UAV Austin | The University of Texas at Austin

Technical Design of *Phoenix III*

AUVSI SUAS 2019

PHOENIX III

Abstract

In this document, UAV Austin outlines and describes the design, development, testing, and operation processes of Phoenix III, the team's 2019 system. Additionally, the mission of Phoenix III, as defined by the 2019 AUVSI Student Unmanned Aerial Systems competition, is presented in the document. The development of the UAS and its mission was accomplished by the 83-member development team whose student engineers' backgrounds span across multiple disciplines, including aerospace engineering, mechanical engineering, computational engineering, electrical engineering, computer science, and applied physics. Phoenix III is a fixed-wing aircraft system capable of autonomous flight, image data acquisition and processing, and unmanned ground vehicle delivery. An iterative design approach and systems engineering were the primary principles incorporated during the developmental phase of Phoenix III, while always prioritizing the safety and security of the individuals.

1 Introduction

Over the past decade, unmanned aerial systems have become increasingly prominent in tasks spanning commercial applications, military purposes, and personal use. As the demand for the UAS continues to rise, the need for innovation in technology, reliability, and safety have simultaneously become key requirements of the engineering field. The AUVSI SUAS competition serves as an opportunity for students to involve themselves in designing and operating a UAS. Incorporating mission objectives relevant to the cutting-edge technology and capabilities of the UAS field, the competition challenges teams from universities across the world to develop exceptional unmanned systems.

Figure 1: Phoenix III

UAV Austin took a systems engineering approach to maximize performance of the UAS under given situational constraints while maintaining professionalism and a mindfulness of excellence in the final project. UAV Austin was able to effectively carry out the process through multiple design iterations, simulation and modeling analysis, component, systems, and flight testing in order to complete the final iteration of the aircraft, unmanned ground vehicle, and ground station. Figure 1 shows Phoenix III—the 2019 unmanned aerial system for UAV Austin. The team is comprised of a diverse group of undergraduate students spanning across various engineering disciplines. This technical report seeks to exhibit the objectives, mission considerations, aircraft components, and design procedures.

2 Systems Engineering Approach

INCOSE defines systems engineering as an interdisciplinary area of engineering that synthesizes multiple considerations simultaneously in order to optimize the design process and ensure that the customer and stakeholders' needs are exceeded [4]. UAV Austin structured the project in a similar manner as INCOSE's systems engineering model, by first stating objectives and investigating alternatives. Through modeling, integrating, launching, assessing, performance testing, and re-evaluating the system, the project was executed with attention to the quality of the final product.

2.1 Mission Requirements Analysis

The foundation of any system is governed by the set of constraints that define the system requirements. UAV Austin analyzed the constraints defined by the 2019 AUVSI SUAS competition rules [2], created a weighted table based on environmental factors (Table 1), and defined a set of derived requirements for the 2019 UAS. The mission, as defined by the competition, requires autonomous flight, obstacle avoidance, target recognition, and air delivery. In addition, the UAS is the conglomeration of the UAV, unmanned ground vehicle (UGV), and ground station.

2.1.1 UAV Derived Requirements

The 2019 system, Phoenix III, and the unmanned ground vehicle, were designed with regards to mission requirements set by the competition and derived requirements based on the team's previous experience, budget constraints, and time constraints. In addition, UAV restrictions include a maximum takeoff weight of 55 lbs, a maximum airspeed of 70 KIAS, and the usage of low-risk, non-foreign fuel or batteries. UGV restrictions include maximum drive speed of 10 mph, maximum weight of 48 oz, and fully autonomous driving capability.

The derived requirements for the UAV are as follows:

1. The UAV shall be capable of autonomous flight.
2. The UAV shall accurately fly to each waypoint while remaining within the flight boundaries.
3. The UAV shall avoid stationary obstacles, whose locations are received from the interoperability system.
4. The UAV shall gather images of approximately 0.1 square miles in under 10 minutes to identify target characteristics and GPS locations.
5. The UAV shall accurately deliver a UGV to the drop location which shall sustain the high-force impact of the drop sequence.

2.1.2 Ground Station Derived Requirements

The ground station requirements are a combination of both competition defined rules and team defined rules to ensure the safety of all flight line members and support mission success.

1. The ground station shall display a map showing flight boundaries, UAV position, speed, altitude, and other competition elements for the judges.
2. The ground station shall receive mission details and submit mission deliverables using the interoperability system.
3. The ground station shall be portable and easy to set up in less than 20 minutes.
4. The team shall have personal protective equipment (PPE) which includes, at minimum, proper tools, gloves, eye protection, and hearing protection when appropriate. Safety risk mitigation shall also be implemented, which includes team training and radios for communication. The team shall have equipment (first aid kit and fire extinguisher) to respond quickly to emergencies.

2.2 Design Rationale

For the 2019 competition, UAV Austin designed and manufactured a custom composite fixed wing vehicle to maximize mission performance. In tandem to Phoenix III's fabrication, our previous competition plane, SVL, was modified and utilized for weekly mission testing at the Austin Radio Control Airfield.

2.2.1 Environmental Factors

The environmental factors that primarily influenced the design were a rigid timeline, budget, and UAV Austin's expansion. A budget of $13,800 USD was distributed between travel expenses and fees, the team's subteams, spare components, and tasks for mission requirements. Increasing the number of members of UAV Austin required a change of structure. The program manager, chief engineer, and chief software engineer created subteams that would specialize in a specific aspect of the mission. For hardware, the *Ground Station* subteam works with Mission Planner to communicate with the UAV and its subsystems, *Internal Systems* designs and manufactures the gimbal and UGV, *Power and Electronics* utilizes circuit theory to power the vehicle's subsystems, and *Structures* manufactures the composite airframe. For software, the *Autopilot* subteam creates obstacle avoidance algorithms, *Infrastructure*

Task (%)	Description	Requirement for Accomplishment
Timeline (10%)	• Complete mission in minimal time, under 40 minutes • Refrain from using a timeout	• Optimization of UAS for flight time • Design the mission for optimal flight time
Autonomous Flight (20%)	• Complete mission without manual takeover • Travel through waypoints as accurately as possible and within 100ft	• Conduct autonomous flights with UAV with integrated autopilot scheme to ensure repeated accuracy of waypoint capture
Object Detection, Localization, and Classification (20%)	• Identify target shape, color, character, orientation, and GPS location • Submit targets autonomously during mission • Submit targets via interoperability system	• Simulate accurate mission environment during testing period • Optimize pixels on target to meet user-defined constraints and UAS flight capabilities
Obstacle Avoidance (20%)	• Avoid stationary obstacles in flight area defined by interoperability system	• Implement obstacle avoidance scheme during test flights to ensure repeated accuracy • Create mission plan around stationary obstacles
Air Delivery (20%)	• Design payload to weigh less than 48 oz • Accurately drop payload to the target zone within 75 ft distance • Drive payload autonomously to destination	• Utilize iterative approach to design weight-optimal payload • Develop a deployment and airdrop sequence systems that accommodate payload capabilities • Simulate and model airdrop sequence to predict accurate drop path • Conduct multiple tests to ensure autonomous driving capability of payload
Operational Excellence (10%)	• Maintain strong team communication and exercise caution	• Develop specific roles and dedicated checklists for mission

Table 1: Mission Requirement Analysis

controls communication and telemetry, *Integration* exemplifies systems engineering by interfacing hardware and software, and *Image Recognition* builds tools for automatic and manual target classification. Members for each subteam were selected based on their qualifications.

