Drone Project Documentation

Siddharth Gummadapu

May 2024

Contents

1 Purpose

The overall purpose of this project is to design and fabricate a custom quadcopter frame for an FPV Drone, basing all design decisions on engineering analysis tools and current market products. In addition to this, I will add consideration to optimal electronics for the drone, and how they would possibly affect flight. This drone is meant for general consumers, so the goal is for electronics to be covered. In addition, my goal is for this drone to be a long distance and high flight time drone, instead of a racing drone.

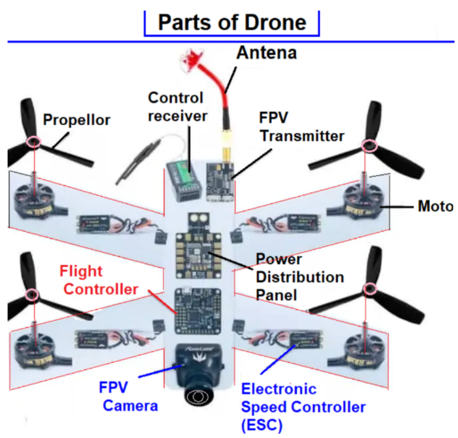


Figure 1: Drone Avionics

2 Avionics/Electronics Research

This section will cover the decision making for the electronics of the drone.

2.1 Motor

I decided to go with a brushless motor (F2203.5 2850KV) with the following specs:

- Dimensions: 26.8*26.45mm
- Lead: 24#AWG 130mm
- Shaft Diameter: 3mm
- Idle Current (10): 0.3A
- Max. Power (the 60s): 734W
- Weight (Incl. Cable): 19.7g
- Internal Resistance: 90 milliohms
- Configuration: 12N14P
- Rated Voltage (Lipo): 4-6S
- Peak Current (60s): 31.4A

2.2 Electronic Speed Controller

I have been given certain specifications for my motor. I will go for the 2850 kV specification since it can handle more torque, and I would like my drone to be able to hold more weight and carry objects. I also chose this specific motor due to its common usage in quadcopter drones as well as available 3D CAD online.
When considering, there a a few specific key considerations:

- Rated Voltage (Lipo) - 4-6S or 14.8V to 22.2 V (1S = 3.7 V)
- Peak Current - 31.4 A
- Max Power - 734 W

The ESC rating should be higher than all of these (i.e. it should be able to handle the max).

It should also have a BEC (battery eliminator circuit) - to power other electronics such as the flight controller, have programmable firmware, and good thermal management.

Hobbywing XRotor Micro 40A ESC

- Voltage: 3-6S LiPo
- Continuous Current: 40A
- Burst Current: 60A (10 seconds)
- Protocols: DShot150/300/600, Oneshot125/42, Multishot, PWM
- BEC: 5V/1.5A
- Features: BLHeli_S firmware, robust and reliable design
- Size: 40x33x5 mm
- Weight: 12 g

One thing to note is that one of the main advantages of this ESC is that the variety of protocols allows it to be compatible with modern firmware and more easily programmable.

2.3 Flight Controller

Holybro Kakute F7 AIO V1.5

- MCU: STM32F745 32-bit processor
- IMU: MPU6000 (SPI)
- Barometer: BMP280
- Current Sensor: Approximately 130 amps maximum measurable value
- USB VCP Driver (all UARTs usable simultaneously; USB does not take up a UART)
- 6 hardware UARTS (UART1, 2, 3, 4, 6, 7)

- Supports serial receivers (SBUS, iBus, Spektrum, Crossfire) only. PPM and PWM receivers are not supported.
- TF card for Blackbox logging
- Dimensions: 35x48x7mm (includes foam-mounted gyro board in height)
- Mounting Holes: Standard 30.5mm square to center of holes
- Weight: 11g

Additional Information: The Flight Controller is compatible with the uses of ESCs and motors in mind. The integrated MPU6000 IMU and BMP280 barometer provide precise and accurate flight data. With a current sensor capable of measuring up to 130 amps, it ensures safety and stability during flight. The USB VCP driver allows all UARTs to be used simultaneously, without sacrificing any ports. With 6 hardware UARTs and support for serial receivers such as SBUS, iBus, Spektrum, and Crossfire, it offers versatile connectivity options, compatible with the ESC and other electronics. Compact in size (35x48x7mm) and lightweight (11g), it is perfect for my application.

2.4 Battery

Considerations:

- Voltage/Cell Count: Want either a 4S or 6S configuration
- Capacity (mAh): Consider Flight Time Vs Weight
- Discharge Rate (C rating): Need to be able to handle maximum demands of all motors

TATTU R-Line 1300mAh 4s 95c Square Lipo Battery Pack:

- Voltage (S): 4S (14.8V)
- Capacity: 1300mAh
- Discharge Rate (C Rating): 95C continuous (123.5A), 150C burst
- Connector: XT60 (common connector type for drones)

- Weight: 162.5 g
- Height: 37.2mm
- Length: 49 mm
- Width: 49 mm
- Wire Length: 100 mm

Additional Information: Note that my chosen FC and ESC do not have any pre-required connector, so the connector for the battery is not restricted. With a voltage of 14.8V and capacity of 1300mAh, this battery pack offers a balance between flight time and weight. Its high discharge rate of 95C continuous (123.5A) and 150C burst ensures it can handle the maximum demands of all motors. The XT60 connector makes it compatible with most drones.

2.5 FPV Camera,Transmitter,Control Receiver, Antenna

FPV Camera:

- Resolution & FOV
- Compatibility: Ensure compatibility with your FPV system and video transmitter.

