International Aerial Robotics Competition Technical Paper

Jonas Buxton | Logan Thomure | Roger Downs | Garrett Bosanko
Missouri University of Science and Technology
Multirotor Robot Design Team

ABSTRACT

This report details our complete aerial robot's hardware and software design, from our current progress to our goals for the near future. Jonas Buxton is the CEO. Roger Downs is the Chief Software Engineer. Garrett Bosanko is the Chief Hardware Engineer. Logan Thomure is the Chief Financial Officer and acts as the Hardware/Software liaison.

INTRODUCTION

Keywords

Sentinel - The name the team has given the aerial multirotor robot designed for competition. For simplicity, the robot will be referenced as Sentinel throughout this report.

Problem Statement

For Mission 7 of the IARC our team was tasked with creating a fully autonomous aerial robot that can guide fully autonomous ground robots. Ten of these ground robots will be placed within a square arena measuring 20 meters on each side. Once the mission begins our aerial robot must locate and direct all ten of the ground robots over a green line that will be marked on one side of the arena by interacting with them physically. The main problem addressed in Mission 7 is that of autonomous flight indoors and without the use of GPS. Our aerial robot is expected to avoid obstacles and navigate itself autonomously without the use of GPS or SLAM techniques.

Conceptual Solution to Solve the Problem

Our approach to complete the IARC begins with object detection; using a downward facing camera to find and determine the current direction of the ground robots. In order to ensure that the camera is parallel to the ground, we use a servo to correct for the current angle of Sentinel. Once the ground robots are detected, they are given a weight of importance so that our aerial robot can decide which ground robots to interact with first. This weight scale changes based on position on the field, trajectory of obstacle robots, and probability of obstacle interference. Probability of obstacle interference will be decided using a LIDAR sensor that will detect the obstacle robot's vertical pipes. In order to save time with obstacles, our aerial robot will detect them using lidar and fly over them while continuing on the originally defined path. Sentinel will make contact with the ground robots using a flat bottom surface, capable of performing both the top touch and front blocking interactions. In order to interact with the ground robots, once a robot has been located and selected for interaction, Sentinel will use the image frame to control flight movements to center the ground robot for interaction. Once the image is centered Sentinel will land on or in front of the robot, depending on which is preferred. Balance and stability of the aerial robot will be maintained by using a PixHawk flight control board and optical flow sensor.

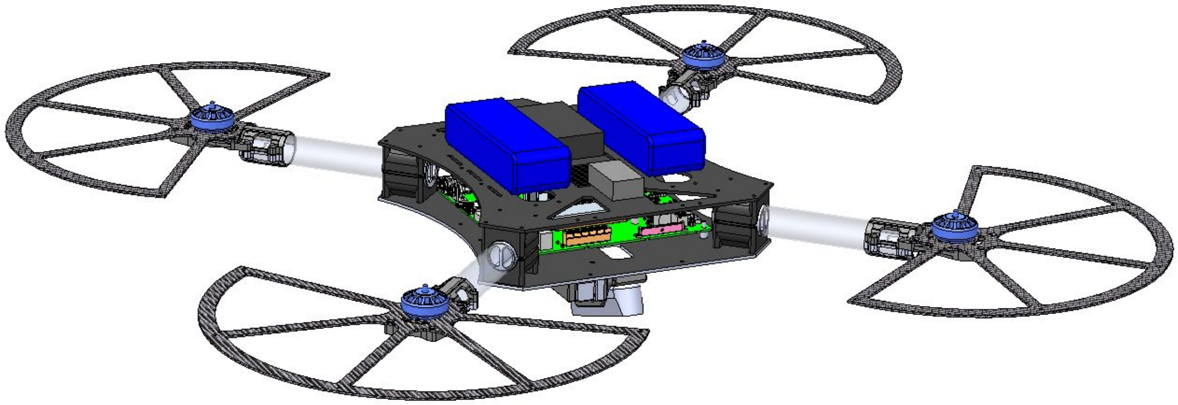

Figure 1. Sentinel's Current Assembly (without cover shell)