2.2.2 Design Selection Criteria

Figure 2 depicts the design decision process inspired from the systems engineering approach.

Creating a custom aircraft allows for all mission requirements to be met without modifying a previous UAV or kit plane. UAV Austin identified the following as design priorities from the mission requirements: UGV payload, object detection, ease of maintenance, and mission time. Since the camera is such a critical subsystem, the airframe was designed around the Z CAM E1 and a spherical 360° rotating gimbal. The Z CAM E1's compact and lightweight structure, high resolution output, and interchangeable lens make it an ideal camera for target recognition.

For conceptual design, the Student Air Vehicle Evaluation program, or SAVE, by Dr. Armand Chaput was used for preliminary sizing and UAV configuration analysis to optimize user defined goals and create airframes iteratively. Accounting for short mission time, low thrust to weight ratio, and payload, a key generalization was made: for a UAV to remain stable after a significant payload drop, the center of gravity (CG) of the UGV must be near the CG of the UAV, necessitating a high wing design. Using the previous competition UAV as a comparison, the fuselage's volume was increased to support the UGV and the associated electrical systems. Between a twin engine and boom-propeller pusher configura-

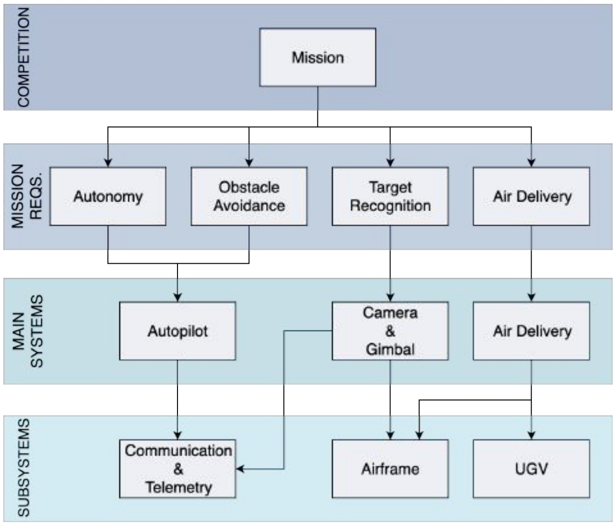

Figure 2: Design Decision Chart

tion, the pusher system was chosen to effectively balance the front loading weight from the gimbal. Design trade-offs had to be taken into account in the process. If more weight was allocated towards the payload systems, the UAV performance would decrease. The UGV drivetrain consists of two independent wheels for sizing reasons, but given larger sizing constraints, a four-wheel drivetrain could be designed. The parachute for the UGV could be larger to reduce impact speed, but space allocated for storage is limited.

After the initial stages of manufacturing data were recorded, the weight estimate grew from 20-25 to 28-30 pounds, thus requiring an increase in the wing surface area. As a result, a one-foot extension to the root of each wing was added, which subsequently called for an increase in the vertical tail. Further iterations and analysis shall be discussed in Section 3.1.1 Airframe and Structure.

3 Systems Design

For the AUVSI SUAS 2019 competition, a custom composite UAV was created to fulfill the mission requirements. The following section describes the design decisions carried by the design team and their encounters throughout the manufacturing process.

3.1 Aircraft

3.1.1 Airframe and Structure

In preliminary discussions, the design team determined the primary mission goals were to find and identify targets, accurately drop the UGV, and to complete the mission as fast as possible, all while being easily maintainable. The camera and gimbal system are critical to the fulfillment of these goals, and all subsequent design discussions kept the camera at the forefront.

With the help of Dr. Armand Chaput, the design team utilized the SAVE program to analyze and perform trade studies on a large number of vehicle configurations. Each vehicle configuration earned a score that was produced by maximizing a user defined cost function. The final handful of configurations had the highest scores with optimizing the thrust to weight ratio for minimum flight time and minimum thrust to weight ratio.

The fuselage cross section is trapezoidal to allow space for the drop bay of the UGV, as well as increase the total internal volume to allow for easy maintenance. The front and back tapers were added to ensure a smoother transition from the camera gimbal to the fuselage and from the fuselage to the motor housing, respectively. The wing-fuselage interface was decided to be a high-wing, as seen in figure 3, because the payload could not fit at the CG with a mid-wing since it would interfere with the wing spar.

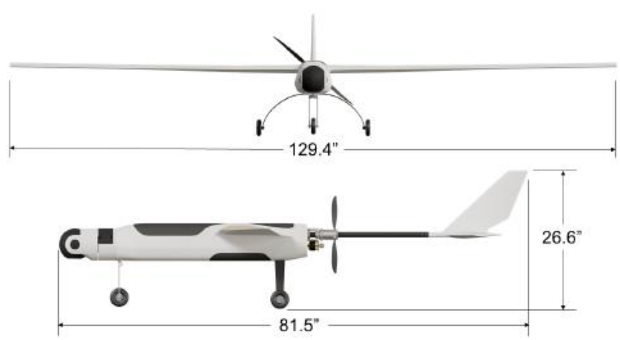

Figure 3: Phoenix III dimensions

Three options were considered for motor placement: wing-mounted propellers, pusher configuration, and off-center motor mounting. Wing-mounted propellers were considered because of the extra thrust it provides with greater motor efficiency. However, the extra structure and electronics systems needed to support propellers in both wings eliminated the idea and shifted focus to a back-mounted push propeller. A back mounted push propeller is attractive because the added weight in the back could easily be manipulated to balance out the weight of the camera gimbal at the front of the plane, as well as the team’s previous experience with the push propeller configuration. The main disadvantage is that it requires a heavier tail and structure. To reduce overall weight, an alternative pusher propeller configuration was developed, where the push propeller could be mounted around a central boom so a conventional, easy to construct tail design could be implemented.

The fuselage was made of two fiberglass foam composites, with 5 layers of alternating 45/45 0/90 3 oz fiberglass with a single foam layer in between the middle. Each part was laid up in two separate molds using resin infusion, and the two were joined by glassing them together using the bulkheads for alignment. The motor assembly was created with a CNC out of aluminum. The landing gear was made with unidirectional Kevlar with 45/45 Kevlar wrapped around it.

