Unmanned Aerial Systems at UCLA

2019 AUVSI SUAS Competition Technical Documentation

University of California, Los Angeles | Department of Mechanical and Aerospace Engineering

Abstract

Unmanned Aerial Systems at UCLA (UAS@UCLA) is a team of undergraduate students from the University of California, Los Angeles (UCLA) working to design, manufacture, and test autonomous flight and ground vehicles to compete in the annual AUVSI SUAS competition. This year, UAS@UCLA expanded upon the capabilities of the 2018 flight system to correct and enhance technical shortcomings at last year's competition; among these are an avionics box, antenna tracker, groundstation box, and an unmanned ground vehicle (UGV). This report gives an overview of the mechanical and software systems and development process for the 2018-2019 academic year.

1 System Engineering Approach

1.1 Mission Requirement Analysis

The mission demonstration component of the competition involves a series of challenges that may be encountered by a shipping company wishing to use autonomous vehicles to assist with its activities. This includes navigation of static obstacles to reach a target, identification of vision targets placed on the ground, and transportation of a package via a deployed ground vehicle.

UAS@UCLA evaluated the mission objectives as outlined in the competition requirements, [1] and then established team goals for the aircraft that would enable their completion. The objectives, with corresponding percentages of total score and requirements for success, are summarized in **Figure 1.1**.

Objective	Description	Requirements for Success
Timeline (10%)	• Ease of set-up and take-down of the quadcopter system • Fly without need for a timeout • Fly the mission in minimal time	• Rigorous practice missions • Sufficient battery life • Algorithm to determine optimal flight path
Autonomous Flight (20%)	• Fly with minimal human intervention • Capture all waypoints throughout the mission • Do not fly out of bounds • Do not crash the flight vehicle	• Calibrated navigational equipment • Stable and secure flight system
Obstacle Avoidance (20%)	• Avoid stationary obstacles placed throughout the flight area	• Obstacle avoidance algorithm
Object Detection, Classification, and Localization (20%)	• Correctly identify and localize objects in search area • Autonomously classify images captured by the drone for submission to interop server	• High resolution camera and gimbal • Well trained computer vision AI
Air Delivery (20%)	• Ability of the quadcopter to carry and safely deliver a ground vehicle • Accuracy of UGV delivery (drop and drive to location)	• Reliable deployment system • Rigid UGV
Operational Excellence (10%)	• Throughout the competition, act professionally as a team with effective communication, consideration for safety, etc.	• Hospitality and Communication • Teamwork development and practice

Figure 1.1: Mission objective weightings

[1] Competition Rules | SUAS 2019, 2019.

1.2 Design Rationale

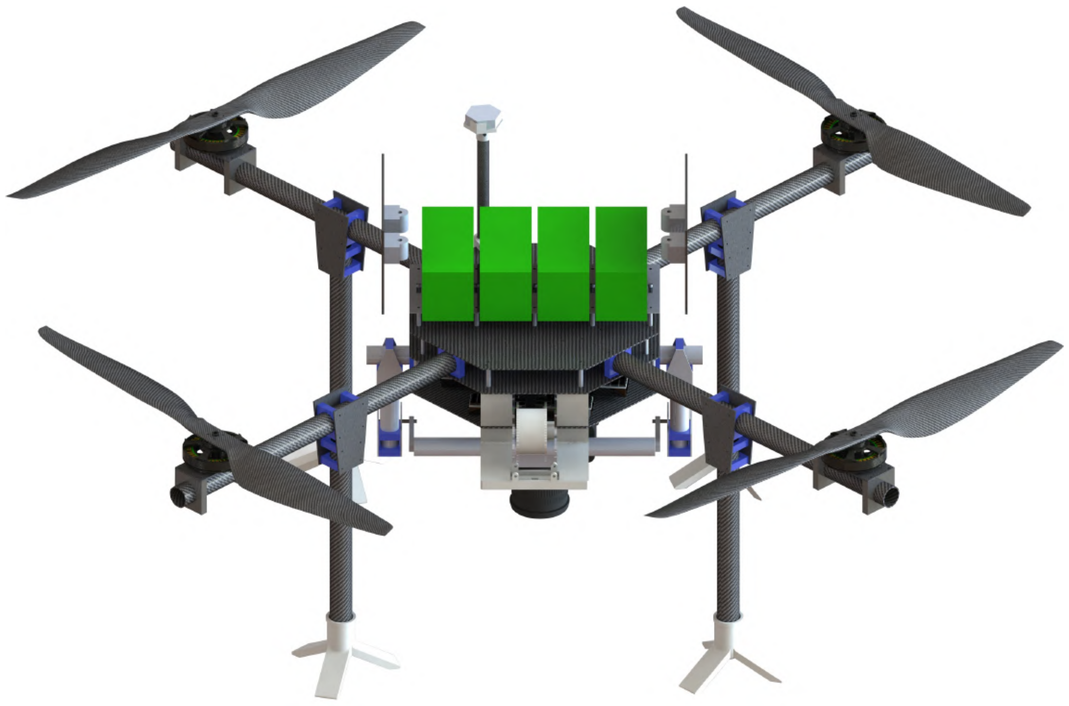


Figure 1.2: Final CAD rendering of Spinny 2.0

Spinny 2.0, the competition vehicle for UAS@UCLA 2019, was designed with the expanded capabilities of our team in mind. Similar to previous years, the largest environmental constraint on our team was linked to the budget and experience of the team. Significant expansion of the team infrastructure and participation in previous competitions allowed for an ambitious retrofit of our previous aircraft. This new aircraft is capable of accomplishing all mission requirements as addressed in **Figure 1.1**. In addition, further development of this aircraft allows for continued investment in member development and research into future projects.

This system was designed to accomplish every mission task outlined in the rules. Our team prioritized iteration and improvement, along with design of new systems which would allow us to stay within budget while also creating a platform which could capably accomplish all tasks. We had the opportunity to be ambitious with our goals because of previous experience and improved support from our sponsors and university.

Selection of a frame was a vital decision which was made early in Spinny's lifetime. Ultimately, the decision revolved around three methods of flight: hover (multicopter), forward flight (fixed-wing aircraft), and a mix (VTOL aircraft). The primary mission objectives, outlined in **Figure 1.1**, highlight obstacle avoidance, survey flight, and boundary fitting as the main objectives of the season. For all three applications, additional maneuverability of a multicopter could only benefit mission performance, particularly obstacle avoidance and survey flight where position targets are dynamic due to the nature of the challenge. This system continues to prove as beneficial for secondary objectives such as autonomous takeoff, landing, and ground vehicle deployment. Autonomous takeoff & landing becomes trivial and more reliable with a multicopter design, and vehicle deployment simplifies the methods for safe and precise landing. The largest debate point for a forward flight vehicle was the improved battery life of such aircraft, and battery life ultimately served the largest determining factor in deciding the configuration of the motors on Spinny. This led to many design decisions made to optimize battery life on the multicopter with resolutions focusing on the powertrain efficiency, battery capacity, and system weight.

