Razgriz Technical Design
TJ UAV

2019 AUVSI SUAS Competition
Thomas Jefferson High School for Science and Technology

Abstract

The Thomas Jefferson High School for Science and Technology Unmanned Aerial Vehicle Team (TJ UAV) worked collectively to create the Razgriz, an aircraft that is capable of flying autonomously, detecting and interpreting targets, and delivering payloads to a drop zone while maintaining safe operation. Twenty-five high school students that specialized in different fields such as aerospace, physics, mathematics, and computer science came together to make the team's first AUVSI SUAS competition project a success.

Table of Contents

Systems Engineering Approach 1.0

Mission Requirement Analysis 1.1

The Razgriz system was developed to autonomously locate potential drop locations and deliver a payload to a customer. The overall mission requirements for the UAV are divided into six distinct sections. **Table 1** describes each task and the requirements for success.

Task	Summary of Sub-Tasks	Requirements for Success
Autonomous Flight (20%)	• Get as close to each waypoint as possible (50%). • Autonomously fly the UAS for at least 3 minutes. (40%) • Follow a given path of waypoints within 100 ft of each waypoint. (10%)	• Proper calibration and configuration of autopilot • Proper setup of flight plan
Obstacle Avoidance (20%)	• Prerequisite: Use the Interoperability server to submit 1 Hz telemetry while airborne. • Avoid stationary obstacles during the flight path. (100%)	• Radio that can receive telemetry at 1 Hz. • Algorithm capable of altering flight path to avoid obstacles. • Method of changing the waypoint commands on the autopilot. • Interface that allows manual creation and upload of waypoints.
Object Detection, Localization, Classification (20%)	• Use the Interoperability System to submit detected objects during the first flight. (30%) • Identify GPS location of objects. (30%) • Characterize each object by shape, shape color, alphanumeric character, character color, and orientation. (20%) • Autonomously provide the aforementioned information. (20%)	• Algorithm capable of autonomously detecting and classifying an object. • Accurate onboard GPS and localization algorithm. • High resolution and FPS Camera. • System capable of allowing team members to manually detect and classify objects.
Air drop (20%)	• Accurately drop the UGV. (50%) • Drive to the UGV destination location while carrying the water bottle. (50%)	• Successfully deploy the UGV with a clean parachute drop
Timeline (10%)	• Minimize flight and postprocessing time. (80%) • Avoid taking a timeout. (20%)	• Thoroughly rehearsed mission demonstrations
Operational Excellence (10%)	• Operational professionalism • Attention to safety • Communication between team members	• Strong leadership • Training of team personnel for safety measures

Table 1. *Mission objectives and their respective tasks.*

Design Rationale 1.2

Development Environment 1.2.1

As a first-year team entering the AUVSI SUAS competition, the team had to work to secure funding. In the winter of 2018, the team was awarded $3,829 from the TJ Partnership Fund as a grant.

Our development timeline allocated time for the training of new members, since part of the purpose of the club is to share the passion for flight with new generations. For freshmen, the first quarter of the school year was spent lecturing and working with remote control aircraft to gain the necessary background experience for SUAS.

Further, the scheduling of high school classes caused time restrictions. The team could only work during school for one 1.5 hour block per week, and could have a maximum of 28 members. In order to spend more time, the team held meetings after school for two 2 hour periods per week, as well as meetings outside of school for one 6 hour period per weekend.

To work within these monetary and time restrictions, TJ UAV decided to pursue a more conventional and reserved design philosophy and timeline.

Aircraft 1.2.2

To meet mission requirements, the aircraft's frame needed to be lightweight and space efficient with ample room for electronic components, a camera gimbal, the UGV, and the UGV drop mechanism. Additionally, the airframe had to be maneuverable enough to perform well in the Autonomous Flight and Obstacle Avoidance parts of the competition. Furthermore, it needed to be a stable platform for the onboard camera.

The Skywalker X8 airframe was generously donated to the team by the school. The X8's efficient flying wing design has a flattened fuselage and no tail section, which reduces drag and utilizes the fuselage as an additional lifting body. While the X8 is more unstable than traditional airframes due to the lack of a tail, the presence of a flight computer adequately controls the aircraft. The team believed the simplicity and spacious interior would help with wiring the numerous electrical components. Lastly, the use of a commercially available airframe was appropriate as the team lacked sufficient experience in designing an airframe from scratch.

The team determined that the X8 did not have the payload capacity for a full UGV, so instead opted for a simpler water bottle drop.

Flight Computer 1.2.3

To manage autonomous flight, the team decided it would need a flight computer that was simple and widely used. The Pixhawk 2.4.8 was chosen as the flight computer as several team members already had experience with the device.