Wing Span	129.4”	**Weight**	27 lbs
Aspect Ratio	10.75	**Length**	81.5”
Cruise Speed	40 KIAS	**Width**	129.5”
Stall Speed	25 KIAS	**Height**	26.6”

Table 2: Aircraft Specifications

3.1.2 Internal Systems and Payload

The payload of the aircraft (see Figure 4) is divided into five sections: the nose cone, the anterior fuselage, the posterior fuselage, the motor housing, and the UGV bay. The nose cone is comprised entirely of the image recognition system, and it houses the camera, lens, gimbal motors, and gimbal controller. To allow for a full range of motion, the camera and gimbal motors are wired through slip rings placed on the gimbal’s axes of rotation. Rather than being housed within a nose cone fuselage, the gimbal itself aerodynamically serves as the nose cone when attached to the UAV. In the anterior fuselage, 3M Dual Lock fasteners are used to secure the Rocket AC, the Jetson TX2, and the main and satellite receivers. The front landing gear is attached directly to the bulkhead, and it is controlled by an adjacent servo motor in the anterior fuselage. The posterior fuselage contains the main and supplemental batteries, the Pixhawk 2, BEC, GPS transponder, Jetson TX2 module, and anchor for the tail boom. The posterior fuselage also houses the electronic speed controller (ESC), which allows users or programs to modulate its effective thrust as needed and records the voltage, current, and temperature. The UGV bay houses the UGV and the mechanism for releasing i and is positioned at the CG of the aircraft, so the CG does not change position after the UGV is released. The motor housing contains the motor for the propeller and the belt system.

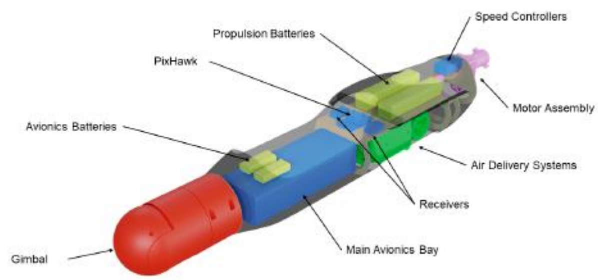

Figure 4: Aircraft Internal Configuration

3.1.3 Avionics and Power System

The avionics design rationale focused on two main ideas: wire management and safety. To ensure efficient debugging and operation during the competition, the team prioritized the planning and wiring of electronics throughout the build process, creating an electronics layout document that adapted to the changes of the UAV during all stages of design and fabrication.

Most wires and electronics were fastened along the inner surface of the plane using 3M Dual Lock fasteners. Mounts were 3D-printed with PLA plastic to secure certain electronics to the curved profile of the interior. Although the UGV bay and CG limitations constrained the electrical blueprint, the layout document ensured adaptability to these changes.

To guarantee successful operation in the event of multiple critical failures, the power system is designed on three layers of redundancy. The Pixhawk flight controller was allocated two redundant power systems via the Pixhawk module, alongside a redundant parallel connection of two propulsion batteries seen in Figure 5.

Regarding the integration of new design standards in this year’s UAV, last year’s Raspberry Pi has been replaced by a NVIDIA Jetson to provide better performance and onboard image processing. A RFD900+ modem was installed in place of last year’s 3DR modem to increase communication range and improve signal strength to the UAV.

The batteries were chosen based on the needs of the system, projected flight time, and battery testing previously recorded by the Air Systems Lab at The University of Texas at Austin. Two 2200 mAh 3S ThunderPower LiPos were selected as the avionics batteries. The propulsion system runs on two parallel 6000 mAh 5S ThunderPower LiPos, which is the limiting factor for endurance at 45 minutes, with the avionics batteries outliving the propulsion system in case a gliding landing is required.

3.1.4 Propulsion

The UAV's propeller motor, Scorpion SII 4020-630, remains the same from last year. To simplify mounting and construction, the propeller was changed to an Aeronaut CAM Carbon 20"x12" pin-mounted folding propeller. As a result, battery specifications have been updated to better meet peak efficiency curves: switching from 4S to 5S resulted in a ~1.5% increase in efficiency.

3.2 Autopilot

Phoenix III uses a Pixhawk 2.0 Flight Controller running ArduPilot to conduct a variety of different autonomous flight operations such as autonomous waypoint navigation, takeoffs, and landings. This presents some issues, such as controller not possessing real time flight path adaption for all of the obstacles. In order to circumvent this issue, the team has developed a secondary system that works alongside the Pixhawk, discussed in Section 3.4. The combination of the two systems provides the GCS with adequate control of the aircraft during all phases of flight.

In order for the aircraft to safely and successfully navigate the different waypoints, the autopilot needs to be adequately tuned to prevent oscillations of control surfaces during flight, overspeeding or stalling of the UAV, and fly through the defined waypoints accurately.

3.2.1 Ground Station Software

The flight software used on the autopilot ground station is Mission Planner (MP), chosen for its ease of access to configuration and analysis tools, allowing for more rigorous testing and information gathering than competing software. Other ground station software, such as QGroundControl, were also considered but not chosen due to MP having tighter integration with ArduPilot, and the safety pilot and team advisor are familiar with both. MP provides the user with access to tools such as a PID tuner, used to ensure steady flight and respond to changes in conditions, and an L1 Navigation tuner, to tune waypoint transition data, as well as other data. Additionally, MP allows the flight line to observe and react to data such as battery voltage, altitude, and airspeed. For the UAV to successfully and efficiently navigate the given waypoints, the controller needs to be tuned to prevent oscillations of control surfaces during flight, avoid overspeeding or stalling the aircraft, and fly the defined waypoints with total accuracy. On the flight line, three display screens are used: one each for the pilot, UGV operator, and convey competition requirements to the judges and mission strategist.

Figure 6: Image of the Mission Planner GUI in flight

3.3 Software Systems

The UAV Austin software team showed great success with the team's microservice architecture from the previous competition year and have extended the previous design for this year. In this pattern, the software is separated into small, modular, and loosely-coupled services whose scope is limited to a particular domain. The design benefits of microservice architectures include more productive software development by allowing different teams to work on separate services simultaneously, increased flexibility in the technologies used by different services such as the programming language, and making maintenance of a particular service easier.

The individual services communicate with each other through platform-agnostic HTTP servers and clients. This enables services to use different programming languages to allow for a language choice that has been found most apt for a particular domain, such as Node.js and Elixir[a] for communications, Python for image recognition, and Rust[b] for low-level interactions and performance.

For deploying the services, each service is containerized using Docker, an operating-system-level virtualization toolkit, which allows for services to be developed and deployed in an identical way across different machines. This brings the development environment to be similar to that used on the flight line.