This year's competition introduced an unmanned ground vehicle with its own mission objectives and given design requirements outlined in **Figure 1.1**. Weight proved to be the largest design constraint for the ground vehicle. For a strong, lightweight, and easily designed frame, we decided to print the chassis of the ground vehicle using a polylactic acid (PLA) 3D printer. The electronics on the vehicle were optimized to save weight, and we determined that a two motor chassis with an unpowered rear wheel would meet design requirements with best efficiency. Several revisions were made to last year's aircraft to arrive at our current flight system. The overall airframe was largely unchanged from the previous competition, so these revisions were made with improvement on previous mistakes and adaptation to new competition requirements in mind. The specifications are shown in **Figure 1.3**.

Blade Span	6'3" Diagonal
Propeller Size	30"
Weight	35 lbs
Clearance	9.5"
Flight Time	45 minutes
Max Thrust	80-100 lbs (depending on charge)
Fully Charged Voltage	50.4 V
Battery Capacity	32 Ah

Figure 1.3: Spinny 2.0's relevant properties and metrics

The first system decision was made with the electronics system, which is contained in a modular box that contains all the necessary avionics for autonomous flight. This decision was made to promote easy software testing and simple, modular wiring schemes.

The second system decision was with the powertrain and battery system. The potential motors and the propeller matchings were then identified that will meet the flight time requirements and selected based on cost and weight. The batteries are encased in a flame resistant container which protects the batteries in the case of the drone flipping mid flight.

The third system decision was in the camera and gimbal system. It was determined that a point-and-shoot camera with a single axis gimbal system would be the most efficient at collecting images of the survey area without adding additional weight or complexity.

The fourth system decision made was in accommodating the new ground vehicle. A deployment mechanism for the ground vehicle had to balance weight, complexity and deployment time. The final product uses fishing line attached to the ground vehicle which is lowered using a motorized spool. The line is cut by melting using a wire heated by a current. This system allows us to hover over the desired landing spot, and deploy the ground vehicle in a quick but controlled manner which also allows for simple disposal of the deployment line and no extra time expended after deployment.

2 System Design

2.1 Aircraft

Spinny 2.0 uses a similar configuration to the 2018 flight system with minor modifications and sub-system additions to achieve the goals outlined in the 2019 competition. The airframe is divided into 4 sub-assemblies: the central airframe, four detachable limbs with motors, a gimbal system, and UGV/deployment system. The structure of the airframe is composed of prefabricated carbon fiber sheets and tubes along with custom designed 3D printed parts and/or laser cut acrylic. Carbon fiber was selected for its low weight and high elastic modulus while 3D printed and laser cut parts offer a lightweight, cheap, and custom solutions to many design problems.

2.1.1 Central Airframe

The central airframe is composed of three 13" octagonal carbon fiber plates with outer and inner mounting points. The first and second layers are joined by 35mm M3 aluminum spacers while the second and third are joined by 50mm spacers. Additional aluminum tube clamps sit between the middle two layers to hold the propulsion. The bottommost carbon fiber plate houses a removable 3D printed avionics box containing our flight controller, Raspberry Pi computer, and other essential electronics. The camera gimbal **(2.1.3)**, UGV, and deployment system **(2.7)** are also mounted to the side and underside of the electronics tray, respectively. The second plate from the bottom houses the electronic speed controllers (ESC) and the series of tube clamps that secure the aircraft's limbs **(2.1.2)**. A custom-made PLA and carbon fiber GPS mast is attached to this plate as well, elevating the GPS compass to reduce interference from the motor controllers attached to the plate below. The remaining electronics, such as the voltage regulators, battery monitors, and secondary GPS compass are mounted to the topmost octogonal plate. A rectangular plate on top secures the aircraft's four 16,000 milliamp-hour 6S 10C Lithium Polymer batteries to the assembly and is covered with a padded, flame resistant battery cover to protect the batteries in the case of a crash and extinguish flames in the unlikely event of a fire.

2.1.2 Aircraft Limbs and Propulsion

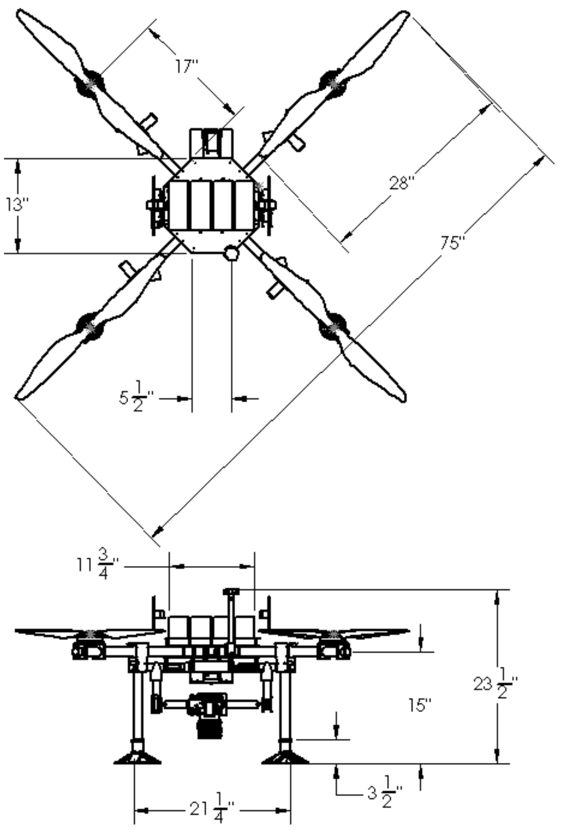


Figure 2.1: Orthographic CAD Views of Spinny 2.0

UAS@UCLA designed the four limbs of the drone to be identical and interchangeable in order to allow for relatively equal motor and landing gear wear and easy transportation. Each limb is composed of two perpendicular, 30mm diameter carbon fiber tubes of lengths 635mm and 500mm, respectively. The shorter of the two is positioned vertically and is attached to the main arm by orthogonal tube clamps sandwiched between two carbon fiber plates. A custom PLA and foam "foot pad" is attached to the lower end of the tube prevent wear on the carbon fiber and 3D printed to support rapid replacement. The longer carbon fiber tube connects to the second octagonal layer of the main airframe via tube clamps. Stress tests were conducted through simulation, using ANSYS and Solidworks, and through physical testing of the tube's flexural modulus, to ensure maximum stress and deflection is below the acceptable level of tolerance for the tubes. Thrust is provided by 30" carbon fiber propellers and the four Hobbywing XRotor 100KV brushless motors, mounted by tube clamps.