FPV Transmitter:

- Output Power: Choose a transmitter with an appropriate output power (measured in milliwatts, mW) based on your flying range and regulatory requirements.
- Frequency: Select a transmitter that operates on a frequency compatible with your FPV goggles or monitor.
- Channels: Consider the number of available channels and select one that minimizes interference with other nearby FPV systems.

Control Receiver:

- Frequency: Choose a receiver that operates on the same frequency as your radio transmitter (e.g., 2.4GHz, 900MHz).

- Compatibility: Ensure compatibility with your radio transmitter and flight controller.
- Number of Channels: Select a receiver with the appropriate number of channels to support all the functions of your drone (e.g., throttle, pitch, roll, yaw, auxiliary channels).

Antenna:

- Omnidirectional vs Directional
- Polarization: Ensure that the antenna's polarization matches that of your video transmitter and receiver to maximize signal strength and minimize interference.
- Gain: Select antennas with appropriate gain values to optimize signal reception and transmission based on your flying environment.
- Connector Type: Check compatibility with your FPV equipment (e.g., SMA, RP-SMA) to ensure proper connection.

Now based on those specifications. I chose certain models.

FPV Camera: RunCam Swift 2

- Resolution: 600TVL
- Field of View (FOV): 2.3mm lens (150° diagonal FOV)
- Size and Weight: 28.5mm x 26mm x 26mm, 14g
- Compatibility: Supports 5V input voltage, suitable for direct connection to the flight controller's camera input.

FPV Transmitter (VTX): TBS Unify Pro32 HV

- Output Power: Adjustable up to 1000mW
- Frequency: 5.8GHz
- Channels: 40 channels, including Raceband
- Size and Weight: 37mm x 25mm x 5.8mm, 8.7g
- Compatibility: Compatible with most FPV goggles and receivers, supports SmartAudio for easy channel and power adjustment.

Control Receiver: FrSky R-XSR

- Frequency: 2.4GHz ACCST
- Channels: 16 channels
- Range: Full range (¿1.5km)
- Size and Weight: 16mm x 11mm x 5.4mm, 1.5g
- Compatibility: Compatible with FrSky transmitters using ACCST protocol, suitable for integration with the Holybro Kakute F7 AIO V1.5 flight controller.

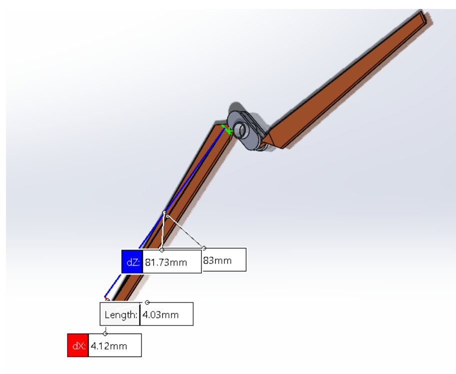
Antenna: Lumenier AXII Stubby 5.8GHz Antenna

- Type: Omnidirectional
- Polarization: RHCP (Right Hand Circular Polarization)
- Gain: 1.6dBic
- Connector Type: SMA
- Size and Weight: 22.1mm x 17.5mm, 4.8g
- Compatibility: Compatible with the TBS Unify Pro32 HV transmitter

3 Version 1 Design Documentation

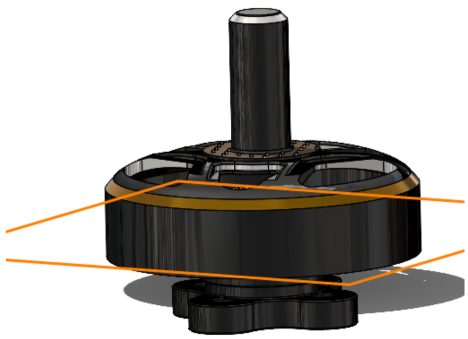
I will now go into an explanation of my design process for the first version of the drone, pre-FEA or CFD simulations. Note that the pictures I include do not show every part in the design process but provide an overall explanation.

3.1 Propellers



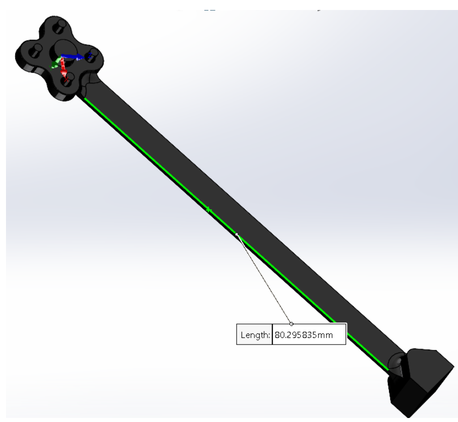
In this sub-assembly, there are 3 individual components: The propeller, the top cover, and the bottom cover. The top and bottom cover fit together, while the propeller is inserted into cylinders on the side, where it wraps around and sticks with a question mark shaped end. This allows for the propellers to be removed and easily replaced if they are possibly damaged, while also serving their purpose. The span of each propeller is about 80mm. Their design was based off a drone I disassembled and analyzed. They are going to be printed in TPU to allow for flexibility.