3.3.1 Fault Tolerance

Software fault tolerance is the ability for software to be able to continue operating after either software or hardware failures. Designing software from the beginning to be fault tolerant is important in that it

[a] Elixir is a programming language built on the Erlang VM, which is famous for its high reliability in telecom and VoIP systems.

[b] Rust is a systems programming language sponsored by Mozilla with a focus on memory safety and concurrency.

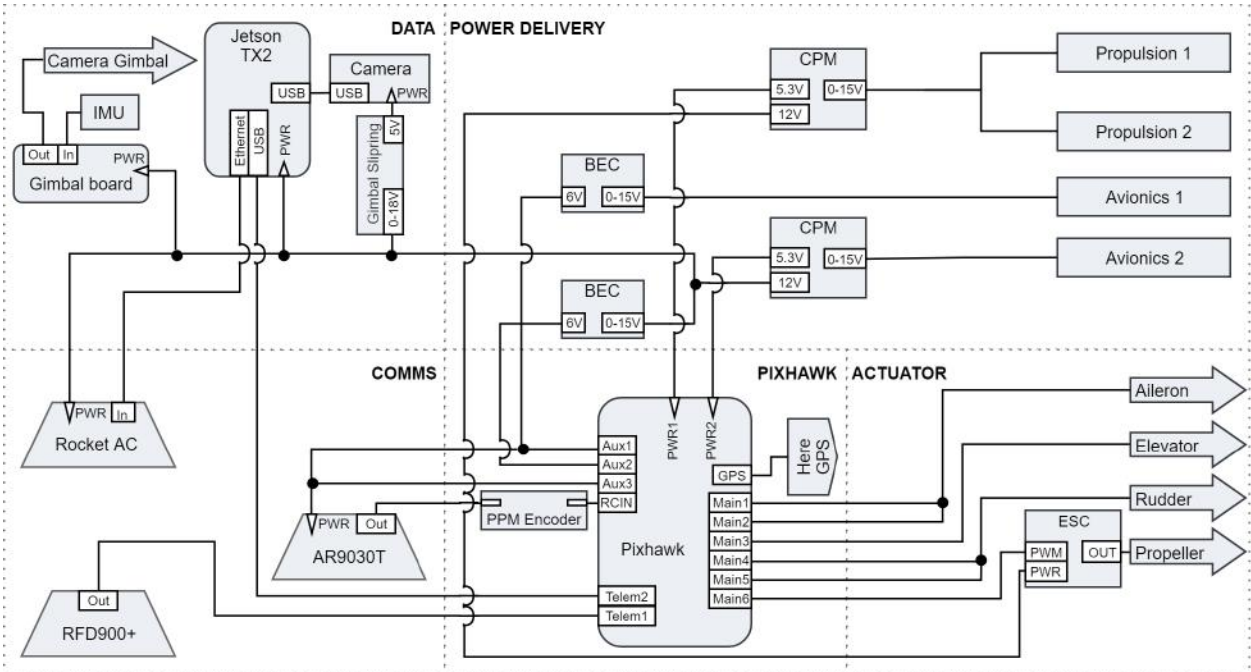

Figure 5: Wiring diagram. CPM (Common Power Module) is a proprietary power module for the Pixhawk. BEC (Battery Eliminator Circuit) is a switching voltage regulator. ESC (Electronic Speed Controller) powers the propeller motor via a PWM signal. IMU (Inertial Measurement Unit) provides sensor data for closed-loop gimbal control.

increases the overall reliability and robustness of a software system. Whereas for pure software failures, it is possible to achieve a degree of fault tolerance by checking for failure conditions of small pieces of code, however, it is often easy for an unexpected failure to still surface. Combined with this, hardware failures are also difficult to mitigate within a piece of software alone. Thus, in the UAV Austin software team's services and libraries, failure conditions are checked if they are likely to occur services are run using the container orchestrator Docker Compose with restart policies in the case they fail unexpectedly; and data, is saved in databases instead of in-memory if it must persist after a software failure.

3.3.2 Ground Station Server

To ease the use of the software stack on the flight line, the team elected to deploy the services onto a portable, provisioned server on the flight line. This allows for the runtime configuration of services to be more consistent, as opposed to running on personal laptops. The server has been mounted in a portable server rack, along with a power strip, router, and networking switch already connected together to decrease setup time and error on the flight line.

3.4 Obstacle Avoidance

The team decided on an automated system of performing obstacle avoidance due to the sheer number of obstacles and limited manpower and reaction time. Given a list of waypoints, the obstacle avoidance system moves the waypoints to generate a path that avoids all obstacles. To accomplish this, the team developed a novel obstacle avoidance algorithm named TAN* (Tangent-Assisted Navigation).

3.4.1 Algorithm

TAN* takes advantage of obstacle geometry to achieve efficient and optimal path generation. By leveraging the circular shape of obstacles, the algorithm can restrict the search space for optimal paths to tangent lines. Furthermore, through treating each tangent as a directed edge and their endpoints as vertices, the competition field elegantly transforms into a weighted directed graph. The best path can then be calculated using the A* algorithm, a highly efficient method for finding the shortest path in a graph using distance heuristics as guidance. This serves as a significant improvement over last year's model because the path exploration can be done in continuous space rather than discrete space.

The team devised an intuitive solution to generate

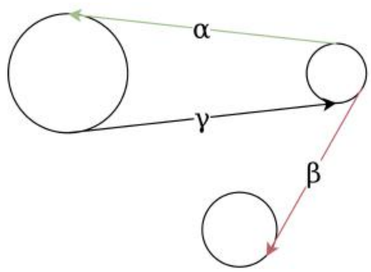

Figure 7: Vertex Polarity

smooth paths in order to minimize sudden turns and instantaneous changes in heading by taking advantage of the circular obstacles. For instance, Figure 7 shows the UAV travelling along edge γ to enter an obstacle from the *right side* with respect to the obstacle. The plane can then slingshot around the obstacle and travel along edge α but cannot travel along through the *left side* edge β due to a sudden turn. Hence, each vertex (i.e the endpoints of a tangent) is marked with *left* or *right* polarity so that TAN* only generates paths between vertices of the same polarity. Consequently, all paths generated are guaranteed to be smooth, continuous paths which the UAV can traverse.

