UCR Unmanned Aerial Systems 2016

Journal Paper for AUVSI Student UAS Competition

University of California, Riverside

Figure 0: UAS on landing approach

Abstract

In preparation for the 2016 AUVSI SUAS competition, the University of California, Riverside Unmanned Aerial Systems (UCR UAS) team has designed, built, tested and improved their UAS and electrical systems beyond that of previous years. This team of undergraduate and graduate engineers have developed a UAS capable of fully autonomous flight, image capture and algorithmic image processing, as well as long range communication. A ground station was also developed to communicate with the UAS and provide the ground team with important in-flight data includes an intuitive user interface for mission tasking. All systems have been designed with safety as a top priority.

1 System Engineering Approach

1.1 Mission Requirements Analysis

The development of the UAS required a complete understanding of the mission requirements and the needs of each subsystem to complete mission tasks. The UCR team broke down the tasks according to specific parameters:

Value - The value of the task in the competition
Budget - How much would it cost to attempt
Complexity - Does the team have the manpower for completing it

From this analysis, the team has categorized the mission tasks into three sections. "Will Meet" means that the team has completed this objective in previous years and has further improved on it during this year's testing. "May Meet" means that the team has developed the system for completing the objective in the current year. "Will not Meet" means that the team has concluded that attempting to complete the objective would be a misappropriation of limited team resources.

Will Meet: Autonomous Flight, Search Area, ADLC, Actionable Intelligence, Air-Drop, SRIC, Interoperability
May Meet: Emergent Target, SDA
Will Not Meet: Off-Axis

1.2 Design Rationale

The focus of this year's platform was to have an aircraft that was lighter, cost effective, and capable of longer flight times, and also one that automated most processes. Those focus points were addressed by several changes from last years platform:

- **Readily Available Aircraft** - Last year the team spent a majority of it's time attempting to build and airworthy platform, rather than testing each subsystem. By using, an available platform, the team does not have to worry about turnaround time in the event of a crash.
- **Higher Durability** - With the long repair time of last year's wooden aircraft, the team opted to utilize a PVC and foam aircraft that could take a hit and still keep going with minimal repairs.
- **Automated Software** - A lot of our software last year was fragmented and needed to be manually run sequentially. So this year we emphasized the creation of software the would increase automation.
- **GUI integration** - Due to the amount of time spent last year manually running a variety of different programs and scripts though the command line we decided that it would make for a better user experience and save us a lot of time and stress during the competition to create a simple and easy to use graphic user interface for controlling the payload and ADLC tasks.
- **Internally Mounted Camera** - Given the amount of cameras destroyed in last year's testing, the team decided to use the new platform's payload capabilities by placing the camera gimbal inside the fuselage.
- **Safety** - With a smaller platform, the risk to personnel and the aircraft is significantly reduced.

1.3 Expected Task Performance

Table 1.3.1 Primary mission and objective criteria

Primary Mission Task	Threshold	Objective
Takeoff	N/A	Will Meet
Flight	N/A	Will Meet
Waypoint Navigation	N/A	Will Meet
GCS Display Items	N/A	Will Meet
Landing	N/A	Will Meet
Localization	N/A	May Meet
Classification	N/A	May Meet
Imagery	N/A	Will Meet
Autonomous Search	N/A	Will Meet
Secret Message	N/A	May Meet

Table 1.3.2 Secondary mission task and criteria of objectives

Secondary Mission Task	Threshold	Objective
Automatic Localization	N/A	May Meet
Automatic Classification	N/A	May Meet
Actionable Intelligence	N/A	Will Meet
In-Flight Retasking	-	Will Meet
Automatic Search	-	Will meet
Target Identification	N/A	May Meet
Release	N/A	Will Meet
Drop Accuracy	Will Meet	May Meet
Bull's Eye Delivery	-	May Meet
SRIC Download Task	-	Will Meet
SRIC Upload Task	-	Will Meet
Automatic SRIC Task	-	May Meet
Download & Display Server Info and Time	N/A	Will Meet
Download & Display Obstacles	N/A	Will Meet
Upload Target Details	N/A	May Meet
Stationary Obstacle Avoidance	N/A	May Meet
Moving Obstacle Avoidance	N/A	May Meet

"N/A" - the team has decided to opt for an Objective score and skip Threshold
"-" - there is no Threshold requirement

1.4 Programmatic Risks and Mitigation Methods

Table 1.4.1 Description of risk and mitigation methods

Risk Factor	Description	Impact	Likelihood	Mitigation Method
Insufficient team training	Team members are not capable of sufficiently operating the system	Medium	Medium	Flight tests, ground rehearsals, and documentation are used to practice flight procedure and train new members
Safety Pilot unavailable to attend test flight	The safety pilot is unavailable to attend a test flight with the team	Medium	Very High	Hold a team work day in the lab to further develop and refine the system and algorithms
System Integration Issues	Team is not ready to integrate all parts	Low	Medium	UAS is kept ready-to-fly even without a finalized payload
Plane crash during testing	Impact causes loss of components and/or airframe	High	Low	A backup airframe is readily available and prepped for a component swap
In-flight hardware failure	Component failure causes partial or total loss of control	High	Low	Using dependable hardware from trusted companies; Components are tested before each flight
Plane RTL above pits	Plane dangerously approaches flight team and spectators	Medium	Low	Safety pilot and flight team maintain communication and mission abort capability

2 UAS Design

2.1 Airframe

UCR UAS's strategy this year is to go as light as possible. That is why we chose the Ranger EX airframe. The PVC plastic injection molded fuselage keeps the structural rigidity of the plane while lowering its weight. Furthermore, the deformable material allows the plane to protect the payload in the event of a belly landing or crash. Also, the abundant space in the front section of the plane is ideal for mounting large payloads. The space also allows for ease of access when assembling the payload and troubleshoot any failures.

The wings are a simple Hershey Bar design that provide a high aspect ratio for lower stall speeds which are better for taking aerial pictures at low cruising speeds. The wings are made of EPO foam which is ideal for weight reduction while keeping structural rigidity with a carbon fiber monotube spar connecting the wings. With a 3 Kg load capacity and cruising at speeds of 15 m/s, the Ranger EX is the ideal airframe for the objectives of UCR UAS. Many important modifications were made to the Ranger airframe.

The front section of the fuselage is reconfigured to allow for an internal gimbal. All stock servos are swapped with reliable, Hitec HS-5070MH units to prevent in-flight failures. The ESC was mounted externally for adequate cooling. All interior cables are routed to facilitate access and decrease wear. The landing gear is modified to increase stability and prevent nose-overs. Additional parts were designed, built, and integrated, such as sensor mounts, internal braces, and the egg-drop mechanism.