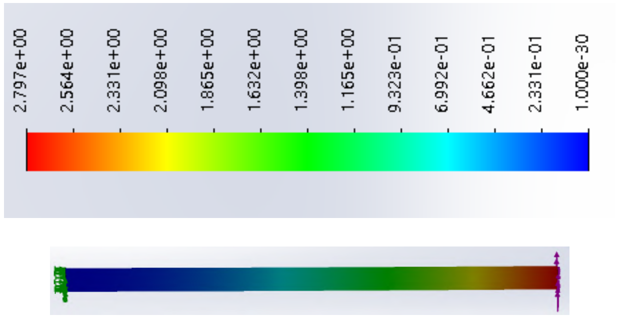


Figure 2.2: FEA showing deflection of motor arm tube at maximum throttle (25 lbs, 3mm deflection)

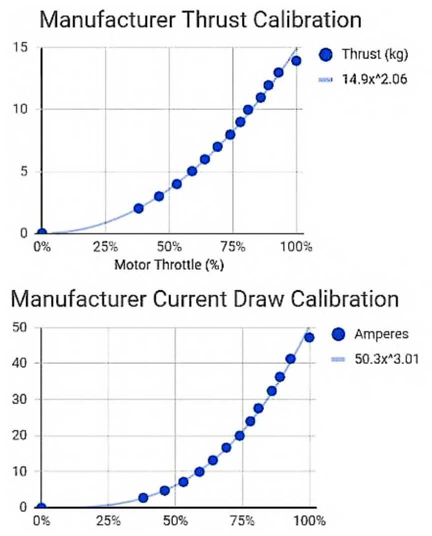


Figure 2.3: Motor thrust curve and power draw.

An estimated flight-time model was created to determine the throttle percentage needed to sustain a hover, and to act as the control when analyzing test flight results. Manufacturer-provided motor calibration data was plotted using spreadsheet software, and power series fit lines were used to derive the following two equations:

$$Thrust = a(throttle)^b \text{ [kg/Motor]}$$
$$Current\ Draw = c(throttle)^d \text{ [Amperes/Motor]}$$

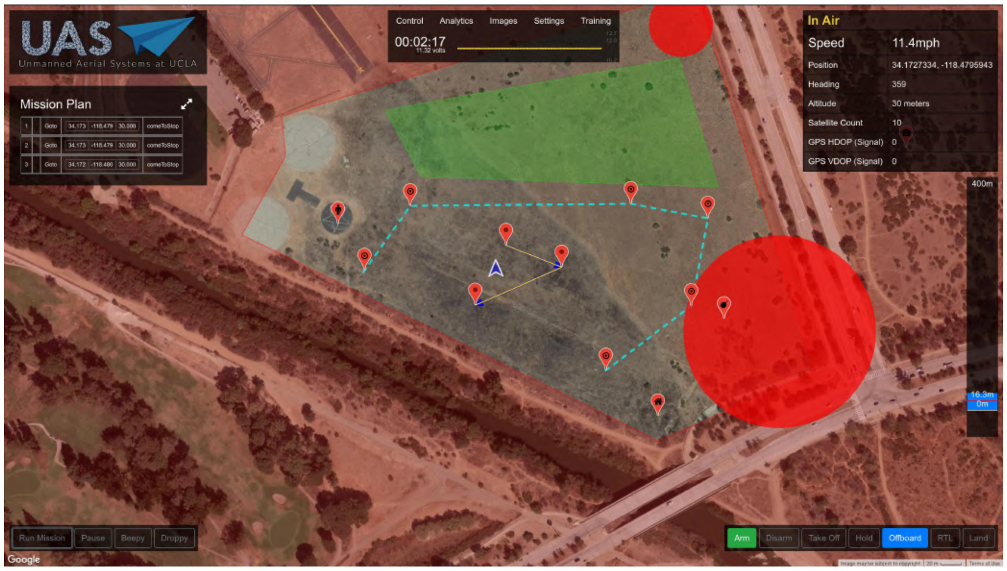
A plot of this model is show in **Figure 2.3**. The values of a, b, c, and d were then used to develop a flight-time calculator that took in the weight of the drone m [lbs] (which was converted to kg), number of motors n, and battery bank capacity A [Amperes/hr], and returned the hover throttle, as well as the flight times at hover and an optional, arbitrary throttle using

$$Est.\ Flight\ Time = \frac{60A}{nc}\left(\frac{m}{an}\right)^{-d/b}$$

2.1.3 Gimbal

The 2019 flight vehicle utilizes the custom gimbal design from 2018 for the acquisition of images used in the image identification and classification portion of the in-flight mission. To satisfy the mission requirements, our team produced several gimbal designs and decided to go with a single-axis design in order to reduce the weight and complexity of the system. Additionally, since the flight vehicle has a programmed forward direction of travel, the gimbal would only require actuation along a single axis. The gimbal is mounted to the airframe using a horizontal 16mm diameter carbon fiber tube that connects to the underside of the electronics tray, using vibration-dampening pads to ensure steady capture during flight. Custom carbon fiber joints connect a perpendicular carbon fiber tube to either end—one houses the iPower 5208 Brushless Gimbal Motor used for maintaining the camera's pitch while the other supports a ball bearing. An additional carbon fiber tube structure connects the ball bearing and the central pin of the gimbal motor and acts as the mounting point for the aircraft's Canon EOS 600D camera. The camera is dynamically held level using feedback from sensors. A Storm32 BGC 3-Axis gimbal controller records the current orientation of the aircraft and allows the on-board flight controller to correct yaw and roll. The pitch of the camera is then adjusted by the gimbal motor to allow for data acquisition in the desired field of view.

2.2 Autopilot



Mission Plan

1		Goto	34.173	-118.479	30.000	comeToStop
2		Goto	34.173	-118.479	30.000	comeToStop
3		Goto	34.172	-118.480	30.000	comeToStop

In Air

Speed	11.4mph
Position	34.1727334, -118.4795943
Heading	359
Altitude	30 meters
Satellite Count	10
GPS HDOP (Signal)	0
GPS VDOP (Signal)	0

Figure 2.4: FlightDeck groundstation interface for planning missions and reviewing tagged photos

The avionics system on Spinny 2.0 was designed around a standard *Pixhawk 1* flight controller running alongside a *Raspberry Pi 3* companion computer. The *Pixhawk 1* controller comes standard with redundant power supplies, inertial sensors, and communication ports, providing the flexibility and reliability required for successive design iterations. Additionally, it is a cost efficient controller, allowing our team to invest in a reliable platform while also being able to afford many flight controller units for developers. This control system is also small enough to fit on a racing quadcopter frame (shown in **Figure 2.5**), allowing our team to debug new software on a smaller scale before deploying to Spinny 2.0. For our competition drone, UAS@UCLA developed an avionics tray to contain any vital computers and electronics required for flight. The avionics tray (**Figure 2.6**) is a 3D printed housing designed to be easily removable from the drone structure and capable of running off a laptop's USB port. Consolidation of electrical components into a removable platform allowed our team to decouple the avionics system from the airframe structure, allowing software developers to work with the avionics system without requiring access to the fully assembled drone. This ultimately expedited software development by giving our software team the tools to begin hardware testing before modifications to the Spinny 2.0 airframe were complete. Additionally, during field tests, developers were able to quickly collect flight controller logs by sliding out the electronics tray to access the Pixhawk's SD card logger slot.