3.5 Imaging System

3.5.1 Camera Selection Rationale

The camera for Phoenix III, the Z CAM E1, was chosen to maximize image resolution at higher altitudes and transfer images quickly during flight. A major advantage of the camera is its HTTP API that is used to capture images. The team's software employs HTTP APIs, allowing for easier integration with the current systems and making it easier to control remotely and capture images. The camera weighs 12.4 oz which is lighter than the camera used last year. A minimum resolution of 13 pixels/ft was necessary to get a clear picture of an object and identify it, this was dependent on several factors, such as flight altitude and lens configuration. A significant effort was made to ensure that the camera provides sufficient clarity: its resolution is 4640 by 3480 pixels. Another variable that was taken into account was the lighting conditions during flight: the Z CAM E1 has a 16-megapixel Ambarella A9 image processor, which enables low-light performance, giving the camera a wider operating range. The camera has an autofocus feature, allowing photos to be taken in a single, continuous center and flexible line. The lens was chosen based on the focal length and shutter speed. The Olympus M.Zuiko 25 mm f/1.8 interchangeable lens features a 25 mm focal length and a 47° diagonal angle of view, which meets the resolution of 12 pixels/ft needed for the image recognition system.

3.5.2 Gimbal Hardware

The main requirements for the gimbal are to obtain the necessary viewing angles for camera operation, which allows for viewing an object outside the flight boundary, and minimize drag. Thus, the nose cone is entirely comprised of the gimbal. The spherical part of the gimbal houses the camera and a Inertial Measurement Unit (IMU). The cylindrical piece, which connects the camera housing to the electronics housing, holds the Tiger GB4106 Brushless Gimbal Motor, which controls the pitch. The electronics housing holds the Tiger GB36-2 Brushless Gimbal Motor, which controls roll alongside a BGC 3.1 MOS Large Current 2-Axis Brushless Gimbal Controller. The team incorporated a bearing and slip ring between the electronics housing and cylinder to ensure smooth movement, 360° rotation in pitch and roll, and improve the overall structural integrity. To save weight, the spherical housing was manufactured using carbon-infused nylon[c]. The cylindrical piece and electronics housing were manufactured using a fiberglass outer surface and a laser-cut plywood internal structure to further reduce the weight.

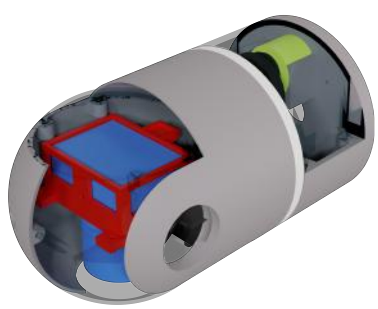

Figure 8: Gimbal Hardware Assembly

3.6 Object Detection, Classification, Localization

The autonomous target recognition algorithm was designed to receive, classify, and submit targets to the interoperability system. No target will be submitted that does not have both a shape and an alphanumeric identified with high levels of confidence. Two software members will do manual image recognition on the flight line to identify the emergent target and targets the algorithm might not have picked up.

[c]A Markforge 3D printer was used to overlay layers of a nylon thermoplastic and carbon fiber to create a stronger, lighter part as opposed to a part printed with solely PLA plastic.

3.6.1 Autonomous Classification

The classification phase determines all of the characteristics of a target. The full image is first cropped into overlapping sub-images. These sub-images are sent through a convolutional classifier which determines if the sub-image contains a target. This step prevents empty sub-images from going through the classification step, speeding up the overall classification time of the image. The target sub-images identified are passed through a Tiny YOLOv3 [7] object detection model which produces bounding boxes for shapes and letters. In the case that a shape gets split across multiple bounding boxes, the boxes will be coalesced into one large bounding box so as to prevent duplicated submissions. The bounding boxes of each sub-image are then merged using the model's confidence as a heuristic. The remaining bounding boxes which have both a shape and a letter are then used.

The classifier model was trained on 80k synthetic targets on background images from the camera. The object detection model was trained on 13k synthetic images for each shape and alphanumeric, totaling 37 classes. On a 50k synthetic image testing set, the model scored a mAP of 0.87.[d]

The autonomous classification is run by workers in parallel that take unprocessed images from a queue. Designing this to work in parallel allowed for classification time to be longer than the rate at which images are captured, making classification time less of a design constraint. This allows for a classification job to fail once, in the case of an error in requesting image data or in submission.

3.6.2 Manual Classification

Manual classification is done through a React frontend, which is designed for operator-friendliness. The *Explorer* page allows users to examine images and image metadata. The *Classifier* page allows users to select and classify targets in the image. Users can drag boxes on an image to designate targets. The right sidebar lets users specify the target's characteristics. The backend then passes this information to the interoperability system through the submission pipeline. The *Targets* page allows users to see the autonomously and manually submitted targets and to delete incorrectly submitted manual targets.

3.6.3 Localization

The team uses a yaw-pitch-roll coordinate system with respect to where the top of the camera is pointing as opposed to a traditional gimbal coordinate system with respect to where the camera lens is pointing. This choice in coordinate system makes the vertical direction of the camera in line with the pitch, and the horizontal direction in line with the roll, and is the same coordinate system as an aircraft with the gimbal pointing down orthogonally.

When a manual or autonomous target is found, the center of the thumbnail submitted is used for the location. The distance north Δx and distance east Δy of the target from the UAV are calculated using the altitude h, yaw ψ, pitch θ, and roll ϕ, as well as the pitch deviation $\Delta\theta$ and roll deviation $\Delta\phi$ from the position in the image

$$\Delta x = h\left(\tan(\theta+\Delta\theta)\sin\psi - \frac{\tan(\phi+\Delta\phi)\cos\psi}{\cos(\theta+\Delta\theta)}\right)$$
$$\Delta y = h\left(\tan(\theta+\Delta\theta)\cos\psi + \frac{\tan(\phi+\Delta\phi)\sin\psi}{\cos(\theta+\Delta\theta)}\right)$$

where $\Delta\theta$ and $\Delta\phi$ are found with the target location in the source image of width w and height h at pixel (i, j) and the horizontal angle of view α of the camera computed with the imaging sensor width d and focal length f.

$$\Delta\theta = \frac{\alpha}{w}\left(\frac{h}{2} - j\right) \qquad \Delta\phi = \frac{\alpha}{w}\left(\frac{w}{2} - i\right)$$
$$\alpha = 2\arctan\frac{d}{2f}$$

The UAV's latitiude and longitude at image capture are then offset with Δx and Δy to localize the target.

3.6.4 Target Submission

Instead of submitting targets directly to the interoperability system, autonomously and manually classified targets are submitted to a buffer. The buffer is watched for new targets and the targets are submitted in the order they are classified by a background task. Another background task watches for requests to delete targets. Autonomous targets are only submitted if they are not similar to one of ten already submitted autonomous targets. Autonomous targets are considered similar if they share any two of the shape, background color, or alphanumeric characteristics with another autonomous target. The cap of ten targets prevents the extra target submission penalty from taking off too many points in the event of many false or repeated target classifications.

The buffer allows for targets to be able to be queued for submission in the case that the interoperability server is unavailable. In the event that backup

[d]The mAP, or mean average precision, is the mean of the average precision for each class.

target submission over a flash drive is required because of an unexpected fault of the interoperability server, an archive of the targets in the buffer may be downloaded from an API using the flash drive target submission format.