3.2 MotorCAD



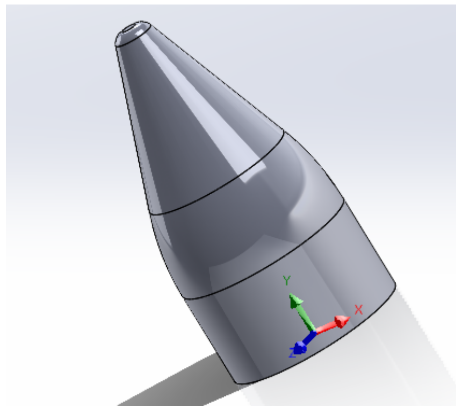
This is an accurate CAD of the motor I found online. I suppressed the M5 Screw feature of the shaft for the purpose of the CAD.

3.3 Arm



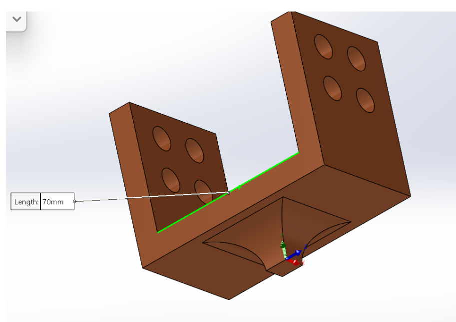
This the arm that connects to the motor on the top left end and the arm connector on the bottom right end. The top left has the positive shape of the motor base, with 5 extruded circles that create a fit into the holes for the motor. Though designed separately for convenience, this is one piece with the arm connector and will be printed together.

3.4 Arm Screw



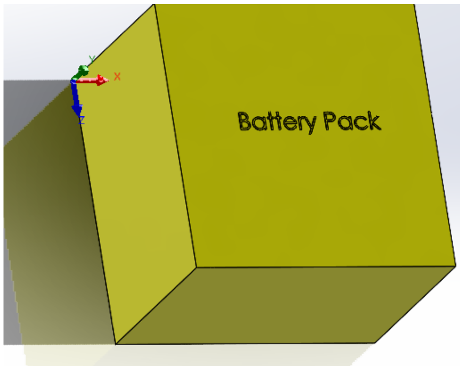
This is the arm screw that will cap the space between the top of the motor screw and the propellers, ensuring that they safely remain in the location they are supposed to. It has an ID of 5mm to match the M5 screw.

3.5 Arm Connector



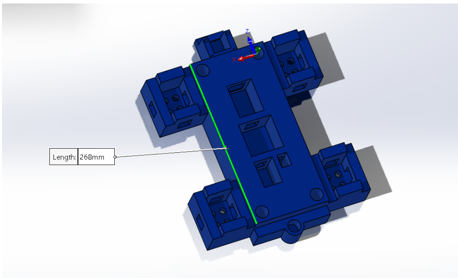
This is the connector from the arm to the ESC holder on the main body. The 8 circular holes are filled with cylinders, meant to act as connectors between the bodies.

3.6 Battery



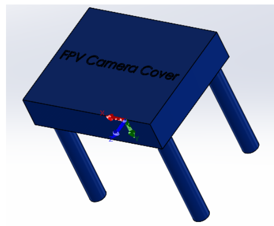
This is an example CAD of the electronics. Each component is colored yellow, and has the dimensions found in the specs online. In addition, I customized the mass properties so they would have their correct mass, allowing for accurate FEA later on.

3.7 Main Body



This is the main body, where the holes are spaces for the various electrical components, with the 4 ESCs on the sides. and holes between various electrical components to account for wiring.

3.8 Camera Cover



This is the cover for the camera in the front of the drone. A similar cylindrical connection mechanism is in place for the ESC covers and the Main Body Cover.

3.9 Assembly V1

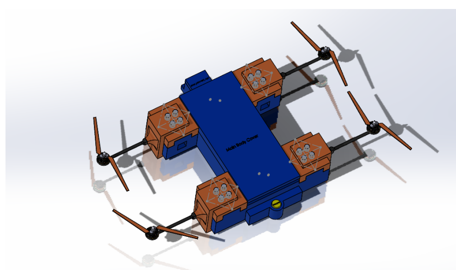


Figure 2: Assembly I

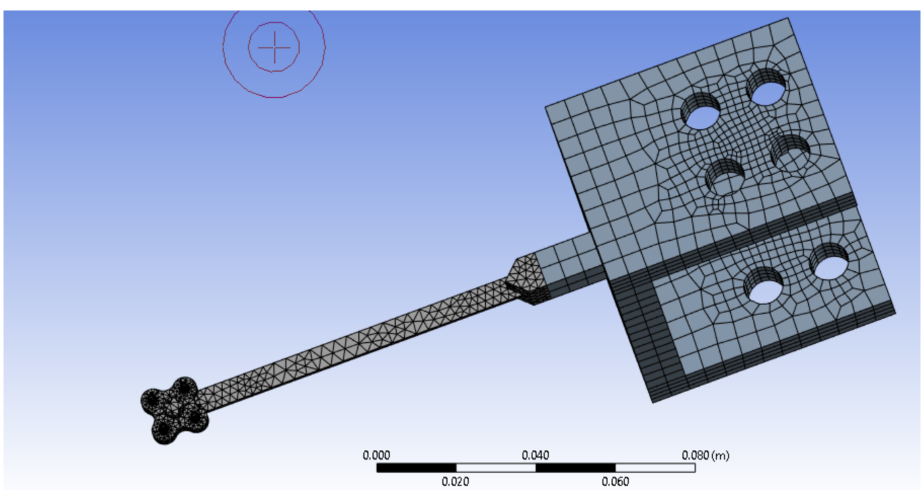
This is an overall assembly of the drone. I believe after analysis, some issues with the overall shape/bulkiness of the drone will arise. In addition, I think the arms need to be re positioned in more of an X shape possibly. We can see what happens after simulation to decide our changes. For now, the goal was to minimize random design decisions, and base all improvements on actual data. Note: The cylinders/holes are not toleranced for this version (when 3D printing this would not work as the shafts and holes would not print material exactly) which will be adjusted in future versions. In addition, all 3D printed parts (i.e. non-electronics) other than the propellers will be made from PETG due to its strength and lack of brittleness compared to PLA.

