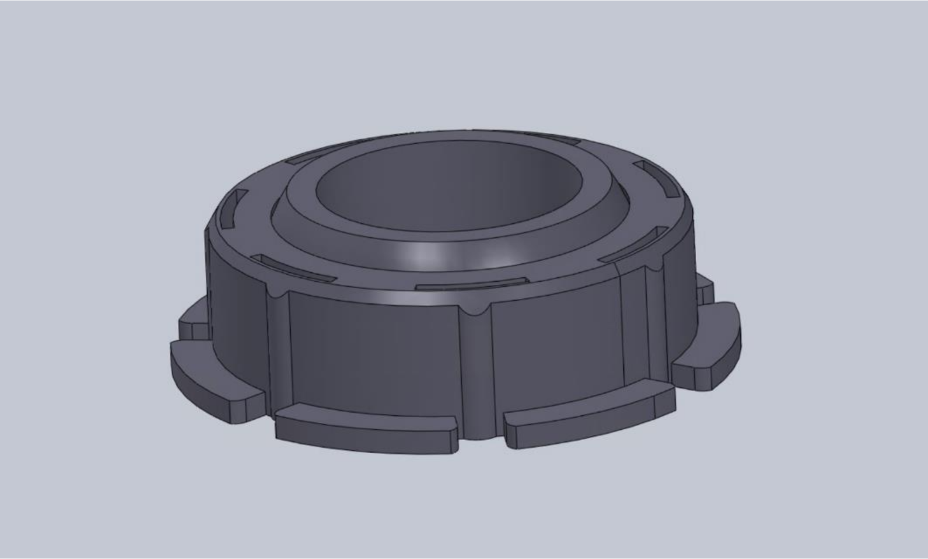


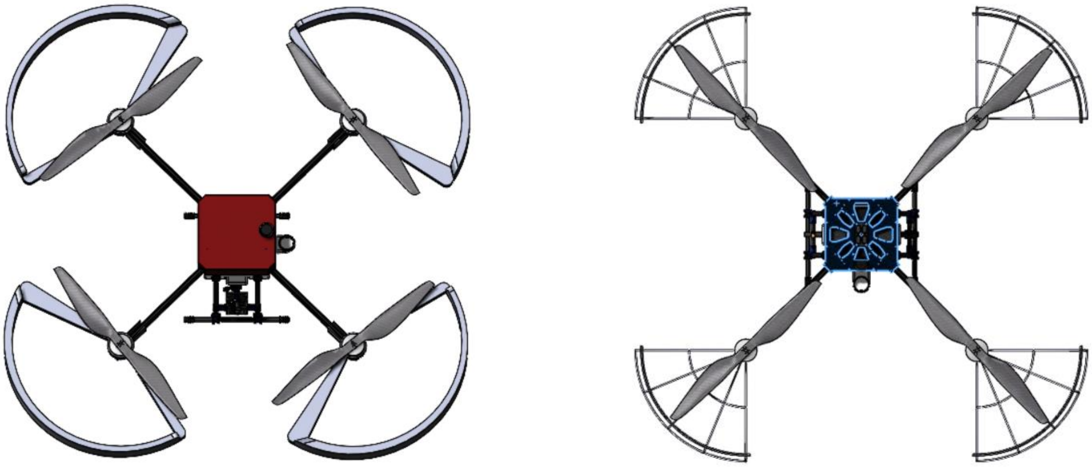
Figure 6: Customized spray-bar bearing design

Additionally, the four-bar and servo housing was attached to a round strut of the landing gear by friction. As the motor or servo rotated, it would exert a torque around this support strut, and rotate the housing, further increasing the binding in the four-bar. A pair of flanges were added to the housing to keep the box in the proper orientation. For a clearer visual of all the components that make up our spray-bar mechanism, see Appendix C: 3D Printed Spray-bar Mechanism.

4.1.2. Propeller Guards

Propeller guards are used to protect drone propellers from damage in the face of minor surface collisions. They also help protect the drone's immediate surroundings from any harm or injury that could be incurred in such scenarios. Typically, they are lightweight, made of plastic, and fixed to the rotor arms. This is to allow for sufficient encasement of the propellers without compromising any weight requirements of the drone.

Using SolidWorks, our team came up with two unique propeller guard models, a cross-comparison of which can be seen in Figure 7: Initial vs. Modified Propeller Guards Design Comparison below.



I. Initial design *II. Modified design*

Figure 7: Initial vs. Modified Propeller Guards Design Comparison

The first design (shown in closer detail in Figure 8: Side view of the initial propeller guard design) provided 270° protection for the drone propellers only in the case of normal collision to flat surfaces, such as walls or windshields. Since that design did not provide protection against curved surfaces, we attempted to design a new set of propeller guards that could do so.

Figure 8: Side view of the initial propeller guard design

The goal was to design guards with the ability to protect the propellers — as well as the colliding object(s) in question — in the event of hitting a wall, a car windshield, and/or pedestrians on the street during indoor and outdoor testing. As Figure 9: Side view of the modified propeller guard design clearly shows, design number two, with its curved exterior, comfortably accounts for these specifications. Although the degree of lateral propeller coverage was slightly reduced for weight and sizing reasons, this was a compromise we were more than willing to make for the improved shock absorption that the mesh structure of this new design would provide.

Figure 9: Side view of the modified propeller guard design

The final propeller guard design shown above was attached to the same base structure as the propellers and, at its point of most protrusion, encased each 14-inch blade with about 2.5 inches of leeway. This was an important consideration to make in order to account for any temporary deformations that might follow a collision.

4.2. Electrical Design

4.2.1. Existing Drone Components

The flight controller that the drone is equipped with is a Holybro pixhawk GPS module that uses a Ublox Neo-m8n and is accurate to about 2.5m. The flight controller has various gyroscopes and accelerometers, which allow the drone to determine its orientation during flight. The drone has four hobbywing xrotor pro 50A ESCs and a holybro pixhawk 4 flight controller. A 22.2V 25C 6S 22000mAh battery powers the entire drone, including the propulsion and flight control system. This battery also powers the spray-bar mechanism, which uses a servo motor. Manual flight of the drone is controlled by a Taranis x9d Plus, paired with the L9R receiver equipped on the drone.

4.2.2. Real Time Kinematic System

While the previous MQP team used an infrared camera along with QGroundControl software to pilot the drone, this year's project team will be utilizing an RTK system to allow the drone to navigate its environment. The team purchased a C94-M8P-2 from u-blox, which consists of two boards, two antennas, two USB wires, and two antennas with two metal discs (see Figure 10: RTK Components below).

