Peter Linder
Engineering Honors
Winter 2022

M-Fly Autonomous Detect-and-Avoid System and Unmanned Ground Vehicle

Capstone Final Report

Introduction

M-Fly is a student design team at the University of Michigan which builds three airplanes for three different competitions. For the 2021-2022 academic year, I held the position of Autonomous Chief Engineer, meaning I oversee the development of the MAT-3, an airplane which will compete in the AUVSI-SUAS competition (Figure 1). The objective of this competition is for an airplane to autonomously fly, navigate a series of GPS waypoints, avoid stationary obstacles and other competing aircraft, identify objects on the ground via aerial photography, construct a map of the flight area with image stitching, and drop an Unmanned Ground Vehicle (UGV), which must land at specified coordinates and then autonomously drive to another set of coordinates to deliver a water bottle. The goal of this competition is to simulate the real-world need for autonomous vehicles for search-and-rescue missions.

Figure 1: The MAT-3.3 airframe, manufactured in March 2022

The computer vision tasks, including object classification and mapping, are within the scope of my position as a lead, but are delegated to our software lead who has more experience with computer vision and machine learning. I have already completed the code responsible for planning a path that navigates the waypoints and avoids stationary obstacles. Additionally, the airplane itself has already been designed and fabricated by my predecessors on M-Fly. Thus, the scope of my work for the previous two semesters was tailored to the other remaining tasks: avoiding other competing aircraft and deploying a UGV. As the chief engineer for the autonomous team and a computer engineering major, my work entailed selecting the electronic components for our obstacle detection system and the UGV, along with the embedded programming which must pair with each to make it functional.

The primary objective of this project for the past two semesters was to prototype a detect-and-avoid system and a UGV to make M-Fly ready to compete in AUVSI-SUAS in June 2022. This entailed solving two large problems. First, I needed to do more research to determine what type of sensor would be cost-effective and good enough to detect another UAV at a range of 50-100 meters. The software approach to detection would simply be determined by whatever sensor I end up choosing. Second, I needed to collaborate with the mechanical engineers on M-Fly to design a UGV which is light enough to be dropped out of our airplane and parachute down to the ground, while housing the altimeter, GPS, IMU, microcontroller, motor drivers, and other electronics needed for it to autonomously deploy the parachute and drive.

My project serves to improve M-Fly's airplane for the AUVSI-SUAS competition, which aims to foster engineering creativity to advance the field of UAVs for applications such as search-and-rescue, surveillance, package delivery, and more. Specifically, a detect-and-avoid system is applicable to any UAV, especially one with a cost constraint such as the one imposed on me by M-Fly's limited budget, and the UGV is relevant to search-and-rescue and package delivery applications. Ultimately, the audience is primarily the defense industry, but there are commercial applications as well.

Timeline

As part of a design team, my capstone was inherently a group project in nature. My individual responsibilities included hardware selection for the detect-and-avoid system and UGV, developing the software myself, and collaborating with our design and manufacturing subteams to create a UGV that fits the competition's constraints. My individual goals were to detect a small quadcopter at one of M-Fly local test flights, and to deploy the UGV at a test flight. My contributions to the project that make this my capstone include the underlying code that I have spent the past two years writing for path planning around obstacles and autonomous flight systems communications, along with the hardware and software I developed in the past two semesters. Most importantly, as the leader of the autonomous subteam, this is a large project which I managed myself.

I divided up my plan to complete the detect-and-avoid system and the UGV into the following series of steps:

- Detect-and-Avoid

  - Program dynamic obstacle avoidance using other UAVs' published telemetry
  - Purchase Nvidia Jetson Nano and multiple MEMS microphones
  - Construct microphone array for testing purposes
  - Program Jetson to localize the sound of a small quadcopter's propellers
- UGV
  - Select microcontroller, GPS, altimeter, IMU, motor drivers, motors, and parachute release mechanism
  - Order those components
  - Meet with structures team to design UGV
  - Help manufacturing team construct UGV
  - Wire all the components within the UGV
  - Program the UGV

With this plan in mind, I devised the Gantt chart shown in Figure 2.