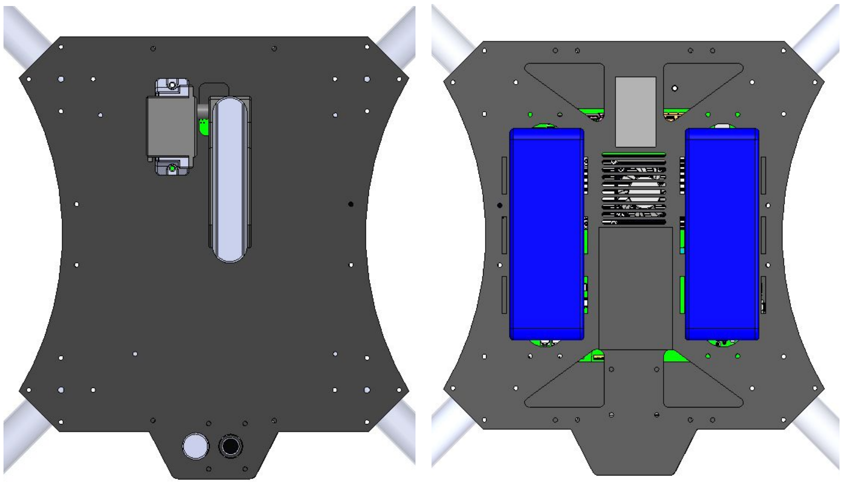

Figure 2. Bottom plate (left) and Top plate (right)

Yearly Milestones

1. August 2017
    a. Frame updated with new, in-house manufactured carbon fiber sheets.
2. October 2017
    a. First prototype frame flight
3. January 2017
    a. First autonomous take-off
4. March 2017
    a. Competition frame built
    b. Fully autonomous movement on prototype
5. April 2017
    a. First flight with competition frame
    b. First autonomous line tracking for movement

AIR VEHICLE

Propulsion and Lift System

Our aerial vehicle uses a standard quad motor design oriented in the "X" configuration as seen in **Figure 3**. The vehicle utilizes four Tarot 4006/620 KV brushless motors controlled by 20 Amp ESC's. Attached to each of motor is a 12-inch carbon fiber propeller with a 5.5-inch pitch. The low KV motors and high pitch propellers were specifically chosen to allow what is a relatively large aerial vehicle to achieve a long flight time. The motor/ propeller combination is incredibly efficient, and was specced by resident aerospace majors on the team. This combination sacrifices the ability to perform more acrobatic or aggressive maneuvers, but is optimal for the more precise and minute movements required for Mission 7.

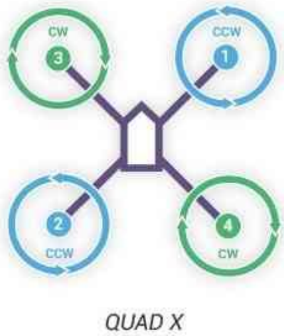

Figure 3. Quadcopter Motor Layouts

Guidance, Navigation, and Control

There are multiple software systems at varying abstraction levels that blend together into one symphony of software for autonomous aerial vehicles. The team built upon last year's experiences as well as online resources to make progress for this year. Several research papers described custom quadcopter flight controller software and, after studying them, the team made an attempt to create a flight controller by implementing an Estimated Kalman Filter, PID Controllers, and various other pieces of software all on an ATMega 2560 with an off-the-shelf IMU. Starting out, this proved to be especially difficult with the young team's lack of general familiarity with such high level concepts.

