BYU | FULTON COLLEGE
ENGINEERING AND TECHNOLOGY

BYU AUVSI Team

Technical Design Paper for the 2018 AUVSI Student UAS Competition

Brigham Young University

ABSTRACT

This technical paper discusses the UAS (Unmanned Autonomous System) designed and constructed by the BYU AUVSI Student Competition team, which consists of undergraduate students from Mechanical, Electrical, and Computer Engineering studying at Brigham Young University (BYU). The paper covers the team's approach to analyzing what the UAS would be required to do, per competition requirements and regulations, and the model designed and created to fulfill those requirements. Included in this model are an airframe, built from scratch, and ROSplane Autopilot, a Robot Operating System (ROS) autopilot system designed here on BYU's campus by the MAGICC Lab, and further built on by the BYU AUVSI team. Furthermore, the paper discusses the testing completed by the team on the plane's individual and assembled systems and explains safety and mitigation features.

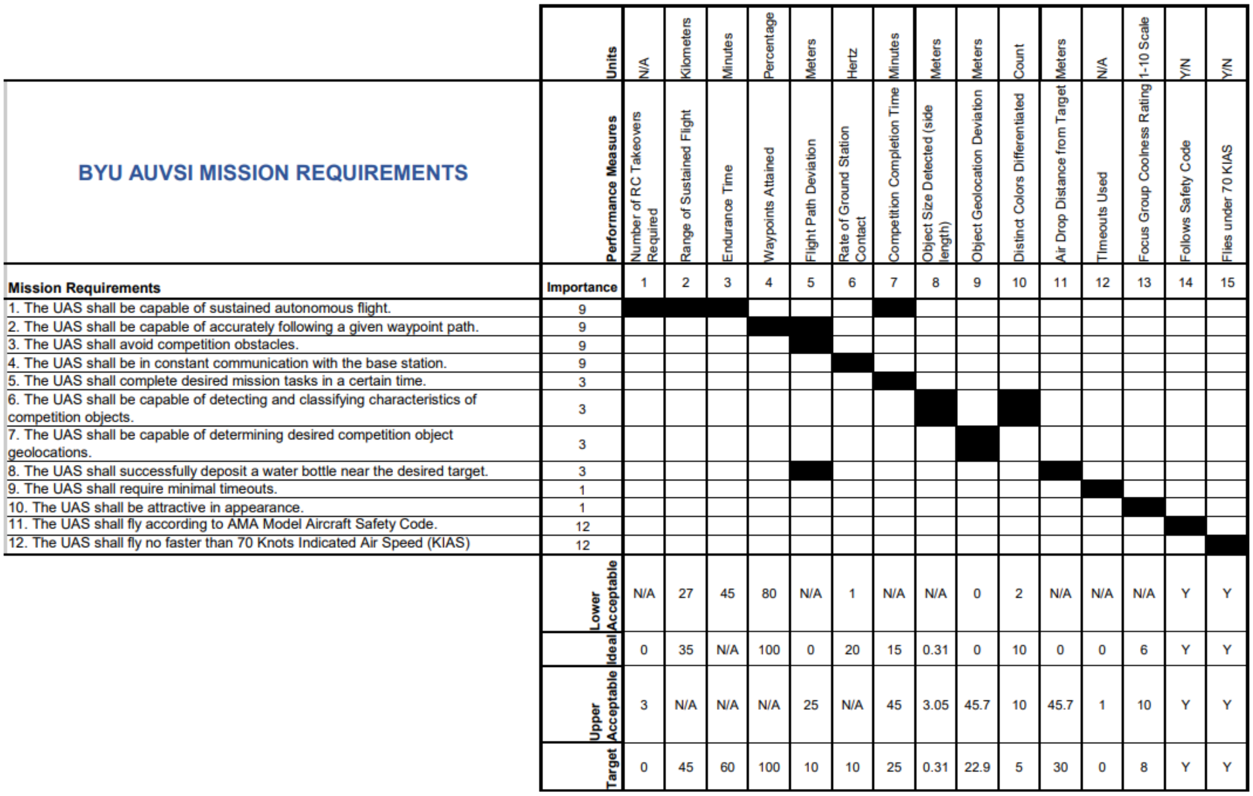
Table of Contents

1. Systems Engineering Approach

1.1. Mission Requirement Analysis

The mission requirements were primarily derived from the AUVSI SUAS 2018 Competition Rules. Their ranks were determined by the relative scoring weight at the competition, with higher-scoring mission objectives receiving greater importance. Requirements 11 and 12 (see Table 1) were considered constraints and therefore received highest importance. The team also determined performance measures relating to each mission requirement to assess how well each requirement was met. Each performance measure had an acceptable range along with ideal (best value in the absence of tradeoffs) and target values. Table 1 contains a summary of the mission requirements, related performance measures (indicated by marked boxes), and associated ranges/values.

Table 1 – Mission Requirements Matrix.