The team also needed a companion computer to complete calculations for obstacle avoidance and object detection tasks, both of which are computationally intensive. For this reason, the team chose the NVIDIA Jetson TX2 for its NVIDIA high-end Pascal GPU architecture as well as 2-core CPU and 8 GB of RAM. Furthermore, the TX2 has extensive documentation available making it ideal for a custom application.

System Design 2.0

Aircraft and Electronics 2.1

The Razgriz is a modified Skywalker X8 flying wing, which has a large electronics/cargo bay, allowing for central location of most components, and is partitioned to separate the components into different areas. A custom nose cone was 3D printed and replaced the existing foam nose of the X8. To provide space for the UGV and drop mechanism, a hole was cut in the bottom of the nose. While the UGV would preferably have been placed directly below the center of mass to eliminate the possibility of the COM from shifting when deployed, the UGV was instead dropped from the nose to maximize the distance between the dropped payload and the propeller.

Several additional modifications were made to the rest of the X8 airframe. Holes and channels were cut into the wings to allow for the RFD900x and Ubiquiti Bullet M2 antennas to stick out. These were connected through SMA extension cables to the RFD900x and Bullet M2, which reside back in the fuselage of the Razgriz. Portions of the wing were then covered in a Monokote skin, which improved its structural integrity. Lastly, larger winglets were added for yaw stability.

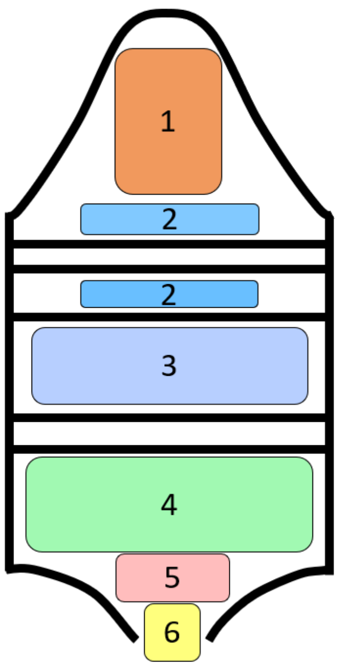

***Figure 1**. Razgriz Space Allocation*

1	UGV Bay
2	4S 5000mAh Battery
3	HobbyKing Quanum 2-axis Gimbal See3Cam-CU135 Alexmos SimpleBGC board Ubiquiti Bullet M2 RFD900x
4	3DR Ublox LEA-6H Pixhawk 2.4.8 NVIDIA Jetson TX2 APM Power Module
5	80A ESC
6	eRC BL46 600kv motor

Wing Parameters		Aircraft Dimensions and Specifications		Performance	
Span (in)	83.5	Gross Weight (lb)	8.6	Cruising Speed (mph)	40
Area (in^2)	1,240	Length (in)	36	Max Speed (mph)	68
Airfoil	NACA 4412	Payload Bay Volume (in^3)	582	Max takeoff Angle (°)	5
Leading Sweep (°)	27.3	Propeller	14x8	Landing Glide Slope (°)	3

***Table 2.** Physical Specifications*

The team originally planned on using the Ublox Neo-6M GPS due to its low cost. However, the team upgraded to a 3DR Ublox LEA-6H due to its greater signal strength and accuracy. This allows the ground station to track the Razgriz more accurately and ensures the autopilot has a strong GPS and compass reading at all times.

The team used an eRC BL46 600kv motor because it was readily available. During early testing of the Razgriz with 12x8 propellers, the team found that the aircraft could only take off at 100% throttle, even without a large payload. Thus, the team decided that a 14x8 folding propeller would be more suitable for greater takeoff thrust. The folding action of the propeller helps minimize the risk of breaking during a belly landing. The folding propeller also enables low drag cruising at idle throttle. The motor manufacturer recommends that the motor is used with a 65A ESC, and for this reason, the team decided upon a HobbyKing Skywalker 80A ESC that was also readily available. Two 4S 5000mAh batteries in parallel were calculated to be the largest safe battery configuration.

The team decided to hand launch Razgriz for the competition due to the team's lack of experience with launcher devices. The necessary precautions will be taken to ensure the safety of the launcher team member.