3.7 Communications

On the flight line, establishing a solid communications infrastructure has proven crucial to converting technical accomplishments into tangible points through the submission of data to the interoperability system. With active monitoring, consistent design, and correct use of automation, the reliability of ground and air equipment can be easily measured, asserted, and maintained.

3.7.1 Communications Hardware

The communications stack is composed of four radios with different functionalities: the RFD900+ Modem, 3D Robotics 3DR Radio, AR9030T DSMX receiver, and Ubiquiti Rocket AC. The RFD900+ Modem allows for Pixhawk communication over a long distance. The 3DR Radio allows the Pixracer on the UGV to communicate with a Mission Planner instance. The AR9030T DSMX is a telemetry receiver that receives commands from a DSMX transmitter to control actuators in the aircraft. Lastly, the Ubiquiti Rocket AC connects the NVIDIA Jetson to the ground station access point. Figure 9 shows each radio within the entire communications hardware stack. A router in the ground station functions as the access point, and the RFD900+ Modem and 3DR Radio are connected to the ground station server.

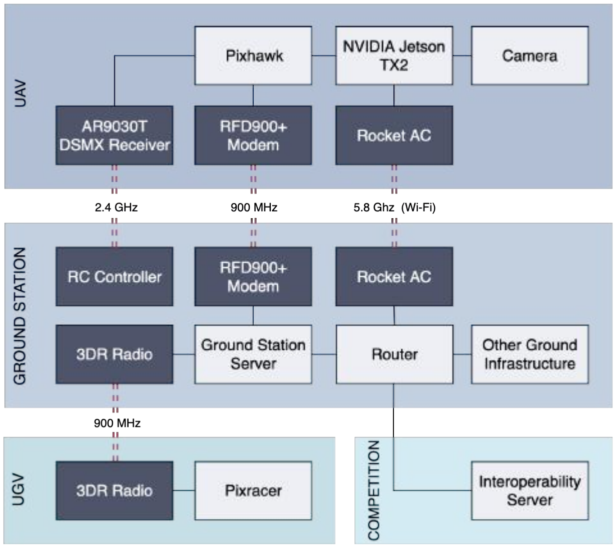

Figure 9: UAS Communications Hardware

The Rocket AC antenna has both a helical and omni-directional antenna. The RFD900+ Modem has a helical antenna and a dipole antenna. Lastly, the 3DR Radio has only a dipole antenna.

For communications stability, the RFD900+ modem and AR9030T DSMX receiver are oriented at 45° away from the Rocket AC transmitter to reduce interference. When the UAV rolls, the signal strength between the ground station and the AR9030T DSMX receiver is significantly weakened. To address this, the receiver connects to three separate satellite antennae on the nose and both wings. In addition, the GPS module (Here GNSS) is enclosed within a Faraday cage to ensure accurate functionality.

3.7.2 Antenna Tracker

The team developed an antenna tracker in order to consistently point the directional communication antennae at the UAV and prevent the need for manual redirection of the antennae. The antenna tracker uses two separate servos to facilitate the tilting and panning motion. The servos are operated through a Maestro servo control board connected to a laptop running a Mission Planner instance. The servo control board uses the GPS data position of the UAV to orient the antenna tracker towards the aircraft.

3.7.3 Software Communications

MAVProxy streamlines concurrent communications between telemetry, Mission Planner, and the UAV by exposing the telemetry via a freely accessible UDP socket. To mitigate congestion and ensure reliability, MAVProxy prioritizes telemetry communications via the Rocket AC over the RFD900+, which has limited bandwidth.

A telemetry service exposes an HTTP API that allows other services or a frontend to retrieve GPS information, as well as modify the aircraft's waypoints in real time.

3.7.4 Monitoring

A monitoring service is dedicated to performing API and ICMP ping health checks for a defined list of services and devices (including the interoperability server). It sends these results to an InfluxDB time-series database, which in turn is visualized in an easy-to-understand Grafana dashboard, as shown in Figure 10. This supersedes the text-based dashboard developed for the 2018 competition, which was limited in both function and utility.

The dashboard helps to detect and prevent extended communication dropouts by alerting the ground station team when a communication problem arises. The telemetry upload rate displayed on the

Figure 10: Communications Dashboard

dashboard breaks down the upload rate by unique and total telemetry messages sent. Even though uploading non-unique telemetry to the interoperability system does not count towards the upload rate for the competition, it is useful because it shows an active connection to the interoperability system.

3.7.5 Protocol Buffers

To minimize message sizes, services communicate with each other exclusively through Google's Protocol Buffers over HTTP. Requests and responses can also be made in JSON to facilitate debugging. The frontend uses Protocol Buffers over WebSockets, a slim wrapper for raw TCP connections used in browsers, for real-time communication with backend services.

Despite the judge's comments from the 2018 competition, the team decided not to pursue gRPC due to its limited software support within the Elixir, Rust, and asynchronous Python [e] ecosystems, whereas communication over pure HTTP boasts the benefits of simplicity and universal support in the ecosystem of any language, with the ability to use plain text and JSON when appropriate in testing.

3.7.6 Interoperability Server Communications

A proxy service written in Elixir serves as a layer of abstraction that facilitates communication between the UAV Austin software stack and the AUVSI SUAS software stack. It accepts telemetry data from the aircraft and forwards the data to the interoperability system conforming to the interoperability system specifications. The proxy service combines target submission with an thumbnail image into a single endpoint, and handles the selection of the current mission, so that other services do not need to handle this logic. For this competition season, older systems have been updated to use the proxy service for their interoperability server communications, to combat communication problems found on the flight line during the 2018 competition year while selecting the active mission in services not using the proxy service.

[e] While Python is officially supported by gRPC, it still does not support the *de facto* `async`/`await` syntax, complicating its use in Python `asyncio` projects.

3.8 Cybersecurity

As discussed in a 2018 Homeland Security report on UAS security, cybersecurity in the UAS domain remains relatively underdeveloped, and concerns err towards the defense of property from unauthorized UAS use as opposed to UAS security [6]. Nevertheless, a UAS is susceptible to compromise during field use, such as in the 2011 crash of a CIA drone in Iran caused by GPS spoofing, a serious vulnerability in UAS technology demonstrated in 2008 by Humphreys [3]. With correctly mounted equipment, GPS spoofing is a *straightforward* attack that is becoming increasingly lucrative with potentially disastrous consequences [3].

While it would be impractical to develop countermeasures against GPS spoofing for Phoenix III, there are other practical risks and mitigations, as listed in Table 3.

A significantly more practical vulnerability is gaining access to the network via a rogue Rocket AC, which would grant free reign for collateral damage. Even a fish tank proved enough of an attack vector to allow exfiltration of data from an American casino in 2017 [5].