2.2 Propulsion System

The UAV features a rear mounted Cobra 3515 1100kv motor with a Cobra 60A ESC with an APC 10x5E propeller in pusher configuration. It is powered by a Pulse 4s 6600mAh 35C lithium polymer battery. This provides the aircraft with 4500 grams of efficient thrust.

Figure 2.2.1 From left to right, power pack, esc, motor and propellor

2.3 Autopilot (AP) & Flight Control System

The 3DR Pixhawk, an upgrade from the previously used APM 2.6, serves as the flight controller and autopilot. It supports better firmware, is more redundant, has greater processing power, and supports a wider variety of sensors. Our UAS uses a uBlox Neo6m GPS, a digital airspeed sensor, and a sonar to give accurate navigational information. We use 433 MHz telemetry radios for long range communication between the ground station and the UAV. The GPS and compass unit is modular, and can be swapped at any time, even on the field. It is mounted internally, since the fuselage consists of mostly RF-transparent PVC. Navigational and radio components are placed externally to reduce the effects of EMI from the servos, ESC, and motor.

Early on, fully autonomous missions, including takeoff and landing, were made possible by extensively testing and tuning the autopilot systems. Takeoffs and landings, as well as waypoint navigation, were successful even with crosswinds. Failure modes, such as loss of RC/telemetry link and battery depletion, were planned and simulated.

The safety pilot uses a Spektrum DX9 Black Edition with telemetry capabilities. The independent telemetry feedback provides redundancy to the autopilot telemetry and allows the safety pilot to monitor plane conditions without having to refer back to the autopilot operator.

2.4 Payload System

The following sections will describe the design of the payload system, including the hardware selection process, code structure, GUI and gimbal design.

The payload system is comprised of an on-board computer, EO sensor, and communications radios. We use an Intel NUC as our onboard computer for its powerful quad-core i5 processor needed for capturing and processing high resolution images from the optical system. We use the Flea3 camera by Point Grey as our camera because of its light weight and large high-resolution image sensor. It is connected to the NUC by a flexible USB 2.0 cable and is mounted on a downward facing gimbal in the nose of the UAS.

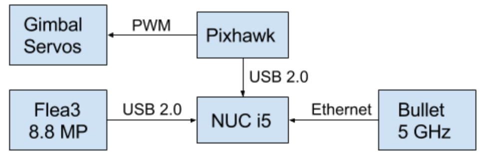

Figure 2.4.1 High level block diagram of the payload system

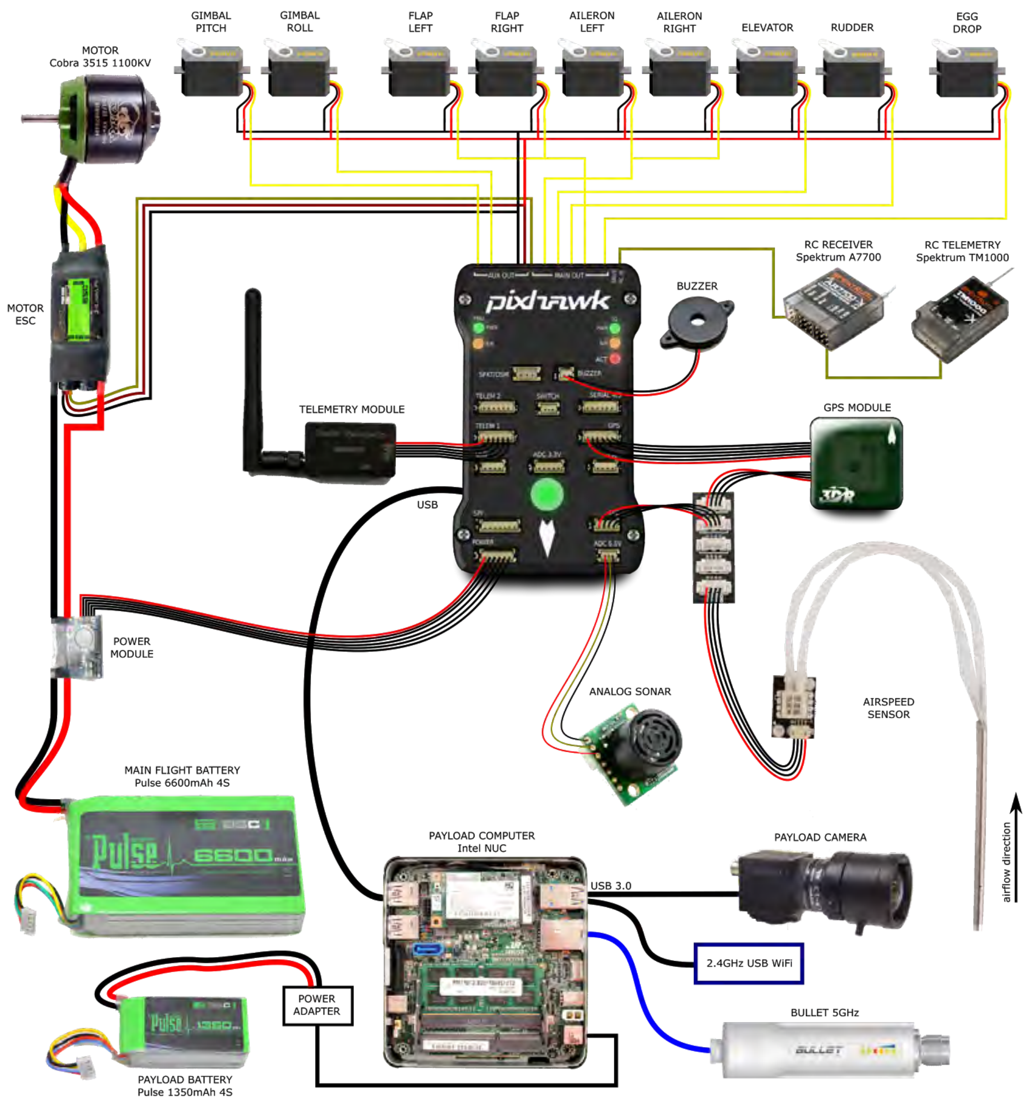

Figure 2.4.2 System wiring diagram

The gimbal uses 2 Hitec HS-5070MH High Torque servos for roll and pitch control. It is designed to minimize vibration and to maintain the camera perpendicular to the ground, ensuring that all images are centered directly under the UAS. The gimbal orientation is controlled by the flight controller. The NUC receives telemetry data from the Pixhawk via a USB 2.0 cable for use in geotagging. To exchange data with the ground station we use a Ubiquiti Bullet 5 GHz Wi-Fi transceiver to connect to our ground station network.