4 Static Structural Analysis

4.1 Pre-Analysis

We will now talk about the set up behind the FEA of the quadcopter. Firstly, I decided to do static structural analysis on the arm, as I want to see if it could hold up the motors and withstand the weight of the main body. The main body is made of PETG with assumed 40 percent infill density. The propellors are made of TPU and are assumed to have a density of 60 percent. By finding the volume of the bodies through the mass properties in SolidWorks, and density values through values in research documents (1.27 g/cm3 for PETG and 1.21 g/cm3 for TPU) I calculated the masses. For the body mass + the electronics, it will output 351g per arm (assuming even distribution to each arm). The motor + propellor combination leads to a mass of 21.3 g.

4.2 Mesh



The above is a general picture of my mesh. To increase my mesh quality, I got rid of the fillets on faces and edges, which should not majorly affect the results, but do higher the mesh quality. In addition to this, I applied a few meshing operations. Firstly, I applied MultiZone meshing on the rectangular body to allow more hex compared to triangular shapes which have higher quality. I then applied body sizing to the rectangular body with each node having a size of .005 m. I applied face sizing to the circular holes allowing for a more refined mesh around them, with a size of .0025 m. I then applied face sizing to the top of the arm body as well as the connection point with a size of .002 m. Then, on every part of the circular end of the arm other than the 4 minor connectors, I applied a face sizing of .0015 m. Finally, on the 4 minor cylinders I assigned a size of .0005 m. These different sizes allow for locally refined mesh based on the size, so I can also try to avoid excessive nodes by making it unnecessarily refined. In addition to this, due to the irregular shape of the left body of the arm, I applied the patch conforming method to allow for tetrahedron shapes to accurately cover this arm.

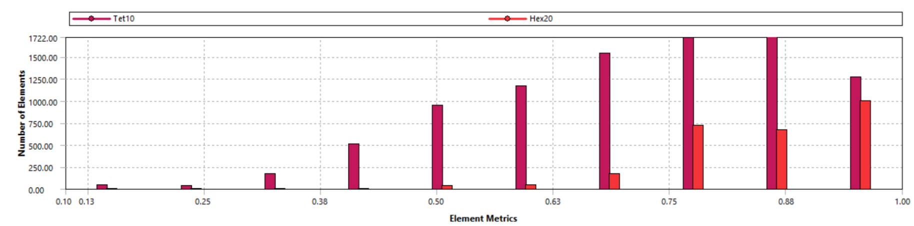


Figure 3: Element Quality

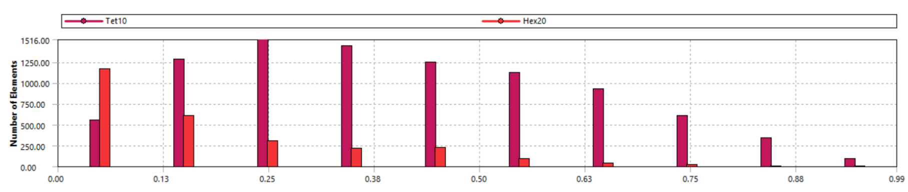
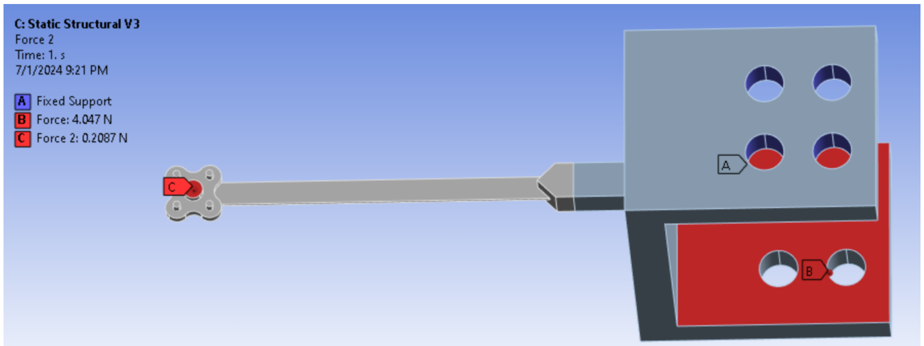


Figure 4: Skewness

The figures of element quality and skewness show that my mesh is of relatively good quality, and the mesh should not skew the results. We can evaluate this more in the verification and validation step.

4.3 Boundary Conditions

With the fixed support at the top holes and the weights at the bottom and end of the arm, we have fully defined our object for analysis



now. Though the forces (derived from F = mg) are very small, the arm is also extremely thin, and as we will see, will not handle this well.