	Units	N/A	Kilometers	Minutes	Percentage	Meters	Hertz	Minutes	Meters	Meters	Count	Meters	N/A	1-10 Scale	Y/N	Y/N
BYU AUVSI MISSION REQUIREMENTS	Performance Measures	Number of RC Takeovers Required	Range of Sustained Flight	Endurance Time	Waypoints Attained	Flight Path Deviation	Rate of Ground Station Contact	Competition Completion Time	Object Size Detected (side length)	Object Geolocation Deviation	Distinct Colors Differentiated	Air Drop Distance from Target	Timeouts Used	Focus Group Coolness Rating	Follows Safety Code	Flies under 70 KIAS
Mission Requirements	Importance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. The UAS shall be capable of sustained autonomous flight.	9	■	■	■				■								
2. The UAS shall be capable of accurately following a given waypoint path.	9				■	■										
3. The UAS shall avoid competition obstacles.	9					■										
4. The UAS shall be in constant communication with the base station.	9						■									
5. The UAS shall complete desired mission tasks in a certain time.	3							■								
6. The UAS shall be capable of detecting and classifying characteristics of competition objects.	3								■		■					
7. The UAS shall be capable of determining desired competition object geolocations.	3									■						
8. The UAS shall successfully deposit a water bottle near the desired target.	3					■						■				
9. The UAS shall require minimal timeouts.	1												■			
10. The UAS shall be attractive in appearance.	1													■		
11. The UAS shall fly according to AMA Model Aircraft Safety Code.	12														■	
12. The UAS shall fly no faster than 70 Knots Indicated Air Speed (KIAS)	12															■
	Lower Acceptable	N/A	27	45	80	N/A	1	N/A	N/A	0	2	N/A	N/A	N/A	Y	Y
	Ideal	0	35	N/A	100	0	20	15	0.31	0	10	0	0	6	Y	Y
	Upper Acceptable	3	N/A	N/A	N/A	25	N/A	45	3.05	45.7	10	45.7	1	10	Y	Y
	Target	0	45	60	100	10	10	25	0.31	22.9	5	30	0	8	Y	Y

As seen in the table, higher priority was given to the tasks of autonomous flight, waypoint navigation, and obstacle avoidance. Lower importance was assigned to other competition objectives such as air delivery and object detection, classification, and localization. These rankings reflect the team's goal of achieving superior performance in the requirements related to autonomous flight while also gaining some points from other objectives.

1.2. Design Rationale

This year's competition team consisted of senior undergraduate students from the Mechanical, Electrical, and Computer Engineering departments. The project was allocated a budget of $2,500 plus $1,000 for competition registration. The team had access to the previous year's test documentation and design. This provided a base for the team to start addressing the tasks and challenges of the competition. The top priorities were to design and build a custom airframe and a suitable autopilot. These subsystems would provide the basic autonomous capabilities of waypoint navigation and static obstacle avoidance. Second to those priorities were to add an imaging system, moving obstacle avoidance capability, and a bottle drop subsystem.

To accomplish the goals stated above and in the previous section, the team subdivided into three groups: airframe, autopilot, and imaging/interoperability. The airframe team worked to design and build a completely new aircraft better suited for the competition. The autopilot team continued to refine the custom ROS autopilot developed at BYU, and the imaging/interoperability team established a reliable system for capturing images, gathering appropriate data, and transmitting results to the judges' server.

1.2.1 Airframe

Initially, the team considered three primary airframe structures: fixed-wing, rotary-wing, and vertical-takeoff-and-landing (VTOL). A fixed-wing structure was determined to be the most promising concept for several reasons. First, the team had more prior experience with fixed-wing aircraft design and control. Second, a fixed-wing aircraft would require fewer modifications to the existing autopilot software. Therefore, more time could be spent perfecting path planning and path management. Another advantage of a fixed-wing aircraft is a relatively higher payload capacity for a given power consumption. A fixed-wing aircraft could provide ample space in the fuselage to carry all the required components for completing the mission. The speed of the aircraft was also critical for success since flight time is worth 10 percent of the competition score, and a fixed-wing design would be faster than a VTOL or rotary-wing of comparable size and power.

While a fixed-wing aircraft may fulfill the requirements best, it is not without compromise. For example, multi-rotor and VTOL designs are able to hover, while a fixed-wing cannot. This hovering ability could allow for cleaner images with more accurate positioning, as well as more accurate payload drop. Autonomous landing would also be easier with vertical flight capabilities. However, these benefits do not outweigh the advantages of simplicity, speed, low power requirements, and weight-carrying capacity that a fixed-wing aircraft offers.

After deciding to use a fixed-wing structure, the team also decided on a relatively stable, safe design to maximize the success rate at the competition. Since the competition missions do not require excessive maneuverability, there is little benefit to having a statically unstable plane. This decision, in turn, solidified much of the plane's remaining design. For example, a high-wing design made more sense than a low-wing plane because the motors and propellers would be put at greater risk upon landing with a low-wing design. The wings are like the "Hershey-bar" design (constant chord throughout, no sweep, rectangular shape), with slight modifications, because this design is known to be more stable at low speeds. The boom design helps keep the tail safe by dispersing any shocks to the body through the carbon-fiber rod (it also reduce drag on the fuselage).

1.2.2 Autopilot

A standard approach to UAS autopilot solutions is to implement a Pixhawk-based system. However, the team has opted to utilize research at BYU that has led to the development of a Robot Operating System (ROS)-based autopilot called ROSplane. ROSplane utilizes deep integration with ROS to enable platform flexibility as well as total customization of controllers.

The open-source and robust nature of the ROSplane autopilot provides a good platform for algorithm customization and optimization. It also uses Gazebo to provide high-fidelity simulations that reduce development and in-air testing time. The accompanying flight controller firmware ROSflight also provides a customizable platform that allows modifications down to the lowest level of pure PWM output to servos. The Pixhawk or other off-the-shelf autopilots can be difficult to adapt the code to the mission requirements.

