

UNIVERSITY OF CALIFORNIA, SAN DIEGO  
TRITON UNMANNED AERIAL SYSTEMS  
2022 STUDENT UAS COMPETITION

---

## Technical Design Paper

---

*Project Manager:*  
*Airframe Leads:*  
*Embedded Leads:*  
*Software Leads:*

Allyson Chen  
Allyson Chen/Brandon Vinh  
Hector Montenegro/Vishaal Vasu  
Tyler Lentz/Anthony Tarbinian

*Competition Team:*

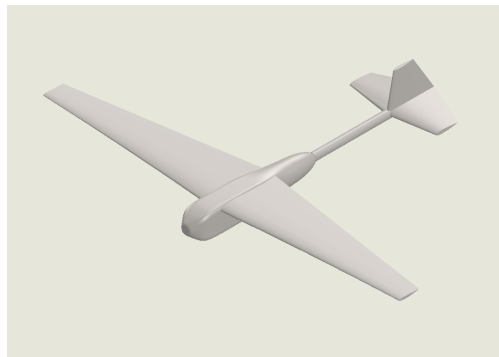
Brandon Vinh, Tyler Lentz, Anthony Tarbinian, Hector Montenegro, Vishaal Vasu, David Nielsen

*Faculty Advisor:*

Dr. Ryan Kastner

### Abstract

Autonomous unmanned aerial systems have a multitude of applications across the package delivery, search and rescue, planetary exploration, and data gathering. Additionally, the requisite aircraft design, communications, path planning, and computer vision are broadly applicable to a vast range of fields. To continue advancing these fields into the future, students are presented the 2022 AUVSI SUAS competition in order to build excitement and experience in these fields. For this competition, Triton Unmanned Aerial Systems (Triton UAS) presents the Carbon Copy. The Carbon Copy is an autonomous, fixed-wing unmanned aerial vehicle that is capable of simultaneously completing computer vision and package delivery missions. The airframe is designed using multidisciplinary design optimization and validated using commercial tools. The communications and power distribution are handled using a combination of off-the-shelf components and custom-designed printed circuit boards. For path planning, the system uses a custom implementation of a rapidly exploring random trees method. The system is also outfitted with a mapping camera and onboard computer for performing computer vision using machine learning methods. Finally, the system includes an autonomously landing and driving unmanned ground vehicle that can be dropped and deployed during flight. Design results show that the system should be safe, efficient, and capable of completing the entire mission.



Carbon Copy CAD Model

# 1 Requirements & Acceptance Criteria

In order to complete all required competition tasks, the system must be able to autonomously fly through waypoints, avoid obstacles, drop an autonomous ground vehicle, and perform computer vision tasks. This requires a software system that dynamically recalculates waypoints, automatically captures and identifies ODLIC images, and uploads this information to the interop server. The captured images must have a minimum resolution of 40x40 pixels to resolve the targets, which was determined by inscribing characters from the MNIST dataset into shapes and determining the overall dimensions. The system also needs to have enough computation power to process the images using state-of-the-art neural networks.

Given that the system configuration relies on high bandwidth communication links to relay images back to the ground control station (GCS), the system must utilize gigahertz range physical links. As a consequence, an implementation of high bandwidth links and intense computation requires robust power systems for the payload. Thus, it is necessary to separate the power system for the motor and the payload to increase safety and reliability. In order to ensure the systems are powered throughout the flight, each power system must have enough capacity to remain active for a minimum duration of 35 minutes.

In order to carry the necessary electronics, the system must be capable of carrying a payload weight of at least 15 pounds. Furthermore, to allow for communication between the aircraft and the GCS, signals must be able to pass through the airframe with minimal attenuation. In addition, when flying at a maximum altitude that allows for the required pixel resolution, the system must be able to cover the entire search area with a maximum of 100 feet in between passes. Therefore, the system must be capable of a minimum turning radius of at most 100 feet and trav-

eling a distance of about 12 miles within 20 minutes. Given the distance and time requirements, the system must have a cruise speed of at least 36 mph and a range of 50 miles. To ensure safety, the aircraft must satisfy all of the stability coefficient constraints and have a sufficiently low stall speed of no greater than 25 mph. Finally, the system must both be manufacturable and transportable.

## 2 System Design

### 2.1 Aircraft

#### 2.1.1 Conceptual Design

To approach conceptual design, high-level design concepts are independently raised and addressed.