As it turns out, however, there is a sprawling, wonderful world of pre-made flight controllers and open source firmwares for flight controllers. ArduPilot is the de-facto standard for autopilot firmwares. It is open source, and offers proven solutions utilizing all of the above pieces of math and control theory that were previously researched in one package. ArduPilot was then paired with the wonderful open source flight controller hardware project by PX4, the PixHawk. These two created a robust guidance and navigation framework that could expand upon the team's indoor flight advances utilizing machine learning and optical flow techniques. The Jetson TX2 Development Board is connected to the PixHawk flight controller over USB and sends an angle as a quaternion for the vehicle to perform. Underlying PID loops running in ArduPilot ensure that the copter actually achieves the desired attitude using the built-in IMU on the flight control board as feedback. This, combined with an optical flow algorithm to measure relative ground velocity, allows us to achieve relatively stable indoor autonomous flight.

Flight Termination System

Our competition robot, Sentinel, has several failsafes in place to manually terminate a flight. The first failsafe is an automatic landing mode. This mode can be triggered remotely from a laptop and will immediately enter Sentinel into a landing maneuver. The robot will also enter this mode automatically when the onboard battery voltage has reached a preset minimum value.

The next failsafe in place is a manual takeover by a pilot. If at any point during a flight the robot is determined to be unsafe, the pilot on standby can flip a switch on a transmitter and take full control. This will internally switch the copter from a flight mode that waits for offboard attitude commands to a flight mode that only accepts controller input. From there the pilot can safely land the robot before an accident occurs.

If a more extreme action is needed, both the pilot and the laptop have the ability to kill the motors using the Emergency Motor Stop, colloquially referred to as "the kill switch." This may cause damage to the robot itself, but it gives us the ability to quickly react to any unforeseen events during flight.

PAYLOAD

Sensor Suite

The sensors on Sentinel are meant to guide and stabilize flight to help accomplish the IARC Mission 7. The sensor used to identify the ground robots is the Intel Realsense D415 camera. The D415 has a video camera and two infrared cameras that can generate stereoscopic 3D images. This diversity enables us to utilize the camera in multiple ways on the robot. The Software Division of the team has implemented an approach based on a Convolutional Neural Network that can identify and detect Haar-Cascade like features of objects using both conventional RGB data in an image, and depth data. The RealSense will mount beneath the bottom plate, attached to a gimbal system to allow for forward viewing angles of the arena. A hole in the bottom plate exists so that the camera can see the ground and is protected whenever the robot needs to make contact with a ground robot from above. The gimbal system is custom designed as shown in **Figure 5**. It communicates over a serial connection with an on-board processing unit, the Jetson.

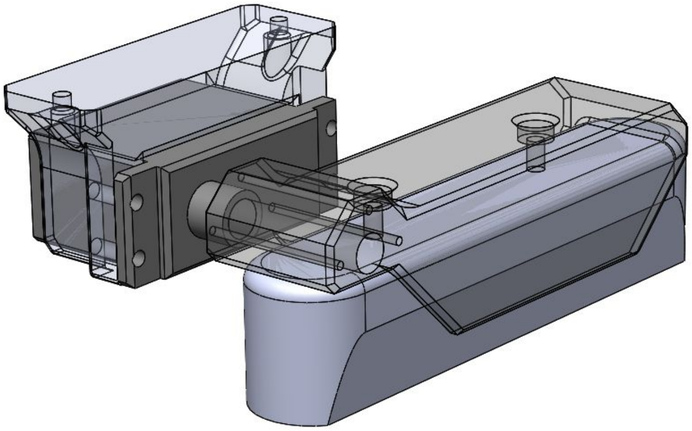

Figure 5. Camera Gimbal Assembly

The Jetson TX2 Development board is a 17cm x 17cm computer with enough processing power to do all the needed calculations and run the game AI. The Software Division has been developing an approach based on the Sum-And-Difference algorithm for Optical Flow provided by the PX4Flow sensor. Optical Flow uses an assumed brightness constancy between differing frames of terrain to determine vehicle velocity and therefore negate drift.