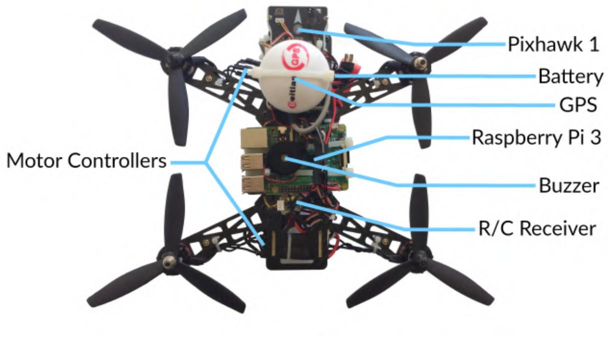


Figure 2.5: Avionics system on Speedy, the UAS@UCLA crash-test drone

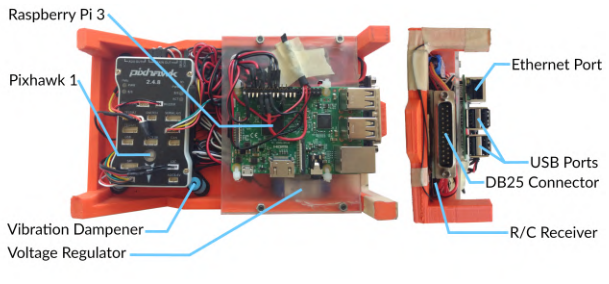


Figure 2.6: Avionics tray on Spinny 2.0

Vital to the success of autonomous flight is a robust monitoring system on the ground. An electric guitar case was repurposed in order to serve as a ready-to-go GCS (Ground Control Station) with all required components (**Figure 2.7**). In addition to laptops, a router, and a monitor, this includes a control panel with buttons, indicator LEDs, and a small LCD display to provide a simple and practical means of interacting with the drone. The panel's aluminum body was generously provided by Protocase.

For more advanced functionality, *FlightDeck*, the UAS@UCLA front-end interface shown in **Figure 2.4**, monitors and controls the drone, and provides a visual summary of everything that is going on at any instant. Mission plans for package delivery can be programmed using this interface, and sent to the drone which executes the series of commands, at which point the drone's progress and performance can be monitored.

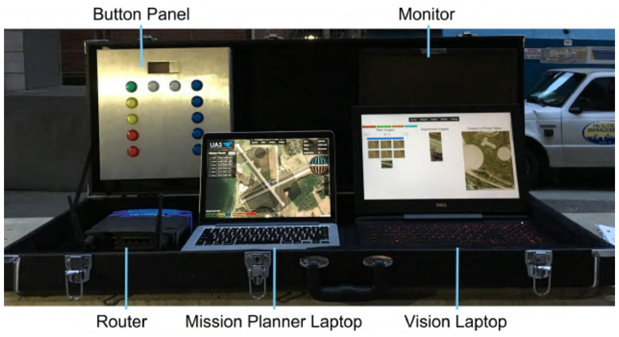


Figure 2.7: Ground Control Station, consisting of a repurposed guitar case with built-in button panel, a monitor, and storage of other hardware.

FlightDeck is also fully functional when the control system is run as a simulation, allowing for full-stack development and refinement before the drone hardware was completed and ready for use. The interface also integrates well with the Interoperability Server provided for testing. A system was developed so that the mission configuration could be easily swapped between various mock missions designed by the team, including some located at a local Los Angeles airfield for real-world tests.

	SITL	Racey	Spinny 2.0
Total Flight Time	200 hours	12 hours	7 hours
Autonomous Flight Time	200 hours	5 hours	4 hours

Figure 2.8: Autopilot testing and performance

2.3 Obstacle Avoidance

Obstacle avoidance routing is performed using rapidly exploring random trees (RRT). The UAS@UCLA interface for this algorithm takes in the raw coordinates of start/end locations and obstacle information for outputting a safe path to follow. By running the algorithm once, Spinny will be equipped with the best path based on the locations of the static obstacles.

The UAS@UCLA interface for the obstacle avoidance algorithm uses linear transformations and the Eigen linear algebra library to efficiently map coordinates to a gridded representation of the obstacles before running it through a 3rd party RRT library. Once a smooth obstacle-avoiding path is produced, the inverse of these transformations are applied to the result to return to the coordinate system. Finally, an extensive series of tests accompany the RRT code to ensure that calculated paths maintain accuracy, do not intersect obstacles, and have reasonable runtime when fed a random series of obstacle avoidance situations. Currently, the system natively supports static obstacle avoidance without requiring the ground station user to manually plan mission paths around the obstacles.

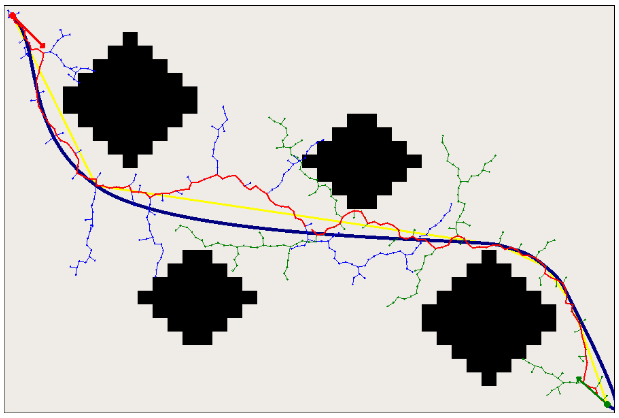


Figure 2.9: Example obstacle avoidance algorithm representation. Colored trees represent the computed paths whereas the dark and thick line represents the actual path taken.

To ensure that the drone does not veer from it's planned path, a control loop on the Raspberry Pi translates drone position and velocity data into the velocity commands that the flight controller should follow in order to stay on course. By asymptotically approaching the planned path, the drone will be able to correct itself if it overshoots a sharp turn or is blown by a gust of wind.