1.2.3 Image System

The Basler Ace acA2500-60uc was chosen for its high resolution, relatively low price, and ability to easily interface with ROSplane, the autopilot of choice. A gimbal was deemed unnecessary because only one target is outside of the competition boundaries. All other targets can be detected by flying overhead in the search area. This targeting system will allow for object detection, classification, and localization of all point-to-ground objects.

2. System Design

2.1. Aircraft

Using XFLR5, an aircraft modeling software, the team designed a plane that could perform well at relatively low speeds, maintaining steady and level flight without much input from the autopilot system. The finalized design can be seen in Figure 1 below.

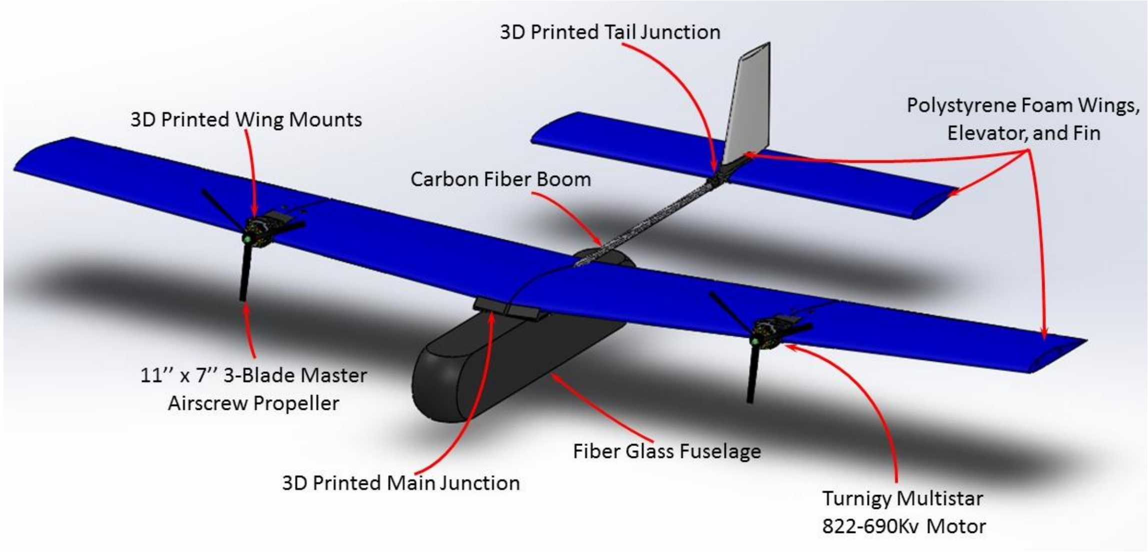

3D Printed Tail Junction
3D Printed Wing Mounts
Polystyrene Foam Wings, Elevator, and Fin
Carbon Fiber Boom
11'' x 7'' 3-Blade Master Airscrew Propeller
Fiber Glass Fuselage
3D Printed Main Junction
Turnigy Multistar 822-690Kv Motor

Figure 1 – Labeled CAD model of the airframe design.

The plane has a span of 2.4 meters with a mean aerodynamic chord of 0.3 meters. Specifications for the lifting surfaces can be found in Table 2 below. The plane is roughly 1.4 meters long, with all the individual pieces being attached through a custom-made, 3D printed junction piece that screws into the fuselage. Additional 3D printed pieces were used for the motor mounts and the tail junction.

Table 2 – Specifications for the designed airframe.

	Main Wing	Elevator	Fin
Span (m)	2.40	0.90	0.55
Area (m^2)	0.72	0.18	0.04
Aspect Ratio	8.00	4.50	3.38

The fuselage is made of foam-core wrapped in fiber glass for additional strength. The wings, elevator, and fin are cut from insulation foam boards (polystyrene) using a CNC foam cutter found on BYU's campus. The wings are reinforced using carbon-fiber rods. The boom attaching the tail to the fuselage (via the 3D printed junction) is also a carbon-fiber rod. A variety of airfoils were tested in XFLR5, including a variety of NACA and Eppler E-series airfoils, the GOE447, and the Clark Y-H; however, using the XFLR5 model, the team decided to use the Eppler E-205 airfoil, largely because of its stability and high lift-to-drag ratio at the desired velocity (see Figure 2 below). To ensure that this structure would be sufficiently strong, the team performed a wing-loading test, where a specified mass was placed along each wing. The wings safely held three times the weight of the aircraft (about 103 N), which is more than adequate for the maneuvers required by this competition.