The optical flow sensor will be mounted securely to the top side of the bottom plate of the frame for protection of the sensor, with a viewing hole allowing it to face the ground. The optical flow sensor is meant to track features across frames and generate a vector based on which way the feature has moved from frame to frame in order to detect ground movement. This will allow Sentinel to counter drifting in any particular direction. For altitude hold capabilities, Sentinel utilizes a TeraRanger range finder. The range finder provides incredibly accurate altitude measurements. The rangefinder will also combined with the sensors built into the PixHawk flight board, and run through an Estimated Kalman Filter to eliminate noise. The PixHawk flight controller includes a barometer for altitude estimation, a 6-axis gyroscope for attitude measurement, and a magnetometer for use as a compass. All of the built-in sensors work in conjunction to form a robust IMU.

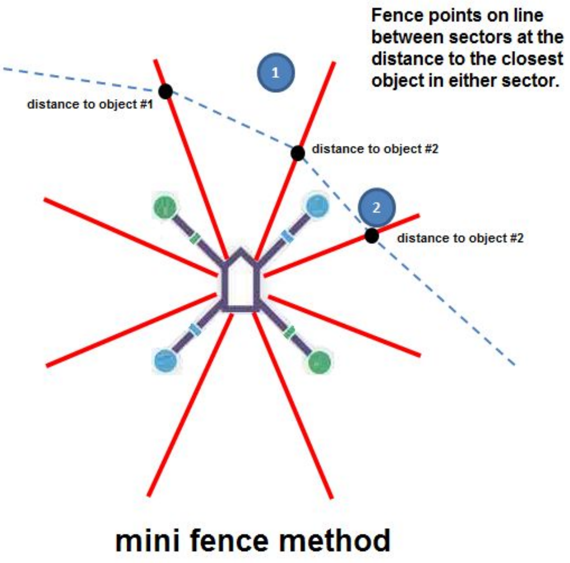

Figure 4. Mini Fence Method for Object Avoidance

Sentinel also utilizes a Scanse LIDAR sensor with 360° field-of-view for obstacle avoidance. LIDAR data is interpreted to first determine a direction within a 45 degree sector around the vehicle as illustrated above in **Figure 4**. The offending sector is then sent to the PixHawk flight controller which takes care of any necessary maneuvers to avoid the obstacle. The LIDAR sensor will be mounted on the top shield by bolts, and will interface with the Jetson.

Communications

The PixHawk communicates with the ESCs using a PWM interface. The PixHawk then provides some relatively high level directives through the MAVLink protocol that the Jetson can take advantage of in order to command the vehicle. The Jetson is typically connected over USB to the PixHawk, but can also be connected using a serial connection. In between this sits the receiver, which is directly plugged into the PixHawk over the SBUS protocol. Some offboard processing will take place. Images and text data may be streamed to a remote workstation over WiFi, with text in a JSON format. The Jetson also communicates with the gimbal over a simple serial connection with text data.

Power Management System

The power on Sentinel branches from two 4 cell batteries placed between the main plates of the body. From there, a centralized lumenier mini power distribution board. All electronic speed controllers are soldered directly to the board, and attached to the Tarot motors using 3.5mm bullet connectors. Also coming off of the main board are two 5V regulators supplying power to the PixHawk over it's Hirose DF13 connectors and the Jetson on pins 4 and 6. Using the 5V regulator that came in the PixHawk's power module, the power drain is able to be monitored. The main receiver is plugged into the PixHawk's RC IN port and the LIDAR sensor is plugged into the I2C for power. For the Jetson, the Realsense R200 camera will be powered over USB Micro B and the servo motors from the gimbal will be powered from pin 2, supplying a 5V output and connected to ground at pin 34. A fully detailed diagram including power and signal connections can be seen in **Figure 6.** With this setup, an estimated 14 minutes of flight time was achieved.