One concept is whether to develop a fixed-wing aircraft or a quadcopter. Quadcopters are more agile and reliable for payload drop. However, they lack the necessary range, speed, and payload capacity to attempt a full score. For these reasons, a fixed-wing aircraft configuration is chosen.

For fixed-wing aircraft, configurations include low-wing, mid-wing, and high-wing. Of these options, high-wing has the best stability and easiest payload access. Therefore, a high-wing configuration is chosen.

For takeoff method, the methods considered include hand launch, bungee launcher, cart takeoff, and belly landing, conventional takeoff and landing, and eVTOL (Electric vertical take off and landing). The Hand Launcher and bungee launcher methods are ruled out due to concerns of safety and reliability. Cart takeoff and belly landing reduces weight and drag, but require constant manufacturing of skids. eVTOL removes the need for a runway which is not a competition consideration and the additional rotors add significant weight and wing drag. Due to these considerations, a conventional takeoff and landing is

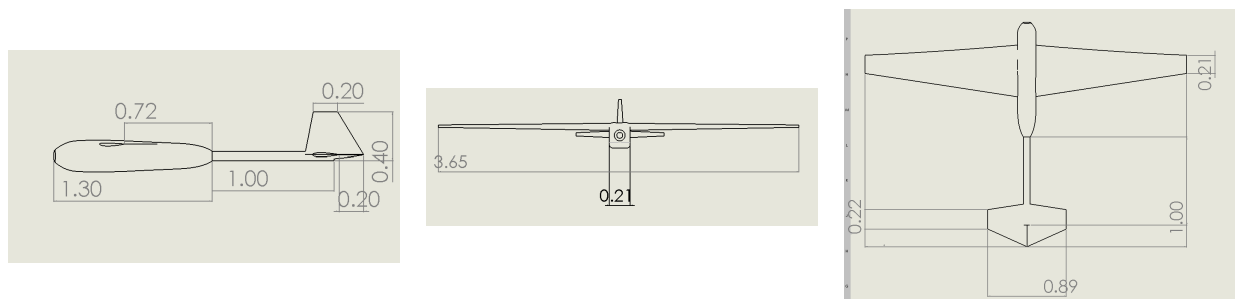


Figure 1: Side, back, and top view of Carbon Copy with dimensions in meters.

chosen.

### 2.1.2 Detailed Design

The detailed design of the aircraft is performed using a custom-built multidisciplinary design optimization program in Python using OpenMDAO. The optimization objective is to minimize turning radius and the constraints are defined by the system requirements. The models used include a density-based weight model, a classical laminate plate theory structural model, a quadratic lift and drag aerodynamic model, a static and dynamic stability coefficients model, a Breguet range model, and a mass properties model. The optimizer used to solve the optimization problem is SNOPT: a gradient-based sequential quadratic programming solver (SQP).

While the optimization does optimize the fuselage dimensions, it provides no insight to the fuselage shape. Therefore, a manual shape optimization process is used with Ansys Fluent, a computational fluid dynamics (CFD) solver, as the model. The objective is to minimize drag while also allowing enough space for the internal components.

The propulsion system is designed iteratively using a tool developed by the RC community. Real components are selected for analysis until a desired design is achieved. While it would be preferred to design a custom propulsion system, manufacturing constraints make this infeasible.

### 2.1.3 Design Results

The final design is a fixed-wing aircraft that is capable of supporting all mission tasks (**Figure 1**). For geometric design variables, the wingspan is 12 feet and the aspect ratio (AR) is 9.12. The horizontal stabilizer has a span of 2.92 and an AR of 2.5 while the vertical stabilizer has a span of 2.62 and an AR of 1.61. Finally, the tail boom is 3.3 feet long, and the fuselage is 3.6 feet long. For the power system, there is an outrunner motor mounted on the nose of the aircraft and an 8S2P LiPo battery pack powers the motor. The power system setup is listed in **Table 1**. An extra 4S LiPo battery powers the rest of the payload devices to isolate the devices from the Electronic Speed Controller (ESC) and motor. For the operating conditions, the aircraft flies at a cruise speed of 33.6 mph and a cruise lift coefficient of 0.725. For performance results, the system design objective, turning radius, is 48.5 feet. The lift-to-drag ratio is 15.16 and the cruise range is 60 miles. The stall speed is 21.5 mph and static stability margin is 0.2.