4.4 Solution & Verification

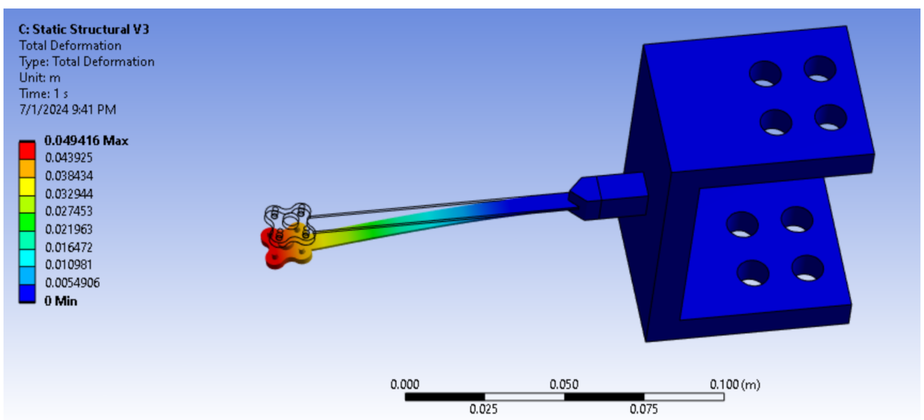


Figure 5: Mesh Solution

From the solution, we can clearly see there is large deformation in the model (a max of 49 mm). The wireframe indicates its original position. From this I can conclude I need to make changes to the arm in order to help it withstand the weight better. I plan to do this through: shortening the arm, adding more structural components to it such as triangular shapes), and making it thicker.

For verification, we need to consider a couple things. First of all, the skewness and element quality showed that the mesh was good. In addition to this, I redid the mesh with smaller nodes, and the max deformation still remained at 49 mm. The only thing of concern here is the Ansys warning that our deformation may be too large to be calculated correctly, but this still validates our point of needing reconstruction. In addition to this, Ansys may miscalculate the properties due to the fact that the PETG will not have 100% infill. However, all these points still further reinforce the fact that the arm is weak, and next we will focus on its reconstruction to properly support the motors.

After adjusting my arm, I was able to reduce deformation by over 99 percent to just .25 mm. (Note in the following picture the units are mm while the previous picture was in m).

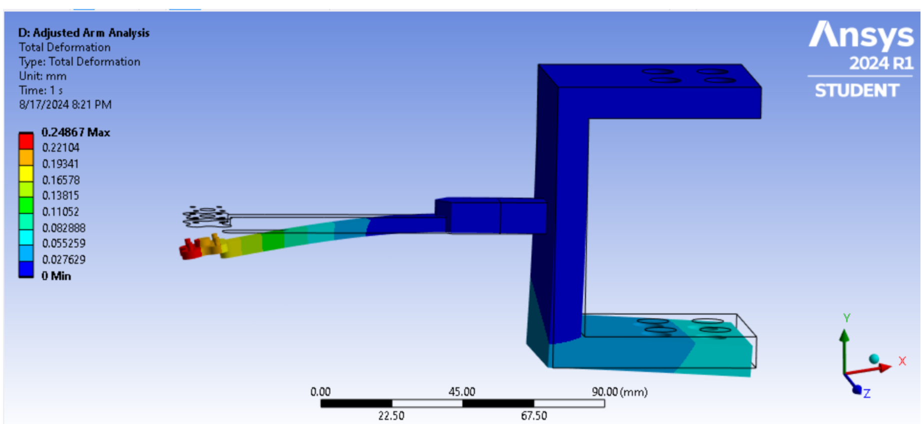


Figure 6: Adjusted Arm Deformation

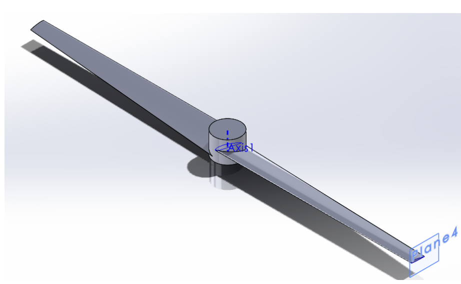
5 Computational Fluid Dynamics

5.1 Purpose

The purpose of this section is to design propellers that will produce enough thrust. The current propellers are just flat and more for show than functionality. I will be designing and then analyzing my propellers, attempting to create an optimal design

5.2 Initial Design

For this, I researched propeller designs for drones as I decided my initial parameters. I decided on doing a 2 blade propeller design, and was limited to a length of around 80mm to make sure my propellers didn't hit my drone body. Next, I had to determine the chord of my drone along with the angle of attack, and twist. What I decided for my design is to simply use the max chord I currently had on my initial design, which was modelled off of real drone propeller blades that I disassembled to analyze. I made the tip of the drone have the smallest chord of 5mm and angle of attack of 5 degrees. I then used the NACA 2412 airfoil in my design, as it is a simple and commonly used airfoil, which is perfect when I am not sure how much thrust I actually can produce. Based on the given plotter chart on the NACA 4 digit generator site, I was able to determine that the max angle of attack should be 15 degrees before it loses effectiveness. Then, I simply lofted 2 NACA 2412 airfoils together, one with the max chord length and angle of 15 degrees and another with 5mm chord length and 5 degrees angle of attack. After this, I created a central hub and circular patterened to create a second propeller blade.



5.3 Setup

Now I will discuss my parameters, input conditions, and the results I hope for from my simulations. First, I did a Reynold's number calculation to verify that doing a turbulent flow model would be best. I did this with the equation Re = pUL/(mu). I got a minimum Reynolds number of 22,313, assuming an RPM of 5000. I am ranging my RPM from a min of 5000 to a max of 40000, all within the range of the motor to see which one produces the thrust I am looking for. This large Reynold's number confirms to me that I will be using a turbulent flow model. In addition to this, I generally chose to do a K-Epsilon realizable scalabale wall function model due to K-epsilon being good for high Re and meant for external aerodynamic problems. Realizable functions are an improvement on K-epsilon equations and make sure the model is physically realizable. Wall functions additionally, are necessary for the behavior near walls and boundary layers, and help deal with the steep velocity differences. The scalable part is an improvement on this, where it is adaptive for mesh resolution, and allows our overall analysis to be lest computationally expensive as our mesh resolution at walls do not have to be as high. In addition to this, I will graph RPM vs efficiency (output power / input power) to determine how optimally my propellers work. My overall drone weight in this design is lightened to 1427g, or 356.7g of weight distributed over each propeller when hovering, which is what I want my drone to be able to do, while having the potential to attain a 2:1 thrust to weight ratio at certain RPMs, which will generally allow for smooth maneuvering.