Autopilot 2.2

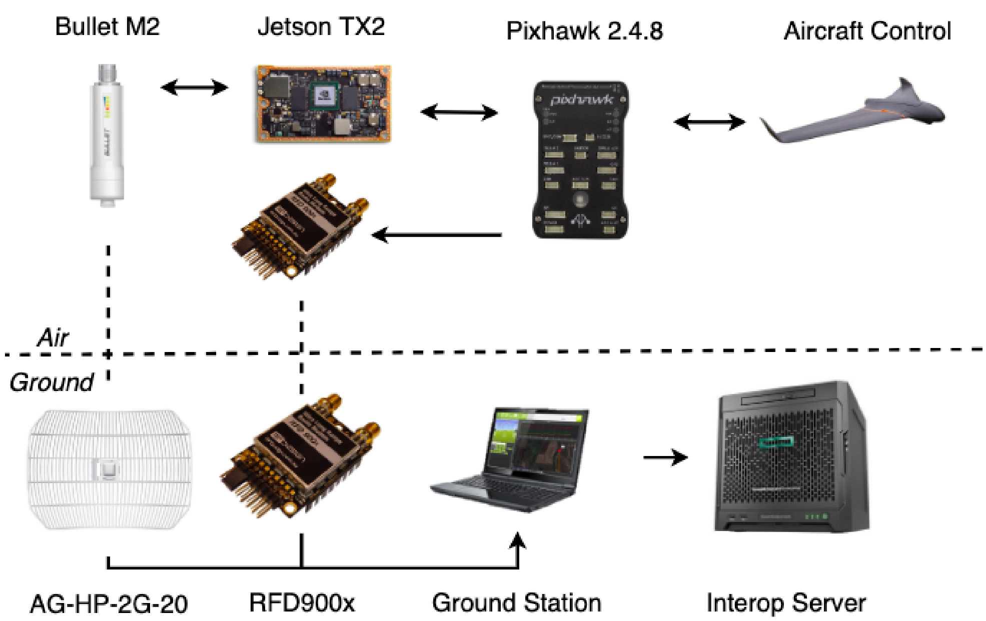

***Figure 2.** Autopilot System*

The Razgriz autopilot system uses a 3D Robotics Pixhawk 2.4.8 with all included sensors. The Pixhawk has out-of-the-box compatibility with a range of sensors, making it easy to test different components before integrating them into the Razgriz. The features of the Pixhawk also satisfy many of the competition safety requirements, including a hardware safety switch and the ability for telemetry with multiple receivers.

The Pixhawk runs a stock version of ArduPilot 3.9.6. ArduPilot was a good choice for the Pixhawk since it is the most popular open-source firmware for autonomous aircraft with a large developer community. The software has the capability for waypoint navigation and autonomous takeoff and landing, so the Razgriz will not need pilot intervention during normal operation. The Razgriz is configured for autonomous runway takeoff and landing, with GPS defining the location of these operations. **Figure 2** is a diagram of the autopilot system.

The team chose an open-source ground control software, Mission Planner, to use during the competition. This Ground Control System (GCS) allows the team to modify mission waypoints in flight, and view telemetry data. This data is relayed to the interoperability server by the team's custom companion software. This software also allows the operator to interchange mission objectives and relay these changes to Razgriz if necessary.

Obstacle Avoidance 2.3

The team developed the Razgriz Obstacle Avoidance Algorithm (ROAA) to complete the Obstacle Avoidance task. The team decided to use this algorithm to draw out the best alternate flight plan, rather than manually tracing a new flight plan. Additionally, one member of the competition team, an Obstacle Avoidance Monitor (OAM), will be monitoring the UAV position in comparison to nearby stationary obstacles using ArduPilot Mission Planner. In the event of a failure in the code or a disturbance to the plane mid-flight, the team can switch to manual flight mode and waypoint placement. In case of an emergency or problem with the UAV, the OAM has the ability to force the return or landing of the aircraft. This is to ensure the safety of mission personnel and the aircraft in emergencies.

The ROAA generates a complete flight path upon receiving obstacle data from the interoperability server. Predicting the exact flight path of the UAV based on its waypoints is important for verifying the plane's avoidance of obstacles. This task is accomplished by simulating the autopilot navigation of the plane through a set of given waypoints. Using the ArduPilot source code along with the dronekit documentation, the team was able to mimic the anticipated flight path of the Razgriz.

The ROAA performs all calculations on a GCS computer. This decision was made so that the OAM could observe the newly created waypoints. The ROAA examines the initial flight path created before takeoff for any vulnerabilities, such as the given obstacles and/or severe angles during turns. The ROAA then recursively subdivides the flight path into several segments, each of which are checked to ensure a safe flight. If an obstacle is detected to be obstructing the path, a new point is added at the midpoint of the blocked segment. The generated point is then shifted a distance proportional to the length of the blocked segment in the direction perpendicular to the segment. This process is recursively repeated on the newly created segments, joining points until all segments avoid collision.

Finally, the new flight path is converted to a .waypoint file that can be interpreted by the autopilot. **Figure 3** displays the ROAA's response in an example situation to avoid obstacles. If time is needed to compute a new path, the plane is set to LOITER mode prior to uploading a new set of spline waypoints to the Pixhawk, allowing the Razgriz to maintain altitude while awaiting new commands.