### 2.1.4 Manufacturing Plan

To manufacture the aircraft, a set of molds are manufactured out of machinable foam using a computer numerical controlled (CNC) mill. Once the molds are completed, carbon fiber and fiberglass wet layups are done to create the primary airframe components. To combine the components, the appropriate connections are bonded using epoxy and microspheres. Lastly, the additional structure and payload mounts are manufactured using a combination of 3D printing, laser cutting, and hot wire cutting.

Part	Item Name
Motor	Scorpion SII-4035 Brushless Motor 250KV
Battery	Tattu 5200mAh 4S 35C
Propellers	APC 18x12e
ESC	Phoenix Edge Lite 75 A

**Table 1:** Power system setup

## 2.2 Autopilot

In order to complete the competition challenges, the autopilot must have the following features:

- Waypoint Following: Necessary for the waypoint navigation
- Geofencing: Necessary to stay inside the competition area

To ensure mission success, it is imperative that the autopilot is well tested and reliable. In addition, an open source software is ideal since it expedites the debugging process and allows for potential improvements to the public firmware code.

The ArduPilot firmware running on the Pixhawk 2 board meets all of the constraints. While other Pixhawk boards only have one undamped inertial measurement unit (IMU), the Pixhawk 2 has multiple redundant dampened IMUs. In addition, it uses locking connectors for a more robust electrical system, and a highly accurate GPS.

## 2.3 Ground Control Station

The primary device for the GCS is an Intel NUC that acts as the hub for communications between the plane and all other devices on the ground. Any device can connect to the network and interact with the GCS as a webapp. Because the custom path planning

and computer vision systems are separate from the plane, the backend of the GCS handles communication between the subsystems, the plane, and the interop server. Each subsystem makes HTTP requests to the GCS in order to receive and submit the information relevant to its task. In turn, the GCS communicates with the plane via MAVLink messages and the interop server. This design provides a layer of abstraction between the different subsystems allowing for extensibility and modularity in the future. In addition, the GCS specifically does not include any autopilot specific functionality such as tuning the plane because it is only necessary during the development phase, and the Mission Planner software already provides an adequate implementation.



**Figure 2:** Early iteration of new custom GCS displaying the mission and current plane position.

## 2.4 Obstacle Avoidance

In order to complete the obstacle avoidance task, three things must be true about the generated path:

- The path needs to accomplish mission tasks
- The path needs to avoid stationary obstacles and potentially other planes
- The plane needs to be able to accurately follow the path

The software system achieves these three goals through the path planning server and the GCS.

### 2.4.1 Obstacle Avoidance Algorithm

The path planning server is set up as a basic HTTP server using Flask—a Python networking library. It receives information about the mission and outputs generated paths in response. This implementation

serves to decouple the managing of the mission (done by the GCS) and the actual path generation, while also allowing for mid-flight path changes in response to unexpected plane behavior or other planes flying in the area. In essence, this separates the logic behind completing the mission and obstacle avoidance.

In order to generate paths that accurately avoid both static and dynamic obstacles, the path planning server uses an RRT (Rapidly-Exploring Random Tree) based algorithm which connects randomly sampled points using Dubin’s curves with a fixed radius. This type of algorithm allows the pathing to be more flexible in the face of dynamic obstacles, and correctly works in edge cases that the previous obstacle avoidance algorithm could not handle. The use of Dubin’s curves ensures that the plane will be able to follow the ideal path closely, since it is based upon the actual movement of a fixed-wing plane.

### 2.4.2 The GCS as a Mission Manager

In this implementation, the GCS serves as the “mission manager” which controls the path planning server. As long as the ground control station knows how to interface with the path planning server, it can focus on making sure all of the mission tasks are completed.

In order to verify that mission requirements are not skipped, the GCS keeps track of which tasks have been completed, which are currently in progress, and which still need to be started. Additionally, before moving onto a new task, it confirms that the current one is completed. For example, the GCS verifies that all the waypoints have been reached before moving onto the next portion of the mission.

## 2.5 Imaging System

For an effective imaging system, the camera must be able to take precise images, easily integrate with other technology, and be within budget. Furthermore, it must also be lightweight to help maximize airframe efficiency and maneuverability. With this in mind, the main heuristics for choosing a camera are:

- Weight
- Hardware interfacing compatibility
- Lens capabilities
- Sensor quality and effective megapixels
- Cost

The Lucid Triton 20 MP Model with a 50mm lens best satisfied these heuristics. With a 20 MP sensor,

it achieves a 14.1 pixel per square inch resolution at a 400ft altitude, giving 93.8 pixels per foot of resolution for a target with a width of 2 ft. The on-board computer then sends images to the ground for completion of the Object Detection, Localization, and Classification (ODLC) task.

