Autonomous Aerial Vehicles in a Dynamic Environment

Logan Thomure
Missouri University of Science and Technology - Chief Hardware Engineer
Clay McGinnis
Missouri University of Science and Technology - Chief Software Engineer
Cole Dieckhaus
Missouri University of Science and Technology - Software Engineer

ABSTRACT

This paper outlines a strategy of accomplishing the tasks required for the eighth mission of the International Aerial Robotics Competition. Through the use of vehicle communication protocols, voice commands, and computer vision, the mission can be solved. The strategy outlines the aerial vehicle builds that will allow for a viable flight criteria and structural support, along with vision and communication algorithms. Combining these strategies for a mobile phone application with human interaction will allow for a simple, yet effective solution for having aerial quadcopters identify and respond to QR codes, humans, and their environment.

INTRODUCTION

Problem Statement

Mission 8 of the International Aerial Robotics Competition proposes new challenges to teams focused on areas of development that are deemed to be ready for further advancement. These challenges are given in the form of a mission that teams work to complete with fully autonomous, multirotor robots. Teams are challenged to develop autonomous flight as the entire mission is to be performed autonomously, they are challenged to develop obstacle avoidance techniques as enemy robots attempt to attack the human operator, they are challenged to develop human / machine interaction as they use a human operator helping to command the aerial robots, and they are challenged to develop aerial robot swarming methods as their aerial robots work together to search for, identify, and decode various codes. These challenges encourage teams to develop nuanced solutions that are aimed to result in generally applicable designs. Mission 8 has formulated these challenges by providing the task of using up to four aerial robots to assist a human operator in decoding QR code in order to unlock a four digit lock, while avoiding being shot by an enemy robot enough times to be taken out of play. Although the operators quadcopters may heal the human operator, a limited number of heals can be given to the operator.

.

Conceptual Solution to Solve the Problem

To accomplish the problem defined above, four aerial quadcopters have been built to use sensor data to execute actions from autonomous code. By having well-functioning stable flight, the copters are able to perform the following strategies with little error to complete the mission. Using a central, offboard computer, human or computer operator commands are processed and the aerial copters' vision is monitored. The four copters are divided into three different tasks, and they are able to be prepared for switching tasks between runs. One task is identifying and decoding the QR codes found near the bins. Another is keeping the operator healthy by using a laser fired programmatically at the helmet of the operator to trigger a heal, and the last task serves as overwatch, and provide a camera feed of the arena to the operator. This overwatch, at its minimum, will provide the operator with a minimap of the arena, allowing strategic movements to be made to avoid being shot by enemy quadcopters. This also allows for further autonomous improvements through the use of computer vision, allowing relative quadcopter positioning.

Since each copter is able to fly stably, each one will maintain its position until it has sufficient information to help it make an action. These actions are built into the copters and processed onboard, and when no information is available it will hover and wait for override commands from the offboard computer. Each copter has a target, whether it be a QR code, the operator, or a full area of the field. Utilizing computer vision, targets are detected and the copters will guide themselves to these targets and perform their routines. If they veer off course or do not have sufficient information, the operator may correct their course using voice commands and the offboard computer can cause actions to manipulate the quadcopter' course.