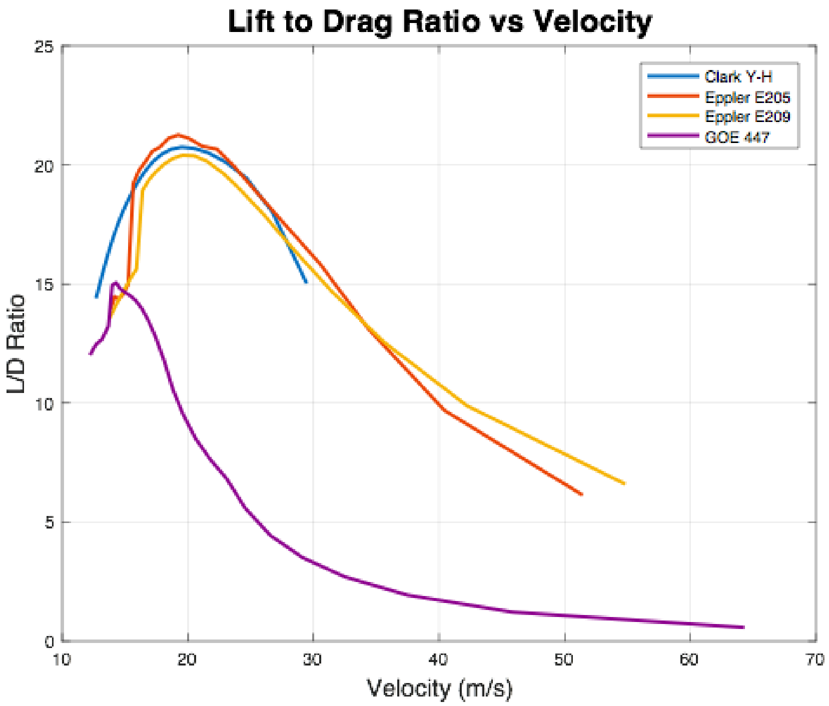


Figure 2 – A comparison of the Clark Y-H, Eppler E205 and E209, and GOE 447 airfoils. The Eppler E205 offers the highest lift-to-drag ratio at the desired velocities (~15-20 m/s).

The airframe is powered using two Turnigy Multistar 822-690 Kv brushless DC motors, each with an 11 by 7 inch 3-blade Master Airscrew propeller. Dynamometer tests showed each motor provides roughly

9.8 N of thrust and only draws a maximum of 22 A per motor, with a working current of approximately 16 A per motor. The speed controllers are rated for 100 A, providing more than enough allowance for current spikes. These parts, along with all the other electrical components in the plane, are powered by a single 4S 16,000 mAh 14.8 V lithium polymer battery that is stored in the nose of the fuselage.

2.2. Autopilot

2.2.1. Autopilot Hardware

The autopilot consists of an onboard ODroid computer, running a lite version of Ubuntu 16.04. The ODroid interfaces with an IntertialSense navigation system and a Flip32+ flight controller. The ODroid carries out all in-flight computations while receiving state feedback from the InertialSense unit and sending command signals to control surfaces via the Flip32+.

This setup differs from last year's system in the method of providing state feedback. Last year the team used the sensors on the Flip32+ as the inputs for the Extended Kalman filter. This year the state estimator is completely contained in the inertial navigation system from InertialSense, except for the airspeed sensor. The InertialSense package provides high-fidelity state estimations by fusing sensor data from MEMs gyroscopes, accelerometers, magnetometers, barometric pressure, and GPS/GNSS into a Multiplicative Extended Kalman Filter.

The selected onboard computer also changed. The onboard computer runs the autopilot on the plane and communicates with the ground station via a Ubiquiti PicoStation antenna. Last year the Brix i5 was selected because it is a very capable onboard computer. It is, however, heavy and requires a lot of power. To reduce the overall weight and power usage of the aircraft, this year the team has decided to use the ODroid XU4. The XU4 is significantly lighter, smaller, and requires less power while still being capable of running the autopilot, communicating with the ground station, storing images for post-processing, and transmitting images to the ground.

2.2.2 Autopilot Software and Architecture

The plane is being controlled by the ROSplane autopilot developed by the MAGICC lab at BYU. The architecture is modeled after the book *Small Unmanned Aircraft Theory and Practice* (Beard & McLain, 2012) as seen in Figure 3. Each of the systems in the architecture is explained below.

Path planner: The path planner loads in the boundaries, obstacle locations, and primary waypoints, then adds intermediate waypoints to safely guide the plane through missions. When a ground-station operator accepts the calculated path, the path planner will publish the waypoints to the plane via the Ubiquiti antenna connection.

Path manager: The path manager determines when the plane has hit a waypoint and cycles through the waypoint list received from the path planner, in order. When the plane has reached the end of a mission, the path manager commands a loiter about the last given waypoint until the list of waypoints for the next mission has been received. The path manager defines lines and arcs for the plane to track.

Path follower: The path follower takes the current path from the path manager and determines the proper airspeed, altitude, and heading commands in order to track the path. These commands are sent to the autopilot.

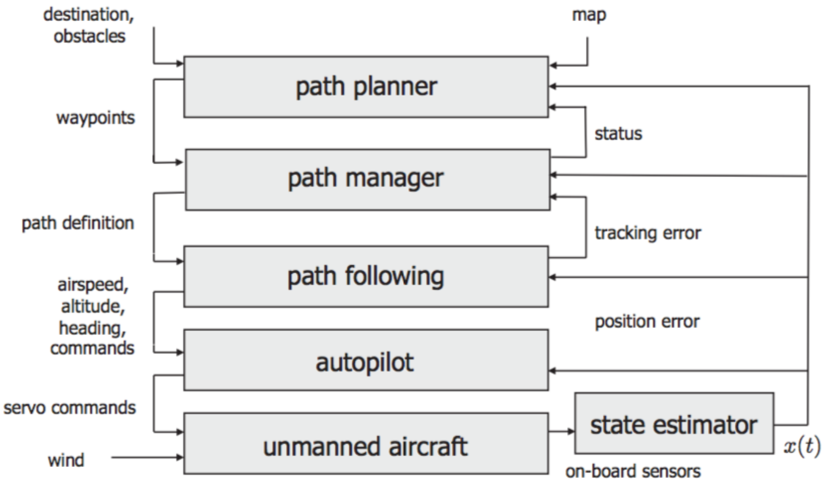


Figure 3 – ROSplane control architecture from Small Unmanned Aircraft Theory and Practice.