## 2.6 Object Detection, Localization, Classification

### 2.6.1 Manual Detection

To account for the inherent inaccuracy in a machine-learning approach, the autonomously detected targets can be overridden through manual classification by a human. To correct possible inaccuracies, an operator can review cropped images of detected targets. If the human detects an error in the machine’s classification, then they can correct it through the user interface and update the classification on the interop server for partial points.

### 2.6.2 Autonomous ODLIC System

The computer vision pipeline serves to autonomously complete all the specifications for the ODLIC task. In order to make this problem more manageable, the pipeline divides it into the following subtasks: saliency, segmentation, classification, localization, and target aggregation. The system is designed to have a central controller node that distributes tasks to workers, allowing for the parallelization of all components of the computer vision pipeline.

### 2.6.3 Target Generation

Mock ODLIC images are generated by creating a shape and character, then overlaying them onto a background image from previous test flights. These targets are automatically annotated with bounding boxes, their shape and character labels, and color information. In the future, this can be expanded to account for more artifacts from the camera such as poor focus, motion blur, and chromatic aberration. This generates tens of thousands of perfectly labeled training images, without needing to spend time on test flights or manual annotation.

### 2.6.4 Saliency

Both Spectral Residual Saliency<sup>1</sup> and Faster R-CNN<sup>2</sup> are possible approaches to saliency—isolating small targets within a larger image. Since the automatic classifier will fail if targets are initially missed, it is very important to have high recall. High recall allows the system to find more targets with the main downside being that it will also detect more junk targets. The comparison between the two approaches is summarized in **Table 2**.

Network	Precision	Recall
Spectral	0.33	0.06
Faster-RCNN	0.90	0.48

**Table 2:** Comparison of two saliency approaches on a dataset of 293 full size images that have 135 targets in total

Faster R-CNN has the advantage of both higher precision and higher recall, but Spectral Residual Saliency is generally much faster and requires less computational power. When the different methods were compared on the NVIDIA Jetson TX2 hardware, the Faster R-CNN proved to have superior accuracy while still being fast enough, so the first step in the computer vision pipeline utilizes this approach to salience images.

### 2.6.5 Segmentation

To improve classification of the targets, the system segments out the shape and character from a target crop received from the saliency component. The segmentation algorithm then converts the crop into two binary images, one where only the character is shaded and another where only the shape is shaded. This greatly reduces the noise in the image and makes the subsequent classification tasks much easier.

Shape and character segmentation is currently implemented using a neural network based on the UNet neural network architecture. While a UNet architecture was originally designed for biomedical image segmentation, it was adapted for target segmentation using transfer learning<sup>3</sup>. This adaptation of UNet outputs a trimap segmentation mask which classifies pixels in a given image as either being part of the shape, alphanumeric, or background. While previous implementations failed at handling edge cases,

<sup>1</sup><https://ieeexplore.ieee.org/document/4270292>

<sup>2</sup><https://arxiv.org/abs/1506.01497>

<sup>3</sup><https://arxiv.org/abs/1505.04597>

UNet can robustly handle a wide variety of segmentation scenarios, such as when the foreground and background colors are similar. The Sørensen-Dice coefficient—more commonly known as an  $F_1$  score—measures training loss and model performance in order to iteratively improve segmentation accuracy.

### 2.6.6 Classification

Using binary masks of both the shape and the character, the system classifies the shape using a comparison metric to reference shapes that make the classifier invariant to noise, rotation, and scale.

The Chars74k fonts dataset provides the training data for the character classifier because it closely resembles the output from the segmentation layer and contains all the uppercase characters provided by the competition. After removing all unneeded characters from the dataset, the training dataset is left with approximately 36,000 characters that are generated in various styles.

To further augment the dataset, each image is randomly rotated as it is loaded. This is required because there can not be any assumptions about the orientation of the character relative to the plane. Because this is randomly applied every epoch, the classifier will never see the exact same image twice in training, which helps reduce the chance it overfits. However, this does limit the maximum possible accuracy since many characters are identical to rotations of others. According to the competition rules, it is allowed to predict any letter-orientation combination that could be easily confused, so this is not a concerning issue.