2.4 Imaging System

The imaging system consists of a Canon EOS 600D with a 18-55mm f/3.5-5.6 lens mounted onto a single-axis gimbal (Section **2.1.3**). During the majority of the flight, the gimbal is configured to hold the camera straight down while the lens is fixed to the 18mm setting for maximum field of view. For the off-axis target and for targets that are positioned within obstacles, the gimbal can be manually controlled to aim the camera at an angle. These off-axis images are automatically rectified with their coordinates and scale corrected by the vision pipeline on the ground station.

The length of an object in pixels captured by the camera can be found with the following equation derived from **Figure 2.10**.

$$f(x) = x \times \frac{f L_{px}}{hs}$$

where f is the focal length of the lens, L_{px} is the width of the photo in pixels, h is the altitude the picture was taken, and s is the width of the camera sensor.

According to this function this camera configuration was determined to be sufficient to resolve targets at approximately 100 pixels across from an altitude of 150 feet.

$$f(1\,\text{m}) = 1\,\text{m} \times \frac{18 \times 10^{-3}\,\text{m} \cdot 5184\,\text{pixels}}{45\,\text{m} \cdot 22.3 \times 10^{-3}\,\text{m}}$$
$$\approx 93\,\text{pixels}$$

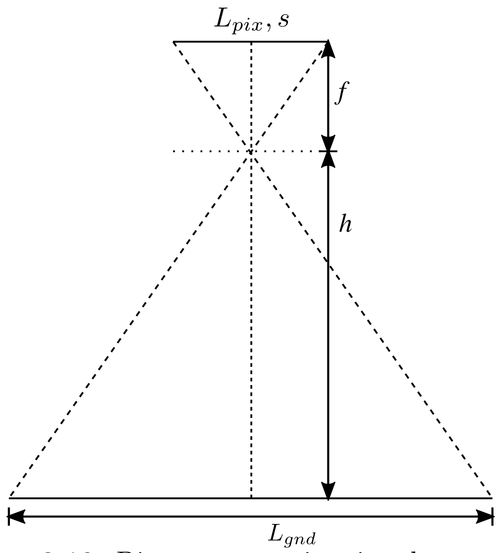


Figure 2.10: Diagram approximating the system as a pinhole camera.

2.5 Object Detection, Classification, Localization

Manual image tagging is performed through the computer vision component of the ground station user interface. Autonomous image processing is accomplished by a computer vision pipeline that can be configured to use various strategies simultaneously for object detection, localization, and classification.

2.5.1 Ground Station Computer Vision Interface

Manual image processing is done through a custom-built image tagging interface that integrates information from the autonomous computer vision system to assist users in quickly sorting through the raw images received from the drone.

This interface consists of three main features:

- The **vision map view** that provides an interface for locating and organizing images.

- The **object annotation pane** that allows users to manually annotate objects in the currently selected image.

- The **pipeline overview**, a visualization of the autonomous image processing pipeline and controls to enable and disable automatic submission.

The vision map provides an alternative view of the search area to the mission planning interface. Here, the user can see an overhead view of the search region with markers indicating the location of detected objects and an optional heatmap layer that indicates regions of interest. The heatmap and markers are calculated from detections that exceed a user-adjustable threshold by the autonomous image processing system. An area on the map can be selected to display a list of raw/rectified photos whose coordinates lie within that region. These images can then be individually inspected in the object annotation pane for more further manual tagging.

In the object annotation pane, the currently selected photo is displayed with annotations indicating the scale and location of where it was taken. The user then has the option of clicking and dragging on the image preview to draw bounding boxes around around objects and further annotate it with the drop down menus below. The position information is automatically calculated based on the GPS location of the drone and the focal length, sensor size, and resolution of the camera.

The pipeline overview displays the progress of images as they pass through each step of the computer vision pipeline. The user can inspect images and data as they are updated in real time for debugging purposes. Additionally, modules in the pipeline can be disabled or enabled to reconfigure the autonomous image processing on the fly.

2.5.2 Object Detection and Localization

Our first approach uses color-based image processing techniques with OpenCV and SciPy to detect and localize the target in the camera image. It operates on a pre-processed image with a perspective transform to normalize the angle and distance of the camera from the target. First, the program computes an approximate histogram of the hue dimension of the image, where peaks in the graph represent the most prominent colors. These colors are likely to correspond to solid regions, such as targets and grassy areas. We extract regions that are composed of these colors from the image, and then filter these regions by their size properties. Because the image has been normalized, the target is expected to occupy a known proportion of the image area. Finally, we use a contour-finding algorithm to trace the boundary of the optimal region, then smooth the environmental noise using the Teh-Chin chain approximation algorithm (**Figure 2.11**). This provides a clean description of the target, allowing it to be classified in a later stage of the pipeline.

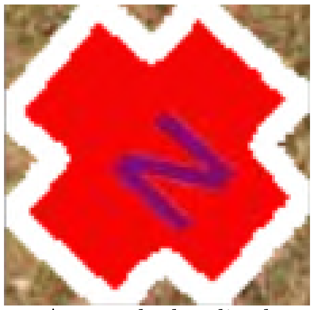


Figure 2.11: A sample localized and segmented computer generated target. The white border is the contour drawn around the target by the algorithm.

As an alternative approach, we implement the Faster R-CNN (Region Convolutional Neural Network) developed at Microsoft research to simultaneously localize and classify the shape of the targets. The Faster R-CNN is based on the VGG 16 model for detecting targets. This approach provides a bounding box, the shape of the target, and a confidence rating for the detection. This confidence rating is corroborated with our histogram-based approach to rule out false-positives. At the same time, this information is also used to augment the manual image tagging interface by indicating possible locations of interest.

2.5.3 Object Classification

In addition to the contour-based classification algorithms, an alternate image classification pipeline utilizes multiple Convolutional Neural Networks (CNN's) and a Fully Convolutional Network (FCN) built in Python using the Keras deep learning library with Tensorflow as its backend. This pipeline consists of the following three steps:

1. Classification begins with the cropped RGB images of the targets being passed through a VGG-16 (Visual Geometry Group's 16 layer classification model) network that classifies each image containing a target into one of 13 possible shapes.

2. The same set of images is then passed through a Fully Convolutional Network that segments the images into grayscale equivalents with white pixels highlighting the letter portion and black pixels masking out the rest of each image.

3. Using the images segmented by the FCN of the previous step, another simple Convolutional Neural Network is used to identify the masked out letter.

This alternate pipeline is used in conjunction with the contour-based detection and classification pipeline in order to improve performance in cases where contours are blurry or otherwise difficult to determine.