Autopilot: The autopilot converts the control commands from the path manager into actuator commands that the servos use to control the plane's motion. The autopilot uses PID controllers to command pitch, roll, course, altitude, and airspeed. The course controller is a nested loop with roll control as the inner loop. The altitude controller switches between controlling altitude by pitch and altitude by airspeed depending on the current altitude zone.

Aircraft: The aircraft uses onboard sensors as well as a state estimator to provide feedback for the overall controller.

2.3. Obstacle Avoidance

The obstacle avoidance system was designed to handle the specific competition environment. The path planner receives the competition boundaries, obstacles, waypoints, bottle drop location, and search area from the judges' server. A rapidly-exploring random tree (RRT) algorithm was designed to plan paths and avoid obstacles. Nodes, or intermediate waypoints along the path, are generated by the algorithm until a list of nodes defines a path that connects each primary waypoint. A high-level overview of the algorithm is outlined below:

1. A map is created with all obstacles, boundaries and primary waypoints
2. A node is initialized at the starting position of the path
3. Randomly, a point is selected and the closest node in the tree is identified
4. A new node is created a specified distance from the closest node and in the direction of the randomly selected point.
5. The link between nodes, or the path, is checked that it is possible for the plane to fly and that it avoids stationary obstacles. If the path is possible, the node is added to the tree.
6. Steps 3 – 5 are repeated until all primary waypoints are attained

As more nodes are added, a network of possible paths is developed, resembling a tree. The algorithm inflates the obstacles and boundaries to maintain a safe avoidance distance. If the algorithm is unable to find a possible path, then the allowable minimum avoidance distance will decrease until a path is possible. A smoothing algorithm eliminates unnecessary nodes along the path. The algorithm is run multiple times to find the shortest path. After a satisfactory path is found, all waypoints are sent to the autopilot via the ROS messaging system. There is no attempt to avoid moving obstacles.

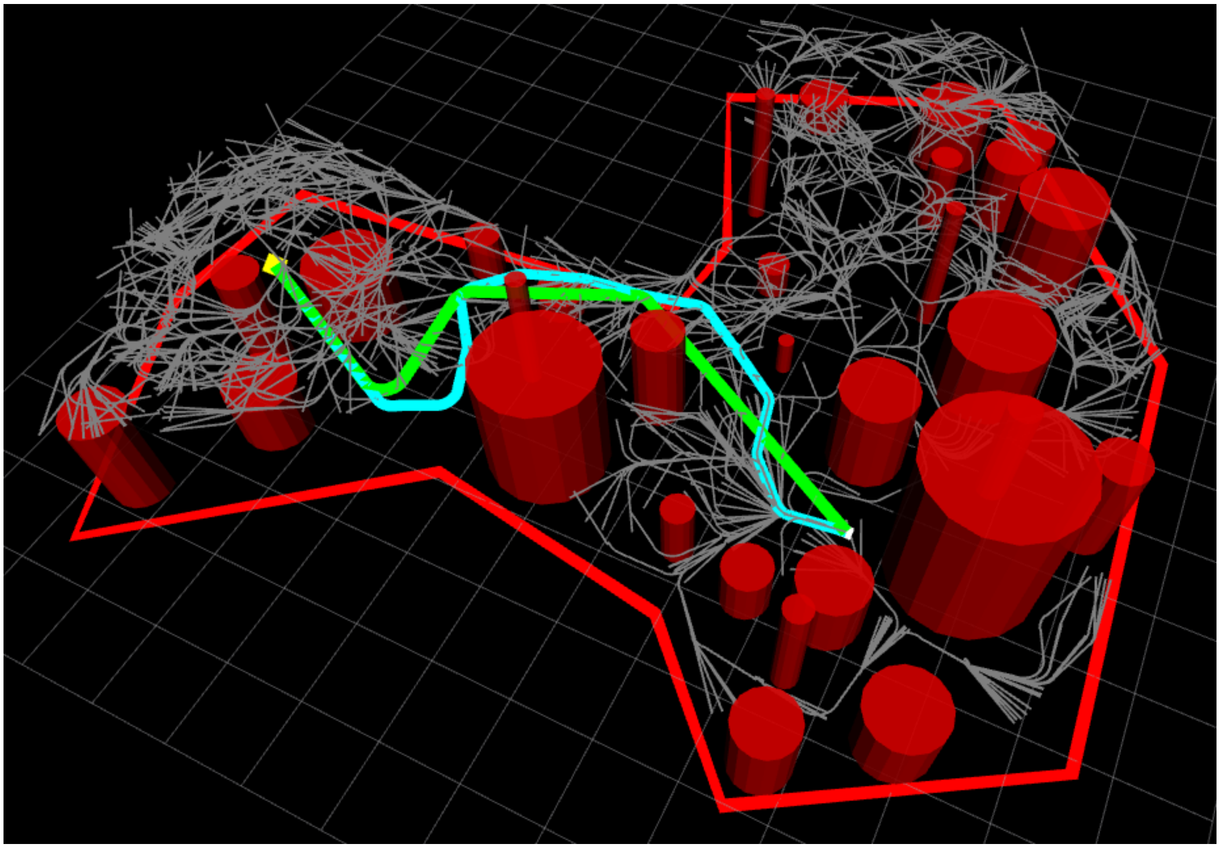


Figure 4 – Above is a depiction of the RRT algorithm. The gray lines represent possible paths that are part of the tree. The turquoise line represents the path produced by RRT. The green line is the smoothed path sent to the autopilot. The boundaries and obstacles are in red. This is an example of the path planner's ability to plan in a congested airspace.