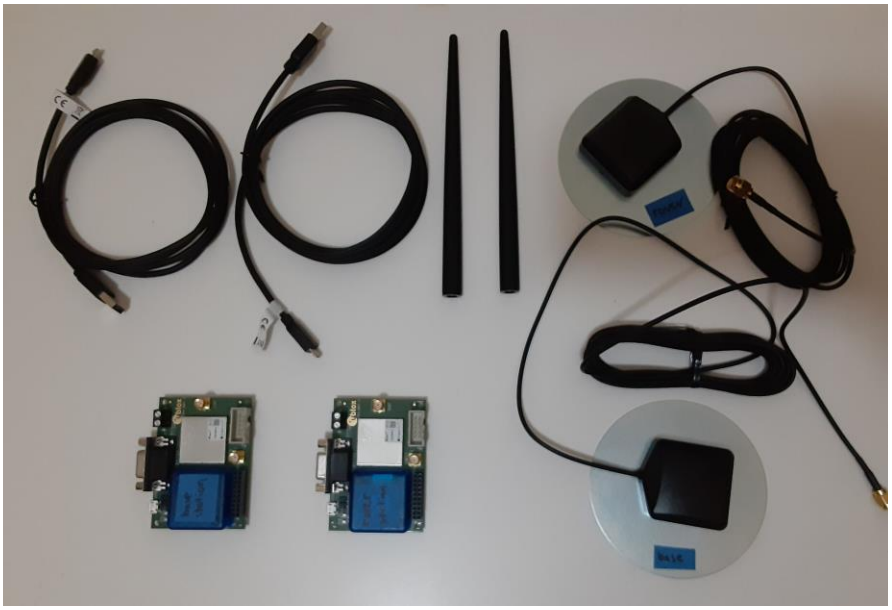


Figure 10: RTK Components

Configuration of the RTK system was used while utilizing u-blox software on Windows. Both boards can be plugged into a laptop and configured to receive satellite data to determine its relative position. Either of the two boards can be configured as the rover or base station. While the rover station is designed to be mounted onto the drone, the base station must remain stationary and parallel to the surface of the ground. The configuration process must be completed outdoors underneath a clear sky, as shown in Figure 11: RTK Outdoor Setup. This will prevent any signal interference from interrupting the pairing of both the rover and base station.

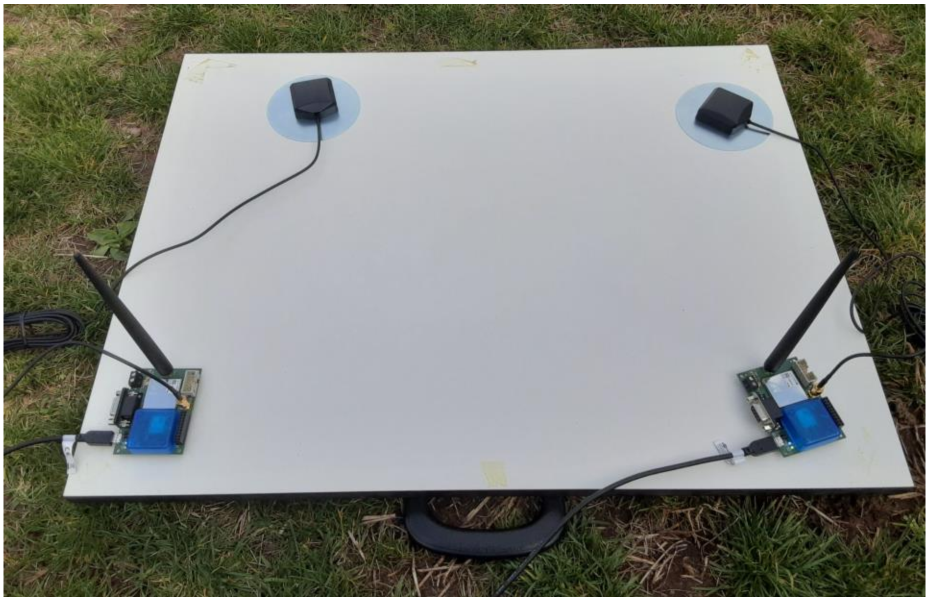
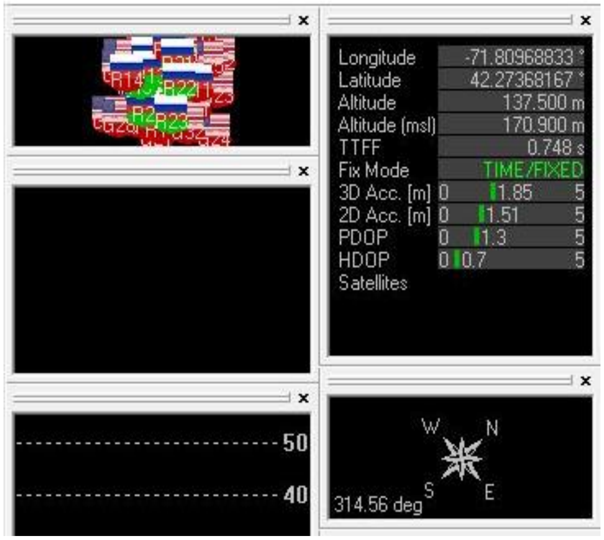
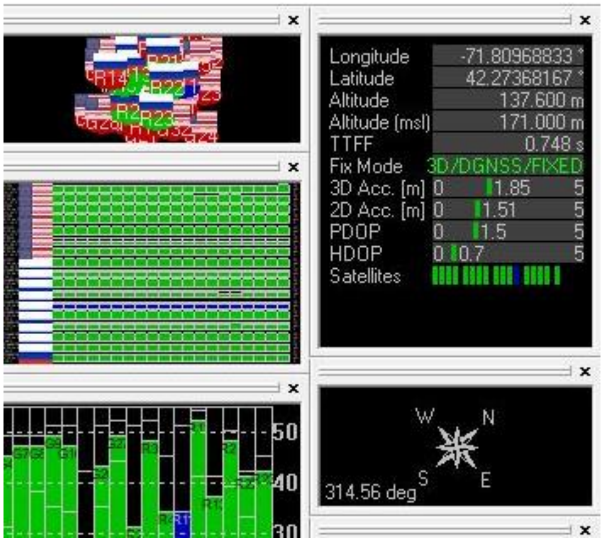


Figure 11: RTK Outdoor Setup

The u-blox software was used to configure one of the C94-M8P boards as the base station, and one as the rover station. To begin configuration of the base station, a survey-in was conducted for 300s, which established its location. Then the base station's radio link was configured while the baud rate was set to 19200, and our RTCM messages were selected to view. A similar process excluding the survey-in was repeated for the rover station. After these configurations were made, both rover and base stations entered "FLOAT" mode, and in the correct outdoor conditions (clear sky, no obstructing buildings, no signal interference, etc.) the RTK system will have corrected any inconsistencies in the satellite data and will enter "FIXED" mode (see Figure 12: Base Station Fixed Mode and Figure 13: Rover Station Fixed Mode). This mode is where the rover station is able to now be moved around while its changes in longitude, latitude, and altitude can be measured in the u-blox software.