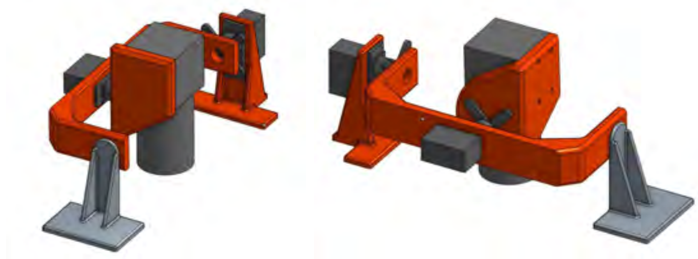

Figure 2.4.3 Gimbal Design

We created a visual interface using Qt5 libraries in C++ to efficiently review images relayed to the Base Station by the UAV and to help use our network in a simple manner. Our interface has three sections, the image viewer, the network widget, and the camera widget.

The image viewer is a window that displays an image starting from the first image received. We can then use the play, next, and previous buttons to cycle through the photos. There are two text boxes to display the coordinates of the target and the target's characteristics. The Network Widget connects our base station to the UAV using the IP address and Port Number. The Camera Widget gives the user convenient control of camera capture, in addition to reporting details about number of images taken and the status of the camera using our custom StarPacket data transfer protocol.

Figure 2.4.4 Payload GUI

2.5 Ground Control Station (GCS)

The ground control station allows the UAS to perform a variety of autonomous missions while still affording the ground team a high level of control over the objectives. In addition, it processes incoming data for analysis. Our focus this year was on increasing the ease of use to minimize complications during a mission. We also intend to increase the modularity of the system for future development and expansion. Our GCS is composed of 4 integral components: flight control, payload control, network control and safety pilot.

2.6 Data Link

For the 2015-2016 competition year, the team has decided to go with a dual data link for the autopilot module. On the autopilot side, the Pixhawk autopilot module is connected to a telemetry radio at 433 MHz. It is also connected to our payload computer via USB, itself wirelessly connected to our ground station via 5.8 GHz. This allows us to connect to the autopilot module both via a direct connection from the telemetry module, and from the network side on the USB module, as shown in figure 2.6.1. The team has decided that this method is very reliable with very minimal latency loss. In fact, connection via 5.8 GHz proved to be more reliable than the telemetry module.

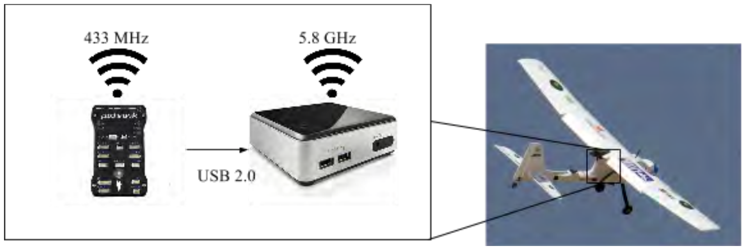

Figure 2.6.1 High level block diagram of data connection

The 433MHz telemetry module provides a maximum of 57600 bps, while the USB 2.0 connection to the autopilot module allows for 115200 bps, which is double the bandwidth. This allows us to receive the telemetry updates as requested in a fast data rate, and helped us achieve the 10 Hz objective threshold for interoperability UAS telemetry upload. The dual data link connection was also selected as a failsafe option. In case the onboard uav computer decides to die, which is highly unlikely, we always have the secondary, telemetry connection via 433 Mhz.

The primary data link for the imagery team consists of the 5.8 GHz connection, which is also shared with the telemetry stream. We reported good results, with latency of 30 Mbps, less than 5 ms latency at 1000 ft away from the ground station. The data link allows our images to be transferred back in less than 1 second, with enough bandwidth for the telemetry side also. The 5.8 GHz data link has proved to be reliable and little to no dropped packets are observed.

2.7 Development Process of Specific Mission Tasks

- ### Autonomous Waypoint Navigation

Before the team attempted autonomous mode, we flew the airframe in full manual mode with simulated payload weight. The autopilot was not installed in the airframe until the safety pilot deemed the aircraft safe and suitable for our competition payload configuration and weight. The next step was to install the autopilot module, or in our case, the Pixhawk. Along with the Pixhawk, we placed simulated payload weights in the aircraft to keep the weight consistent.

The first step for tuning involves flying the airframe almost acrobatically. The Pixhawk autopilot system learns the requested rolls, yaw and pitch values and automatically applies the parameters for the best response time. After the roll, pitch and yaw parameters were tuned, we moved on to tuning the energy controller of the aircraft. This allows for very tight and sharp turns, and energy efficient flights.

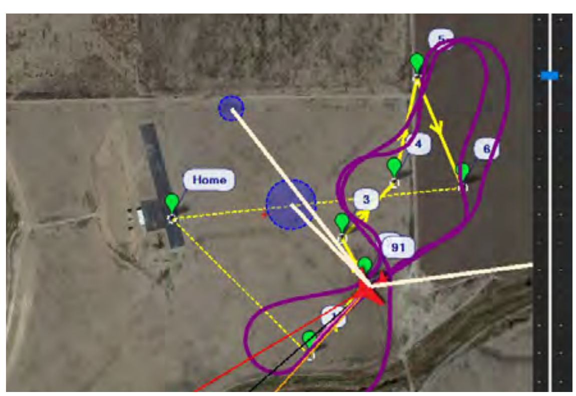

Figure 2.7.1 Waypoint navigation, interoperability obstacles and directed lines drawn to interoperability

The above figure 2.7.1 shows our airframe in autonomous mode, navigating the same waypoints in a loop. We set different parameters until we settled on a set of parameter we felt was safe for the airframe without compromising performance. A few parameters we had to consider was maximum bank angle, maximum requested roll, max pitch and minimum pitch, and the navigation turn rate. Each parameter was changed one at a time, and involved a lengthy tuning process. This is to ensure that a change in one parameter did not affect a change in another parameter. After each loop, parameters were saved and discarded based on which one we decided was necessary.

We set a conservative 50ft turning radius and 50 degree turn bank angle for navigation. A maximum of 35 degrees and minimum of 25 degrees was requested for pitch. This allows for precise waypoint navigation, which was further tuned by decreasing the waypoint radius. The waypoint radius is set to 16 feet, which is much less than the threshold of the 50ft in the competition. During testing, our airframe did not deviate off the course of a planned mission due to wind gusts. We are confident our airframe navigates waypoints autonomously and accurately.