Figure 1. Example of the Conceptual Solution

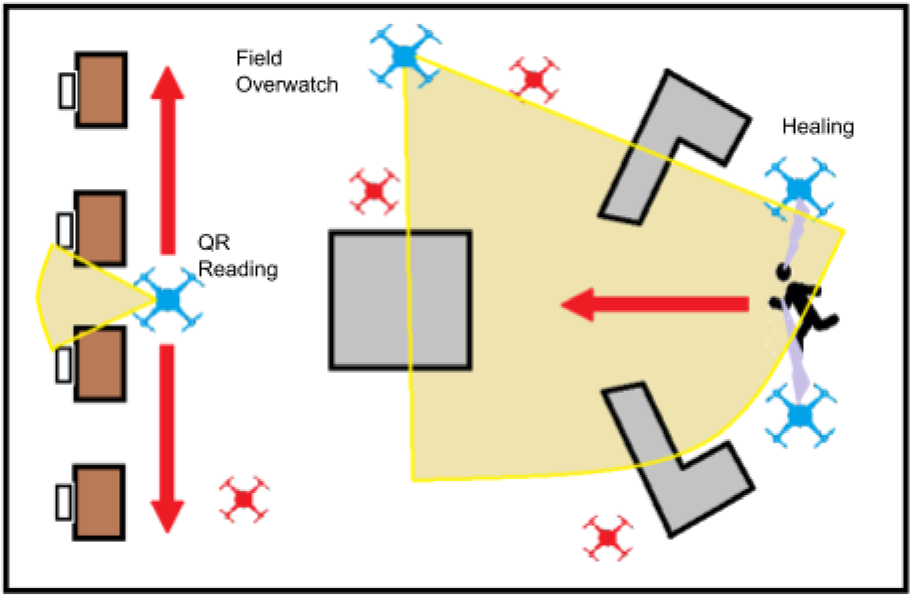
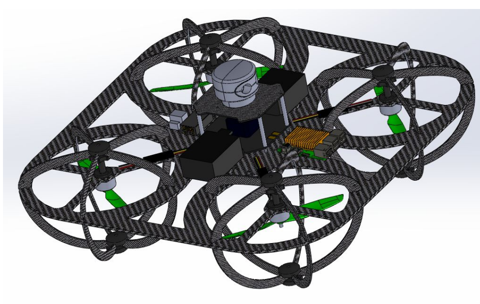