When preparing the geometry, I used the method of creating a rotating cylindrical domain around the propellers and a box static domain to show the outer walls of my simulation. My other conditions are as follows: a pressure outlet with the gauge pressure being 0 and a velocity inlet with the condition being .1 m/s, which is meant to simulate a hovering drone. This velocity condition changes in some simulations to 10 m/s to see how the drone would react if it was moving at some reasonable speed. For the cell zone conditions, I used the mesh motion method on the rotating domain, meant to simulate as if the nodes there were moving at the given RPM. Additionally, I did a transient analysis with gravity present. Finally, I used hybrid initialization with 26 time steps of size .15s and 10 iterations per step.

5.4 Results and Iterations

The first iteration was done with a NACA 2412 airfoil lofted together. The results were a thrust of around -.05 N. This result was suspiciously low, so I ran simulations slightly varying the parameters, making sure my inlet and outlet conditions were right, paying specific attention to the direction of flows, making sure everything was the correct sign. From this, to further verify that my parameters were reasonable and would work, I was able to find a propeller design online that someone made, and who verified their results through CFD. After running my own simulation, I was able to get a reasonable thrust force production of 9.7 N. This verified to me that while my method seems to work, my propeller is the issue. My second iteration of my propeller used a different airfoil: the NACA 4412 which is generally known to produce higher thrust. After doing analysis on this, I faced the same issue of a negligible thrust force. My third iteration, I decided to work on changing the offset of the propeller so it wouldn't have a straight edge, as this seems to be a possible cause of issues. This version had better streamlines, but still was unable to produce any sizable thrust. Therefore, my next iteration I decided to vary my twist on my blade, starting with a much higher initial AOA (initially 15 degrees), to make a much sharper twist, as this seems to be how other successful propellers are designed. For reference, here are my different propeller designs, in order, below:

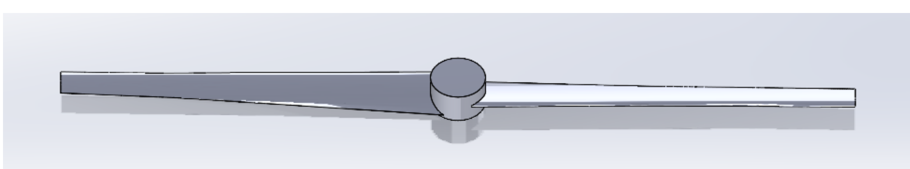


Figure 7: 1st Propeller: NACA 2412 Airfoil

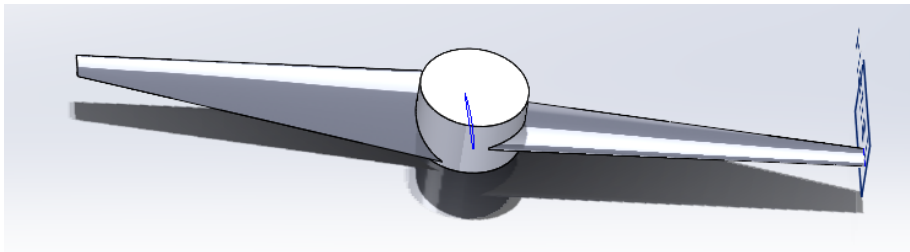


Figure 8: 2nd Propeller: NACA 4412 Airfoil

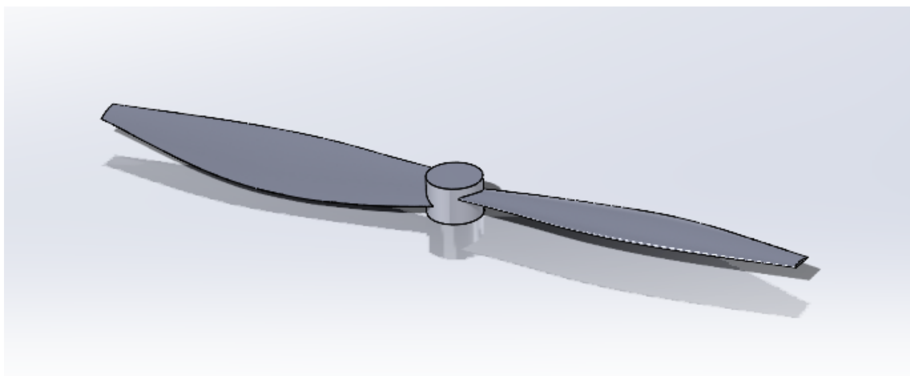


Figure 9: 3rd Propeller: Offset X NACA 4412 Airfoil

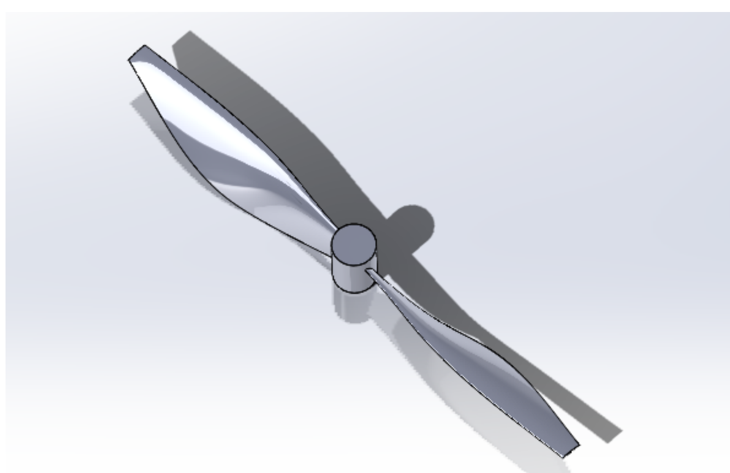


Figure 10: 4th Propeller: Higher Twist

5.5 Final Design

For the 4th propeller, I decided to start with an angle of attack of 80 degrees. After running the CFD simulation, this finally resulted in a successful trial, where thrust was generated. I then varied RPMs doing trials for 10000, 15000, and 20000 RPM. They generated approximately 2, 3.2, and 5.8 N of thrust respectively. Theoretically, this means that my propellers can produce enough thrust to hover my drone as well as have good control of it (can produce a thrust to weight ratio of 2:1). An example chart of the 20000 RPM version is shown:

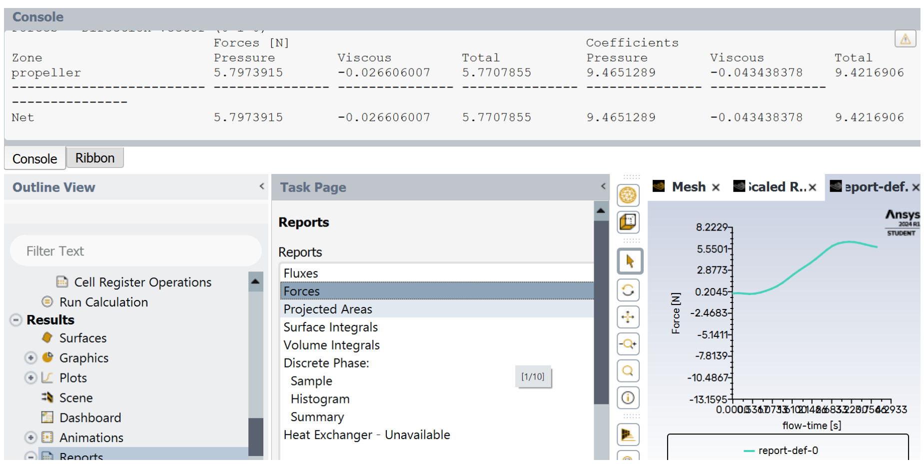


Figure 11: 20000 RPM trial

To discuss possible drawbacks, the first one the fact that the propellers will not produce as much thrust as the simulation, due to efficiency losses. However, considering that my motor is capable of producing up to 40000 RPM with 70 percent of max output, this is not an issue, as they need to provide a max of 20N theoretically for a 2:1 TW ratio.

Another issue arises when viewing the velocity contours for the propeller. The flow of the velocity looks very messy, instead of smooth contours that may typically be generated. Considering that my simulation parameters should be accurate, along with my mesh quality being overall very good for this propeller (the lowest orthogonal quality is .21, which is only very few elements), the reasoning behind the inconsistent flow patterns points to the possibility that my propellers have produce a lot of resistance and have high drag. I suspect this is because I did too much twist, and that having a lower initial angle of attack, for example around 40 degrees, will allow for more consistent flow patterns. However, since my propellers are good enough for my purposes with my drone, this is not a huge issue that I will focus on optimizing. My velocity streamlines, viewed from above, are shown below.

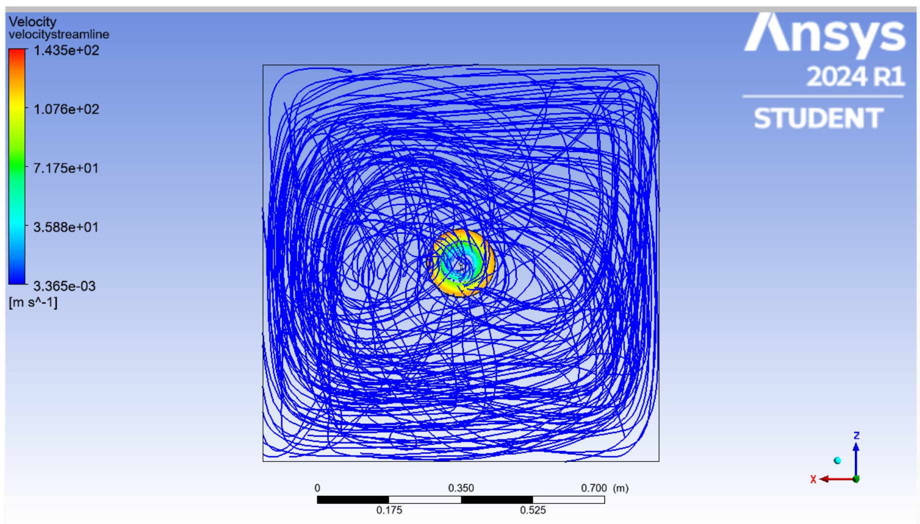


Figure 12: Velocity Streamlines

Overall, these propellers are effective in producing the level of thrust I need, while fitting within my size constraints, so I will be moving forward with these into my last stage, actually 3D printing the drone.