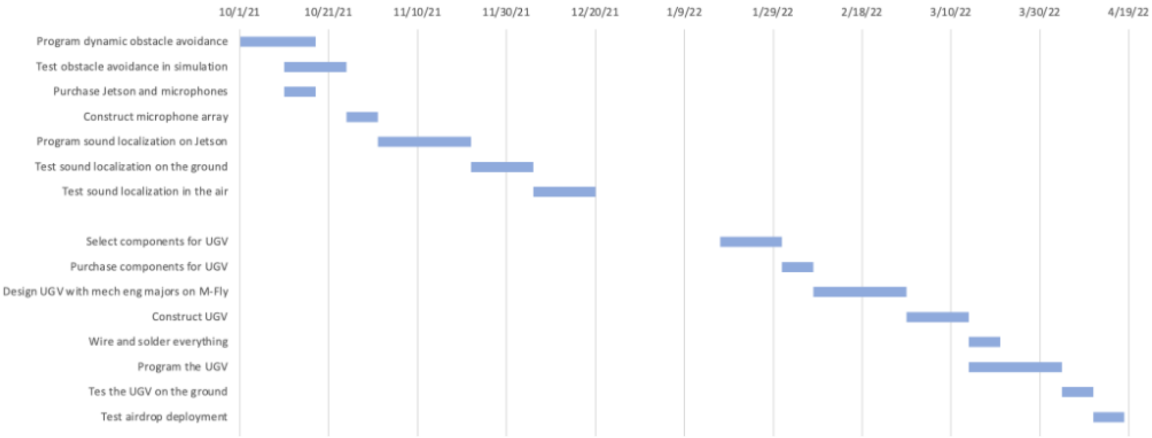

Figure 2: Initially projected timeline for my capstone project

My project did not closely adhere to this timeline, for reasons to be discussed in subsequent sections.

Detect-And-Avoid System

As mentioned above, the objective of the detect-and-avoid system was to develop a cost-effective system that could detect other aircraft during flight from a range of 50 to 100 meters. This range was selected based on the cruise speed of the MAT-3 and its turning radius, so it would have sufficient time to plan and execute an alternative trajectory. A state-of-the-art system would use Lidar, a laser-based sensor which has become the de facto standard of autonomous vehicles. However, while Lidar for small robots or even cars is affordable, it becomes prohibitively expensive to purchase a Lidar with a range of 50 meters and a wide field of view. Such Lidars are used in military and even some high-end commercial applications, but I had to pursue alternatives to work within M-Fly's budget. The cheapest Lidar with such a range that would be suitable was the Livox Mid-40, which costs $600 and still does not provide as wide a field of view as would be ideal.

Stereo Vision

While I was intrigued by using sound source localization, I started with a more conventional approach in robotics: stereo vision. This entailed using two cameras on the same plane, separated by some translation, and computing a disparity map from the resulting two images. The depth accuracy for stereo cameras is proportional to the baseline distance between the cameras. A commercial stereo camera might have a baseline of about 30 cm, and a usable depth accuracy of about 10 meters. For the level of accuracy I needed, I planned to take advantage of the MAT-3's 11-foot wingspan to provide a large baseline, as depicted in Figure 3.