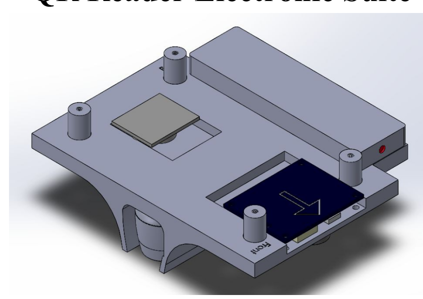
Figure of overall system architecture

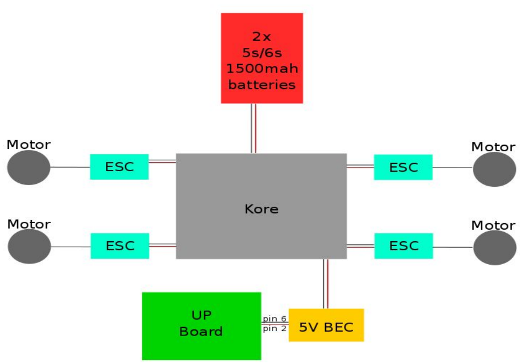
Figure 2. Frame Hardware Layout

Main Frame

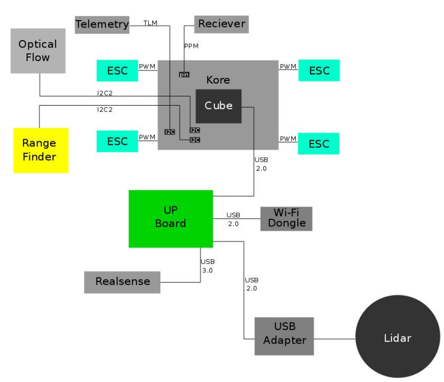


QR Reader Electronic Suite

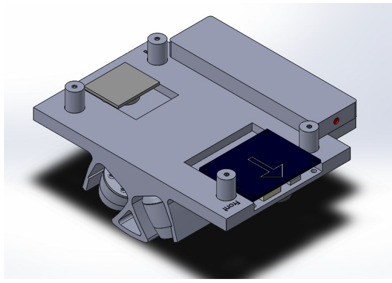


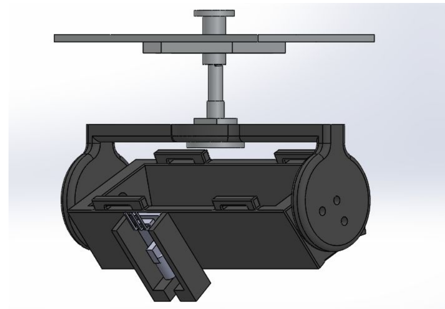


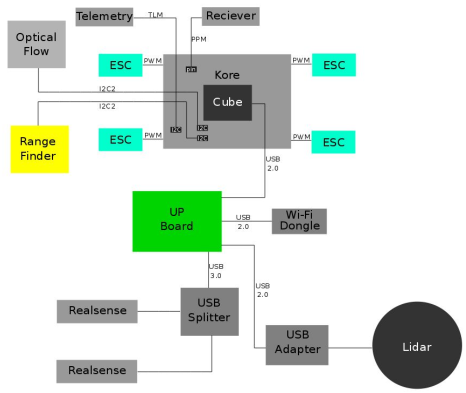
Overwatch Electronic Suite

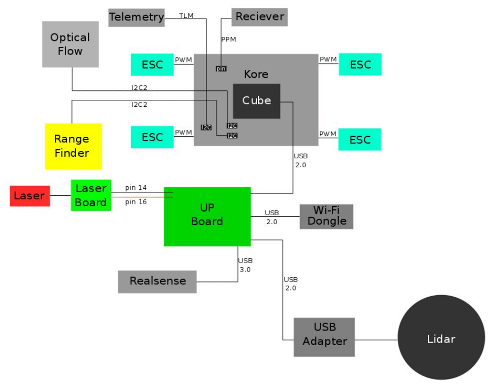


Healing Electronic Suite









Yearly Milestones

With the clean slate provided by a new mission, everything from hardware to software, everything was remade. This involved a full rewrite of the flight code and development of new auxiliary tools to aid in the process of testing new and improved strategies.

Some benefits of the new software came from new documentation, modularization, new testing methods and standards, and the removal of old or deprecated code. These and more refinements lead to safer and more efficient testing strategies. Rather than all tests being run manually on the copters, the tester now connects wirelessly to the copters and sends commands through a remote interpreter or remote script. This allows the team to connect once and test multiple changes. A lot of changes dealt with abstracting the autopilot software into a system that is easier to work with for less experienced, new members. Refreshing the teams perspective between missions allowed the team to create its most advanced and capable aerial copters to date.

Hardware milestones included learning professional designs and manufacturing methods. A complete redesign for all sensor attachments have been made along with the first successful attempt for an enclosed mesh around each propeller and motor system to allow for maximum safety. Methods for creating carbon fiber layups have been improved, and brought to an industry level of quality. This has helped the designs have less epoxy, lowering the overall weight without hindering the strength of the carbon fiber. The team has also incorporated a preflight checklist that covers everything from the hardware to software prior to any flight. Every flight is logged with written notes on the quadcopter's behavior and what changes have been implemented between runs. These logs are saved for the team to use and learn from in future tests.

AIR VEHICLE

Description of configuration/type

For this mission, the use of an X-Frame quadcopter is used as the ideal aerial vehicles. This allows for easy maneuverability in all three physical dimensions, along with the use of different open source, premade flight boards. These flight boards integrate both customized codes and strategies, while also containing the native, premade code for the conversion of movement commands into physical movement via the motors.

Main Frame Design

For the main frame of all the aerial vehicles, a single carbon fiber plate composed of four layers of 3K carbon fiber, with a divinycell core is used. Divinycell is chosen to lessen the overall deflection in the frame, while keeping the weight to a minimum. The overall size of the aircraft is a 22 inch square, using 8 inch propellers, with vertical height of 6 inches. The motors have been mounted upside-down such that the landing gear and propeller guard system can be combined, allowing further weight reduction, while maintaining safety requirements. The landing gear system is made of five, C-strut components, connected in the equally spaced pattern shown in **Figure 2,** and held together by a compression fit between two nylon clamps on either side. These C-Struts are made of two-layer 3K carbon fiber with a 1/16th inch birch core. This core provides strong impact rigidity when landing, without snapping, while still allowing some flex for vibration dampening. A plastic mesh is formed around the outside of the C-struts, with ½ inch holes to impede contact between the environment and the propellers. Lastly, a mount is added on top of the motors connecting the C-struts to the motor, adding rigidity and preventing vibration caused from the motors spinning.