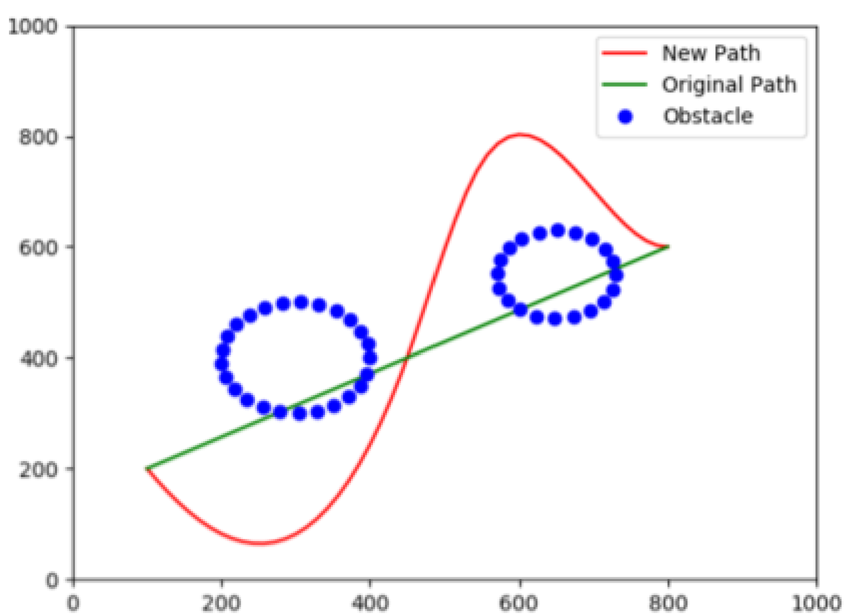

Figure 3. *Example of new path created by ROAA*

Imaging System 2.4

Our camera choice was primarily driven by two aspects: cost and resolution. Secondary considerations included the physical aspects of the camera unit itself (dimensions, weight, connector type), as well as total operability (the ability to be controlled by software and/or multiple devices).

To get a better understanding of each camera's optical quality, an additional metric defines pixels per foot (px/ft), useful for estimating the resolution of the target. Using data obtained in preliminary tests, it is possible to render a target at 12 px/ft, and consistently above 18 px/ft. Above this value, additional pixels are considered supplemental. This metric is calculated with the formula $\frac{arctan(\frac{1}{alt}) \times Res._{hor.}}{FOV_{hor}}$. If no FOV (Field of View) for a horizontal image is provided, it is assumed a 67-degree diagonal lens is to be used.

To get a better understanding of the cost-effectiveness of each camera, the metric "Value" is used to judge the prices of each camera. This metric takes into account the exponential price increases often found in electronic devices when comparing cost and resolution, with the total formula as $\frac{1}{\sqrt{Price \div (Res_{hor} \times Res_{vert})}}$.

For Example, See3Cam-CU135

$$\frac{1}{\sqrt{264 \div (4208 \times 3120)}} = 223$$

For Example, Flir Blackfly S 20MP

$$\frac{1}{\sqrt{695 \div (5472 \times 3648)}} = 169$$

Camera	Cost	Resolution	px/ft, 200ft	Value
See3Cam-CU135	$264.00	4208x3120	22.50	223
GoPro Hero 7 Refurb	$229.99	3840x2160	20.53	190
Logitech Brio	$199.99	4096x2160	14.74	210
Raspberry Pi V2 Camera Module	$24.99	3280x2464	17.54	569
ROCK64Pro M12 Camera	$21.99	4208x3120	22.50	773
FLIR Blackfly S 20MP	$695.00	5472x3648	27.31	169

Table 3. *Camera Choice*

By price, the ROCK64 and Raspberry Pi Systems proved excellent deals. However, the RPi camera did not rate high enough on the px/ft metric and the ROCK64Pro is only compatible with one specific System-On-Chip (SoC), which has OS instabilities. The FLIR Blackfly, GoPro, and Logitech Brio all had too low a value rating or px/ft benchmark for us to consider them. As such, the final selected system is a See3Cam-CU135, a device manufactured by e-Con Systems. This unit comes with a lens also manufactured by e-Con, equivalent to a 30mm lens on a standard camera.

Gimbal 2.5

Target detection involves being able to locate a target within a large field of view; therefore, it was critical that the camera has the ability to pan and tilt independently of the movement of the aircraft itself. To do so, the team decided to utilize HobbyKing's Quanum 2-axis gimbal. The servos are controlled by an Alexmos SimpleBGC 8-bit gimbal control board, which will use an IMU sensor to make sure the camera is always nadir pointing.

Object Detection, Classification, Localization 2.6

The autonomous Object Detection, Classification, and Localization (ODCL) system is performed in multiple stages. **Figure 5** outlines the entire ODCL system.