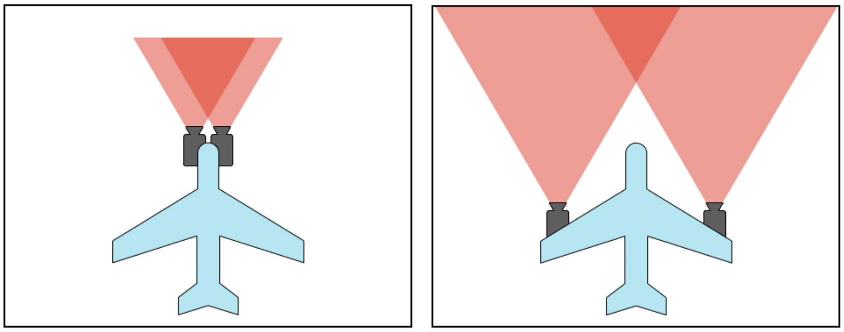

Figure 3: Concept of short versus long baseline for stereo vision

I created a test bench by mounting two ELP 2.1mm lens USB cameras to a spare piece of aluminum, with a baseline of 30 cm. The total cost of this setup was just under $100, which would have made it a very cost-effective method. However, problems immediately arose. Stereo cameras are extremely sensitive to camera alignment. Despite each camera being mounted firmly with nuts and bolts at each corner to the piece of aluminum, the aluminum was not perfectly flat and had some give, causing the cameras to easily get out of alignment. The accuracy of the disparity map drops precipitously at even a small shift in alignment, causing the poor quality exemplified in Figure 4.

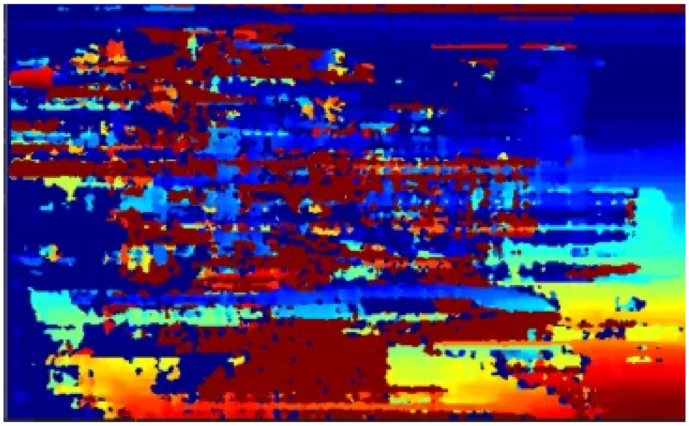

Figure 4: Disparity map of uncalibrated cameras

The problem persisted when the cameras were actually mounted on the wings, as the 11-foot structure had even more room for imperfections and bowing. Due to the inherent problems of the system, I never tested stereo vision in flight, though I predict the results would have been even worse due to the wind and vibrations from the wing-mounted propellers.

Sound Source Localization

The next approach was sound source localization. The underlying principle to this approach was trilateration, or the geometric estimation of an object's position given its relative distances to beacons. This is similar in principle to triangulation, by which a cell phone can be localized given the signal strength relative to nearby cell towers. Figure 5 from the Factory Intelligence Laboratory illustrates trilateration with respect to sound-source localization. Just as signal strength can be measured from a cell phone to a cell tower, the speed of sound is slow enough that a computer can measure the difference in time between a signal arriving at one microphone versus another. This time difference of arrival (TDOA) can be used to approximate the object's distance from each microphone given the speed of sound, which can in turn be used to approximate the object's position.