Payload Layout

The Pixhawk Cube has been chosen as the primary flight board, along with the Kore carrier board for power and signal distribution. It is connected directly to an Intel UP board over usb for communication. Two 1500mAh batteries are connected in parallel, supplying the main power directly to the Pixhawk Kore. A 360 degree-spinning LIDAR is mounted above the Kore, with a secondary receiver and relay connected between as an external kill switch. A primary receiver is added to allow manual control by use of a transmitter. A WiFi adapter is connected to the UP Board, and allows the copters to connect to a wireless signal. Mounted underneath, different sensor suites will be attached depending on the specific quadcopter's purpose, as defined in **Figure 1.** Each one will include at least one realsense camera as the primary vision detection device, an optical flow sensor, and a laser range finder. For the QR reading sensor suite, it simply has every sensor mounted directly downwards. The Overwatch copter has two realsense cameras mounted at 55° angle between them to allow the maximum field of view to be possible. Lastly, the heal suite has a camera mounted at an angle of 27.5° with respect to the vertical so that the camera can see in front of it and directly below it. Along with the camera, the addition of the healing circuit board and laser with the camera are attached below the main frame. A diffraction lens is then added to the front of the lens to give the laser the same field of view as the camera, which shapes to a scattered cone with an angle of 55° . Further information on the sensor suites, power distribution, and signal connections can be found in **Figure 2.**

Flight Control System

Navigation / State Estimation System

Using the built-in features of the flight controller the team chose, the Pixhawk Cube, two methods of navigation are specified. Both of these methods will allow the copter to travel in three-dimensional space, but each have their pros and cons for their use case.

The first of these methods is sending velocities in the x, y, and z directions. Using this method, you can control the speed of the craft very precisely, even when expecting to travel long distances. In the current case, this is mainly used for hovering, as giving zero velocity in all directions forces the copter to stop in place and maintain position.

The second method, which is used most often in this strategy, is responsible for how accurate the copters' movement is, as positions are sent to the copters. Positions are simply distances to travel in each plane in each direction. The amount of control granted by using positions is paramount to the effectiveness of the strategy. The Pixhawk Cube keeps track of its estimated error on each movement, which allows the copters to almost always know where it is in real space. The team's networking strategy makes it possible to send commands to the quadcopters which they then use to make positional movements, and it is under some assurance that the copter will only go as far as it is told.

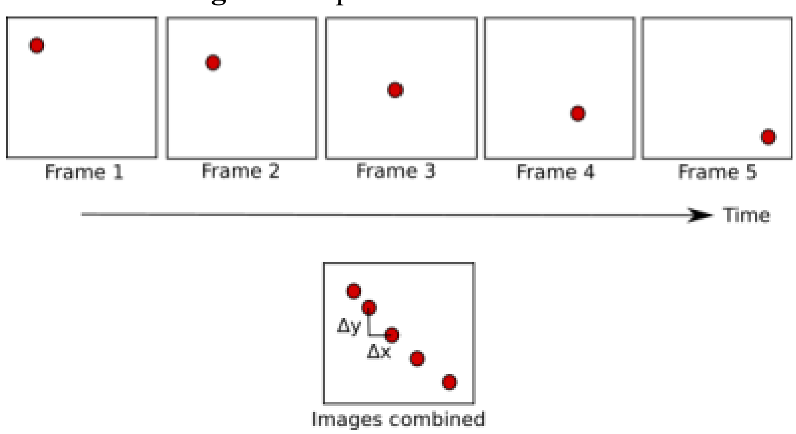
For state estimation, the ground control station is responsible for keeping track of all the aerial robots. It mainly is only aware of the existence of each individual copter unless one of the copters signal that it has come near to another, in which case messages are sent to the ground control station to keep the new state up to date with where the swarm is.

Attitude / Position control system

The Pixhawk Cube, the flight controller used in all of the copters, is key to the successful position control system and its effectiveness. Using a specific selection of sensors, the Pixhawk allows the copters to maintain stable flight with minimal effort. It was built for stability to be achieved with GPS, which is not a good practice indoors, so instead, a lesser-known feature of the Pixhawk is leveraged to replace the GPS with a compatible combination of sensors: Optical Flow, Rangefinder, and Compass. What the GPS does is provide three dimensions of positional data to the Pixhawk, without which is is not able to calculate the copter's estimated position in three dimensional space. This information is necessary for maintaining steady flight and hover. Using Optical Flow, which takes images of the ground below the quadcopter at high frequency, and calculates the distance delta in the x and z direction, the Pixhawk can calculate the error in movement which is accrued over time. This feature of the Pixhawk is nearly plug-and-play, meaning there is not much setup required. To achieve truly stable flight, parameters on the flight board allow the copter to control how much compensation should be expected that the Pixhawk should exert when it finds it has traveled too far, or drifted.