The first stage is a Haar Cascade (also known as a Haar-based Classifier), which determines a Region of Interest (ROI) based on the images retrieved from the camera. The Haar Cascade is trained using data developed from a series of test videos. There are two types of training data: positive and negative. The positive training data is generated by taking a continuous video of the training data from the air. Then, the team made thousands of images by manipulating every ten frames of the video. For each of those images, the team manually defined the ROI. Negative images are generated from the same set of images with a selected ROI that contains only grass.

The Target Isolation Program (TIP) is run on a 400 by 400 pixel image, centered on the ROI obtained in the previous step. The TIP k-Means color quantization is performed on the image to decrease the number of colors to work with. This process involves binning of similar colors, where clusters can then be identified to find similar colors that often appear despite variations in the image, due to noise and other factors. An additional 200 by 200 pixel crop is taken as a tighter crop of the target; the remaining area is used as training data to identify grass and other irrelevant areas. Using the data obtained, the target can be isolated by removing areas of colors identified previously in the training data. The letter follows much of the same process, using k-Means Dominant Color (kDC) to remove the base target, leaving a single letter ready for further processing. **Figure 4** shows this process on an example ROI image. The shape and character color are derived as a result of steps involving k-Means.

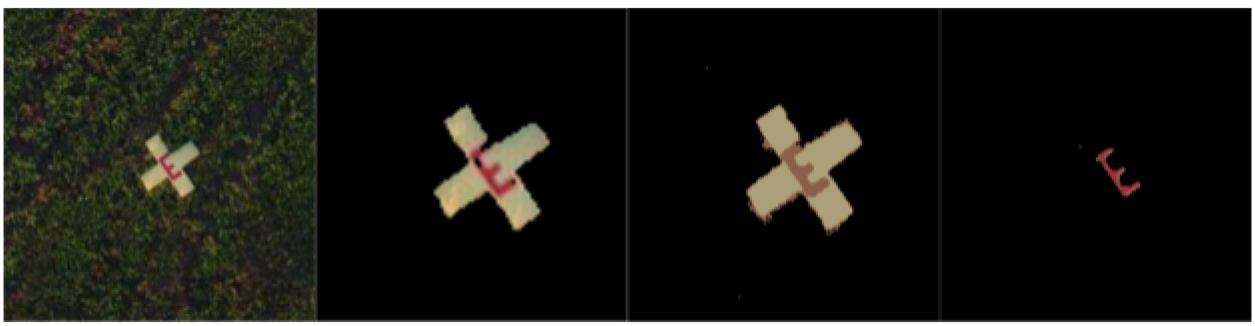

***Figure 4**. TIP Steps*

The steps above show the object isolation process into a target and letter. The first cell depicts the captured image. The second displays the results of cropping, quantization, and area sampling algorithms. In the third cell, further color quantization helps to isolate the letter, target, and their colors. Lastly, cell four is the result kDC, removing the most common color and leaving an outline of the letter, in this case an "E."

The next stage in image processing is a Convolution Neural Network (CNN), which determines the shape of the object using the image outputted from the TIP. The TIP helps improve the accuracy and speed of the CNN by returning a smaller image and filtering out the pixels that are not part of the object. After the CNN receives an ROI from the Haar Cascade and TIP, it classifies the image to one of the 12 possible shapes. It also has the ability to classify an image as nothing, thus helping to prevent false positives.

After the ODCL system determines the color of the alphanumeric character, it filters out all other colors. Then the processed image is sent to the ground station where the rest of the processing takes place. The partially processed image goes through a spatial transformation network which overcomes a CNN's inability to deal with spatial variance. This image is then passed to another CNN for Optical Character Recognition (OCR). From the CNN's output, the ODCL system determines the orientation of the letter relative to the UAV. Once combined with GPS data from Razgriz telemetry, the absolute orientation of the target can be determined. The team considered using Tesseract for OCR because it can determine the orientation of the object simply, but Tesseract is unreliable for blurry letters. Therefore, the team decided to create a CNN using Python's Tensorflow and Keras libraries. The model was then trained using testing data obtained from a series of camera drone flights.

For geolocation, the ODCL system first finds the pixel coordinates of the target in the image of the ROI returned by the Haar Cascade. Based off of these coordinates, the ODCL system calculates a component form vector that describes the location of the target relative to the center of the image. Based off of the height of the aircraft and the FOV of the camera, the ODCL system determines how many pixels in the image are equivalent to one degree, and uses that scalar to convert the units of the vector from pixels to degrees. Finally, the code adds the relative vector to the GPS coordinates of the image to get the absolute position of the target. Manual detection is done by two team members using a custom built GUI. The onboard computer sends once every 10 frames to the ground station, where they are analyzed by the team. If the team finds a frame with a previously undetected object, the team will submit it via the interoperability server.

If the TIP or CNN returns unexpected results, the ODCL system will assume the object is the emergent object. The current ODCL system is not capable of classifying emergent objects, so it will use a description from a list of possibilities and use the color information from the TIP to describe the emergent object.