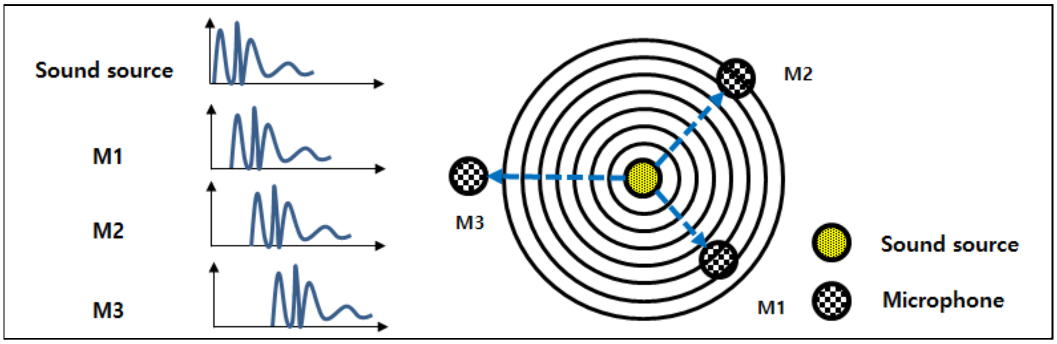

Figure 5: An illustration of sound-source localization using TDOAs

To test this principle, I first purchased a ReSpeaker USB Mic Array, a $70 array of four microphones. The ReSpeaker was designed for Internet of Things (IoT) applications, such as a smart home speaker like an Amazon Echo or Google Home, which can use multiple microphones to filter out noise and focus on a person's voice. The ReSpeaker computes onboard the direction of arrival of the dominant sound, but also outputs the raw audio channels from each microphone. I used these raw channels to try localizing a sound given a target frequency. In my tests, I played a 260 Hz sine wave on my smartphone while moving it around the ReSpeaker. My code was able to successfully identify the direction of arrival of the sine wave, but consistently estimated a distance of a few centimeters. I discovered the cause of this ambiguity was rooted in the fact that the ReSpeaker has a diameter of 77 mm and a maximum sampling rate of 16 kHz, which meant the speed of sound was too fast for it to accurately estimate the distance.

To solve this problem, I decided to build a custom microphone array that would more accurately represent its final configuration if successful. I procured a leftover carbon fiber fuselage from a previous plane M-Fly had built which was approximately the size of the MAT-3. I then purchased multiple Adafruit I2S MEMS Microphone Breakouts, a $7 digital microphone which could be mounted on the outside of the plane. I mounted 4 of these to the fuselage, two on the left and two on the right, in a non-planar configuration, so that they could be used to localize a sound in 3D. I2S (Inter-IC Sound) is a digital communication protocol very similar to SPI (Serial Peripheral Interface). In order to use these four microphones to localize a sound, I would need to take samples from each microphone at the same time. A typical microcontroller could not do this, as this would require 4 separate I2S buses. I used an Arduino MKR Vidor 4000, an $80 PCB that includes an ARM Cortex-M0 microcontroller and an Intel Cyclone 10 FPGA. With the FPGA, I programmed a module which simulated four I2S buses and latched values that it samples from each microphone at the exact same time. Then, the microcontroller would fetch these synchronized samples from the FPGA over SPI. Then, this data was passed to a computer over USB via the Arduino's UART adapter, and the computer performed the same data processing as with the ReSpeaker. The architecture of this system is more simply illustrated in Figure 6.

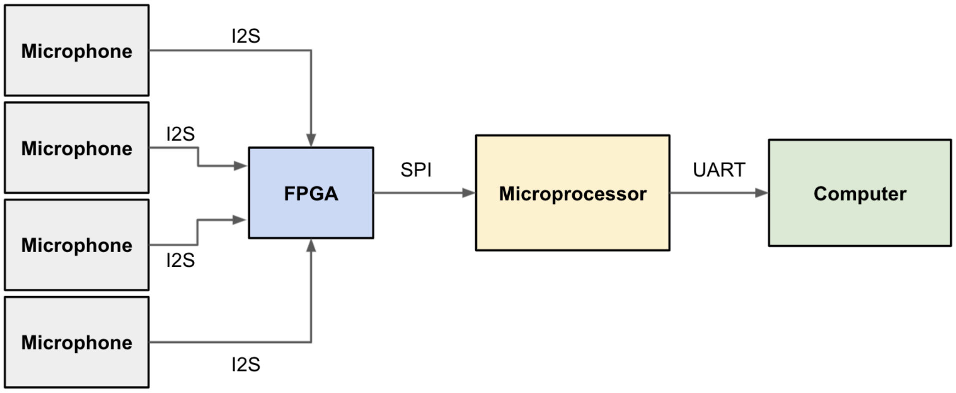

Figure 6: System architecture of custom microphone array for sound source localization

The microphones had a sampling rate of 15 kHz, and after the overhead of latching the values on the FPGA and passing these to the microcontroller, the overall system sampled at 10 kHz. This proved sufficient to detect noise from other aircraft. I performed tests flying a small quadcopter near the fuselage test bench, and found that the quadcopter was generating the most acoustic energy at 2.5 kHz. However, at such a high frequency, a new issue came to light: phase ambiguity. Due to the periodicity of the sound from a propeller, there was no way to distinguish whether the signal arrived at one microphone before or after another. Especially at 2.5 kHz, much higher than the previously tested 260 Hz, the period is so short that assumptions cannot be made. For example, Figure 7 shows two signals, where it is ambiguous whether the blue signal arrived slightly earlier or much later than the orange signal.