- ## Autonomous Takeoff and Landing

Autonomous takeoff and landing was not attempted until after the aircraft was fully tested and flown with competition payload weight, and after the waypoint navigation tuning was complete. A rigid testing methodology was set to test autonomous takeoff, as this is one of the more critical parts of the mission. A few parameters were measured and set, including many measurements of takeoff speeds, take off angles and throttle input. On the ground, the turning radius of the plane was set to enable autonomous taxi on the runway.

In order to take off, the safety pilot reported about 75% throttle is needed for takeoff. This was verified in the logs, along with a takeoff pitch of 20 degrees, and 30 mph taxi speed before takeoff. The elevator is in full up position until the autopilot module measures a 30 mph threshold. After the threshold is met, the plane takes off and keeps heading straight at 20 degrees until a desired altitude is met. In order to read the airspeed accurately, we used Pixhawk's airspeed sensor.

During one of the first autonomous takeoffs, the pilot always had full manual override. There was one instance during autonomous takeoff where the aircraft veered dangerously close to a wingtip and the safety pilot decided to intervene. Back on the ground, the autopilot team verified that all the parameters were normal, but the mechanical team found the tail wheel to be bent. A quick fix was corrected the problem and testing resumed.

After more than 10 autonomous takeoffs, the autopilot team and the safety pilot decided the autonomous takeoff was safe. The next item on the list to be tested was autonomous landing. Autonomous landing was trickier to test than autonomous landing, and the safety pilot had to use manual override a few times.

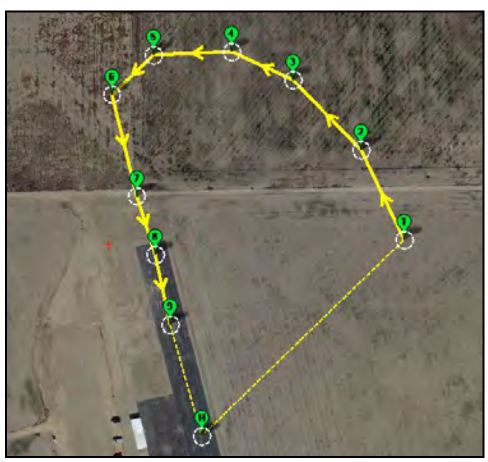

Figure 2.7.2 Autonomous landing sequence scenario with wind from south to north

During autonomous landing, a few scenarios were tested. Depending on the weather during the moment, more specifically the wind direction, the autopilot team created different flight plans to land north to south, or from south to north on our airfield. The figure above shows the following scenario: landing from north to south. Each of the waypoints serve a purpose from guiding the plane and lowering the altitude. It is impossible to use just one command and tell the autopilot module to land the plane. If that were the case, the plane could come in at any direction and potentially, land behind the flight line where spectators are.

We utilized the MaxSonar-EZ1 for accurate distance readings above the horizon. This allows for very precise flap deployment and flare procedure. The first landing procedure was achieved using the default parameters. However, the plane seemed to be in the glide slope for too long and overshot the runway, and the safety pilot had to take over. While in the air, the safety pilot changed a few parameters to lower landing altitude via waypoints, and increase flap percentage as well as landing angle. Our final parameters included 100% flaps at 3 meters above the ground and a nose up pitch of 5 degrees. For every autonomous landing, we also had autonomous takeoffs which we were monitor closely for any deviations. After more than 10 autonomous landings, the team felt comfortable and concluded testing for autonomous landing. Autonomous landing and takeoff procedures were completed on February 20, 2016.

- ## Image Processing (ADLC)

In order to reduce bandwidth on the network, image processing starts on-board computer in our system. We use a residual saliency algorithm written in C++, using OpenCV, to find regions of interest and crop them from an image. This process allows us to reduce the total amount of data transferred over the ground link by as much as 90%.

The residual saliency algorithm works by analyzing the log-spectrum of an input image, extracting the spectral residual of an image in spectral domain, and constructing the corresponding saliency map in the spatial domain. This filters out elements of the image that occur frequently while simultaneously allowing us to amplify features that deviate from the norm.

In previous years, we relied on a color gradient-based method. However, that method, while reliable, was computationally expensive and returned an excessive amount of false positives. This new algorithm is far less computationally expensive and promises to return far fewer false positives.

Figure 2.7.3 Images cropped by residual saliency as regions of interest, from a data set of images taken during flight tests and previous competitions.

After images are sent back to the base station they are segmented using a shadow and highlight invariant color segmentation. The shadow and highlight invariant segmentation algorithm is the first stage in automatic target recognition pipeline. Its purpose is to return separate images of a cut out of the shapes and the letter for use by the algorithms further down the pipeline. The algorithm uses simple color thresholding in the HSV color space to find initial seed points for the a region growing algorithm. This algorithm helps to reduce the effects of changes in color representation due to dynamic lighting conditions. However, the algorithm is limited by the resolution of the image captured. For our algorithm we found that we need approximately 4 pixels/inch to distinguish the letters.

Figure 2.7.4 Cropped images after running image detection

This algorithm transforms the cropped image into a binary image, and then outlines the convex hull of both the letter and the shape. The convex hull helps to reduce the effects of under segmentation and produces a much cleaner segmented image for the shape recognition process. The segmented letter is passed to a letter classification algorithm that uses Tesseract OCR. Which is an open source optical character recognition library.

Afterwards, we run the images determined to have characters through a fuzzy-logic shape detection algorithm. This algorithm uses known ratios of geometric features for a given shape as the features that are provided to the classifier. For the classifier we use a simple fuzzy recognition algorithm. The fuzzy recognition algorithm relies on thresholds and decision trees to distinguish between multiple sets of shapes.

The final process in the recognition stage is the color segmentation which is done by creating a histogram of the image, masking out the background, and taking the peaks. Localizing the target is taken care of by getting telemetry data from the Pixhawk and EXIF tagging those coordinates into the original image when it's taken. When the image is cropped, all images that come from the parent image are tagged with the same geocoordinates.