Figure of control system architecture

Figure 3. Optical Flow Overview



Kristoffer Richardsson: "Feature Tracked Over Time"; 6 November 2017, https://www.bitcraze.io/2017/11/optical-flow/

Flight Termination System

Each copter's termination systems may be accessed through three methods: the transmitter, emergency kill switch, and each copter's flight board. The transmitter allows for an emergency motor stop as well as the built in land mode, which safely executes a landing. The external kill switch and receiver has been added to the system as an upper level inhibit, through the use of a relay between the batteries and the main power distribution board. This system will be a last line of defense in the event a termination message to the copters through both the ground station and the transmitter has been unsuccessful. The flight board allows for use of land mode from the software and from commands sent from the offboard computer.

Within the software onboard the copters, a number of checks are in place to maintain that flight does not become dangerous. The operator could only know whether something has gone wrong because of physical behavior, however, in the code, the copters look for a number of conditions: excessive pitch, roll, and altitude, loss of connection to the other copters, and loss of connection to the collision avoidance package. Using these safeguards, the quadcopters will be able to either orchestrate a reconnection or reset of their movement protocol, or make a safe landing. This is to prevent accidents from occuring in the first place.

MISSION PACKAGE

Perception System

Two sensors are used in the perception system for each copter: lidar and Realsense cameras. The lidar is primarily used in the detection of horizontal obstacles. The sensor can only detect things level to the copters. Using computer vision, the offboard computer is able to use the depth and rgb capture features of the Realsense camera to get information about objects beneath the copters.

Target identification and behavior

Two copters will be responsible for collecting images off the iPads displaying QR code segments. First the copters will be instructed towards, and behind, the outermost iPad bins to start, and after reading the codes, towards the middle iPad bins. This will ensure that all four corners of the QR code will be captured. The copters will be directed by using information gathered by the overwatch copter viewing the whole field, or voice commands if the operator wishes to override the autonomous functions. When a copter is assumed to have an iPad in its field of view, the given copter will then detect the iPad by looking for a rectangle that is a different color than the floor, with correct proportions, then, center itself over that rectangle and drop in altitude to capture images. To find the sections of QR codes in the given images, first the colors of the images will be blurred and set to grayscale. Then regions of interest will be found where the code section is expected to lie within, in which, the algorithm will look for edges. These edges will be used to find lines via the PC Lines algorithm. Given that the algorithm only tracked edges where a section of code exists, these lines should represent the borders to each code and will then be used to crop the image down to just the section of QR code. The individual sections will be placed into the correct corners in the final image based off of where they lie in the pictures taken. The final image of the QR code will be read and the resulting number will be sent to the operator.

Operators will be identified using simple color filtering techniques to group sections of images from the healing copter which appear to have predominantly red pigment. This will signal that the operator is wearing the helmet specified by IARC. Using depth data from the Realsense camera, it can also be detected that the helmet is at the expected altitude. The 360° spinning lidar will assist with edge detection of the arena. Using the size of the arena as a reference, the lidar will use the wall distance as a base reference by flying higher than most obstacles in the room, and treating the walls simply as obstacles it will not fly within a predefined safe range of via the obstacle avoidance system of a two meter radius.

Threat identification and behavior

Enemy robots will only be detected using lidar. The reason for this is that the copters have a lack of sensor data, and aside from detecting object below the copters, which they currently do, there is currently no better way to detect horizontal threats. Lidar will be demonstrated most effective against walls and structures, but small aerial robots run a high risk of being in a sensor dead zone. Because of this, the copters will take measures to pool their intelligence and stagger their altitudes. All altitudes will be above the presumed height of the structures used in competition as to avoid a majority of obstacles with little effort. An attempt to identify the relative x-y position of enemy robots from the overwatch copter's perspective will also be done, and thus communicating these positions to other friendly copters. If done successfully, obstacle avoidance can be improved so that no friendly copter will be near the same x-y position of the enemy robots at a given time.