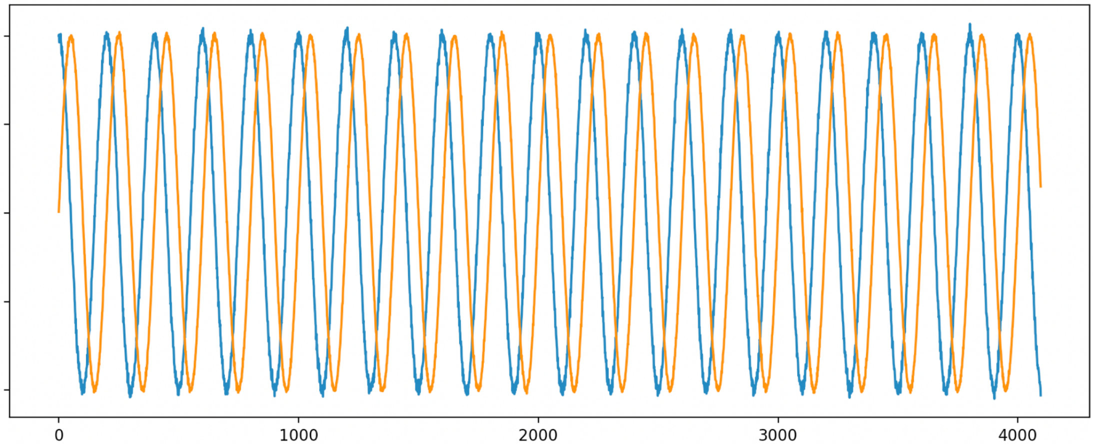

Figure 7: An example of a phase ambiguity for a 2.5 kHz wave at two microphones

I attempted to rectify this by using a wider bandpass filter for focusing on a specific frequency in order to leverage noise in the signal to make it appear less periodic, but this did not work because the noise at each microphone was independent. Ultimately, my attempt to use sound source localization failed.

Radar

Given that the two most ambitious and cost-effective methods of stereo vision and sound source localization proved ineffective, I began exploring radar, a more traditional method although not as potentially cost effective. Radar, like Lidar, can be prohibitively expensive, so I limited my search to cheaper commercially available radars. Many of these radar sensors were designed for traffic-related applications, such as detecting the speed of a car, and did not report the position of the object. I ruled these radars ineligible, because the position of an obstacle, or at least the general vicinity, would be necessary in order for the MAT-3 to plan a path avoiding it. My search led me to OmniPresense, a company that sells radars priced around $200. Some of their models only provide the speed, while others provide the speed and distance of an object, as illustrated in Table 1.

Table 1: OmniPresense Radar Feature Comparison

Product	Type	Speed	Direction	Range	Regulatory Certification
OPS241-A	Doppler	Yes	Yes		
OPS242-A	Doppler	Yes	Yes		FCC/IC
OPS243-A	Doppler	Yes	Yes		FCC/IC
OPS241-B	FMCW			Yes	
OPS243-C	FMCW and Doppler	Yes	Yes	Yes	FCC/IC pending

However, the latter models lack FCC certification. The AUVSI-SUAS competition requires all radio frequency communications to be FCC compliant, which meant none of OmniPresense's products would be suitable for the detect-and-avoid system.