2.6 Communications

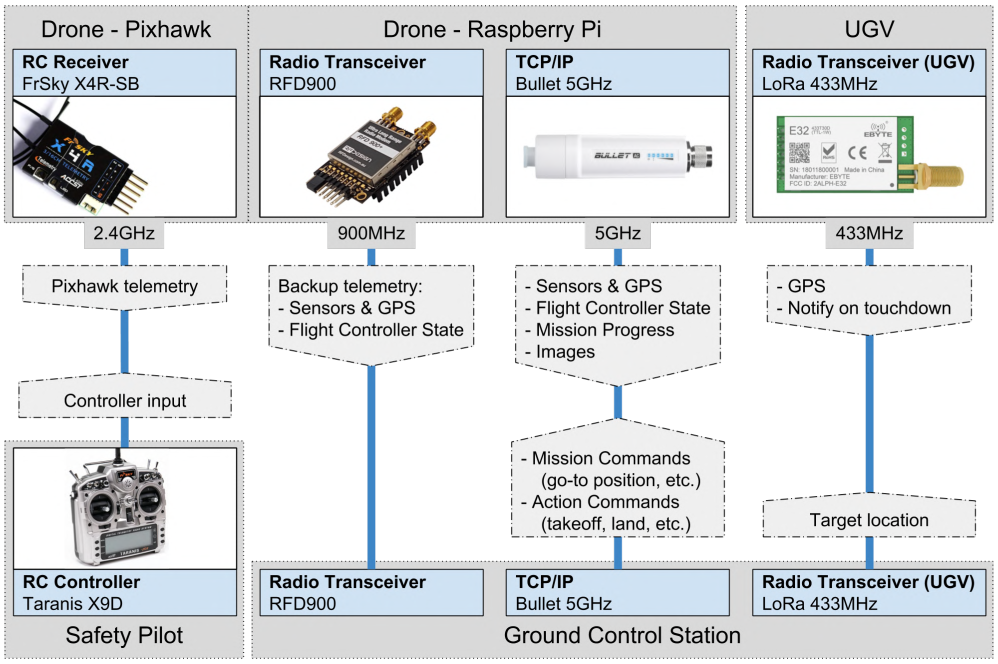


Figure 2.12: Communications System Diagram

There are three separate communication links between the drone and the ground, in addition to UGV communications (**Figure 2.12**). First, there is the RC controller. This year, the decision was made to purchase the FrSky Taranis X9D, a controller with a built-in display, providing for much richer feedback from the drone for the safety pilot, including the flight controller state and other telemetry. The controller communicates over 2.4GHz with the FrSky X4R-SB telemetry receiver connected to the Pixhawk. This receiver is capable of sending telemetry directly to the controller.

Our team has worked on a TCP/IP communication system to manage the interactions between hardware in order to ensure a successful autonomous mission demonstration. One benefit of using TCP/IP is that the TCP (Transmission Control Protocol) transport can be used. TCP provides reliable, ordered, and error-checked transmission, meaning that messages are guaranteed to be delivered in order, and failed transmissions are reported or re-sent until successful. TCP potentially introduces more latency in communications, but our testing has shown that it has little impact on the time scales at which we are required to send messages (e.g. sending telemetry at a rate of 25Hz). The reliability of message delivery is much more important – it allows us to easily send commands to the drone, and provides a means to transmit captured images. To make development convenient, we use Node.js and Socket.IO, which were chosen due to their popularity and support for web communications. Socket.IO is an abstraction of WebSocket, which is itself an abstraction of TCP for sending messages between server and client over a single connection. A computer at the ground station is designated as the “ground server,” which is connected to a router that can be accessed via ethernet or WiFi (2.4GHz). Several other devices are connected to the router, including a button panel, an antenna tracker (below), and at least one other computer, which will run tasks such as image processing. The router also enables a connection with the Interoperability Server. In addition, UDP is used to route MAVLink messages from the Pixhawk to QGroundControl.

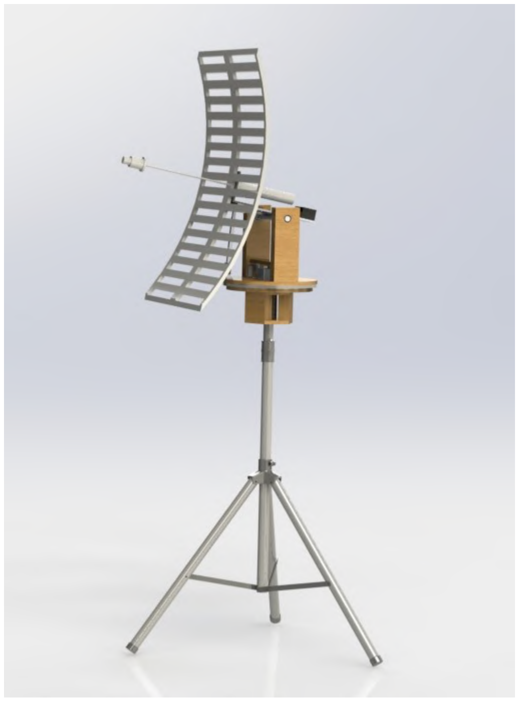


Figure 2.13: Antenna tracker CAD rendering

To make WiFi communication possible, a pair of 5GHz Ubiquiti Bullets is used to establish the connection between the drone and the ground, and a multi-directional antenna is fixed to the Bullet on the drone. Because a strong connection is needed for 5GHz Wifi, a high-gain directional antenna is used on the ground. This year, UAS@UCLA developed an automatic tracking system to direct the antenna towards the drone, using the drone's GPS position and altitude. The antenna tracker was built using a wood fixture mounted to a metal turntable which allows the tracker to spin freely with 360 degrees of yaw using a stepper motor. The correct yaw position is determined as in **Figure 2.14**. The pitch of the antenna is controlled using a linear servo along with a counterweight to decrease the moment required for pitching. The entire assembly is mounted on a prefabricated metal speaker stand that can be transported easily for field testing. A CAD rendering of the tracker is shown in **Figure 2.13**.

In case WiFi fails or is not available, an RFD900 (900MHz) radio is used as backup for transmitting telemetry to the ground station at a rate of 5Hz.

Finally, the UGV communicates with the ground station using a 433MHz LoRa, a low-power long-range device. LoRa devices use a specific spread spectrum modulation technique to enable this, and the low power consumption is especially useful for the UGV which has limited power available.

$$\Delta\theta_{\text{yaw}}(t) = \arctan\left(\frac{y_{\text{drone}} - y_{\text{antenna}}}{x_{\text{drone}} - x_{\text{antenna}}}\right) - \theta_{\text{yaw}}(t-1)$$

$$\text{where, } y = R_e \times \cos(\text{latitude}) \times \sin(\text{longitude})$$
$$x = R_e \times \cos(\text{latitude}) \times \cos(\text{longitude})$$

Figure 2.14: Calculation of yaw angle of the antenna tracker based on the drone's position

2.7 Air Drop

This year's airdrop task is meant to simulate the autonomous delivery of a package to a consumer by air and ground. This process involves deploying an unmanned ground vehicle from our unmanned aircraft to GPS coordinates provided by AUVSI. After deployment, the UGV must autonomously deliver an 8 oz. water bottle to target coordinates on the ground. During the design process, the airdrop task was separated into two categories: the development of the UGV itself and the development of the deployment system.