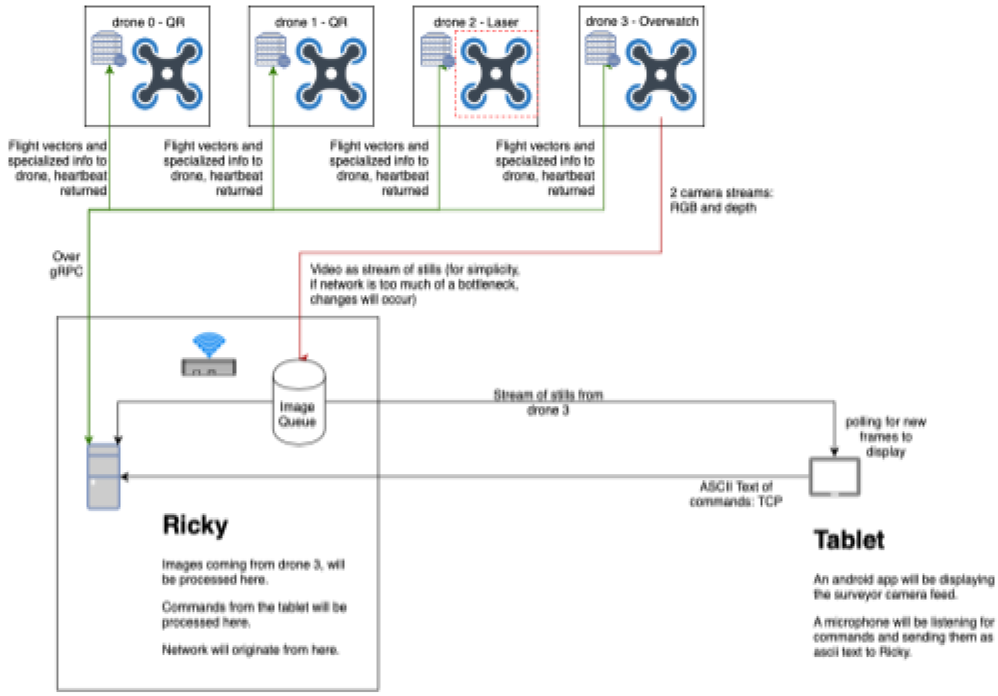
Gesture identification and behavior

For this year's competition, the team did not use gesture identification as a form of control for the copters. It was decided that implementing and using voice control would be simpler than gesture based control, which is a viable replacement because any information that can be communicated with gestures can also be communicated with voice. Although the player's tablet has a built in camera, pointing the tablet's camera at the players free hand requires hand-eye coordination, whereas voice control simply requires holding the microphone to the players mouth, and at times, shielding it from outside nose. It would be possible for gestures to be read by a copter following the operator, head on, capable of interpreting gestures, but a lack of human capital to commit to this added implementation complexity. Implementing voice required only adding voice recognition software to the existing tablet app. This was made easy thanks to a tool called PocketSphinx, which is optimized for on-device voice transcription.

Communications System

The quadcopters are guided by two different mechanisms. The first is an onboard routine, which guides their movement using vision and simple movement algorithms. Second, a ground control station acts as an off-board computer that keeps a complicated link to each of the copters to allow for swarming capabilities. The ground control station is connected to each copter with TCP, a network protocol responsible for web traffic and much more. The reason that TCP was used is because of its innate ability to detect network disturbance and signal the death of any connection. In the case of this competition, a connection failure is very bad news, so having that information available in a fraction of a second is incredibly important. The team's wireless solution is simple, using an offline router connected to the ground control station by ethernet. Ethernet provides an incredible amount of bandwidth to the network link and removes the bottleneck of wireless from one of the links in the network. The 2.4GHz broadcast frequency provides stability at range and suffers from low physical interference. One downside of 2.4GHz that poses a challenge is the lack of available channels to broadcast on, which mean that other wireless networks in the arena could cause interference that may decrease the network throughout. TCP being used for network commands and heartbeats has added a lot of flexibility, but one challenge it had was performance with lots of data due to the implementation used. To complete this competition, it was required to employ computer vision techniques in identifying the operator and QR codes, and so the ground control station needed to get live video streams from the copters. Achieving this with TCP was not practical so in this particular case the frames of video are sent one by one to a service on the ground control station called RabbitMQ.. Frames are then pulled off the queue and processed. This allows the ground control station to receive images from multiple sources with little latency. Aside from the networking strategies used in the arena, using a tool called Ansible, the quadcopters are kept in sync with the codebase. It has allowed the team to test features from different segments of the team's version control system and update their UP Boards. With a tool like Ansible, you can bootstrap any number of machines simultaneously.