I then discovered millimeter-wave (mmWave) radar, a type of radar technology quickly becoming popular in autonomous vehicles. While OmniPresense radars operate in the 24 GHz band, Texas Instruments manufactures mmWave radars that operate in the 77 GHz band. By the Friis transmission formula, the range of a wireless signal is inversely proportional to the frequency of the signal, implying that mmWave radar has a shorter range than traditional radar. This also means that mmWave radar causes less interference with nearby devices, and thus mmWave radars have more easily obtained FCC certification.

I purchased the Texas Instruments AWR1642BOOST, a $300 mmWave radar with a range of about 80 meters. Though not as cost effective a solution as I had initially hoped, mmWave radar is clearly a viable solution which is more affordable than Lidar, so the development of a detect-and-avoid system was a success.

Unmanned Ground Vehicle

In parallel to my work on the detect-and-avoid system, I was working on the unmanned ground vehicle (UGV). The objective for this part of my project was to design a ground vehicle which weighs under 4 pounds, including an 8 oz water bottle, per the AUVSI-SUAS competition requirements. The competition also requires that the UGV be deployed from at least 100 feet in the air, after which it must land within 5 feet of specified coordinates and autonomously drive to another set of coordinates to deliver the water bottle. Given these competition rules, the practical constraint arises that the UGV must fit within the MAT-3's fuselage.

With this in mind, the first idea I came up with, which a mechanical engineering major helped me to design in CAD software, was a tank-like vehicle robust enough to fall from 100 feet and survive, depicted in Figure 8.

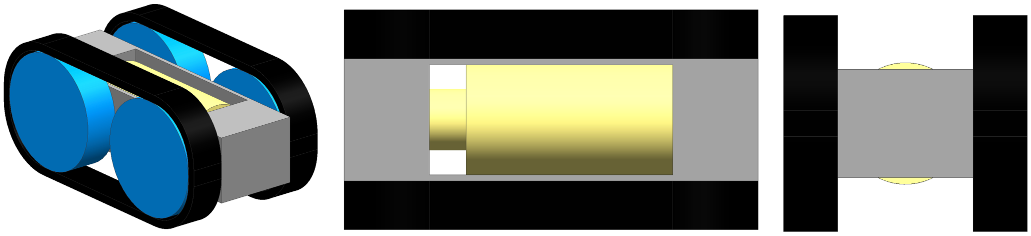

Figure 8: Design for a tank-like UGV

I was hopeful this would be able to land either upright or upside down and still drive. Ultimately, my teammates and I ruled this design out because it wasn't robust enough. After all, it could land on its side, in which case it would not be able to drive. Additionally, the weight constraint meant this would need a very lightweight frame, an optimization which conflicts with a frame robust enough to protect the internal electronics on impact.

The next design I worked on was a design leveraging a parachute, shown in Figure 9. The parachute problem is a common example shown in differential equations classes, so I wrote code which could predict the trajectory of a UGV with a parachute deployed based on the drag of the parachute. In simulation, this worked well. However, the trajectory of the UGV is dependent on the wind, and in reality the MAT-3 has no way to measure absolute wind speed. It can measure relative wind speed with a pitot tube, and can approximate the absolute wind speed using ground speed measured via GPS or an IMU, but not accurate enough for the UGV to land within 5 feet of the specified coordinates.

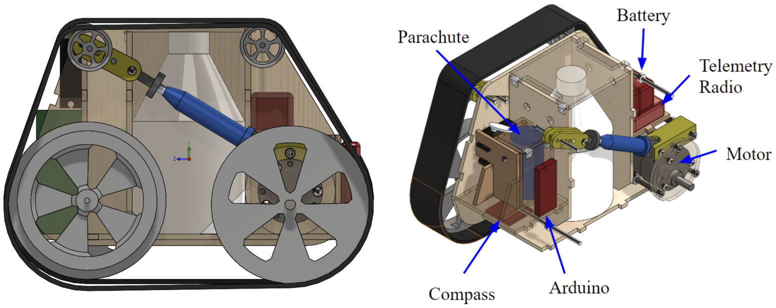

Figure 9: Design for a UGV with a parachute

After ruling out a parachute as too passive and unpredictable a mechanism, I pursue an active approach to slowing and steering the UGV's descent. My teammate designed a UGV with spring loaded armatures that can extend once the UGV is deployed, revealing propellers mounted to drone motors (Figure 10). I consulted with a member of Michigan Autonomous Aerial Vehicles (MAAV), another project team at the University of Michigan which has experience with quadcopters, and he validated that the design would likely be able to actively fly, rather than merely slowing down its own descent. By adapting the control scheme for a conventional quadcopter, the UGV could realistically land directly at the competition's specified coordinates.