Longitude -71.80968833 °
Latitude 42.27368167 °
Altitude 137.500 m
Altitude (msl) 170.900 m
TTFF 0.748 s
Fix Mode TIME/FIXED
3D Acc. [m] 0 1.85 5
2D Acc. [m] 0 1.51 5
PDOP 0 1.3 5
HDOP 0 0.7 5
Satellites



Longitude -71.80968833 °
Latitude 42.27368167 °
Altitude 137.600 m
Altitude (msl) 171.000 m
TTFF 0.748 s
Fix Mode 3D/DGNSS/FIXED
3D Acc. [m] 0 1.85 5
2D Acc. [m] 0 1.51 5
PDOP 0 1.5 5
HDOP 0 0.7 5
Satellites

Figure 12: Base Station Fixed Mode

Figure 13: Rover Station Fixed Mode

4.2.3. Battery Eliminator Circuit

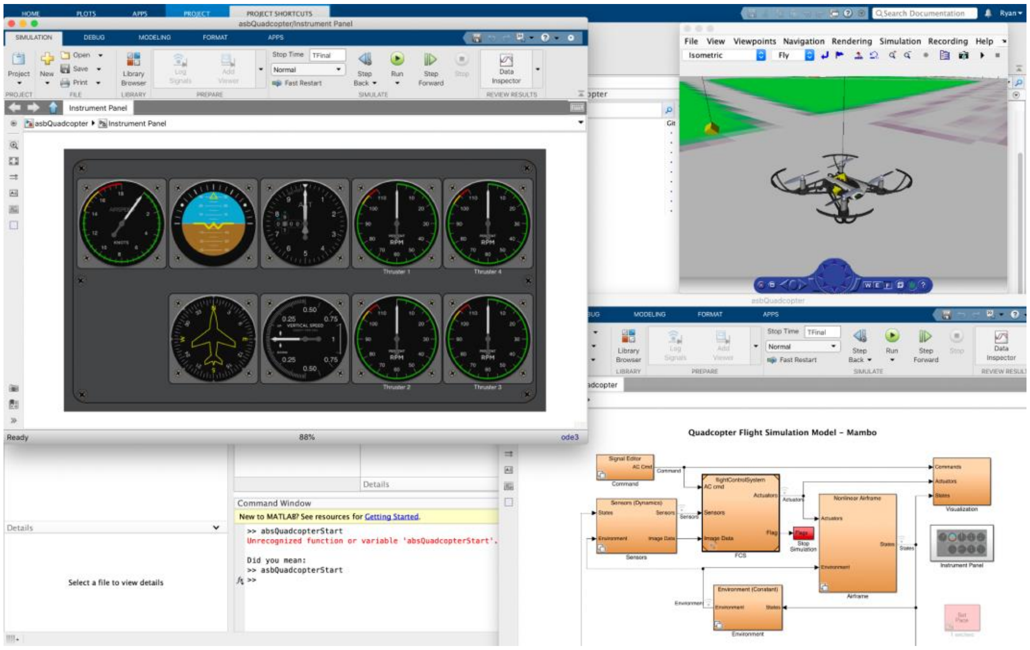
When preparing to transition the spray-bar to a servo, it was found that the pixhawk setup does not natively supply power to its servo pins. The drone operates off a 22.2V battery, but the servo only requires 5V. In order to power the servo, we used a battery eliminator circuit (BEC) to transfer appropriate power to the servo pins. An alternative would have been to attach an additional battery, but a BEC is substantially lighter than most batteries, and keeps the circuit simple, as we only have to worry about charging one battery.

4.3. Software Design

We started out using the Pixhawk flight controller and QGroundControl as the base station. Since the project had been on hiatus since the end of the previous academic year, we downloaded the drone's flight telemetry logs. These extracted simulations gave us an idea of how much progress the previous team had made in the flight process. The data matched the reports that the drone was unacceptably unstable in the air. After corresponding with last year’s team, we found that they used gains generated automatically by QGroundControl given the basic airframe of the drone, little to no tuning had been performed due to time and availability constraints of the pandemic.

We began researching QGroundControl and found that all the gains in the flight controller could be manually adjusted. The QGroundControl manual also included a guide for PID tuning and ESC calibration. However, these guides assumed that the drone was already flyable, but unstable. We decided that a better ballpark estimate of the drone’s gains was needed before this fine-tuning stage. Due to the project’s time and expense to design and build the drone body, we did not want to risk damaging the drone in repeated test flights. Thus, we turned to simulation.

MATLAB’s package for parrot mini drones, combined with extensive documentation and modding community, gave us a good start for simulating quadcopter reactions in a controlled environment. We began by downloading the project, then following the documentation and guides to learn how the control system operated, and how to automatically derive gains. The simulation was largely broken up into four major sections, the estimated input, the flight controller, the output, and the environment (see Figure 14: MATLAB Simulation). The input section would record the drone’s position and orientation, introduce noise, and inform the flight controller. This section made these estimates using a GPS, internal gyroscopes, and a camera. Since this iteration of the drone did not include a camera, we removed that section from the calculations. These estimates were then fed to the flight controller. This is where we made the majority of our calculations.



Command Window

New to MATLAB? See resources for Getting Started.

```
>> absQuadcopterStart
Unrecognized function or variable 'absQuadcopterStart'.

Did you mean:
>> asbQuadcopterStart
>>
```

Figure 14: MATLAB Simulation

QGroundControl provided in-depth diagrams of how they arranged their control loops, making it relatively easy to modify the different elements of the parrot mini drone control system to reflect the way QGroundControl builds its control system more accurately. We also modified the airframe data to reflect the SolidWorks model of the actual drone. With this in place, the flight control software should generate outputs that reflect the drone. These outputs are then passed through the environment to see how they affect the drone. The environment panel was largely untouched but could handle external factors that may affect the drone, like wind and air pressure. This data was fed back into the input panel, and the loop continues. Once this was completed, we linearized the system and had MATLAB generate gains for us to test experimentally.

While we were generating the simulation, we also had to ensure the connectivity between the base station, the drone, and the analog remote. QGroundControl required an update, while the

Pixhawk firmware also needed an update. Next, the remote control was paired with the drone, and the telemetry receiver was connected to the computer hosting the base station.

Once everything was connected and the simulations complete, we were able to begin experimental tests. The first round of tests found that the simulated gains were just as bad, if not worse than the ones automatically generated by QGroundControl (see Figure 15: Flight Recording). We went back to the simulation to attempt to try to find the issue, but in the meantime, we began to follow the QGroundControl tuning manual. To our surprise, we managed to make a notable improvement after a few rounds of tuning. QGroundControl plotted the recorded input for the roll, pitch, yaw, position, and altitude with the drones estimates of those values (see Figure 16: Roll Tuning Graph). These graphs were another useful metric in judging the tuning of each gain. However, we were still a long way off. We attributed part of the problem to a lack of user experience.