2.7.1 Unmanned Ground Vehicle

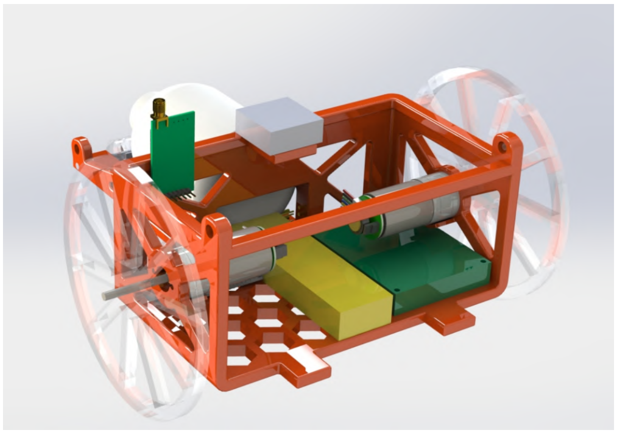


Figure 2.15: Unmanned Ground Vehicle CAD rendering

The ultimate goal of the UGV is to autonomously deliver its payload to a target location. However, in the process of doing so, it must satisfy several design considerations: the UGV must be robust enough to survive its landing, light enough to stay below the 48 oz (3 lb) weight limit, and drive accurately enough to stop within 10 feet of the specified location. A major part of the design process was determining the appropriate balance between weight and strength of the chassis. Due to the 3 lb total weight limit for the drop and the decision to use a single motor to deploy the UGV, we decided to prioritize weight reduction over the strength of the chassis. 3D printed PLA was selected as the material for the chassis due to its low weight and relatively high strength. Additionally, 3D printing the chassis allowed for a complex, weight-saving design to be manufactured and iterated upon with relative ease. After several iterations of chassis designs, a box-like chassis with honeycomb holes in the base and semi-triangular holes in the walls was devised. The main box is large enough to fit all of the electronic components, including motors that drive two large-diameter wheels, an on-board computer, a motor controller, and a lithium polymer battery. A platform extends behind the main box to hold the 8 oz water bottle. Three small wheels are attached on an axle at the end of this platform to reduce drag as the UGV drives to its target location. Two small holes are present at the top of the UGV for the fishing line from the deployment mechanism to go through.

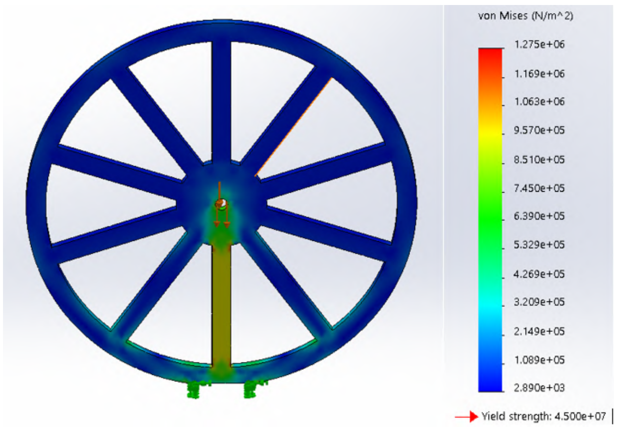


Figure 2.16: Wheel Finite Element Analysis

As the wheels of the UGV are the first to hit the ground when the UGV lands, they must be strong enough to withstand the collision without breaking. However, strength is not the only consideration for the wheels: they must also allow the UGV to move quickly and reliably to its target destination. 5 inch diameter chariot-style wheels were designed to provide considerable speed without much added weight. Acrylic was decided as the material due to its light weight and high impact resistance. Stress simulations using a 20 N force were performed on the wheels to ensure that they would not fail in the case of a hard landing. Rubber treads were glued to the outside of the wheels to increase traction and impact toughness.

Using the maximum speed of 10 mph as a guideline, we were able to determine that each motor would need to provide a torque of 55 oz-in and an angular frequency of 970 rpm given the 5 inch diameter of our wheels. Once the motors were selected, we focused on choosing a battery that would be able to provide the 12V and 3.8 A that each motor required.

2.7.2 Control system

In order to accomplish the task of autonomously navigating to the target location, we decided that the UGV should be able to measure its orientation relative to Earths magnetic field, as well as its position. The latter was easily done using an of the shelf GPS module, but the former was more involved. We selected a 9-axis MEMS device which can measure magnetic flux, angular rate and gravitational acceleration (MARG). The values from this sensor are used as inputs to a sensor fusion algorithm[2] which finds the orientation of the UGV as it moves.

Purpose	Selected Hardware
MCU	WeMos ESP32 Dev Board
RF Communication	EBYTE E32-433T30D LoRa Transmitter
Orientation Sensing	MPU9250 IMU
GPS	NEO-M8N GPS
Motor Driver	Cytron 10A Dual Motor Driver

Figure 2.17: UGV hardware summary

2.7.3 Deployment Mechanism

The UGV must remain functional after its landing, which necessitates a deployment mechanism to deliver the UGV from the drone to the ground. The main considerations in designing such mechanism are: safety (how to avoid damaging the UGV), accuracy (how to deliver it to the stipulated location), and speed (how to perform the task in a swift manner). Some of the foreseeable disruptive factors are: wind, motion of drone (causing the UGV to sway and land with low precision), and speed of descent (snapping of fishing line due to sudden deceleration). To address these problems, we

[2]S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, June 2011.

decided to have a porous chassis for the UGV in order to minimize the surface area exposed to wind and use a motion sensor to limit the speed of descent.

Early in the design process, the idea of using a parachute was eliminated to reduce the unpredictability in the payload landing location. The mechanism that would be further developed would be using a tether to create a controlled descent. A 3D printed spool is fastened to Spinny 2.0 with enough fishing line to lower the UGV to the ground from the minimum flight altitude of 100 ft MSL. We use A low infill percentage and a small radius for the 3D printed spool to reduce the weight of the spool and is attached to the aircraft using laser cut acrylic plates that connect to the standoffs in the core of the drone. A small 12V Brushed DC motor is used to drive the spool, and torque calculations were made to ensure that the moment around the motor axle generated by the UGV's weight does not exceed the stall torque of the motor; from this information, the diameter of the spool was designed accordingly. To detach the UGV from the deployment mechanism, a 3 inch, 26 GA Nichrome A wire touching the fishing line is heated to 550^oF, well beyond the melting point of the monofilament material. From a reference table the current required is 2 A and the resistance per foot (RPF) is 2.5 [3]. The total resistance of this wire is 0.65 Ω and the voltage required is 1.3 V, calculated from the two equations below.

$$R_N = L \cdot RPF \ [\Omega]$$
$$V_P = I \cdot R_N \ [\text{V}]$$

With this voltage and current, the wire is able to melt the fishing line in under 5 seconds after power to the wire is turned on and was tested using a prototype rig hooked up to a DC generator. An LED strip on the drone is used to indicate whether the power is on for safety considerations. The optimal drop time is trivial for a quadcopter like Spinny 2.0 because of its ability to hover. To perform the drop, Spinny will fly to the drop location and wait to ensure that the UGV is not swaying.