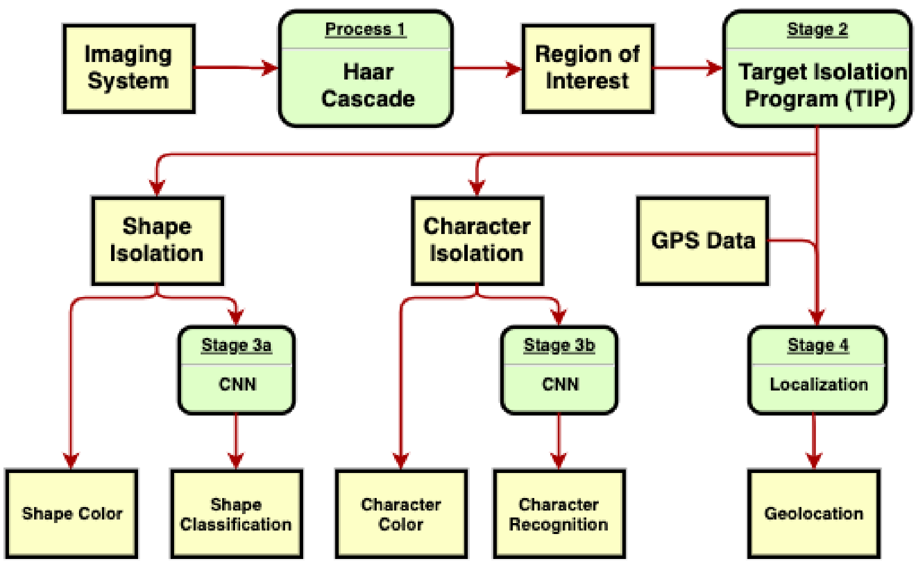

Figure 5. *Diagram detailing the different stages of the ODCL system.*

Communications 2.7

The Razgriz has three wireless links between the aircraft and the ground station: one 2.4 GHz X8R RC link, a 2.4 GHz WiFi link established by a Ubiquiti Bullet M2 linked with a Ubiquiti AG-HP-2G20, and a 915 MHz telemetry link established by a pair of RFD900x's. The 2.4 GHz X8R RC link allows for manual override of the plane and allows for a ground station pilot to navigate the Razgriz in the event of autopilot failure. The safety pilot, a member of the competition team, will constantly be ready for manual takeover in the event of a flight anomaly.

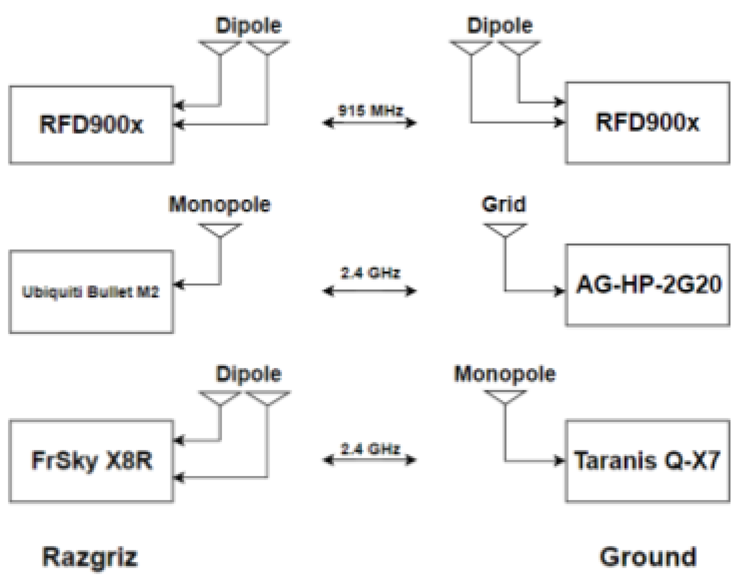

Figure 6. *Communication System*

WiFi Link 2.7.1

The Ubiquiti Bullet M2 onboard the Razgriz works in conjunction with a Ubiquiti AG-HP-2G20 antenna on the ground station to establish the primary WiFi link. Using this link, the ground station is able to establish a TCP connection with the onboard Jetson TX2. The TCP connection allows the team to upload new mission details to the plane, as well as send important commands to the plane's system. Additionally, the TCP connection allows the ground station to receive data from the Jetson including the plane's status, object images, and object classification data.

The team originally looked to utilize the Ubiquiti Bullet M5, the 5 GHz counterpart of the Bullet M2, due to its higher bandwidth. A 5GHz signal would also avoid interference with the 2.4 GHz X8R RC link. However, the extra monetary cost of the Bullet M5 outweighed the aforementioned benefits, and for this reason, the team decided to use the Bullet M2. Furthermore, concerns were raised on the stability of the 5 GHz connection, due to an inability to maintain a connection without a direct line-of-sight between the plane and ground station. Obstacles such as trees ground clutter needed to be easily dealt with.