Figure 15: Flight Recording

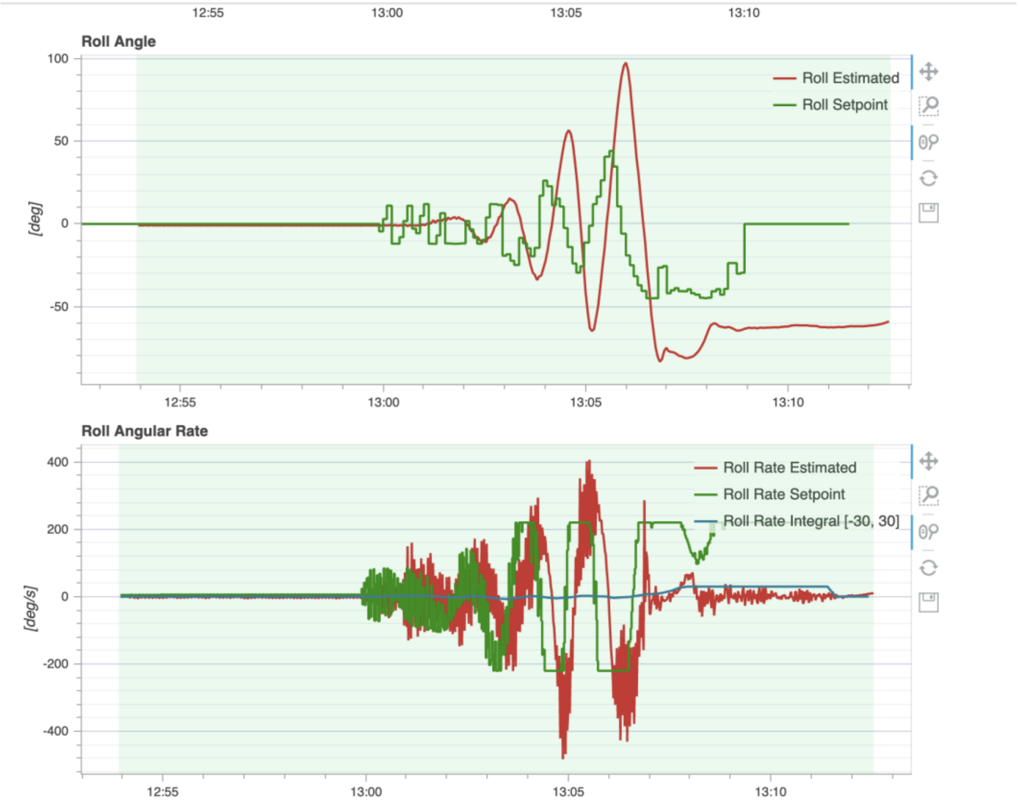
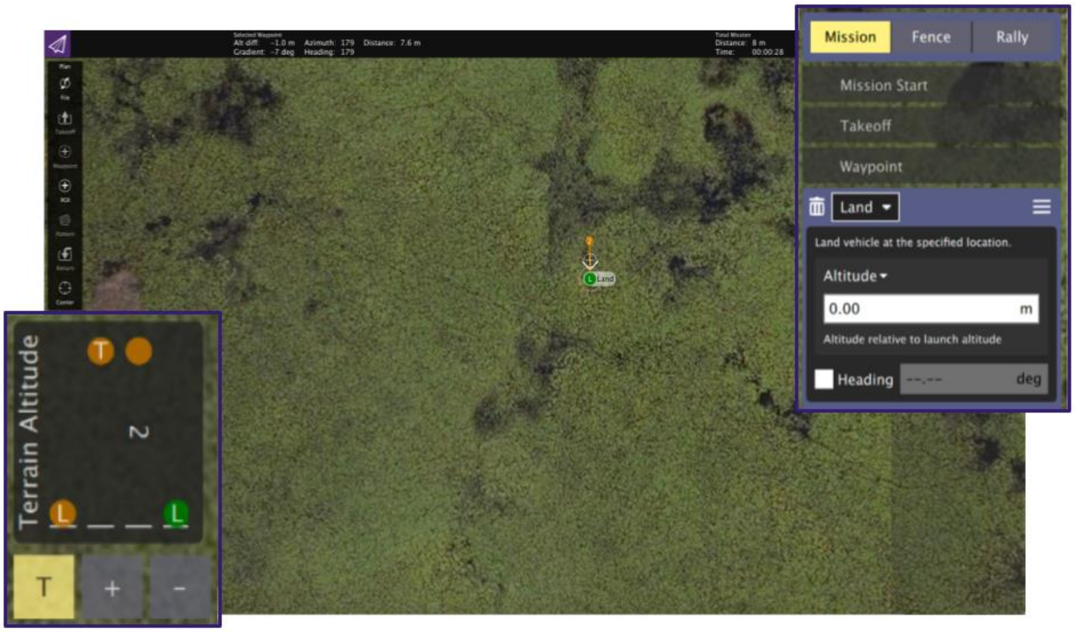


Figure 16: Roll Tuning Graph

Due to trouble controlling the drone, many of the flights were short. We reached out to WPI's UAV club and asked for advice on how to practice flying drones. They directed us to the Drone Racing League Simulator (DRLS). The DRLS is an online game that allows the user to connect a remote to a virtual drone, and practice flying it around virtual environments. Some environments are clear, while others have obstacles to navigate around. Further discussion with the UAV club helped us pick out a drone that most closely represents our situation. After being given a description of the drone, they recommended we use the "ORCA" for our simulations (see Figure 17: DRL Simulator Practice Flight). Unfortunately, the game does not provide access to the gains or control systems on these drones, but it has been a valuable resource for improving our flight skills.

Figure 17: DRL Simulator Practice Flight

Once we felt confident with the improved gains, we moved on to piloting the drone using QGroundControl. The QGroundControl ground station supports the generation of flight paths using a waypoint system. Under the planning tab, the user can generate a takeoff, landing, and a list of waypoints in between. The drone's position, height and orientation can be specified at each waypoint, and QGroundControl generates the flight code (see Figure 18: QGroundControl Flight Planner).



Selected Waypoint
Alt diff: -1.0 m Azimuth: 179 Distance: 7.6 m
Gradient: -7 deg Heading: 179

Total Mission
Distance: 8 m
Time: 00:00:28

Land vehicle at the specified location.

Altitude: 0.00 m

Altitude relative to launch altitude

☐ Heading --.-- deg

Figure 18: QGroundControl Flight Planner

Unfortunately, during our first test of the automated flight system, the drone impacted a wall and damaged two of its propellers. Recent snowstorms had hampered our ability to test outside, and the decision was made to do the test in a large open space. However, the building was not equipped with a GPS repeater, so the GPS on the drone was unable to get a fix inside the building, resulting in unexpected behavior.