2.8 Cyber Security

The ground station is capable of sending commands to the drone over a WiFi link, including takeoff, land, RTL, and even throttle cut, so it is vital to protect this connection. To prevent attackers from gaining access, the WiFi network is WPA2 protected; a password is required to join the network and network activity is AES encrypted, as is the modern standard. The antenna tracker and button panel also communicate over this network. While this prevents attackers from eavesdropping and controlling our systems, it does not prevent what is known as a deauthentication attack, in which an attacker disconnects clients on the network. Deauthentication frames (in which a router instructs a device to disconnect) are not encrypted as per the 802.11 standard, so this attack is almost impossible to prevent.

Radio is used for communications between the ground and the UGV, as well as the drone. Access to these communication links provides precise control and knowledge of the UGV and the drone, so transmitted data is obfuscated. This is done by using base64 encoded Google Protocol buffers, which can only be decoded if one has access to the specific protobuf schema that is used.

3 Safety, Risks, and Mitigations

3.1 Developmental Risks & Mitigations

Since UAS@UCLA will be using a modified version of its existing 2018 flight vehicle, many risks associated with manufacturing hazards were bypassed. Everstill, new members were given walkthroughs of the lab space and were required to take an online safety course administered by the university. All members, including returners to the team, who would participate in any manufacturing were also given training on the drill press, bandsaw, grinders, and other power tools, even though use of them was limited this year. Lastly, the UAS@UCLA labspace was always monitored by a director or UAS@UCLA board member when any members were inside. Flight-related hazards, such as crashes or propellor injury, were mitigated through a strict software-testing protocol. All flight software was tested first on ground station simulation, then on small test drones with identical on-board hardware as the main flight vehicle to ensure all fatal bugs were caught. Once software updates were deemed safe, Spinny hover and stability tests were conducted in a controlled, isolated space outside the UAS@UCLA labspace. Finally, the full flight vehicle was tested several times at a UAS airfield located close to the university. With every flight, the direction of propellor rotation, tightness of bolts, and sensor calibration were all checked to ensure the safety of the team, as well as any possible bystanders or property.

[3] Nichrome wire calculator.

Developmental Risk	Occurrence	Severity	Mitigation
Manufacturing Injury	Infrequent	Medium	• Mandatory safety training for all new and returning members • Manufacturing spaces equipped with PPE, including goggles, gloves, and face masks • Enforcement of rules in lab space
Crash Injury	Rare	High	• Rigorous testing in simulations and smaller test drones before flying competition drone • Hardware redundancies • Failsafe feature • Pre-flight safety checks for hardware and structural components • Testing of drone in official airfield • Safety whistle and LED strip to notify spectators
Propeller Injury	Rare	High	• Spectators must stand 30 feet away during takeoff and landing • Safety whistle and LED strip to notify spectators • Labeled propellers to ensure correct orientation
Battery Related Injury	Rare	High	• Serial code tracking system to monitor battery charging • Li-Po safe storage bags

Figure 3.1: Developmental risk analysis

3.2 Mission Risks & Mitigations

Flying a mission poses risks to not only to the flight vehicle, but also to team members and spectators. UAS@UCLA wants to ensure that we identify the most significant risks and understand how to to mitigate them. A summary of our findings are shown in **Figure 3.2**

The general trend that we can see from **Figure 3.2** is that mission incidents are not very common. This is largely due to the great deal of work that is done in both a simulation environment and with smaller test drones before Spinny flies its missions. However, if an incident were to occur, the results will most likely result in the disfiguration of the drone. Ensuring that an incident doesn't occur is one of the team's top priorities this year because accidents can greatly disrupt the drone's development cycle, and furthermore, will cost the team valuable time and resources. This is why UAS@UCLA has such rigorous testing before flying the competition drone and safety features like a manual override.

There are smaller inherent risks in our system that our team has had to accept and work around. For instance, our high speed WiFi link for communicating with the drone will occasionally drop out, especially at long distances away when the drone is scanning the area where vision targets appear. These risks have been minimized by adding redundancy to our system to provide a reliable failover in the event of a primary system failure. At the core, all of the controls used on our drone can be bypassed by manually sending a signal to initiate a return-to-launch if the drone behaves erratically. This signal goes straight to the flight controller unit that we chose, bypassing our custom control system completely.

Mission Risk	Occurrence	Severity	Mitigation
Loss of WiFi communication with the drone	Frequent	Medium	• Pilot can control mission execution via R/C controller • Return to home if connection cannot be re-established • Redundant telemetry interface (RFD900)
Propeller injury	Rare	High	• Propellers only installed before test and competition flights • Human operators remain far away from drone when propellers are installed and safety switch is enabled
Autonomous error	Infrequent	Medium	• Continuous integration implemented on the codebase to ensure code on the production branch is constantly maintained and tests still pass after new code is added • Controls were first tested in simulation before being tested on our secondary UAS, and were only pushed to our primary UAS after successful flight on both of those previous testing platforms • Manual override by a human pilot and Return-To-Launch modes always available throughout flight
Pilot error	Rare	High	• Return-to-launch mode enabled on connection loss • Controls were first tested in simulation before being tested on our secondary UAS, and were only pushed to our primary UAS after successful flight on both of those previous testing platforms • Manual override by a human pilot and Return-To-Launch modes always available throughout flight
Damage to LiPo batteries	Rare	High	• Batteries stored in Li-Po safe bags. • LED Strips to display battery power state • Batteries placed in easily accessible area and can be easily separated from flight system with velcro

Figure 3.2: Mission risk analysis

4 Conclusion

UAS@UCLA has spent the last few months updating the flight system to fix the shortcomings from the 2018 competition and to meet the new requirements presented in the 2019 competition. The flight system, a heavy-lifting quadcopter, uses the same airframe as last year with minor adjustments to the structural components to allow for new subsystems, such as the Unmanned Ground Vehicle, deployment mechanism, and avionics box. Using a similar method to last year, these airframe components were first developed with computer-aided design software and iterated upon with prototype and field testing to ensure mission capabilities. With the new upgrades to the system, we believe that our system has the ability to effectively perform the tasks at competition and improve upon our performance from last year.