Three different network architectures were attempted: transfer learning on VGG-16, VGG-11, and a much smaller convolutional network trained from scratch. This custom network has two convolutional layers with 200 filters each, and three fully connected layers with 1000 neurons, 100 neurons, and 25 neurons. These networks have three output layers, one for shape, character, and rotation. Using the same network for all classifications reduces the training required, since all three tasks will share low level feature extraction. All used the stochastic gradient descent (SGD) optimizer with a negative log-likelihood loss function for classification. The results after convergence are summarized in **Table 3**.

An RGB pixel value is calculated from the median of pixels inside the respective shape and character masks. A  $k$ -nearest-neighbors classifier identifies a color name based on its closest RGB value mappings stored in a large labeled dataset of colors.

Network	Test Accuracy
VGG-11	0.9235
VGG-16	0.9251
Custom CNN	0.9267

**Table 3:** CNN accuracy comparison

### 2.6.7 Localization

To determine the real world latitude and longitude of objects, several transformations are applied to convert the image coordinates of the object to its real world coordinates. The position of the object within an image is given by the saliency stage of the pipeline. These coordinates in the image plane are transformed using the intrinsic parameters of the camera, orientation of the camera, and geolocation of the plane. The resulting homogeneous coordinates represent where the target exists in a Cartesian coordinate space with an arbitrary origin which is then converted to its corresponding latitude and longitude.

## 2.7 Mapping

Due to the decision to allocate our limited resources towards refining the other components of the computer vision system, the mapping task has not been fully implemented. As of the writing of this paper, it is currently in its planning phase where various approaches are being researched and evaluated. One prospective implementation utilizes a neural network to detect unique features of images and the SIFT feature matching algorithm to stitch together two images at a time.

## 2.8 Communications

As shown in **Figure 3** there are three communication links between the GCS and the aircraft: a manual radio control (RC) link, an autopilot telemetry link, and a Wi-Fi data downlink.

The RC control system provides the ability for the safety pilot to override the autopilot in case of autopilot malfunction, and therefore this link is critical to the overall safety of the mission flight. To provide this link, the team uses the 2.4GHz-band FrSky X8R, a favorite of RC hobbyists due to its long range performance. In addition, it employs frequency hopping technology to prevent jamming and minimize the impact of interference.

The autopilot telemetry link provides real-time feedback of the autopilot state, as well as the ability for the autopilot operator to send commands and waypoints to the aircraft. The team selected the RFD900+ modem, which operates in the 900 MHz band, due to its demonstrated performance at 60 km without requiring high gain, directional antennas. The RFD900+ modem uses one 2dBi monopole antenna and one 3dBi dipole antenna for their omnidirectional radiation pattern, placed orthogonally for polarization diversity.

The Wi-Fi data downlink enables the transfer of images from the onboard computer to the GCS and provides an interface for the payload systems. This Wi-Fi link must maintain relatively high bandwidth at long ranges. Carrier grade wireless equipment from Ubiquiti Networks satisfies this requirement due to its ability to sustain a connection up to several miles. The team chose the Ubiquiti Rocket AC Lite over the Ubiquiti Bullet M due to its higher transmission power and multiple antenna connection points, which made the Rocket AC's higher price point a good investment. The Rocket AC utilizes two 3dBi monopole antennas for their omnidirectional radiation pattern, placed orthogonally for polarization diversity.

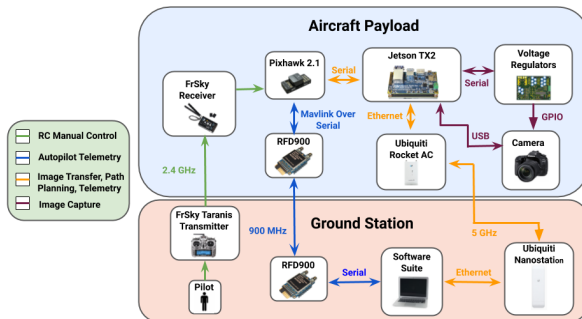


Figure 3: Communications systems diagram.

The team uses the Ubiquiti Nanostation M on the ground for its high gain 16dBi directional panel antenna to compensate for the significantly lower ranges associated with the 5.8GHz spectrum. Due to the high directivity of the antenna, it is vital that the plane is kept within the area of peak gain of the antenna. To accomplish this, an antenna tracker is placed at the ground station which uses the telemetry data of the plane and its own location to determine the proper pitch and yaw of the antenna to keep a strong connection with the plane. Testing at distances from 0 to over 1 km demonstrated that this pair of Wi-Fi modems possess data transfer up to

60Mbps, dropping to 3Mbps at longer distances. Figure 4 summarizes the results of the testing. However, even with this setup, the communication system will not be able to transmit the full sized images to the ground at a reasonable speed. Thus, the onboard computer is used to crop the images to only contain relevant data, which is then sent to the ground. This greatly reduces the bandwidth demands of the computer vision system.