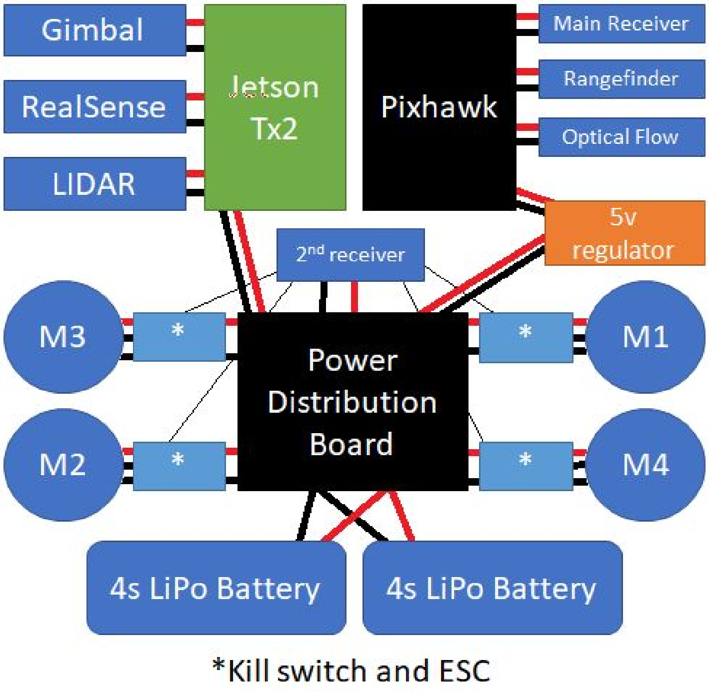

Figure 6. Wiring Diagram

OPERATIONS

Flight Preparations

We have developed a pre-flight check that occurs in three phases: Frame Evaluation, Software Setup, and Situational Safety. Each phase includes a series of specific steps that ensures the aerial robot is operable and safe to fly.

Frame Evaluation Checklist:

1. Check that all wires are fully connected and are correctly plugged in. Their placement should match the diagram in **Figure 6.**
2. Check that all bolts on the frame are tightened properly and that the Jetson and PixHawk are secured to the frame in the formation shown in **Figure 2**.
3. Check that the propellers are properly mounted in the correct orientation, and that the mounting bolts are tightened.
4. Check that the arms are tightly secured and that the motor mounts are level with the frame body.
5. Insert the two 4S 5000 mAh batteries carefully between the plates and fasten them securely.
6. Rotate the propellers manually to be sure that there are no wires or obstacles obstructing their motion.
7. Plug both batteries into the power distribution board.

Software Setup Checklist:

1. Check that the systems are all powered on properly.
2. Using a computer, connect to the robot's wireless network. (The robot broadcasts a wireless network automatically when the Jetson boots.
3. Initiate a remote console connection to the Jetson via the SSH protocol.
4. Perform a motor check to ensure that each motor's rotational direction matches those outlined in **Figure 3.**
5. Ensure that there is a strong WiFi connection between the computer and the robot with minimal spectrum noise.
6. Ensure that the Emergency Motor Stop is functional.

Situational Safety Checklist:

1. Visually check the area in which you are about to fly for any potential obstacles or dangers.
2. Be sure to clear the area of any people or animals that may be in the flight area.
3. Stand at least 15 feet away from the robot and declare that you are preparing for take-off as you arm the robot.

4. Perform a quick test spin of the motors to ensure that they are spinning in the correct duration and to ensure that there are no obstructions by slowly increasing the throttle to about 5%.
5. Declare that you are taking off as you begin to increase throttle, and gain altitude.
6. Maintain visual contact with the robot at all times during the flight, and never take your hands off of the transmitter if it is a manual flight. Always be prepared to initiate an emergency landing or to kill the motors.

Man/Machine Interface

The Jetson on the vehicle runs Ubuntu Linux and can therefore be connected using the SSH protocol from another computer. The Jetson is configured to broadcast a wireless network and uses detachable antennas to make connection more stable. Once connected to the WiFi network, and then a remote terminal session, the MAVProxy software allows for command line control of the vehicle. Multiple ground control stations (GCSs) with GUIs support ArduPilot and the MAVLink protocol as well. For quick interactions, the team typically uses QGroundControl or APM Planner. The software division is working on a custom web-based ground control station with an easy to use interface with support for live streaming video feeds from the vehicle and visual overlays of the robot's computation. The custom web-based ground control station will also have the ability to control custom systems like the vehicle's gimbal and issue remote commands for indoor flight during competition. Implementing a custom GCS is a necessary step as most off-the-shelf softwares do not have support for the custom control systems on the vehicle, or the custom commands the vehicle will need to execute for indoor flight. The custom web-based ground is implemented with the Flask web framework.