Figure 4. Networking Routes



User interface / Man-Machine Interface

As discussed above, the team decided to go with voice as the interfacing technology for many reasons. This allowed the operator to do such things as subject-noun chaining, which allows the operator to specify a copter to perform a specific action. This would have been more difficult with gestures and has allowed the operator to perform many actions. The philosophy behind the voice commands was to allow the operator to make up for any lost functions in the competition. Where the copters may not be able to find the operator if they were on the other side of the arena, the operator can instruct the copters on how to get to the operator and begin to track them. This allows the team to feel fairly safe when in the arena, even if during competition the algorithm experiences low accuracy. Another benefit of the tablet is that it allows the operator to have access to a camera feed from above. One of the copters will be positioned high above the arena such that it will be able to send live video of the three other copters in motion below. Using this strategy, the operator will have more information they can use to help direct the copters and tell the status of progress. The tablet will also be the place where the pin will display once it is solved. Using this combination of features, the tablet should allow the operator to see, solve, and direct the copters in this game.

RISK REDUCTION

Vehicle Design

Propeller guards are fully encapsulated around the propeller with a mesh of half inch squares wrapped around each motor mount. The weight has also been kept to a minimum, which allows low impacts during crashes that allow for little to no damage in the event of a heavy landing or crash.

EMI/RFI Solutions
Through the choice of the Pixhawk Cube, the EMI and RFI have been already brought to a minimum based on the built in sensor protection. This is done through both metallic shielding and a high insertion loss filtering system done onboard ("Cube (Pixhawk 2)).

Shock/Vibration Solutions
Using a birch-carbon fiber core for the landing gear, and a divincell-carbon fiber core for the frame, motor vibrations were dampened throughout the frame to ensure minimal noise for flight dampening. Internally, the Pixhawk Cube has a foam dampening system that helps further reduce noise.

Safety
As mentioned in the vehicle design and the flight termination sections, many safety protocols have been put in place in order to reduce the risk for the operator and environment in the arena and during testings. Each member of the team has also been instructed on flight preparation and basics, and a preflight checklist has been made mandatory for the team. This checklist requires a leads approval before flight and to ensure all physical and software safety features are checked. This includes, but is not limited to, the emergency stop switch, checking wiring and continuity for negative to positive wires, and ensuring that sensors give accurate readings in a ground control software. Preventing issues not related to custom built code has kept software unit testing smooth for problem solving.

Modeling and Simulation
For all structural designs, Solidworks is the base 3D modeling software. This software helps us create information to help estimate design sizes, weights, and manufacturability. By analysing different designs, this software has helped with the final formation of all of the copters designed. Solidworks assembly files are then used to transfer to a collada file via Modo so that further flight simulations can be done in gazebo.

Using Gazebo, which can do three dimensional physics simulations with simulated hardware, the team is able to do flight tests inside the simulation before attempting to fly with new code. This has allowed the software division to make great strides without hardware, and also safer flights overall.

Physical Testing

For the first two years, the team utilized small locations that were never consistent and outdoor locations that proved to have less than optimal weather conditions. Because of this, it was critical that the team found a consistent, close to base testing location that was both indoors and safe. A location that fit this criteria was eventually met in January 2019. This zone was indoors, fully enclosed by a net, and was right next to the team's machine shop. This testing gave the ideal environment with consistent weather and constant access to lighting and electricity. With this new testing location came new testing procedures and protocols. The framework for these new procedures came from a preflight checklist and a written summary of the flight test(s) along with high definition video recordings of each flight. Every Saturday for the entire second semester, the team was in the testing location working diligently to test new developments from the previous week. This combined with the new flight code allowed for rapid development as well as continuous improvement both in hardware and software. Each physical test began with a with these preflight checks and ended with multiple flight logs.

CONCLUSION

Through the use of computer vision, voice communication, and strategic coordination between four aerial robots, Mission 8 of the International Aerial Robotics Competition can be accomplished. This will be done through the use of three different tasks assigned to each quadcopter, while having one extra quadcopter help with the slack of any leftover task needed. These tasks include overwatch, healing, and decoding the QR codes. Although autonomous functions will be built in using vision algorithms, the human operator can assist each of the quadcopters for moving to their destinations or activate different actions by using voice commands. These quadcopters have been custom built and coded with the use of the arducopter library, allowing for a sleek design and smart autonomous flight.

REFERENCES

"Cube (Pixhawk 2)." *Cube (Pixhawk 2) · PX4 v1.9.0 User Guide*, docs.px4.io/en/flight_controller/pixhawk-2.html.