2.4. Imaging System

The challenges presented in the imaging system include gathering high quality images taken at high altitudes, mitigating motion blur, ensuring efficient image overlap, and obtaining sufficient resolution. Along with these challenges, the data bandwidth between the UAS and ground station must be kept at a sufficient rate to not lose images or delay image transmission. Finally, the camera needs to be carried by the airframe, so the camera must have a compact sensor and relatively low weight.

After doing preliminary tests with last year's camera, the Pointgrey Chameleon 3 was determined to not provide the desired resolution to effectively resolve objects during the competition and did not fulfill the mission requirements. With these requirements in mind, the camera to be used on the UAS is the Basler Ace acA2500-60uc. The Basler Ace features a 5 MP 1 inch CCD sensor, a focal length of 12.5 mm, and supports a standard USB 3.0 interface. The Basler Ace, with a horizontal pixel per foot of 12.64 PPF at 200 feet , will allow us to effectively resolve the minimum size object of 1 foot wide with 1 inch thick lettering.

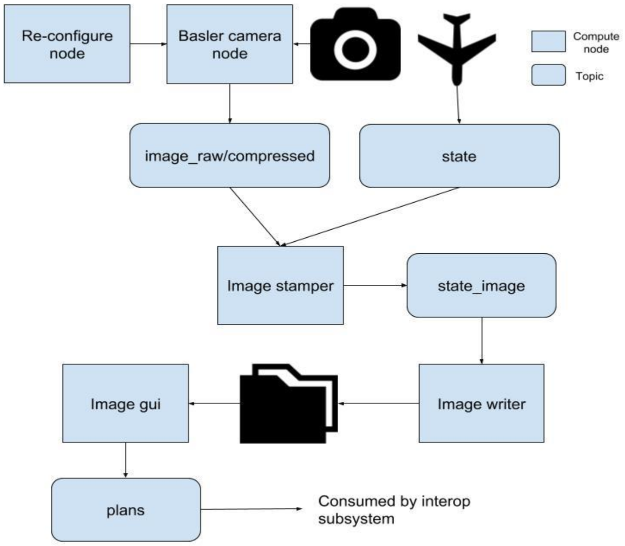


Figure 5 – Image System Diagram.

The Basler Ace, which will point directly out of the bottom of the airframe, will provide high quality images through which geolocation can be determined while the UAS flies a lawnmower path over the search area. The process of detection, initial localization, refined localization, and target characterization will be discussed in detail in the following section.

2.5. Target Detection, Classification, and Localization

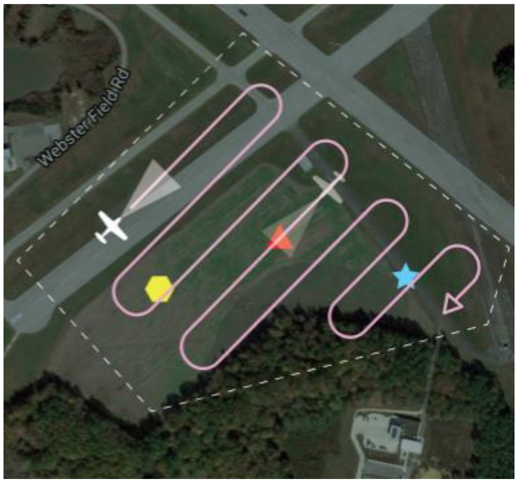


Figure 6 – Search area scan using lawnmower path.

Target detection will be based on operator input. As the UAS flies through the search area, images will be streamed back to the ground station at a rate of 1 image per second. The operator on the ground will cycle through the received images and click on the target in the image frame. With each click a rectified pixel coordinate is returned. This pixel coordinate, along with pose data from the autopilot, is then used to generate an estimate of the target's location. The operator can increment or decrement the target counter at any time as new targets move into view. It is advantageous to click multiple times on the target because the localization algorithm computes an average of all the calculated locations for each target. This minimizes operator error. These calculated locations will then be submitted via the image classification GUI (see Figure 6).

An image classification GUI serves as the main tool for submitting cropped images and target characteristics. The GUI displays all the images associated with a specific target and allows the user to select the best image. Every image displayed on the GUI has corresponding geolocation and heading data. Upon selecting an image, an average of all geolocation values (omitting outliers) is calculated. The user then uses the GUI to rotate, crop, and classify the target. The rotation of the image on the GUI is combined with the stamped heading data to calculate the final target heading. Cropping the image produces a target which is clearly visible and fills more than 25 percent of the submitted image. To classify the target, there are input fields for each characteristic (alphanumeric, alphanumeric color, target shape, etc.) that are packaged along with geolocation data at submission time. Once all these values have been calculated and entered, they are submitted via the interoperability system through the click of a button. There will be no attempt made for autonomous detection, classification, or localization.