Nevertheless, the best approach to mitigating security issues is developing a methodical approach to software development and testing; UAV Austin has followed this spirit of consistency through software sprint planning, protected Git branches, and continuous integration.

While implementing SSL/TLS and shared secrets (private keys shared between services) would be important steps toward securing service communications from man-in-the-middle attacks, the team has assessed that the risk in the context of the competition is far too low to implement an automated public key infrastructure (PKI) and associated authentication system.

3.9 Air Delivery

The air delivery system consists of 2 main hardware systems, the UGV and the delivery system. There have been five major iterations of the UGV, each aiming to reduce weight and increase internal space. The final weight of the vehicle package with the water bottle payload is 43.2 oz. The delivery sys-

Target	Attack method	Impact	Response
Wi-Fi access point	Jamming	No impact: Wi-Fi access point is only used for convenience and fallback purposes	None needed; all ground station connections are wired
Rocket AC	Jamming	Medium: Loss of communication with the plane	Manual takeover
	Interception	Very high: Grants access to telemetry and MAVProxy, which allow full control of the plane	None possible
RFD900+	Jamming	Medium: Loss of communication with the plane	Fallback to communication via Rocket AC
	Interception	High: Control over UGV drop	
GPS	Jamming	Medium: Loss of autonomous and RTL ability	Manual takeover
	Spoofing	Very high: Can put UAV in danger before manual takeover	None possible
Switch	Physical access	Very high: Grants full access to network	Enable MAC filtering on router to evict unauthorized device

Table 3: Likely attack vectors and appropriate responses

tem has had two major iterations, improving the system's functionality. Most parts are made by 3D printing with PLA.

3.9.1 Unmanned Ground Vehicle (UGV)

The frame of the UGV comprises of three pieces: a main chassis, an integrated parachute capsule, and a back panel. All frame components have load-bearing lugs to sustain loads during the mission, and the parts are connected securely by L-shaped pins for ease-of-access during maintenance. All load-bearing components are shelled and have stiffeners added to save weight without compromising structural integrity.

Within the UGV, there are four main subsystems (Figure11): the parachutes, drivetrain, electronics array, and payload. The parachute system lies at the front of the vehicle. The UGV drivetrain has only two wheels that steer by differential steering because the team determined that having four wheels would introduce complexity and add weight. To compliment the wheels in the front, a fiberglass composite skid is attached to the back of the main chassis. The electronics array consists of a shelf fastened in the center of the frame that holds the power module, motor controller, and an mRobotics Pixracer. The GPS sits on the back panel, and the radio sits on the floor of the main chassis. The payload has its own designated space in the back of the UGV with a hook-and-loop strap that can fasten it to the frame.

The main electronics systems, including the Pixracer, power module, and GPS, were chosen for

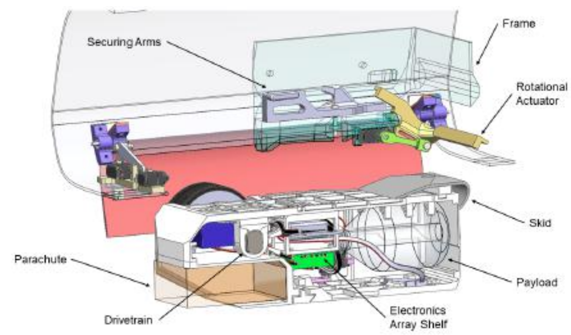

Figure 11: Delivery and UGV System Diagram

their versatility and the team's familiarity with the product. Additionally, those systems could be integrated with the team's existing ground control infrastructure. The motors were chosen based on size and torque output, along with their corresponding motor controllers.

3.9.2 Delivery System

The ejection system consists of a frame, two securing arms, and is servo-actuated (Figure 11). The actuator is made with carbon-infused nylon to resist bending moments at joints. The two securing arms are under constant torque from rubber bands in the unlocking direction to release the UGV. The rotational actuator provides a barrier to prevent the arms from opening. At the time of release, the servo rotates the actuator, allowing the rubber bands to open the arms.

After the UGV is released from the UAV, a pilot parachute opens by air pressure, pulling out the main parachute from the parachute capsule. The main parachute takes, on average, 0.9 seconds to open once pulled from storage. The parachute is attached above the center of gravity of the UGV via an internally-secured, actuating pin, so the upward drag force will result in the UGV falling approximately horizontal to the ground. The maximum measured impact acceleration was determined to be 1.33 g's from parachute-assisted drop tests from a 45-foot building. Each wheel and major impact points of the skid were successfully loaded with 6 lbs of sandbags (2 g's of force at maximum weight) each to test durability.

To determine when to release the UGV, a numerical method is used to calculate the trajectory of the descending UGV in three dimensions, incorporating various wind parameters, varying drag force from the parachute, and the velocity at which the vehicle was released. The simulation uses kinematics and drag equations to accurately predict the vehicle's position, velocity, and acceleration at a given time during its fall. The team created a flight path from the data. With initial conditions of a 164 ft drop height, 36 KIAS horizontal initial velocity, and 2.74 kt northeast wind with the UAV flying north, the UGV takes 9.6 seconds to land with a ground travel of 109 ft.

4 Safety, Risks, and Mitigation

UAV Austin prioritizes safety during the development, testing, and operation of the UAS. In order to recognize and mitigate risks, the team created various checklists and procedures that account for realistic abnormalities.

Gantt charts are utilized to coordinate tasks and ensure critical deadlines are met. Checklists and procedures are created ahead of mission tests for flight preparation, transportation, and configuration. The upcoming sections illustrate the two main areas of risk—developmental and mission—and their divisions. Table 4 is the Fever chart that classifies the likelihood and severity levels of risks. Severity level 1, or *marginal*, creates a short developmental setback of zero to two days from low degree equipment damage. Severity level 2, or *significant*, refers to low degree of personnel injury, and small equipment damage leading to three to five days of setback. Severity level 3, or *critical*, refers to a medium degree of personnel injury, and structural damage that does not allow for immediate flight, thus having one to two weeks of setback. Severity level 4, or *catastrophic* refers to a long setback from a permanent personnel or equipment damage.

		Severity Level			
		1	**2**	**3**	**4**
Likelihood	**Rare: 1**	1,1	2,1	3,1	4,1
	Unlikely: 2	1,2	2,2	3,2	4,2
	Probable: 3	1,3	2,3	3,3	4,3
	Likely: 4	1,4	2,4	3,4	4,4

Table 4: Fever Chart: Definition of Severity

4.1 Developmental Risks and Mitigation

Developmental risks are risks during the design and development of the UAS, and the mitigations of these risks emphasize personnel safety. Table 5 describes the main developmental risks, their severity, and ways to mitigate them.