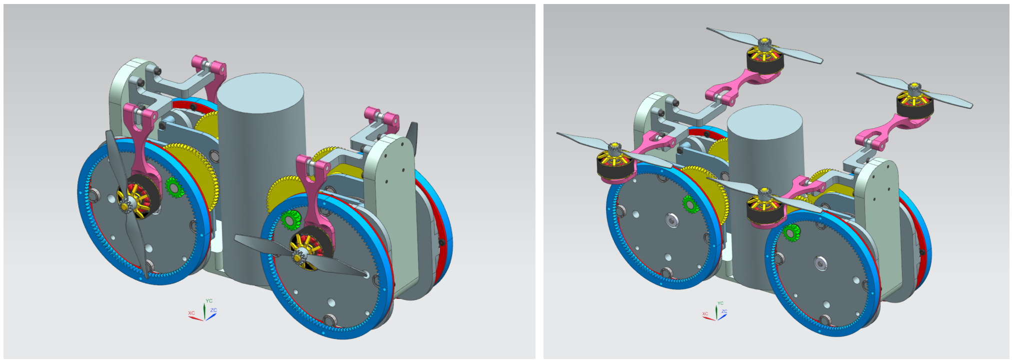

Figure 10: Design for a UGV using drone motors to steer its descent

The design currently weighs 3.6 pounds including a water bottle, but excluding a battery and electronics. The next step is to design a custom PCB with a microcontroller, IMU, altimeter, and motor drivers which can fit inside this design. Given the tight volumetric constraints, I deemed it more practical to create custom hardware to fit into the mechanical design, rather than create a mechanical design to fit existing hardware.

Conclusion

The detect-and-avoid system was a success. After testing stereo vision, sound source localization, and mmWave radar, mmWave radar proved to be a viable solution for detecting another UAV during competition.

Unfortunately, my progress on the UGV fell short of my initial expectations. While the design was completed, it was never manufactured or tested. This can be attributed to the structure of the Autonomous subteam of M-Fly this year, which lacked dedicated manufacturing support. Rather, it relied on the members of M-Fly's Regular and Advanced class subteams to help when they could. This meant that I did not have the resources to manufacture this UGV design.

In fact, the airframe was not manufactured until the end of March, instead of November as I had initially planned. At this point, there was not enough time for test flights and systems integration, and so I never tested the mmWave radar in a competition-like environment. As the Autonomous Chief Engineer, I had to make the difficult decision for M-Fly not to compete in the AUVSI-SUAS 2022 competition.

There still remains much future work. Although I am graduating, I plan to return to M-Fly as a member while I pursue a Master's at the University of Michigan. To prepare for next year, I met with other members of the executive board of M-Fly and we have decided to expand the Autonomous subteam to have dedicated manufacturing support. Additionally, now that the MAT-3 airframe is completed, testing can commence at the start of the Fall 2022 semester. With respect to software development, the next steps are to integrate the output of the mmWave radar with the existing path planner, and to develop the autonomous controls for the UGV. Another large hardware-oriented task is to design a PCB for the UGV.

From my capstone, I have learned the importance of testing early, but more importantly I learned how to lead better. In hindsight, I could have pushed harder to manufacture the plane earlier, and could potentially have avoided falling behind on manufacturing and testing. Nevertheless, I am hopeful for the state of M-Fly Autonomous next year and eagerly await the AUVSI-SUAS 2023 competition.