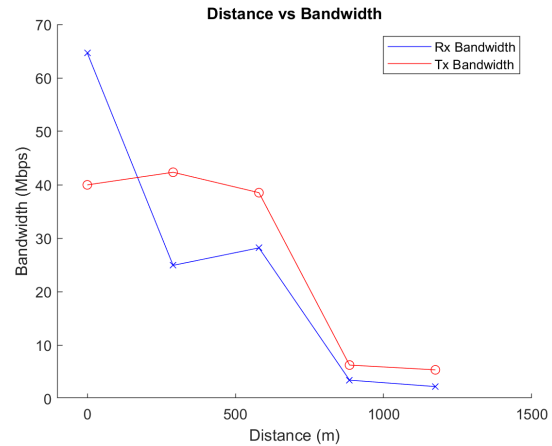


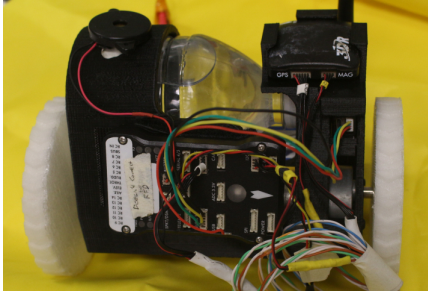
Figure 4: Distance vs Bandwidth of Ubiquiti Rocket AC to Ubiquiti Nanostation datalink.

## 2.9 Air Drop

The unmanned ground vehicle (UGV) needs to be lightweight, compact, and easily reproducible while staying within the competition and airframe constraints. Furthermore, since the UGV is frequently at risk of a failed airdrop, it is important that the chassis be easily adaptable and reproducible. To accomplish this, the custom chassis is 3D printed to include a 2-wheeled drivetrain and custom mountings for the electronic payload ( Figure 5 ). The components of the UGV payload are strategically arranged to minimize the overall volume of the system and allow for easy access to the electronic payloads. The UGV uses a Raspberry Pi Zero for the autonomous controller, a Here GPS for navigation, and a 433 MHz radio transceiver for communication with the GCS. For the power system, the UGV uses a small 3S LiPo battery, two ESCs to control the motors, and two low-cost brushed DC motors.

The UGV uses a Raspberry Pi Zero with custom software to serve as both the autopilot controller and the driving controller. The autopilot controller controls a pair of motors that actuate the parachute

strings in order to autonomously guide the system to the drop location. The controllers are designed by solving the infinite time linear quadratic regulator (LQR) optimization problem and tested using a custom-built simulation.



**Figure 5:** Unmanned ground vehicle prototype

### 3 Alternatives Considered

#### 3.1 Aircraft

The Fiber One, a previously designed and manufactured aircraft for competition a couple of years ago, was designed using an arbitrary design process. Additionally, the aircraft was not capable of carrying the UGV due to its smaller size.

The Swallow, a purchased pre-built aircraft, reduces manufacturing and assembly time. However, it cannot carry the UGV, is expensive, and is not optimized for the competition.

#### 3.2 Autopilot

The team chose the ArduPilot firmware over the PX4 firmware because its developers focus on flight-tested, feature-rich releases, while the PX4 developers focus on developing a flexible and easily extensible platform. Ultimately, the team decided that the features provided by ArduPilot satisfy the competition’s autopilot requirements, making the flight-tested nature of the ArduPilot firmware more valuable than the extensibility of the PX4 firmware.

#### 3.3 Obstacle Avoidance

The older method of using a pure Dubins based algorithm presented a viable alternative because it was already implemented and could potentially be improved. However, because the older codebase was extremely difficult to understand for new members, it proved to be more efficient in the long term to

implement a new algorithm that we were more confident could handle the various edge cases that would always be a challenge under the old approach. The main problem left was the possibility that the RRT algorithm would be too slow, but it seemed more feasible to optimize the time of such an algorithm rather than learn the old codebase and improve its accuracy.

#### 3.4 Imaging System

The camera options and attribute comparisons that were considered were shown in **Table 4**. Though the Canon EOS Rebel SL1 was the previous camera used and had worked in the past, it was within the budget to upgrade to a new and better camera. Among the camera options the team considered, the Lucid Triton 20MP Model provided superior weight and resolution while still remaining in our price range.