Chapter 5. Discussion/Conclusion

5.1. Spray-bar

To test the spray-bar system, the drone was mounted on an elevated platform. A glass panel was placed at an angle and covered in snow. Warm water was used as the deicing fluid. The pump and the bar's rotation were controlled with the Taranis remote controller. For this test, the spray-bar was restricted to 45 degrees of rotation. During the test, the bar exhibited the full range of motion expected. The pump also worked as expected. The previous team had elected to use nozzles that produced a fine mist over a large area. The mist did an excellent job of distributing deicing fluid over the target area, however the lack of force led to concerns about the effects of the downwash on the spray from the drone's propellers. The test was repeated at 4 feet, 3 feet, 1 foot, and 3 inches. Even at the closest distance, the spray had a very limited effect on removing the snow. This snow was collected in early spring, and was already saturated with water, which may have reduced the effectiveness of the tests.

To ensure the new servo provides sufficient torque to power on the spray-bar mechanism, we ran multiple SolidWorks motion studies, which all turned out to be successful. Figure 19: SolidWorks Spray-bar Torque Plot shows the torque plot of the final spray-bar iteration with the new digital servo.

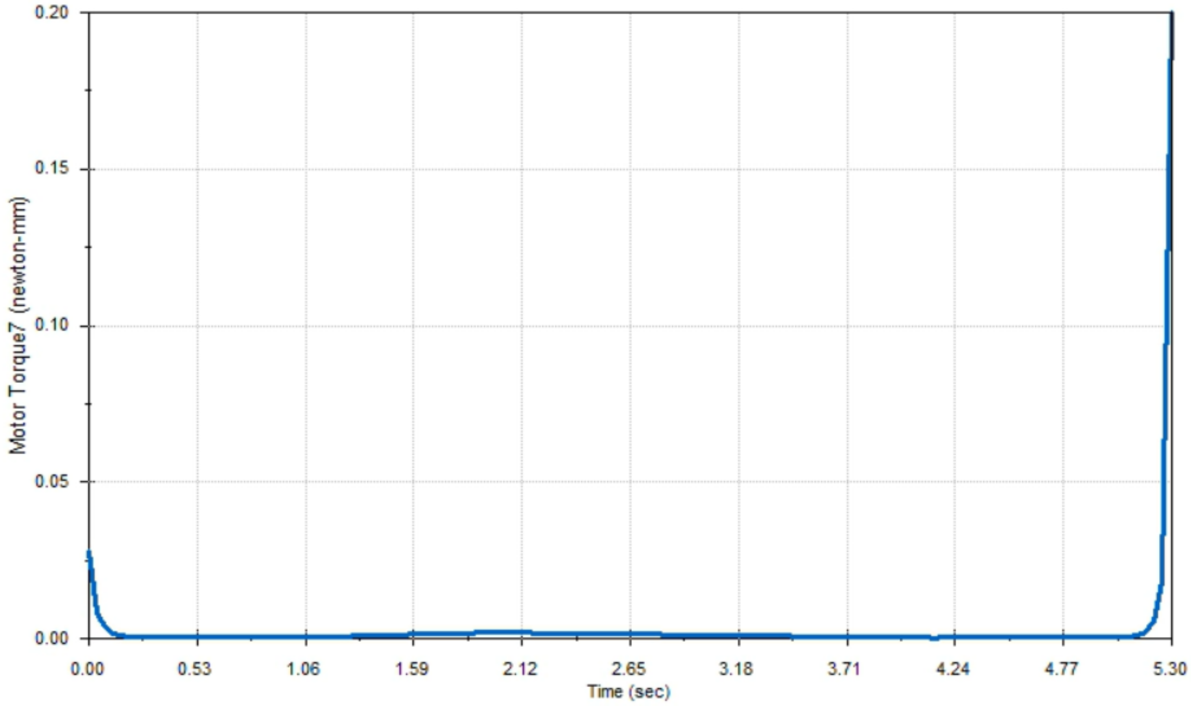


Figure 19: SolidWorks Spray-bar Torque Plot

5.2. Propeller Guards

Although our team was unable to physically develop and test our designed propeller cages in a real-time environment, we opted to utilize the SolidWorks' built-in simulation tool to our advantage. By performing a static finite element analysis (FEA) on the part, we effectively mimicked the physical conditions of a real-world replica of the design.

First, we defined the material properties of the cage as constituting steel alloy, a tough yet light and flexible material. Then we set fixtures at the four screw points at which they would be attached to the rotor arm. As Figure 20: SolidWorks stress analysis of the propeller cage shows, applying a net force of 67 N — a value which far surpasses anything we could expect in reality — to the cage led to minimal deformation.

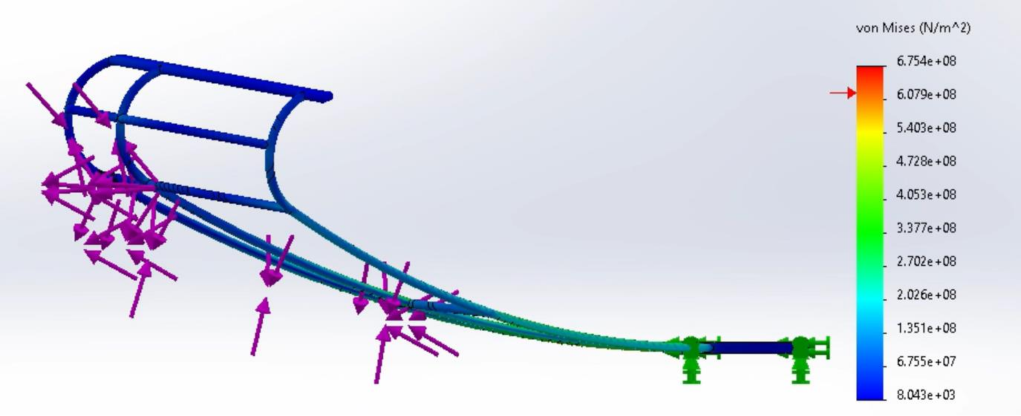


Figure 20: SolidWorks stress analysis of the propeller cage

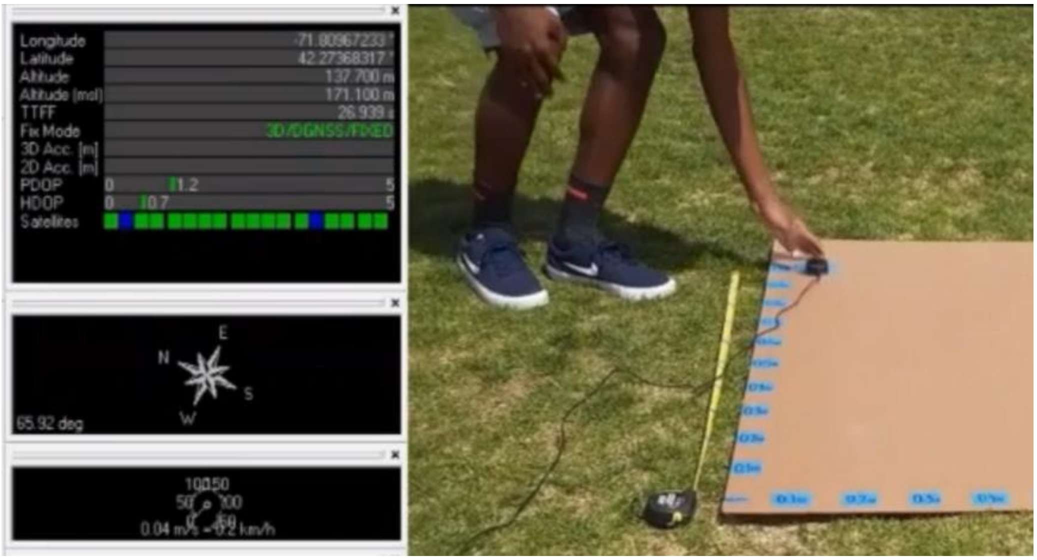
The results of this SolidWorks Simulation confirmed to the project team that the proposed design for the propeller guards, along with the design of the fixturing collars and the spacing between them, was suitable for this application. Though our time constraints didn't allow us to implement this design, we recommend that future teams use this model to develop a tough, yet flexible, cage using simple wire bending techniques, and begin testing immediately.