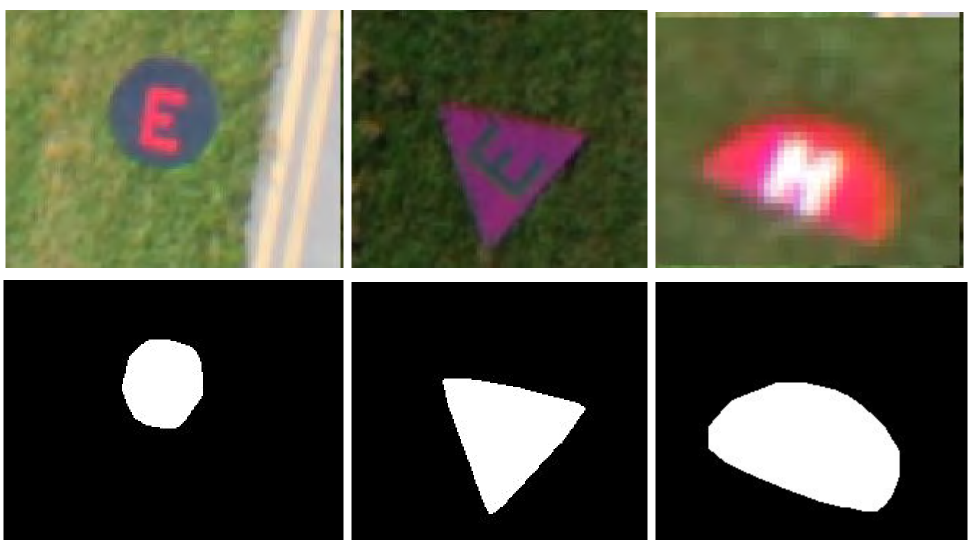

Figure 2.7.5 Image processing using color segmentation

- Interoperability

The interoperability team would like to give a huge thank you to the AUVSI SUAS organizers for providing interoperability sample code for everyone to use. We utilized the Mission Planner and clientproxy scripts and custom modified them to fit our needs. We added API calls to obtain server time, stationary and moving obstacle data at 15 Hz. The interoperability team decided to use the 2D map on Mission Planner's interface as shown in figure 2.7.1. Because the interoperability obstacle also has a height component, the height of the obstacle is drawn on the map and is toggled on or off based on the closeness of the uav to the obstacle. We achieved good datalink with the interoperability server and are requesting and drawing the interoperability obstacles at 15 Hz without any issues.

Uploading UAS telemetry data to the interoperability server proved to be trickier. The first GPS module we used only supported 5 Hz update rate and was verified in the evaluate team csv output. The team made a decision to upgrade the GPS module to the newer uBlox m8n module, which supports up to 18 Hz navigation rate. We used 10 Hz update rate for accuracy and compatibility with the Pixhawk flight stack.

The interop client was also tested using software in the loop. A Linux machine running Ardupilot code was used to simulate real flying conditions. From the simulated flight, we observed telemetry rates from 9-11 Hz.

- Sense, Detect and Avoid (SDA)

As shown in figure 2.7.1, we calculated the distance away from every obstacle, and drew the distance from the center of the obstacle to the current position of the uav. The basic algorithm is as detailed. The heading of the plane determines which direction to avoid the obstacle. For example, if the uav is traveling north into an obstacle, and the next waypoint is to the north east, the uas “dodges” to the right. We utilized the built in guided mode of the Pixhawk firmware, essentially “dragging” the uav out of the obstacle’s path. Guided mode is a “click-to-fly-here” mode, which turns into loiter after reaching the desired GPS coordinates. Because we do not actually want to loiter, we had to set the waypoint far enough, then resume the pre-existing mission by switching back to autonomous mode.

However, if there is enough room to increase/decrease altitude, the algorithm will change altitude accordingly and descend to its previous altitude.The algorithm always checks to make sure the plane has enough distance to ascend and descend based on our climb rate and speed, that the uav will always be within the constraints of a minimum altitude of 100 ft and maximum altitude of 750 feet. The algorithm will not request a huge increase in altitude in favor of just turning to avoid an obstacle.

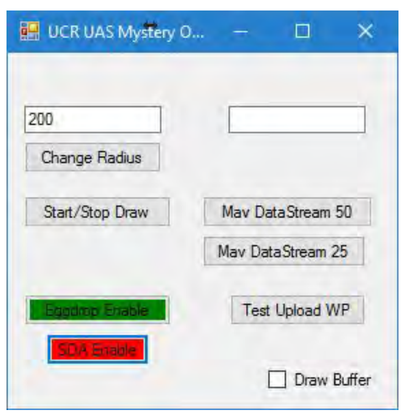

Figure 2.7.6 Custom GUI to enable and disable eggdrop, SDA, and various other parameters. Green means enabled and red means disabled.

All of the above was tested in the software simulator rigorously. The risk for SDA is one of the highest in this competition, and extra precautions were set to make sure SDA is functioning properly. Software overrides are in place to enable and disable SDA. As shown in figure 2.7.3, the SDA enable button is off in red, and is only enabled when the autopilot operator presses the button. At any point, the autopilot operator can press the SDA enable button to enable or disable the SDA algorithm. The color changes from red (disabled) to green (disabled). The safety pilot also has a dedicated switch to take full manual control from the autopilot and disable SDA remotely.

Real life testing was very similar to software testing. The SDA code worked as requested, and the SDA team is happy to demonstrate SDA to the judges during the competition .

- Simulated Remote Information Center (SRIC)

The SRIC communication is handled by a usb 2.4 GHz Wi-Fi radio connected to the NUC payload computer. The stock 4 dBi dipole antenna is oriented inside the fuselage so that the signal strength is constant when the aircraft is loitering above the SRIC ground station.

We configured the Linux NetworkManager service to continuously attempt to connect to the SRIC ground station. A script uses the *pftp* utility to upload and download the SRIC text file once it detect the connection. We also have the option to initiate the transfers manually.

- Air-Drop

To implement the air-drop, we used an effective but rather simple design. Attached to the bottom of the plane is a servo and a 3D printed mount. In between the mount and the servo there is a servo that keeps the bottle snug with the bottom of the plane. We discovered that this setup was slightly unstable, particularly along the axis of the plane. So we decided to re-evaluate the mount to increase stability. When the servo is triggered, the first strap quickly retracts allowing the projectile to easily fall from the plane.

2.8 Mission Planning

The autopilot team is in charge of mission planning for the uav. For takeoff and landing sequences, the direction of wind is a factor in deciding the direction of takeoff. As for capturing the waypoints in sequence, there really is only one set of waypoints to be uploaded. We assumed the SDA is working as appropriate and the UAV will not deviate more than 100 feet from the desired waypoint path.