6 Final Design and 3D Printing

6.1 New Body and Assembly

One of the main changes I made was that I changed the structure of the overall drone body to a more circular structure. This allowed for some major improvements. Firstly, it allowed for my arms to be in an X position, leading to a hypothetically better overall distribution of thrust and flight. In addition, I got rid of unnecessary walls and made the design smaller which allowed for me to drop a lot of the drone weight. The finalized assembly appears as below.

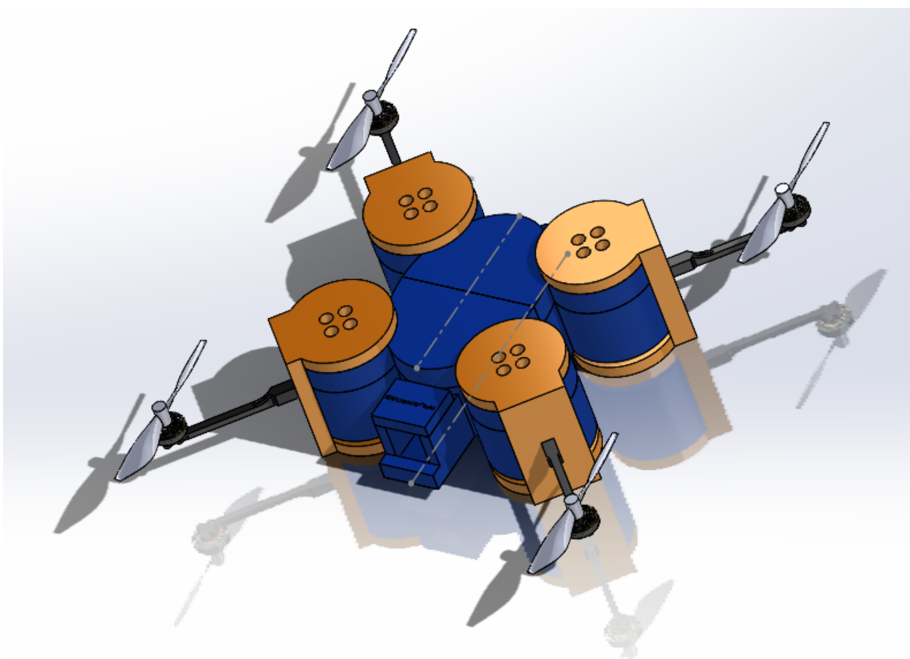


Figure 13: Final Drone Assembly

6.2 3D Printing

When 3D printing for the drone, I generally used 40 percent infill density, and followed the following rules for the PETG printing which are from the official Cura site:

Nozzle temperature: 240 °C
Bed temperature: 70 °C
Layer height: 0.2 mm
Print speed: 40-60 mm/s
First-layer speed: 20 mm/s
Travel speed: 110 mm/s
Retraction: 0.9 mm at 40 mm/s
Build plate adhesion: Brim
Brim width: 4 mm

The only change I made was changing the print speed to about 75 percent of the recommended, as the large drone body was not printing accurately at that speed. Some example 3D printed parts are shown below as proof of concept.

Figure 14: 3D Printed Parts

6.3 Future Improvements

In this section, I will discuss some issues that I would improve on my next iteration if I were to do so. Firstly, I would vary the twist in the propellers to identify the optimal range to generate the thrust necessary at lower RPMs. In addition, I would definitely aim for a smaller drone design. The current design's weight is more than I expected, due to inaccurate assumptions for the density of PETG. I would definitely aim for a design that would thin out the walls more. Additionally, I would consider finding smaller electronics, most notably the battery which may not be the best fit for the type of drone I am trying to design, and decide the trade off between flight time and drone weight and size. On top of this, I would likely remove the arm connector component entirely, as it is possible to make a design where the arm is directly connected to the main body. Overall, my changes to the design are aiming to increase the thrust to weight ratio, as well as reduce the size of the drone as it is bigger than I initially planned. However, regardless of these possible future changes, the drone still should generate enough thrust to fly well, and it fulfills my goals of a consumer-friendly drone.

7 References

https://rotorriot.com/products/t-motor-f2203-5-2850kv-motor?variant=42877872636134
https://grabcad.com/library/t-motor-2203-5-kv2850-bldc-quadcopter-motor-1
https://cfdflowengineering.com/working-principle-and-components-of-drone/
https://www.jouav.com/blog/electronic-speed-controller-esc.html
https://www.hobbywingdirect.com/products/xrotor-20x20-esc?variant=40784381804659
https://www.getfpv.com/holybro-kakute-f7-aio-flight-controller.html
https://www.getfpv.com/tattu-r-line-1300mah-4s-95c-square-lipo-battery-pack.html
https://shop.runcam.com/runcam-swift-2/
https://www.team-blacksheep.com/products/prod:unifypro32
www.getfpv.com/frsky-r-xsr-2-4ghz-16ch-accst-micro-receiver-w-s-bus-cppm
https://www.getfpv.com/lumenier-stubby-axii-5-8ghz
https://journals.sagepub.com/doi/full/10.1177/26349833211016477
https://www.simscale.com/blog/how-to-optimize-propeller-design/: :text=Propeller
https://iopscience.iop.org/article/10.1088/1757-899X/455/1/012018/pdf
https://m-selig.ae.illinois.edu/pubs/DetersAnandaSelig-2014-AIAA-2014-2151.pdf
https://all3dp.com/2/ender-3-s1-cura-profile-settings/: :text=We