#### 3.5 ODLC

For the ODLC task, the primary design decision was between when to use machine learning methods as opposed to classical computer vision algorithms. It was decided to primarily use machine learning algorithms due to the experimental success that was achieved compared to the aforementioned computer vision algorithms.

#### 3.6 Air Drop

An alternative that was considered for the configuration of the UGV was using a 4-wheeled drivetrain. Ultimately, a 2-wheeled design is chosen in order to reduce weight and volume. Another consideration is an unguided parachute system, but the team opted for a guided parachute system in order to provide disturbance rejection and flexibility in the drop time.

## 4 Testing & Evaluation Plan

### 4.1 Developmental Testing

#### 4.1.1 Aircraft

To test the primary structure of the aircraft, a wing loading test will be performed. This test will constitute placing sandbags on the airframe wings to model the design loading case with a factor of safety of 1.5.

#### 4.1.2 Embedded

Substantial amounts of simulations are performed in order to verify initial designs of both the signals

Camera	Weight	Hardware interfaces	Focal length	Effective Resolution	Resolution at 400ft alt	Price
Lucid Triton 20MP Mode	2.36 oz	GigE PoE, GPIO M8 connector	50 mm	20 MP	14.1 px/in <sup>2</sup>	\$1166
Canon EOS Rebel SL1	14.36 oz	mini-USB, 2.5mm aux jack	40 mm	18 MP	3.8 px/in <sup>2</sup>	\$417
Sony A5100	9.98 oz	micro-USB	40 mm	24.3 MP	4.6 px/in <sup>2</sup>	\$548
Nikon D3200	16.05 oz	USB2.0, 2.5mm aux jack	40 mm	24.2 MP	4.7 px/in <sup>2</sup>	\$447
Samsung NX500	10.30 oz	USB2.0	40 mm	28.2 MP	5.3 px/in <sup>2</sup>	\$1,595

**Table 4:** Camera comparison

board and power distribution board within the plane. In development is an electronic load, so that the circuitry is able to be tested in a more robust manner.

#### 4.1.3 Obstacle Avoidance

In order to test the robustness of the RRT obstacle avoidance algorithm, there will be a combination of stationary and dynamic obstacle avoidance scenarios created. Stationary obstacles can be tested by generating mission configurations with a set of known obstacle locations and applying the obstacle avoidance algorithm accordingly. Dynamic obstacle avoidance will be tested by utilizing multiple instances of the ArduPilot Software in the Loop (SITL) to simulate additional UAVs. These adversarial UAVs will be provided sample missions to execute while the team’s UAV will be utilizing its obstacle avoidance algorithm. This will allow the team to create robust tests that allow for objective benchmarking of the overall performance.

#### 4.1.4 ODLC

A full ODLC pipeline will be created in order to iteratively test each component of the system. The system will be designed in a way such that each individual component for ODLC can be easily replaced, allowing for the team to test how a single component affects the overall performance for the ODLC task. Initially, this pipeline will be using generated images from the custom target generator that was created; however, once the team is able to execute full test flights, more test data will be gathered.

#### 4.1.5 Airdrop

To test the dropping mechanism, repeated tests will be performed while on the ground. Once the dropping mechanism has been validated, the dropping mechanism will be attached to the flight plat-

form and tested during a test flight. To test the autonomous landing system, the UGV, with its autonomous parachute, will be repeatedly dropped off the side of a building into an area without people. Once the landing mechanism has been sufficiently tested, tests will be conducted during test flights. For the driving mechanism, the controller will be tested by running sample missions in a local parking lot.

## 4.2 Mission Testing

Planned test flights will allow the team to assess the performance of the fully integrated system out in the field. During test flights, procedures, similar to ones that will be performed at the competition, are executed in order to examine performance of the systems. The team will develop a mission scoring system that utilizes the predefined formulas set by AU-VSI SUAS, allowing for direct comparison between previously performed mission tests.

## 5 Safety & Risk Mitigation

### 5.1 Developmental Risks & Mitigations

In order to mitigate the risks associated with developing a UAS, the team takes the following precautions:

#### 5.1.1 Plane Handling Procedure

The team has a reference sheet with clear instructions to decode the Here GPS indicator lights, RFD900+ modem lights, RC transmitter channels, and flight channels during test flights. This allows for quick diagnosing of errors and prevents unintentional throttle when plugging in wires and operating the RC controller. There is also a disarm policy in place; members do not approach the plane until the

GCS or pilot confirms that the throttle, servos, and shunt keys are disarmed.