Further mission planning is conducted after the search grid is commenced. A standard search grid of 100 feet distance between adjacent waypoints is created, and images are captured. If imagery team wishes to fly at a higher or lower altitude for a wider/narrower field of view, the autopilot team easily re-tasks the UAV to fly at such altitude. Most of the missions were pre-made and uploaded the day of during the competition. Immediate tasks, such as emergent target, requires in flight re-tasking. Higher weight is placed on primary objects, such as capturing waypoints in sequence. Secondary tasks, such as SRIC, and eggdrop are weighted lower on our mission planning decision making. For example, if the time it takes to complete SRIC surpasses 2 mins, we have no choice but to abandon SRIC and move on to other objectives. The autopilot team lead keeps an eye on the mission timer and decides when to move on to the next objective. As shown in table 2.8.1, each task will be terminated after the pre-allocated time. We estimated the total mission time in the air will not exceed 16.5 minutes.

Table 2.8.1 Mission planning termination times

Task	Terminate after time (minutes)
Waypoint Navigation	4.5
Air-Drop	2.0
Search Area	6.0
SRIC	2.0
Emergent Target	2.0
Maximum Total Time In Air	**16.5 Minutes**

3 Test and Evaluation Results

3.1 Aircraft Performance and Airworthiness

Our aircraft performs very well in flight. Payload, flight time, glideslope. The plane's specifications state that its max carrying capacity is 4kg. Our plane weighs in just over that at 4.1kg, but this has had no noticeable effect on our aircraft's performance. For each fully charged battery, our plane is able to fly for about 20 minutes. It is outfitted with reliable and tested metal gear servos, as opposed to plastic servos.

3.2 Mission Task Performance

Table 3.2.1 Achieved primary mission task performance

Primary Mission Task	Threshold Achieved	Objective Achieved
Takeoff	Did not attempt*	100%
Flight	Did not attempt*	100%
Waypoint Navigation	Did not attempt*	100%
GCS Display Items	Did not attempt*	100%
Landing	Did not attempt*	90%
Localization	Did not attempt*	100%
Classification	Did not attempt*	100%
Imagery	Did not attempt*	100%
Autonomous Search	Did not attempt*	100%
Secret Message	Did not attempt*	N/A

Table 3.2.2 Achieved secondary mission task performance

Secondary Mission Task	Threshold	Objective
Automatic Localization	Did not attempt*	
Automatic Classification	77%	11%
False Alarm Rate on Classification	23%	23%
Actionable Intelligence	Did not attempt*	100%
In-Flight Retasking	N/A	100%
Automatic Search	N/A	100%
Target Identification	100%	
Release	Did not attempt*	100%
Drop Accuracy	80%	40%
Bull's Eye Delivery	N/A	10%
SRIC Download Task	N/A	100%
SRIC Upload Task	N/A	100%
Automatic SRIC Task	N/A	100%

Download & Display Server Info and Time	Did not attempt*	100%
Download & Display Obstacles	Did not attempt*	100%
Upload Target Details	Did not attempt*	60%
Stationary Obstacle Avoidance	Did not attempt*	70%
Moving Obstacle Avoidance	Did not attempt*	70%

*The team has decided to opt for an Objective score and skip Threshold

3.3 Autopilot System Performance

- ### Waypoint Navigation

Waypoint navigation varied significantly between different flight plans. If care went into plotting waypoints, with significant altitude changes and direction changes minimized, the plane would have a high success rate with hitting waypoints. Likewise, if the plotted waypoints involved radical changes in direction and altitude, the plane would have more difficulty navigating to the waypoints, and our success rate dropped. Auto takeoff usually worked well regarding waypoints. Auto landing, however, was more difficult. The plane would either come in too steeply for landing and would be saved by the safety pilot, or would not be low enough to land without running out of runway. Large amounts of flight tuning was done in order to make landing more reliable and consistent. We also added sonar to increase reliability in landings.

- ### Autonomous Search Area

During autonomous search area testing, the autopilot system performed exceptionally well. Because our airframe was tuned well and our waypoint navigation did not deviate from our flight plans, we were able to accurately geotag your images within 20 feet accuracy. Autonomous search area consists of full autopilot control, and we had the ability to retask the autopilot mid flight. If the imagery team requests a second pass over a questionable target, we were able to retask the plane to fly to the target area quickly and efficiently. We are confident that the autonomous search area meets the objective.

- ### Interoperability

Interoperability is one of the more critical components involving the ground control station and communication with autopilot module. For this reason, the interoperability client was always piggy backing on a mirrored data link from the primary unmodified Mission Planner build. This was to insure that if the interoperability client crashed, the main communication link with the autopilot module would not be lost. There were very rare occasions during the alpha builds of the interop client integration with Mission Planner where the code would crash the entire Mission Planner build. However, due to the isolated setup of the interoperability client, the critical communication link between the uav and primary Mission Planner computer was not lost. After debugging code, the interoperability computer is now the primary Mission Planner computer. The system now has an excellent track record of stability with no crashes so far. Server information, time, and obstacle data are all being retrieved and displayed at 15 Hz. Due to hardware limitation in GPS, the UAS telemetry data is uploaded at 10 Hz.

- ### Sense, Detect and Avoid (SDA)

Rigorous testing was done on stationary obstacles, and are avoided at 90% success rate. Further fine tuning is needed to avoid the stationary obstacles at 100% success rate. However, the moving obstacles are proving more of a challenge. Using the stationary obstacle avoidance algorithm, the moving obstacle avoidance rate is about 50%, depending on the direction of the moving obstacle. A moving obstacle algorithm increased this avoidance rate to about 70%, which can still be improved at this time of writing. One of the biggest factors for the success of dodging moving obstacles involves the ability to predict next location of the moving obstacle and the current path of the UAV. A more nimble UAV would also make SDA an easier task.

- Simulated Remote Information Center (SRIC)

After verifying operation on the ground, the SRIC task was tested in the air by loitering the UAV at a 200 ft. altitude above a simulated SRIC station. The network was detected and connected to within 10 second of the UAV entering loiter. The link speed was tested by uploading a large image file to the server and was determined to be 344 kbps. Upload and download of the SRIC text files were subsequently attempted and, given their small size, finished practically instantaneously as expected.

- Air-Drop

A kinematics calculation was done to determine the rough estimation of the release time of the projectile. The calculation was done with the set altitude to be 400 feet above the ground and with a cruising of 49 ft/sec. This lead to the estimation of projectile to be released 364 feet before the target while spending 4 seconds from release to impact. We were unable to account the drag of the projectile through the air and the cross winds that may change the trajectory of the projectile; however, the weight of the projectile in comparison to the cross sectional area is significantly higher thus any drag and/or cross wind will have minimal affect to the projectile's trajectory.
From the initial testing, we determined that the UAS had to be tracking straight and not be in a turn as it approached the drop point. Several attempts yielded drops that were inconsistent and inaccurate. After further tuning, were were able to get consistent drops within 5 feet of each other.