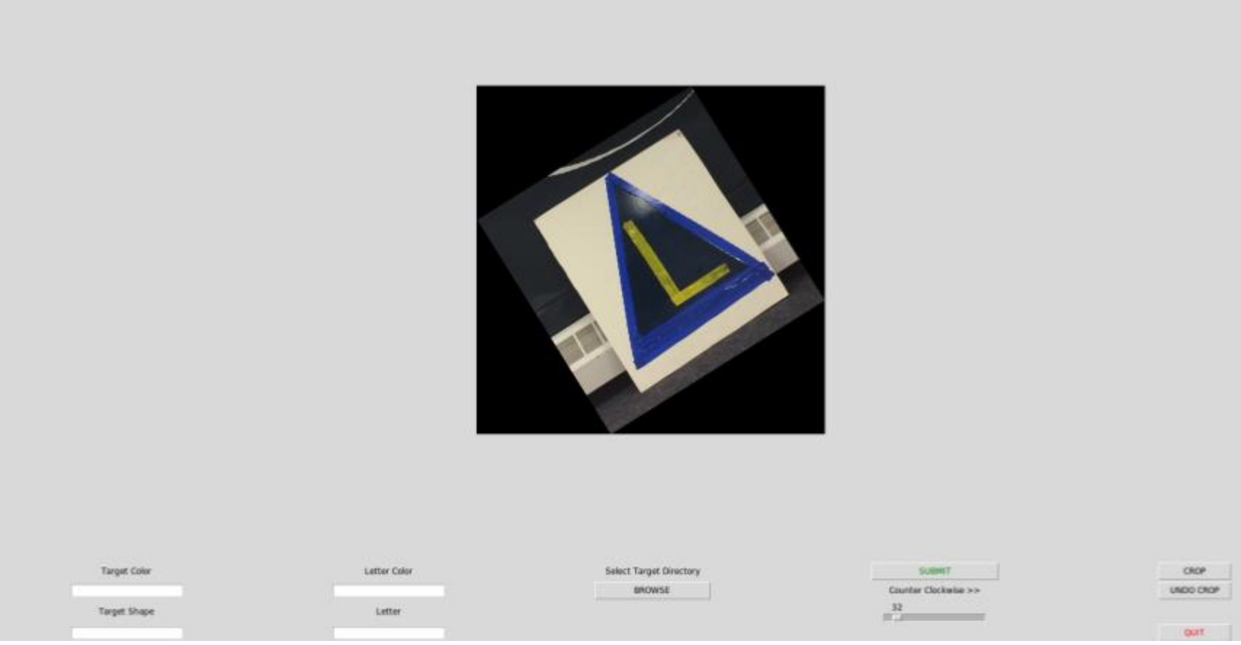


Figure 7 – Image editing and classification GUI.

2.6. Communications

The UAS contains two independent communications systems described below.

900 MHz FHSS: To ensure reliable RC control, the safety pilot communicates with the UAS on an independent frequency hopping spread spectrum. This communication occurs between an R9 receiver on-board the UAS and a Taranis Q X7 transmitter using an R9M module. Only the raw RC commands are sent over this network throughout flight.

2.4 GHz: The main data link between the UAS and the ground control station occurs over a 2.4 GHz connection. A Ubiquiti airMAX Sector antenna on the ground and a Ubiquiti PicoStation onboard the UAS provide the bridge for all data transfer through the ROS network. The UAS publishes telemetry data and images while subscribing to waypoint commands from the ground station. Communication between systems on the ground also happens via the ROS network through an Ethernet switch.

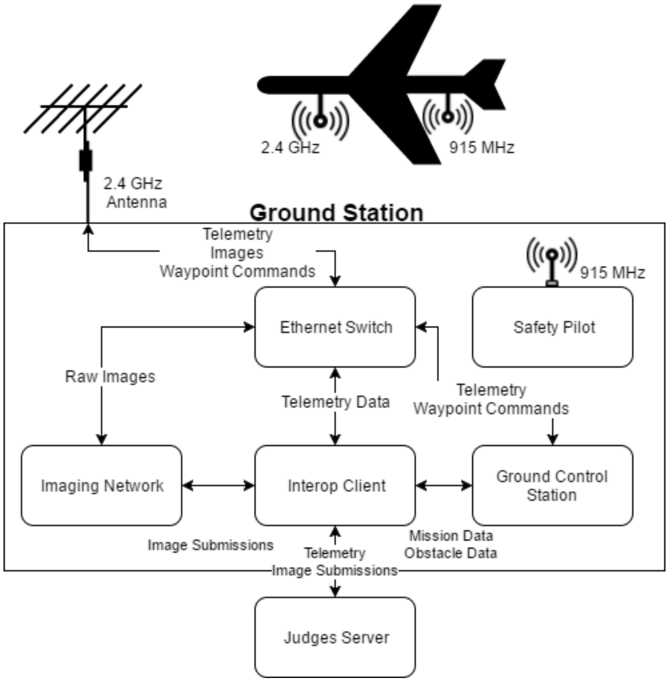


Figure 8 – Ground Station Communication.

2.7. Air Delivery

The payload is a standard 8 oz water bottle that is stored inside the fuselage until deployment. The bottle rests directly on top of a door that is secured by a servo. The weight of the bottle presses the door firmly against the servo horn. At deployment, the servo horn rotates so that the door is free to open from the weight of the bottle. The bottle drops, and the force of the air pushes the door back closed. The servo horn then rotates back to its original location, holding the door shut for the remainder of the mission. This mechanism was tested in a wind tunnel at various wind speeds (10-20 m/s), and the wind blew the door closed 100 percent of the time after deployment. Figure 9 illustrates the mechanism.