Attached to the Ubiquiti Bullet M2 is a 3 dBi omnidirectional VERT2450 antenna, mounted on the left wing of Razgriz. The omnidirectional pattern of the antenna allows for a stable connection with the ground station. The ground station will use a directional Ubiquiti AG-HP-2G20 antenna to increase the signal strength.

Telemetry Link 2.7.2

The 915 MHz radio link uses a pair of RFD900x telemetry radios on the Razgriz and the ground station. They allow for transmission of telemetry data so the ground station team can monitor the status of the Razgriz. Each RFD900x utilizes two omnidirectional quarter wave monopole antennas, ensuring a stable connection regardless of the plane's orientation. The team has experimentally tested the radios for distances up to 2km, however, the listed manufacturer range is 40km or more.

Ground station 2.7.3

The ground station consists of four machines: the Communications computer, the Telemetry computer, the Automatic ODCL computer, and the Manual ODCL computer. The ground station will utilize a Netgear network switch to allow for data communication between all devices. These devices provide the team with an interface to communicate with the NVIDIA Jetson TX2 and Pixhawk. Each computer in the ground station will be assigned a certain task and will have a member of the team monitoring it.

The Communications computer will establish a stable connection with the interop server. This connection will be used to receive mission data from the interop server as well as submit objects. The team created its own server using the AUVSI API and tested the various functions on it to ensure success during the competition. Additionally, the computer will maintain a TCP communication with the TX2 using the WiFi link. The team developed a header system to identify messages conveyed between the ground station and the TX2. Upon receiving data from the TX2, the computer analyzes it and performs the appropriate task based on the message's header.

The Telemetry computer will be hosting the RFD900x antenna and will display the telemetry data using the ArduPilot Mission Planner software. The software enables the team to monitor the Razgriz's status at all times. This computer will also run the ROAA, which will create a .waypoint file that contains a new flight plan if needed. The OAM can load the file into Mission Planner and upload it to the Pixhawk via the RFD900X.

The Automatic ODCL computer will be performing the ground station portion of the ODCL System. Upon receiving an ROI image, the ODCL computer will run the TIP and the CNNs to isolate and classify the object. It will then request GPS data from the Telemetry computer to perform localization calculations. Once all of the information is determined, a JSON file will be created and sent to the Communications computer along with the ROI image for submission.

The Manual ODCL computer will be used to manually detect targets in case of a malfunction within the automatic target detection program. Two team members will be searching for targets in the frames sent to the computer, and will manually classify the target and input data to create a JSON file. The JSON file, along with a cropped image of the object, will then be sent to the Communications computer in the same manner as the Automatic ODCL computer.

Air Drop 2.8

As this was the team’s first year, TJ UAV decided to pursue a simpler UGV design. The team planned to have a water bottle drop mechanism that would fulfill the drop accuracy requirement, but decided not to develop a system for driving. The mechanism hangs the water bottle from a pin within the drop bay. Once permission for a drop is given, a ground station command retracts the pin, allowing the water bottle to drop out of the bay.

Given the localization method described above, the GCS telemetry computer takes the relative position of the drop site to the plane, the velocity of the plane, and wind velocity to estimate the parabolic arc the payload will travel along. The computer will then use ROAA to generate a path that will maneuver the aircraft such that the drop arc will eventually land within the target zone. Once the estimated landing position of the bottle is within a specified tolerance of the target, the Razgriz will drop on the target, assuming ground station permission has already been received.

Cyber Security 2.9

The Bullet M2 access point receives data throughout the duration of the Razgriz’s flight. This constant transmission of data can be exploited; although an attacker would have to be physically located between the plane and the ground station to do so due to the directionality of the AG-HP-2G20 antenna. Some potential security risks of an actively listening access point are:

1. Denial-of-Service attacks on the M2 or AG-HP can block off critical data, telemetry, or commands from the TX2 or GCS
2. Man-in-the-middle attacks can intercept the credentials of communicating nodes.
3. DNS poisoning attacks can cause the ground station server to crash and return sensitive user information.

To defend against these threats, the team used directional antennas for communication between the flight computer and ground station. Furthermore, instead of dynamically assigning IPs, the Jetson specifies its own static IP to block routing table attacks. Additionally, user IDs and passwords are safely stored and hashed in the TX2. Upon connection, the TX2 rehashes all login credentials.

These security measures effectively thwart any cyber security threats while offering fast operation. The use of TCP connections diminishes the speed of data transfer in exchange for more secure transmission. However, the team’s Python script divides data into blocks before wireless transmission to enhance communication speed. Furthermore, Ubiquity (the manufacturer of the M2 and the AG-HP), offers its own protection for packets sent between AirMax® supported antennas. It uses special headers and a proprietary packet format to improve security.