3.4 Payload System Performance

- Camera

Last year, we had many issues with the camera, from broken lenses to randomly-changing settings. This was mostly due to the camera SDK's poor documentation as well as the limited support for the Debian operating system that we were using. This year, Point Grey released updates and improvements for their drivers, SDK, and documentation, leading to increased camera performance. Despite the improvements, using this camera has been an ongoing struggle since all of the settings need to be set manually. Ultimately, we have it taking clean images and have a list of setting values that we can assign it manually, depending on the lighting conditions and height that we are flying at. We also have a backup point and shoot camera that we have been working with to swap out in the case of a malfunction. We also intend to use it to replace our current camera for next year as it provides an easier coding interface for new team members to learn on.

- Data Transfer

Our network is designed to facilitate communication between the base station and the UAS. Using our custom StarPacket protocol, we send vital status and control information to the Camera and ADLC systems from our Base Station. To achieve an acceptable transmission success rate we constantly stream packets at 3 Hz. To ensure total data transmission accuracy we have redundant bits embedded in the message as well as a BitDebounce algorithm that looks at every 3 messages to test for errors. With this method we have achieved a consistent 100% data transmission rate for our custom StarPackets.

● Automatic Detection, Localization and Classification (ADLC)

Salient Detection

The primary purpose of this algorithm is to filter out redundant or unnecessary information and only send back relevant information i.e. the targets. Therefore, main criteria by which we judge the performance of our salient detection algorithm is the false negative rate. In order to test this we used a dataset of 26 images each one containing a target for recognition. Preliminary results show that we are achieving a 44% false negative rate. However, with further tuning and processing we are confident that we can decrease this number further.

Shape recognition

Out of a dataset of 26 images, each containing a target, our shape segmentation algorithm finds and segments the target in 9 of them. Although this seems like a low number, the purpose of the segmentation algorithm is to both prepare the image for classification and to reduce the number of false positives received from salient. The algorithm ignores black, white, green, and yellow to vastly reduce the number of false positives. Our shape recognition algorithm correctly classifies 7 of the segmented images, giving us a False Alarm Rate of 23%. We expect to be able to achieve this objective by the time of the competition.

Color Detection

Out of the previously described data set, our color detection algorithm correctly identifies the color of the shape and letter of 7 of the 9 segmented images, giving incorrect results on 23% of the segmented images. The incorrect results are due to the fact that the algorithm has difficulty with classifying green targets. However, with a little more tuning and preprocessing, the rate of incorrectly classified targets can certainly be reduced. We expect to be able to achieve this objective by the time of the competition.

Letter Recognition

Preliminary results show that Tesseract-OCR is not reliable for our purposes. Using the same dataset of 26 images we were only able to achieve a recognition rate of 11%. This large error is due a lack of rotational invariance as well as a high sensitivity to noise. In order, for us to achieve this objective, new algorithms will have to be investigated that can account for these error sources. We do not expect to be able to achieve this objective by the time of the competition.

4 Safety Considerations

Given current regulations and protocols governing the use of Unmanned Aerial Systems in the national airspace, the UCR UAS team has developed a thorough checklist and rules of conduct to ensure the safety of the team personnel, and for safe flight of the test aircraft.

4.1 Operational Safety Criteria

There are many different aspects of the aircraft. All team members must work together to verify that everything is functioning properly and in a predictable manner. If any situation arises, the safety pilot is to be notified immediately. Unmanned Aerial Vehicles must be dealt with as machines. They are very complex and are capable of causing serious dismemberment and/or death. The UCR UAS team executes extreme caution when around aircraft.

In addition, the propeller is always removed from the aircraft when not ready to takeoff to prevent accidental spool ups and injury.

4.2 Design Safety Criteria

In order to successfully develop a fully functional unmanned system, emphasis on safety was essential. During the design, manufacturing and verification of the aircraft, continuous risk assessments were carried out in order to minimize possible harm to personnel and equipment. These assessments dictated design restraints and guided team operational policies, which enabled a safe and reliable operation of the system.

Data link also showed a safety design criteria. The UAS is designed to have redundant telemetry data links. In case one data link fails, we have a quick and immediate method to hop onto the active transmitting telemetry link, as shown in figure 2.6.1. We also designed the data link so there is no noticeable latency change from switching from one telemetry stream to the other.

4.3 Safety Risks and Mitigation Methods

The safety pilot must use proper judgment and common sense in order for safety to be ensured. All checks are executed at the discretion of the safety pilot. If the safety pilot declares the situation to be unsafe, flight tests will not be conducted under any circumstances until hazards are dealt with. It is only until then that flight testing may continue.

In addition to personnel safety, the UCR UAS also has to take special care when sharing its airspace. The team's test site is located across from a freeway from a skydive facility which regularly has aircraft taking off and landing. In addition, the UCR test site is on approach for the March Air Reserve Base. There are frequent landings of large scale aircraft such as C-17s and KC-135s. Whenever a full scale aircraft is seen approaching, the team immediately halts the mission and the safety pilot brings the plane in for an emergency landing.

During loss of control from the RC transmitter side, the autopilot module instructs the airframe to circle for a short failsafe session of 30 seconds as the team attempts to regain RC linkage with the airframe. After 30 seconds of no RC transmitter connection, the uav returns to launch and the autopilot team can instruct the uav to land if necessary, as shown in table 4.3.1.

Table 4.3.1 Safety Risks and Mitigations

Contingency	Risk	Mitigation	Action
Telem Link Loss (<3 s)	Low	Telemetry antennas are secured to the fuselage. Ground side telemetry antenna is highly directional and is always pointing at the plane.	Plane will automatically return home after telemetry link is lost for more than 30 seconds
Telem Link Loss (>3s)	Medium		
Remote Control Link Loss	High	Distribute the antennas and satellite receivers for maximum signal strength.	Plane will return to home after 30 seconds.
Bad Compass Health	Medium	Connections between Compass module and AP module are secure and snug. No compass means loss of accuracy to hit waypoints.	Return to base and check connections

Bad GPS Health	High	Connections between GPS module and AP module are secure and snug. No GPS means no flight.	Return to base and check connections
Bad Barometer Health	High	Ensure the autopilot module is secured and vibration is minimized. The mission cannot continue without a healthy barometer.	Return to base and check connections
Payload Code Failure	Low	The system is designed so that the payload and the autopilot are independent systems in order to ensure the plain is still flyable in the event of payload failure. The software module is designed to be modular and independent from all other modules.	Restart code wirelessly and proceed with the mission.
Automatic Target Detection Failure	Medium	In order to mitigate the risk to the success of the mission a human is kept in the loop at all times in order to verify all of the targets.	Restart code wirelessly and proceed with the mission.
Payload Communication Failure (<2 min)	High	Utilize a directional antenna in order to increase the range of the payload communications network.	Attempt to reconnect with the network in flight
Payload Communication Failure (>2 min)	High	Utilize a directional antenna in order to increase the range of the payload communications network.	Land the aircraft in order to reconnect with the network.