### 5.1.2 Safety of Members

During the build process, all members involved in the manufacturing process of the plane are required to abide by certain safety rules. Risks to members include skin irritation from composite and process materials, eye injury, power tool related injuries, inhalation of dust and fumes, and slip and fall hazards. All members present during manufacturing are required to wear closed-toed shoes and long pants and have hair past shoulder length tied back when using equipment. Personal Protection Equipment (PPE) and adequate ventilation is provided to members during the appropriate tasks. For example, team members wear goggles and gloves during bonding of composite materials, and respirators during sanding or using chemicals. The project room is swept and cleaned after

every meeting to minimize the risk of slipping.

### 5.1.3 Airframe Visibility

The airframe is striped with black markings on the bottom of the wings to assist the safety pilot in orienting the plane. The battery packs are brightly colored to increase visibility in the event of separation from the airframe upon a crash.

## 5.2 Mission Risks & Mitigations

The team recognizes that flying an autonomous vehicle can be very dangerous and takes several precautions to ensure that missions are carried out safely. The mission safety risks and their respective mitigations are summarized in **Table 5**.

Risk	Probability	Severity	Mitigation Strategy
Miscommunication on flight line	MEDIUM	LOW	The team established detailed standard radio communication protocols to minimize the possibility of miscommunication during the mission.
Power failure in the GCS	LOW	MEDIUM	If the generator or connection to the generator were to fail, the team always keeps fully charged backup batteries on hand to power the necessary devices on the GCS. Furthermore, the team's laptops can hold a charge for longer than a full mission.
Onboard computer reboots or server crashes in flight	LOW	MEDIUM	Configured onboard computer to boot on power and set up server program as a service so it automatically restarts.
System integration issues	MEDIUM	MEDIUM	The team specified one member of the club to be in charge of all integration tasks. The team also made sure to leave time to test everything thoroughly.
Software system crash	MEDIUM	MEDIUM	The ground station provides monitoring for each software system, immediately alerting the operator to a software failure and allowing the software to be quickly restarted.
Poor signal quality between payload and ground	HIGH	MEDIUM	Payload operation to continue whenever connection is available.
Unexpected Throttle on ground	LOW	HIGH	The plane employs a safety shunt that requires a key to connect the motor to the power system. Without the key, the propulsion system is entirely disconnected from the power.
Payload components are improperly connected	MEDIUM	HIGH	Before takeoff, a preflight reboot shutdown, control surface checks, and throttle test are initiated, catching any connection issues before takeoff.
Autopilot failure	LOW	CATASTROPHIC	The team made sure that it is easy for the safety pilot to recover control of the plane, and made sure the fail safe of the plane is moving control surfaces correctly.
UAS crash	MEDIUM	CATASTROPHIC	The team purchased and assembled an exact copy of the primary airframe to minimize turnaround time in the event of a crash.

**Table 5:** Summary of mission risks and mitigations

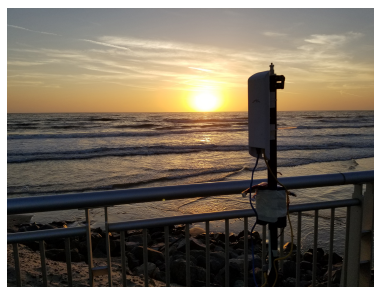
## 6 Acknowledgements

Triton UAS would like to acknowledge the generous support of our corporate and UCSD sponsors:



The team would also like to thank the following individuals for their continued support and guidance:

- Ryan Kastner Ph.D., Faculty Advisor, Computer Science & Engineering
- Mark Liu, ECE Makerspace Assistant Director, Electrical and Computer Engineering
- Colin Zyskowski, EnVision Assistant Director
- Eric Lo, Engineers for Exploration
- Jennifer Eller, Undergraduate Student Affairs Advisor, Structural Engineering
- Andrew H. Fletcher, Ph.D. student, UIUC
- Jessica Y. Chan, General Atomics
- Tim Wheeler Ph.D., Kitty Hawk
- Vinh Phan, Traptic
- Bryan Ritoper, Traptic
- Zack Grannan, OrderMetrics
- James Y. Wilson, Qualcomm
- Shane Grant, inVia Robotics



**Figure 6:** Wi-Fi Testing at at Del Mar Beach