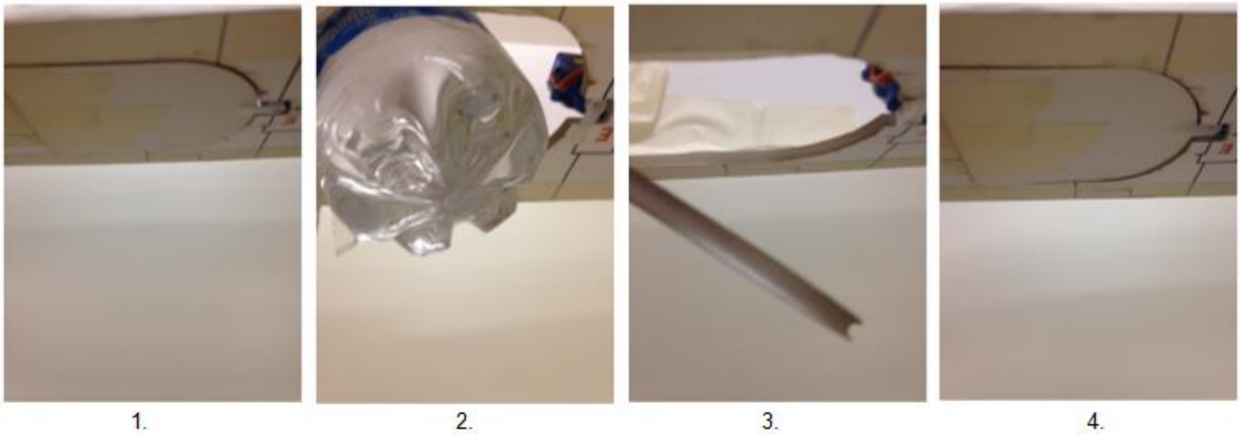


Figure 9 – Illustration of air delivery mechanism. The front of the aircraft is to the left. 1) Payload door rests on servo horn. 2) Servo horn rotates, allowing weight of water bottle to push door open. 3) Force of air pushes door back to closed position. 4) Servo horn returns to original position, keeping door closed.

Although air delivery system has not yet been tested in a mock competition setting, future tests will soon be performed to refine the release algorithm and delivery accuracy.

2.8. Cyber Security

Using a STRIDE threat model, important assets and threats were identified. This knowledge allowed for decisions to be made with security in mind. For instance, the wireless radio is one critical asset that an attacker could target to sniff data transmitted from the UAS. To counter that possibility, a Ubiquiti Rocket controller that supports AES encryption is being used. AES will ensure an encrypted communication channel between the ground station and Ubiquiti controller, thus preventing any unwanted parties from gaining access to the controller.

Authentication of users who interact with the onboard computer (ODroid) is achieved through an SSH connection, a cryptographic network protocol, and the WPA2 security protocol to ensure a secure WiFi connection. In addition, a multitude of threats are countered with a simple firewall that supports Network Address Translation. Without NAT, all systems would be exposed, and potential hackers would have a wide attack surface. Through these methods, there is protection against a variety of possible threats.

3. Safety, Risks, and Mitigation

Throughout all phases of development and testing of the UAS, safety has been a top priority. In all circumstances, safety controls were put in place and strictly adhered to by members of the development team. The follow sections outline potential safety hazards and detail safety controls to mitigate risk.

3.1. Developmental Risks and Mitigation

The development process posed several safety risks, described in Table 3 below.

Table 3 – Developmental Risks and Mitigations.

Developmental Safety Risk	Likelihood	Impact	Mitigation
Dynamometer Test Stand Safety Hazard	Low	Medium	-A fiberglass shield installed to block the dynamometer from the rest of the room -Safety glasses worn during testing -Testing personnel stood at safe distance during all tests -Software cutoff failsafe active during all tests
Autopilot/Communications Malfunction	Medium	High	-Experienced safety pilot ready to take manual control in case of autopilot malfunction -UAS kept within safe range of safety pilot line of sight -Audible, low-signal warning should UAS approach range limit
UAS Crash Damage or Injury	Medium	High	-Test flights conducted in large, remote area -Test personnel stationed near safety cover structure -Personal kept safe distance during launch, testing, and recovery of UAS

3.2. Mission Risks and Mitigation

The competition mission and autonomous flight pose a number of safety risks outlined in Table 4 below.

Table 4 – Mission Risks and Mitigations.

Mission Safety Risk	Likelihood	Impact	Mitigation
Autopilot Malfunction	Low	High	-Experienced safety pilot ready to take manual control in case of autopilot malfunction
Loss of Communications with UAS	Medium	High	-30 second communications loss triggers RTL -3 minute communication loss triggers flight termination
Unintentional Water Bottle Drop	Low	Low	-Water bottle mechanism disarmed by default -Only armable when safety pilot switch enabled
Unintentional Equipment Drop from UAS	Low	Medium	-Internal components securely fastened inside UAS -External components (water bottle assembly, antennas, etc.) securely attached and checked prior to flight -Locking fasteners and thread-lock used where applicable
Electrical Fire	Low	High	-Batteries balance-charged and inspected for damage -All charging done in fireproof box
Loss of Power During Flight	Low	High	-Battery checked for full voltage before flight
Loss of Propeller	Low	High	-Propellers inspected before flight to ensure secure attachment and no blade damage
Loss of Control of Control Surfaces	Low	High	-Control surfaces inspected before flight -Servo linkages reinforced and inspected before flight

4. Conclusion

The BYU AUVSI team has designed, built and tested a UAS ready to excel at mission tasks in the 2018 AUVSI Student UAS Competition. Throughout the course of the year, a system engineering approach has been valuable to converge on a successful design and to mitigate development risks. The final UAS design features an innovative autopilot, a simple yet effective imaging system, and a robust communications network. Extensive testing has been performed to guarantee the successful completion of competition tasks. Frequent testing will continue during the preparation period to ensure reliability and satisfactory operator training.

References

Beard, R. W., & McLain, T. W. (2012). Small Unmanned Aircraft Theory and Practice. Princeton: Princeton University Press.