5.3. Landing Gear

Our plan was to redesign our landing gear to give the drone a softer and safer landing after flight. However, due to time constraints we were unable to reach this point in this project iteration. Additionally, upon attempting to complete a stress study of our existing landing gear in SolidWorks, the study unexpectedly failed multiple times.

5.4. RTK: Rover Station Positioning Test

To demonstrate and test the effectiveness of the RTK system's ability to determine its location, we performed multiple longitude, latitude, and altitude tests, snapshots of which can be seen in Figure 21, Figure 22, and Figure 23. Using a large sheet of cardboard, we marked the distance in 0.1 meters up to 1 meter. We then had the rover station of the system travel along the y-axis of the board to measure its change in longitude, and along the x-axis to measure its change in latitude. We then held a tape measure 1 meter above the ground and had the rover station travel up and down while the change in altitude was measured.



Longitude -71.80967233 °
Latitude 42.27368317 °
Altitude 137.700 m
Altitude (msl) 171.100 m
TTFF 26.939 s
Fix Mode 3D/DGNSS/FIXED
3D Acc. [m]
2D Acc. [m]
PDOP 0 1.2 5
HDOP 0 0.7 5
Satellites

65.92 deg

0.04 m/s = 0.2 km/h

Figure 21: Longitude Test

Longitude -71.80968233 °
Latitude 42.27367967 °
Altitude 137.700 m
Altitude (msl) 171.100 m
TTFF 26.939 s
Fix Mode 3D/DGNSS/FIXED
3D Acc. [m]
2D Acc. [m]
PDOP 0 1.2 5
HDOP 0 0.7 5
Satellites

145.04 deg

0.31 m/s = 1.1 km/h

137.700 m x100

18:01:45 UTC

Figure 22: Latitude Test


Longitude -71.80968417 °
Latitude 42.27367967 °
Altitude 137.800 m
Altitude (msl) 171.200 m
TTFF 26.939 s
Fix Mode 3D/DGNSS/FIXED
3D Acc. [m]
2D Acc. [m]
PDOP 0 1.2 5
HDOP 0 0.7 5
Satellites

Figure 23: Altitude Test

Some of the issues we encountered during testing of the RTK system included various sources of interference that ended up removing the system out of "FIXED" mode and back into "FLOAT" mode. For example, during testing, when trying to position the cardboard that served as our x-y axis upright, the obstruction tended to interfere with the RTK signals and put it into "FLOAT" mode. We ended up resorting to using a tape measure to measure the altitude while the board laid flat on the ground, eliminating any interference to the RTK signal. Often, passersby would walk by the RTK system, and at 6 feet distance away or closer the RTK signal would be interrupted. We decided to correct the issue by moving the rover station's antenna further away from our base and rover station area, which would allow us more room to walk around and complete our testing without causing any signal interference.

For a real-time demonstration of this testing phase, and a clearer depiction of our accurate localization protocol, please refer to our RTK demo video contained in our final eCDR submission.

5.5. Flight Testing

As noted previously, the MATLAB simulation was not able to generate appropriate gains for the drone, making the simulation too complex to model with our restricted time frame and resources. We then resorted to rely primarily on flight testing and manual tuning. When a manual flight was tested, the drone was unable to lift off without full throttle, at which point it would flip over its port side. Subsequent tests using manual gains fared much better. During an indoor test, the drone made unwanted contact with a wall, but managed to maintain its orientation and relative stability throughout the encounter. These mid-flight complications resulted in expensive damage to our propellers, leading to us using our budget to purchase a new set of propellers that took nearly a month to arrive. In response, we researched different methods that improve the safety of the drone from collisions during flight, for example a wire cage prop guard.

Chapter 6. Recommendations for Future Work

We have many suggestions and recommendations for future work on the drone. Some general suggestions we had for the next project team includes building and attaching the prop guards to the drone. We recommend testing deicing fluid on snow to determine the effectiveness of the solution. We also recommend researching ways to improve the drone's landing gear design to allow for a softer and safer landing after flights. Additionally, we suggest researching ways to improve the gains of the drone's flight controller.

One suggestion we have for the next project team is to continue the integration of the RTK system with the Pixhawk 4 Flight Controller and QGroundControl. This should allow for more precise autonomous flight and reduce the risk of unwanted collisions. The improved accuracy can allow the drone to be used in greater proximity to the car, which has the potential to improve the effectiveness of the spray-bar as well. Further, we suggest the next team use QGroundControl software, as it supports RTK integration, works well with the Pixhawk 4 Flight Controller, and has successfully been used and documented in this report.

During our spray-bar testing, we were unable to clear snow off of the glass surface we used. This was mainly due to the mist pattern the nozzles produced. We suggest researching or designing different nozzles that can project the spray onto a surface with greater volume and/or force. We also recommend testing these sprays on different kinds of snow and ice. Due to how late in the year we ran our test, we were only able to demonstrate the effect on saturated snow. However, the effect could be drastically different on lighter snows. The team may also want to experiment with the effectiveness of other de-icing fluids, if changes to the spray pattern are ineffective. Finally, the spray-bar housing has had flanges added to prevent unwanted rotation, but these flanges only protect against backwards rotation, encountered primarily when the servo rotates counterclockwise. While the effect is less prevalent when the servo rotates clockwise, it is a good idea to stop this rotation as well to prevent torquing or damaging the four-bar. We would suggest that the team create a simple piece of material that could be screwed or unscrewed on to the current flanges to maintain ease of assembly.

One of the factors that we didn’t account for or focus on during this project iteration included how knock-back from the spray-bar spraying the deicing fluid would alter the stability of the drone mid-hover. We deemed it a low priority due to the lack of force produced by the misting effect of the current design, but changes to the spray configuration could introduce greater forces. It is important to take these forces into account, and may require the team to manually add a counter rotation in the drone’s flight to maintain stability.