The team encrypted any data sent using AES (Advanced Encryption Standard) encryption. The U.S. Government adopted this encryption standard in 2001. The Rivest-Shamir-Adleman (RSA) encryption algorithm initially appeared as an ideal standard for data protection, but the team eventually chose AES due to the following reasons: AES is a symmetric block cipher, whereas RSA is an asymmetric cipher directly dependent on mathematical operations. Some of these operations, such as modular exponentiation and modular multiplicative inverse, are computationally expensive. Since AES was designed to work with padded blocks of a fixed length, the cipher has a stable encryption speed.

A test conducted by Ali, Esparham, Marwan, and Nazeh Abdul Wahid (2018) quantified the differences between several encryption algorithms. The researchers determined that AES had a fairly consistent encryption time even as the files to be encrypted grew larger. RSA, on the other hand, encrypted files much slower as the size of the files increased. RSA also proved to use more memory than AES while executing unit operations. Additionally, public key ciphers like RSA are primarily used when several clients are communicating, whereas the team only needed to send packets between two machines. AES is also fully functional with the PyCryptodome library. RSA must be integrated with another library, PKCS1_OAEP, to be used with PyCryptodome. These factors convinced our team that AES encryption was more suitable.

Safety, Risks & Mitigation 3.0

Developmental Risks & Mitigation 3.1

In order to ensure the success of the mission and the safety of all members and bystanders, the team examined potential risks. The team then determined how to mitigate the risks. **Table 4** describes how the team assessed these risks. Probability is gauged as low, medium, or high. These are risks that could occur during testing and construction of the Razgriz

Risk	Probability	Severity	Mitigation
Injury to team members during the construction process	Low	High	● Wear safety glasses and closed toed shoes during use of heavy machinery ● Wear facemasks when interacting with dangerous chemicals ● First aid kit kept readily available
Accidental throttle-up during flight preparation	Low	High	● Keep throttle cut active until ready for flight ● Do not attach propeller until ready for flight ● Alert team before arming
Incorrect assembly or construction	Low	Medium	● Have experienced members supervise ● Discuss plans before executing
Unable to meet progress deadlines	Medium	Medium	● Use Slack® for team communication ● Keep team morale high ● Host team meetings on weekends and after school

Table 4. *Developmental Risks and Solutions*

Mission Risks & Mitigation 3.2

Similar to developmental risks, the team examined possible mission risks. **Table 5** assesses potential failures and how the team would mitigate them. The most important factor in severity is safety, followed by impact to mission. The team has also created a preflight checklist in order to eliminate any preventable problems during flight.

Risk	Probability	Severity	Mitigation
Pilot Error	Low	High	● Each pilot has sufficient flight hours with the aircraft ● Practice manual takeover in order to recover
Loss of Power	Low	High	● Check all flight and controller batteries before flight ● Utilize low battery failsafes within Mission Planner
Electrical Fault	Low	High	● Reference prepared wiring diagrams before connection ● All components are tested to ensure that current draw does not exceed nominal amounts.

Accidental Throttle-Up during Flight Preparation	Low	High	• Keep throttle cut active until ready for flight • Do not attach propeller until ready for flight • Alert team before arming
Communication Loss	Medium	High	• Check communication status throughout flight • Program a fall back system in case of failure
Launcher Injured during Takeoff	Medium	High	• Practice launching plane in safe way • Provide a hard hat for the launcher
Slow Mission Demonstration	High	Low	• Practice fast setup routines • Setup checklists for procedural tasks

Table 5. *Mission Risks and Solutions*

Conclusion 4.0

The members of TJ UAV began as just RC Aircraft enthusiasts in 2017. As the members of the team learned more about aerospace engineering and UAV technology, they realized that they wanted to attempt something more ambitious. Beginning in the summer of 2018, TJ UAV began working with flight computers to modify the team's RC aircraft into UAVs. After being accepted into the 2019 SUAS competition, the team developed a cheap, simple platform; the Razgriz. The Razgriz is capable of flying autonomously for extended periods of time, processing and reacting to images, and dropping a payload onto a target. Through the development of the Razgriz, the team has gained invaluable knowledge and experience in engineering, programming, project management, and the engineering design process. Most importantly, the team has gained incredible experience in teaching one another vital skills to make this year's and future projects a success. TJ UAV is proud of how far it has come, and is looking forward to expanding its program to compete in AUVSI SUAS for the years to come.

References

Ali, A., Esparham, B., Marwan, M. D., & Nazeh Abdul Wahid, M. D. (2018). A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention. *J Comp Sci Appl Inform Technol. 3(2)*: 1-7. DOI: 10.15226/2474-9257/3/2/00132