RISK REDUCTION

Vehicle Status

The current status of Sentinel is near completion. Structural issues with our landing gear design has sent us back to the drawing board to find a more robust way to handle harder impacts. Finding a more suitable material or different design altogether should give Sentinel the strength it needs. Plans have also been made to create a protective / aesthetic shell for the top portion of Sentinel that would cover any unnecessarily exposed electronics. If time allows, improvements to the propeller guards and vibration dampening will also be accomplished. The LIDAR Sensor has been planned for and has support with the PixHawk's prebuilt obstacle avoidance algorithm. Object detection is currently being reprogrammed for both the green and red roomba plates, and shows a sign of high accuracy.

Shock/Vibration Isolation

One of the complications that arises with propeller driven aerial vehicles is the vibration created by the propellers spinning at such high rates. To combat this Sentinel has several methods for dampening vibrations incorporated into its design. One of these is the shock absorbing pads that the PixHawk flight board rest on. These pads allow the onboard accelerometers to negate the noise that would be created by the vibrations. Additionally, the main body plates on Sentinel are made of a 4 layer 6k density carbon fiber composite.

EMI/RFI Solutions

The vehicle rarely encounters issues with RFI from the RC transmitter to the receiver unless the vehicle is completely out of range. The WiFi connection between the vehicle and a computer can sometimes be spotty. To improve the connection, the jetson connects to a mobile hotspot which then the computer connects to as well. This allows us to have the jetson connecting to the hotspot rather than having to broadcast to the jetson, which was dropping connection in testing. It also extends our testing range because the hotspot can be set up in between the computer and drone. In terms of negating EMI, ArduPilot provides a built in utility that will automatically compensate for compass motor interference. To calibrate the compass for motor interference in ArduPilot, we first reverse the propellers on the vehicle and ensure that it is secured. Then, we arm the motors, and slowly raise to roughly 50% throttle. We monitor the compass readings as the robot approaches 50% throttle and for each tick in throttle, we also note the corresponding change in compass readings. At the end of the calibration, the offset in compass readings per throttle tick is saved to the flight control board as a way of mitigating EMI.

Safety

To ensure the maximum safety of the aerial robot, as well as other objects or people, both physical and encoded safety measures have been put into place. Propellor guards have been designed to prevent the spinning blades from coming into contact with an object in the event of a collision. These propeller guards also enable safe landings to interact with the ground robot's front bumper. Another physical safety design is the wire management. All wires have been routed and secured to prevent any possible collision with the propellers. Additionally, all of Sentinel's wires have a protective PET braided cable sleeves placed around them to both help shield the wires from being damaged by propellers or other impacts, and keep the wires organized together.

The obstacle avoidance system, elaborated upon in the sensor suite portion of this paper, will also be used to avoid any walls, people, or solid objects that the vehicle may encounter during flight.

Modeling and Simulation

Before last year's competition, we focused on making a 2-D simulation for us to visualize the competition. We also attempted to use machine learning techniques to optimize the drone's strategy in the 2-D world, which we were largely unsuccessful at. For this year's competition, we decided to seriously attempt 3-D simulation while continuing to dabble in our 2-D simulation. Early on in the fall semester, while experimenting with the 2-D simulation, we realized that we could score enough roombas to beat the competition simply by tracking them one at a time across the green line, instead of attempting to score multiple roombas at the same time. This made us realize that our strategy for determining the roomba to fly to would not necessarily have to be complex, and bolstered our efforts for 3-D simulation. We began to learn how to use Gazebo, a 3-D simulation software designed for robotics. Over the course of the year, we created 2 gym environments, roombas with realistic meshes, a mesh of our drone borrowed from the hardware division, and we used them in Gazebo. In December, we first achieved roomba following in Gazebo, though not perfectly. In the Spring semester however, development was held up because only one team member in the software division had a fast enough computing to run Gazebo at acceptable speeds. Near the end of the Spring semester, our design team bought a powerful PC designed for testing, and the furthest we got was having the drone land when it saw a roomba driving away from the green line.