Another factor to consider is how the spray affects passersby. A major concern with the mist configuration is that the mist is easily affected by wind, and can be carried long distances. The team should consider how dangerous the de-icing fluid is to individuals, and how to minimize losses to wind.

On the topic of safety, a set of prop guards were designed, but never made it to prototyping or testing. These pieces still require some degree of real-world testing. Further improvements to the effectiveness of the safety these guards provide should also be considered.

During flight testing, the drone was exposed to water and snow during an accident. However, after the drone and electronic parts were cleaned and dried, it was functional without any issues. To prevent any potential damage to electronic components in the future, we suggest designing and implementing a lightweight cover that will protect them from the elements.

Chapter 7. Appendices

Appendix A: Digital Servo Specifications

Table 1: Digital Servo Specifications

Dead band	2µs
Maximum Pulse Width	900-2100us
Maximum Angle	120°
Working frequency	900-2100us
Motor	high-quality core motor
Operating Voltage	DC4.8~6.0 V
Operating Speed (4.8V)	0.09 sec/60 degrees
Operating Speed (6V)	0.07 sec/60 degrees
Stall Torque (4.8V)	6.8 kg.cm
Stall Torque (6V)	8.2 kg.cm
Dimensions	40.5 x 20.2 x 38mm / 1.59 x 0.80 x 49 in
Weight	55.6 g (1.96oz)
Connector Wire Length	JR 265 mm (10.43in)
Bearing	2BB