5 Conclusion

This paper summarizes the work done by interdisciplinary undergraduate and graduate students students of the University of California, Riverside in preparation for the AUVSI SUAS 2016 competition. Over the course of the year, the UCR UAS team has developed, designed, and tested their UAS. It incorporates an improved payload with automated imaging capabilities, a smaller and lighter platform with longer flight times, and increased safety mechanisms with its motor layout. Advanced algorithms were developed to accomplish new objectives for this year while building off of successful systems from previous years. Painstaking testing procedures with many ground and in-air tests were conducted to assure UCR UAS will excel in this competition.

Appendix A - Cybersecurity

Introduction

Cyberattacks, especially ones targeting powerful physical systems such as UAS's, have the potential to cause serious harm to life and equipment. Therefore, security is always a consideration in the design of our team's systems and we aim to reduce the risks whenever practical. This document will explain some of the ways that UCR UAS does, and sometimes does not, handle cybersecurity.

Because of limited personnel and money, our team is not and probably never will be able to patch every hole in our system. Many of the hobby-level components that we use were not built with any malicious intent in mind. However, this is not an excuse to at least be aware of state of our system, ground station, aircraft, and beyond, as it concerns to cybersecurity.

Wireless Security

The most obvious weakness in our system is the autopilot command and control link. To control the plane we use a pair of open-source 3DR 433MHz telemetry radios. These radios are cheap, widely available, and relatively high performance. However, they are completely unauthenticated and unencrypted. Their omnidirectional antennas allows attackers to gather and send signals from any direction. This leaves the underlying MAVLINK protocol open to any interception or spoofing an attacker wishes to do. In fact, there are already demonstrated attacks against MAVLINK that can cause UAVs to disarm in mid-air and plummet to the ground before the operator can even start to respond (Cammilleri 2015).

To mitigate this, other teams use other solutions for their telemetry link. Examples of these are radios meant for commercial UAS or long-range XBee modules (CUAir 2015). These are better, but not the perfect solution. As an example, the Zigbee protocol used by the XBee can be vulnerable to replay attacks which can quite harmful to a UAS.

Consider the case that an attacker observes that a particular encrypted packet causes the UAV to fly to a waypoint. If they keep replaying that packet, they can cause the UAV to stay in the air until it runs out of power. This can be done without even decoding the packet. In any case, this is a hard problem. The development of a communications system that is resistant to both attacks and is reliable in the face of normal packet loss is worthy of a project by itself.

However, of additional concern to to us, these other solutions are an order of magnitude or more expensive. As a team we could not justify the cost given the relatively friendly environment at the AUVSI-SUAS competition. However, this may change in the future as the competition evolves.

Nevertheless, there are still options available to us. If needed, we can tunnel telemetry, autopilot control, and manual control through the payload communication system. The payload and SRIC communication systems use Wi-Fi with industry standard WPA2 security and long passphrases, which we consider much more secure.

In addition, the system uses directional antennas, which (in addition to increase our link speed) make it slightly harder to perform attacks that require communication with both sides of the link. However, this can be easily defeated with greater transmit power. But because of the increased directionality and complexity we consider it less reliable than the 3DR radios. Therefore, we do not intend to use it unless absolutely necessary.

Electromagnetic Interference

GPS spoofing or jamming is a real and hard-to-defend threat. The consequences of a possible signal hijacking is very serious as it can lead to complete loss of control without the operators realizing it. This has previously resulted in UAVs being damaged or captured with one case occurring back in 2011 with Iran managing to redirect a UAV to land in a different area than it was scheduled to land in (Peterson).

Naturally, with our consumer grade GPS receiver, we are unable defend against such as attack short of perhaps placing the UAV in a non-GPS, inertially-guided mode and guiding it back manually. Given the scope of the competition, and the expected operating environment of the UAS, we have placed GPS spoofing as less of a consideration in our threat analysis. However, this will not be the case in UAS operating in actively hostile, or even just electrically noisy, environments.

Interference does not always have to come from malicious sources. The 2.4 GHz ISM band is crowded. Almost every team is using it for something, let it be RC control or payload communications. In the case that interference causes our UAS to lose communication with our ground station, it will return to it's launch point where our signal is clear enough to reestablish normal communication. To reduce the chance of this hazardous and non-ideal situation from happening in the first place, we use spread-spectrum radios or spectrum analysers to select less occupied channels.

Network Security

Although the various wireless interfaces of the UAS are the biggest surface to defend, our wired network was not forgotten. Generally, clients on the network are trusted as only team members that are involved with running the UAS should be connected. Even though that is the case, network devices and access are secured with non-default passwords.

The only external connection from the wired network is the connection to the competition interoperability server and network. Even though that should be a safe network, we do not control it, and as such, that connection is made via a designated and firewalled port on our ground station router.

Human Security

In addition to the technical considerations for security, there exists the human element as well. The benefits of securing a system are lessened if the users with access are careless or socially engineered into giving others their access credentials. We have witnessed teams in past years write their Wi-Fi passwords in giant letters on their antennas. Whether or not that was just a decoy, we will never know, but it is undoubtedly a bad idea for your users to be broadcasting secrets to the world. Therefore, UCR UAS team members are taught that the integrity of the system they operate are also up to them.

Works Cited

Cammilleri, Casey. "Drone Code Execution (Part 1)." #_shellintel. N.p., 25 Sept. 2016. Web. <http://www.shellntel.com/blog/2015/9/25/drone-code-execution>.

CUAir: Cornell University Unmanned Air Systems Team. *Eos Unmanned Air System*. N.p.: n.p., 21 May 2016. PDF.

Peterson, Scott. "Exclusive: Iran Hijacked US Drone, Says Iranian Engineer (Video)." The Christian Science Monitor. The Christian Science Monitor, 15 Dec. 2011. Web. 18 May 2016.