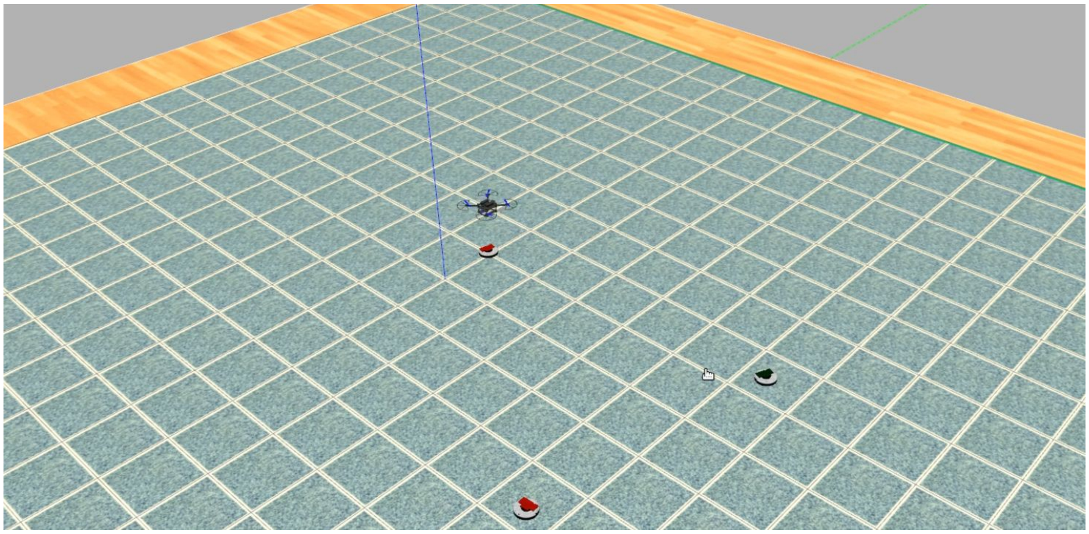

Figure 7. 3D IARC Mission 7 Simulator

Testing

Most of the initial testing for our vehicle took place outdoors in preliminary stages and was very limited due to weather. We then transitioned indoors once we were able to nail flight maneuvers, but it was tough to secure a location large enough. We plan to build a mock arena in our University’s gym for more accurate testing.

CONCLUSION

The documentation from this report describes a viable strategy and aerial robot to complete the tasks for Mission 7 of the IARC. Using both physical and encoded safety measures, the aerial robot designed can be trusted to perform fully autonomous tasks safely. Our approach to hardware is innovative, attractive, and safe. The software approaches put forth in this paper are right in-between bleeding edge and obsolete which ensures that the technology is ready for prime-time and is relatively robust.

REFERENCES

Honegger, Dominik, Lorenz Meier, Petri Tanskanen, and Marc Pollefeys. "An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications." 2013 IEEE International Conference on Robotics and Automation (2013): n. pag. Web. 5 January 2017.

Chao, Haiyang, Yu Gu, and Marcello Napolitano. "A Survey of Optical Flow Techniques for UAV Navigation Applications." 2013 International Conference on Unmanned Aircraft Systems (ICUAS) (2013): n. pag. Web. 5 January 2017.

Eitel, Andreas, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. "Multimodal Deep Learning for Robust RGB-D Object Recognition." 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015): n. pag. Web. 30 Dec. 2016