4.2 Mission Risks and Mitigation

The following outlines the risks encountered at mission tests. Table 6 depicts the major risks and how the team seeks to mitigate them. For mission tests, checklists and procedures are created for flight preparation, material loading, transportation, configuration, and takedown. For example, the flight plan, designed for both pilot and ground crew, lists the objectives for a specified flight, and records the ones that are not met in order to prepare for the next flight.

4.2.1 Checklists

The team employed a set of checklists to simplify preparation and provide uniformity for flight. The checklists serve to assist the members in ensuring that all necessary steps are performed in the correct order, so that the UAS is properly prepared for task execution.

Since checklists have a human interface component, many factors such as readability, functionality, and length need to be considered. In order to develop efficient checklists, the design rationale adhered to the criteria outlined in the NASA research paper detailing cockpit checklist concept, design, and use [1]. This includes factors such as subdividing the checklist into smaller lists so that the user does not get lost in the process or having the checklist show the desired status or action for the item, to name a few. There were a variety of different checklists developed for the UAV, Ground Station, and the Mission Planner Pilot.

Risk	Description	Severity	Mitigation	Alternative Solution
Loss of Vehicle	Complete loss of UAV due to mechanical or communication anomalies	4,3	Communication Redundancy and check attachment points and linkages before flight	Outfit a Senior Design UAV and execute integration and HITL tests
Not enough funds	Cannot buy critical UAS components	4,2	Track expenses in a spreadsheet	Fundraise for more money
Operational injury: Chemical and Mechanical	Chemical injury from battery defects or fire. Mechanical injury from propeller defect or direct contact	3,2 - 4,2	Implement mission testing safety protocol, ensure everyone is trained on proper UAV assembly and handling	Ensure testing is done as a group
Manufacturing injury: Chemical and Mechanical	Chemical injury from composite toxic fumes and adhesives. Mechanical injury from lab equipment	3,2 - 4,2	Execute lab safety protocol, wear PPE, mandate safety training, and elect a safety officer	Seek help from experienced leads or peers
Design or Manufacturing delays	Critical UAV hardware/software is not finished on time	3,3	Set deadlines and perform regular checkups	Focus on functionality over complexity or aesthetic
Aircraft is overweight	The UAV's weight exceeds estimations, affecting performance	3,3	Track component mass and location in mass properties spreadsheet	Optimize motor and propeller size
Electrical Malfunction	Excessive voltage or current damages electrical components	3,3	Ensure the correct voltage is supplied. Follow a circuit diagram for installation	Have backup electronics
Inclement Weather	Inclement weather delays flight testing	2,2	Have backup flight test days	Improve flight test scheduling
Unbalanced Center of mass (COM)	COM must be within a margin of error on the quarter-chord of wing	1,3	Rearrange internal components using VSP model	Add ballast as a last resort

Table 5: Developmental Risks and Mitigations

5 Conclusion

UAV Austin applied the systems engineering and iterative design approach to design, manufacture, and test Phoenix III for the 2019 AUVSI SUAS competition. Each mission test was conducted to ensure that the UAS performs efficiently and safely, and meet all mission requirements. Throughout the development and testing cycles, the team gained valuable hands-on experience with the UAS and systems engineering, ultimately preparing them for the cutting-edge field of engineering. UAV Austin looks forward to participating in the competition with its new UAS.

6 Acknowledgments

Special thanks to all technical writers who contributed to this paper: Adam Ali, Kiara Alvarez, Santosh Balachandra, Bradley Bridges, Rahul Butani, Anthony Chang, London Darcé, Lalith Gannavaram, Benjamin Jesseph, Mihir Kamble, Hyeonseung "Shawn" Lee, Dillan McDonald, Giavanni Lewis, James Manimala, Aaron Park, Bennett Ramirez, Nicholas Riveira, Jacob Roy, Marc Sanchez, Shrivu Shankar, Rhythem Sharma, Pranav Sidhar, Evan Wilde, Paige Williams, Alex Witt, Haoyang Yu, Tommy Zhang, and Steven Zhu.

Special thanks to Dr. Efstathios Bakolas, Dr. Ar-

Risk	Description	Severity	Mitigation	Alternative Solution
Loss of Manual Control	The ground crew may lose control by exceeding the UAV's max range or battery depletion	4,3	Range tests before flight. Test battery voltages before flight. Plan to land with a 20% battery reserve	Safety pilot controls the UAV. If connection fails the UAV returns home. After 30 seconds the UAV fail safes
Pitot Tube Failure	Pitot tube does not return accurate value for airspeed	3,2	Pitot tube will be calibrated and tested before flight	Change setting in flight control software to rely on GPS for ground speed
Loss of Wi-Fi Link	Wi-Fi link is for imagery and is primary telemetry link	3,2	Use antenna tracker and ensure clear line of sight	Rely on RFD900+ failover link for telemetry. Download imagery from UAV after flight for processing
Loss of RFD900+ Link	The failover telemetry link	3,3	Maintain clear lines of sight and use USB current booster	Initiate safety pilot takeover if the primary Wi-Fi link is also down
Failing to meet the mission time limits	Failing to meet the 20 minute setup or 45 minute mission time limit	2,2	Mission will be rehearsed before competition. Ground station will record time for setup and mission	If the 45 minute mission time limit is about to be reached, cease all mission objectives and land
Wind Interference	High winds will alter UAV's flight path and may cause it to miss waypoints or violate no-fly zone	3,2	Monitor wind direction and speed. Adjust the flight plan to ensure the UAV does not violate the no-fly zone	If no-fly zone is violated, quickly return inside boundary. If a required waypoint is missed, adjust flight plan to hit it
Air Delivery Mechanism Malfunction	Malfunctions from servo burnouts or frame interference	3,3	Perform testing to ensure proper functionality	Perform the rest of the mission as usual

Table 6: Mission Risks and Mitigations

mand Chaput, Mark Wilde, and Mark Maughmer II for design, manufacturing, and operational expertise and advice, and to the Department of Aerospace Engineering and Engineering Mechanics at The University of Texas at Austin for logistical and financial support.

References

[1] Earl L. Winer Asaf Degani. Cockpit checklist: Concepts, design, and use. *Human Factors*, 35(2):345–359, 1993.

[2] AUVSI Seafarer Chapter. Competition Rules | SUAS 2019. `http://www.auvsi-suas.org/static/competitions/2019/auvsi_suas-2019-rules.pdf`, 2019.

[3] Todd E. Humphreys et al. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *Radionavigation laboratory conference proceedings*, 2008.

[4] INCOSE. What is systems engineering? `https://www.incose.org/systems-engineering`, 2019.

[5] Lee Mathews. Criminals hacked a fish tank to steal data from a casino. `http://archive.is/cyBEw` (archive), July 2017.

[6] U.S. Department of Homeland Security. Cybersecurity risks posed by unmanned aircraft systems. `http://archive.is/h2u1z` (archive), May 2018.

[7] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.