Appendix B: MATLAB Simulation Models

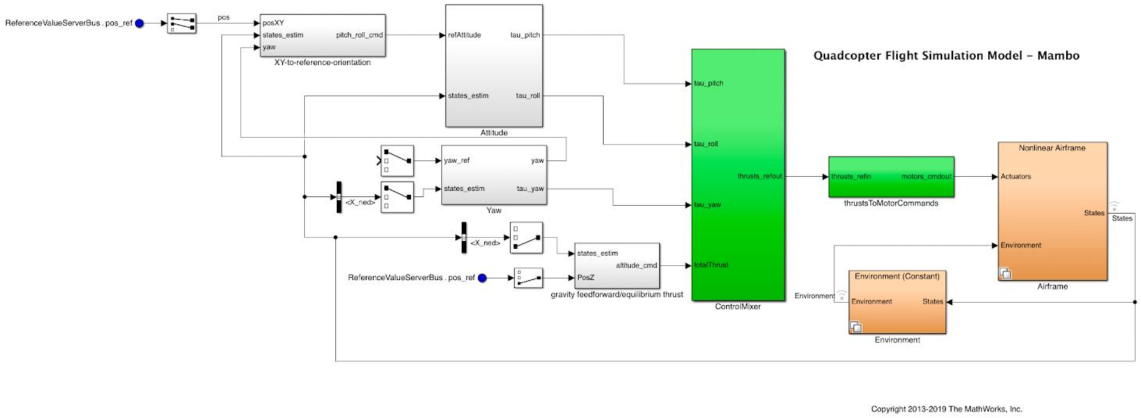


Figure 24: Overall Linearized Simulation Model

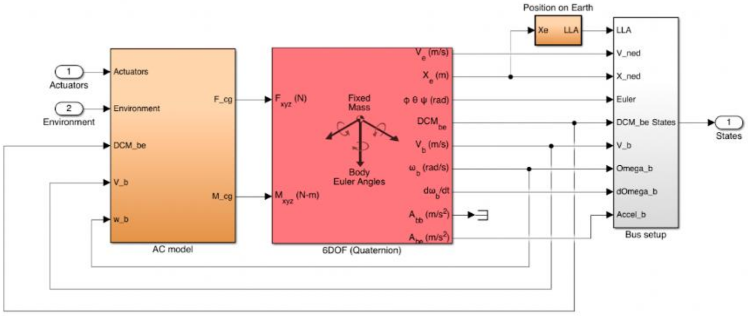


Figure 25: Airframe Module

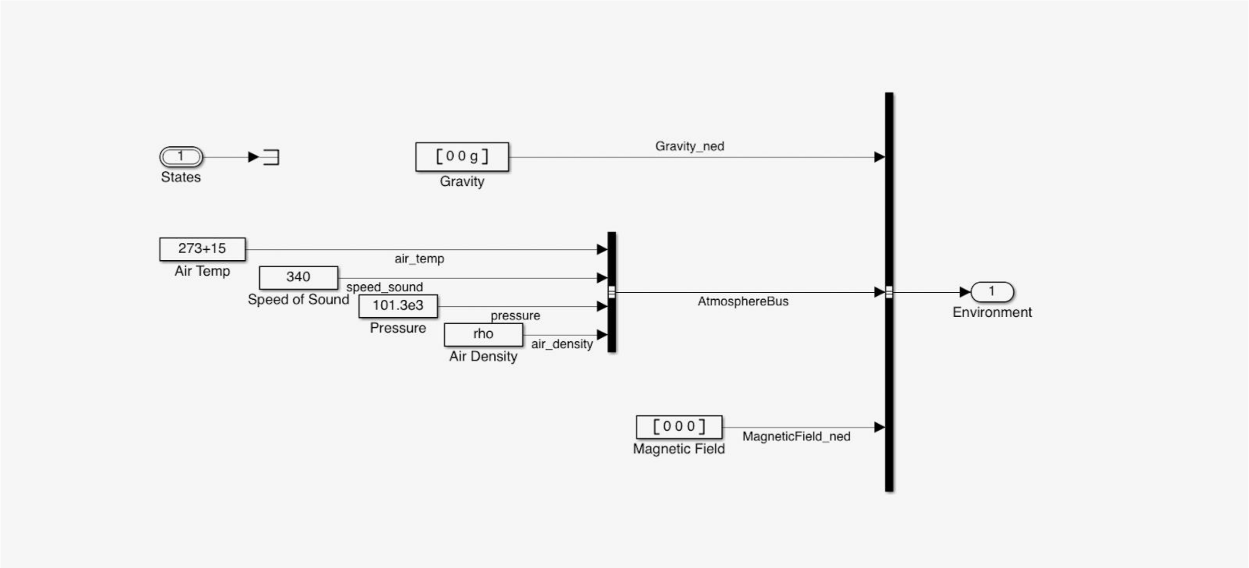


Figure 26: Environment Module

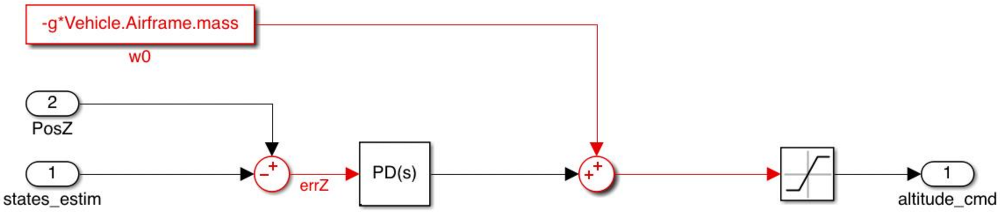


Figure 27: Altitude Linearized

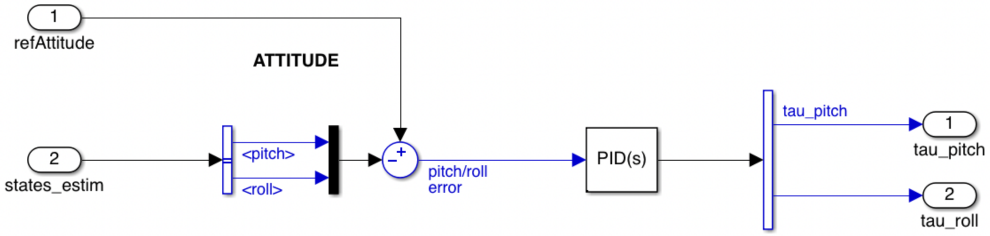


Figure 28: Pitch and Roll Linearized

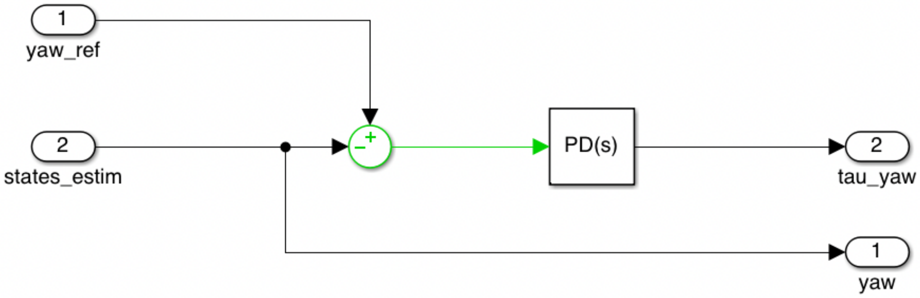


Figure 29: Yaw Linearized

Appendix C: 3D Printed Spray-bar Mechanism

Figure 30: 3D printed model of final spray-bar mechanism

Figure 31: 3D printed customized spray-bar bearings

Appendix D: Contributions

Table 2: Contribution Table

Section	Subsection	Contributor(s)
Abstract		All
Introduction		Jeremy + Chioma
Background	Existing Work	Razan
	Mechanical System	Razan
	Electrical System	Razan + Jeremy
	Software System	Ryan
	Purpose of Work	Ryan
System Overview	System Overview	Razan
	Mechanical Overview	Razan
	Electrical Overview	Jeremy
	Software Overview	Ryan
Methodology	Mechanical Design: Spray-bar	Razan + Ryan
	Mechanical Design: Propeller Guard	Chioma + Razan
	Electrical Design	Jeremy
	Software Design	Ryan
Discussion/Conclusions	Spray-bar	Ryan + Razan

	Prop Guards	Chioma + Jeremy
	Landing Gear	Jeremy
	RTK	Jeremy + Chioma
	Flight Testing	Ryan + Jeremy
Recommendations for Future Work		Ryan + Jeremy

Chapter 8. References

Hilliard, R. (2020). Ready to shovel? Tips to stay safe. *nbc25news*. Retrieved from https://nbc25news.com/news/local/ready-to-shovel-body-might-say-no

Joseph, B. P., Robert, D. D., Rodriguez, G., Burd, M. C. B., & St. George, N. W. (2020). *Car-snow clearing drone.* (). Retrieved from https://digitalcommons.wpi.edu/mqp-all/7506

Douglas, B. (2018, December 12). *Trimming and Linearization, Part 2: The Practical Side of Linearization*. YouTube. https://www.youtube.com/watch?v=K1oMF4684OQ&ab_channel=MATLAB.

Douglas, B. (2018, December 5). *Trimming and Linearization, Part 1: What Is Linearization?* YouTube. https://www.youtube.com/watch?v=5gEattuH3tI&ab_channel=MATLAB.

Douglas, B. (2018, November 13). *Drone Simulation and Control, Part 4: How to Build a Model for Simulation*. YouTube. https://www.youtube.com/watch?v=gEmGfo36INc&ab_channel=MATLAB.

Douglas, B. (2018, November 20). *Drone Simulation and Control, Part 5: Tuning the PID controller*. YouTube. https://www.youtube.com/watch?v=BqrRfoH-19s&ab_channel=MATLAB.

Douglas, B. (2018, November 5). *Drone Simulation and Control, Part 3: How to Build the Flight Code*. YouTube. https://www.youtube.com/watch?v=3Gtb5Eq1Lvk&ab_channel=MATLAB.

Douglas, B. (2018, October 16). *Drone Simulation and Control, Part 1: Setting Up the Control Problem*. YouTube. https://www.youtube.com/watch?v=hGcGPUqB67Q&ab_channel=MATLAB.

Douglas, B. (2018, October 23). *Drone Simulation and Control, Part 2: How Do You Get a Drone to Hover?* YouTube. https://www.youtube.com/watch?v=GK1t8YIvGM8&ab_channel=MATLAB.

Halligan, B., & Gardener, M. (2020). *Best Image Recognition Software*. G2.com. https://www.g2.com/categories/image-recognition.

Iforce2d. (2018, March 7). *Neo-M80 RTK setup and demo (courtesy u-blox).* Youtube. New Zealand. Retrieved from https://www.youtube.com/watch?v=n8PUyOtiGKo

MathWorks MATLAB Hardware Team. (2021, April 26). *MATLAB Support Package for Parrot Drones*. MATLAB Support Package for Parrot Drones - File Exchange - MATLAB Central. https://www.mathworks.com/matlabcentral/fileexchange/70363-matlab-support-package-for-parrot-drones.

Meier, L. (2020, December 16). *QGroundControl*. Mavlink/QGroundControl. https://github.com/mavlink/QGroundControl.

Meier, L. (2020, December 17). *RTK GPS*. RTK GPS | PX4 User Guide. https://docs.px4.io/master/en/gps_compass/rtk_gps.html.

Meier, L. (2021, March 10). *Multicopter PID Tuning Guide (Advanced/Detailed)*. PX4 User Guide. https://docs.px4.io/master/en/config_mc/pid_tuning_guide_multicopter.html.

Mike. (2019, May 22). *Taranis QX7 - Setting up switches*. YouTube. https://www.youtube.com/watch?v=R51UjP8t6BU&ab_channel=FPV_EvilMunkey.

UAV Coach. (2021, April 27). *Drone Laws in Massachusetts (2021)*. UAV Coach. https://uavcoach.com/drone-laws-massachusetts/.

u-blox. (2017, July 07). *C94-M8P: u-blox RTK Application Board Package.* u-blox AG.

Willee, H. (2020, December 2). *PX4 Quick Start*. GitHub. https://github.com/PX4/PX4-user_guide/blob/master/en/assembly/quick_start_pixhawk4.md.