The University of Texas at Austin

Department of Aerospace Engineering and Engineering Mechanics

UT Umanned Aerial Vehicle Group

The 2012 AUVSI SUAS Competition

Patuxent River, MD

Authors: Ben Harrison, Christopher Cale, Wiley Mosley

Project Manager: Ben Harrison
Target Acquisition System Lead: Jason Kish
Ground Control Station Lead: Wiley Mosley
Avionics Lead: Christopher Cale

ABSTRACT

This report provides a description of the Phoenix II UAS designed by the UT Austin UAV Group from the University of Texas at Austin for the 2012 AUVSI SUAS competition. The airplane is a student designed and built airplane. It uses an Ardupilot Mega autopilot. Phoenix II UAS uses a gimbal mounted video camera to transmit real time video to the ground control where the mission is controlled with dynamic capabilities. Target detection and analysis is performed on the ground during flight. This report provides a system overview and discusses mission operation procedures. Lastly, an overview of the system contingency plans is provided to demonstrate efforts to mitigate operational risks.

Table of Contents

1. INTRODUCTION

1.1 Team

The UT Austin UAV Group is composed of undergraduate students from the Aerospace Engineering, Electrical Engineering, and Computer Science Departments. This will be the fourth year that the UT Austin UAV Group has attended the annual AUVSI competition.

1.2 Mission Requirements

The goal of the flight demonstration is to have our aircraft gather intelligence on the surrounding area under realistic military operations circumstances with the greatest amount of autonomy possible. The competition rules require the following items to be demonstrated in flight:

1. Successful takeoff, landing, and navigation
2. Maintain situational awareness
3. Identify targets and their characteristics
4. Retrieve data from Simulated Remote Intelligence Center (SRIC)

1.3 System Overview

Designed and built by students of the UT UAV Group at the University of Texas at Austin, Phoenix II consists of three primary subsystems: the airframe, the avionics and autopilot, and the imagery system. The autopilot is an Ardupilot Mega, which is open-source and provides an easy to use interface. The ground station Mission Operation Console (MOC) computer is used to manage the flight computer through the Mission Planner software. The imagery system consists of a security camera mounted on a two degree of freedom gimbal, which transmits video to the ground station antenna. The video is displayed in real time on a monitor and a laptop running target analysis software (TAS).

Our approach for this year’s competition was to reduce complexity, interfaces, and communication links. The work that was done throughout the year to build a new system to achieve these goals has not yet been completed, so certain advances that are under development will not be implemented at this year’s competition. Therefore, the objectives for the UT Austin UAV Group’s flight demonstration this year are as follows:

1. Execute autonomous take off
2. Fly the predetermined course with the ability to add additional waypoints mid-flight
3. Survey the terrain for targets and identify them using a human in the loop
4. Analyze identified targets using autonomous target analysis
5. Execute an autonomous landing.

The new flight computer that is under development along with its Wi-Fi link is not ready for competition; hence the new competition objective of connecting to the SRIC and retrieving data will not be attempted.

1.4 Mission Operations

The Phoenix II UAS is operated by a team of four:

a. MOC Operator: The MOC Operator is responsible for managing the flight plan and interfacing with the onboard flight computer. He verifies connectivity with the UAV, aircraft status and position at all times and alerts the Mission Director if there are any problems. He is also the liaison between the Safety Pilot and the other ground station team members.

b. TAC Operator: The TAC Operator is responsible for controlling the camera and identifying targets. The TAC Operator coordinates the airplane location with the MOC Operator to maximize the target acquisition system's effectiveness.

c. Safety Pilot: The Safety Pilot is responsible for flying the UAS in the event of an inoperative autopilot, and has the final say on the airworthiness of the aircraft. The Safety Pilot is also responsible for the safety of the flight crew and spectators during the flight. Control of the aircraft is managed through a defined three-step hand off procedure between MOC operator and Safety Pilot.

d. Mission Director: The Mission Director is in charge of overall flight operations. He is responsible for maintaining all pre- and post-flight checklists, and issues clearances for aircraft operations. He is the chief liaison between the team and the judges.

The following are the concatenated list of procedures that the team uses to execute the mission:

1.4.1 Pre-Flight Procedures

a. Prior to the mission, the UAS undergoes an initialization procedure. Initialization is complete once all aircraft and gimbal controls have been checked, communication links are verified, and the TAC operator has verified that the camera is outputting a video signal.

b. The mission waypoints and no-fly zones are loaded into the MOC and the operator notifies the mission director that the system is operational.

c. Once the judges issue the command via the Mission Director, the Safety Pilot taxies the aircraft to the center of the runway.

d. The motor circuit breaker is closed once takeoff clearance has been issued, and the autopilot autonomous take off is initiated.

e. On the Mission Director’s command, the in-flight procedures are initiated.

1.4.2 In-Flight Procedures

a. Once sufficient altitude has been reached, the autopilot proceeds to follow the waypoints as commanded by the MOC.

b. The TAC Operator actuates the camera as necessary in order to identify as many targets as possible. The target analysis is carried out with autonomous software and is overseen by the TAC operator. Utilizing the real-time video, the TAC Operator is able to provide the judges with actionable intelligence of any target that is fully recognized, pending the approval of the Mission Director.

c. Once all the targets have been found and detected, the TAC Operator can request the airplane to be redirected to unrecognized targets for autonomous re-identification. Upon approval from the Mission Director, the MOC Operator issues new waypoints to the flight computer.

d. Throughout the mission the Mission Director can instruct the MOC Operator to execute any retasking as necessary.

e. On the Mission Director’s command, the descent and landing procedures are initiated, and autonomous landing is executed by the autopilot.

1.4.3 Post-flight Procedures

a. Once the aircraft is landed and taxied safely, the Mission Director authorizes the shutdown procedures.

b. The Safety Pilot opens the motor circuit breaker and the avionics is powered off. Battery packs are disconnected and handed to the Safety Pilot.

c. The TAC Operator hands the target list to the Mission Director for a final review. The target list is handed over to the judges.

d. The Mission Director notifies the judges that the mission is complete.

2 AIRFRAME

Phoenix II uses a custom built airframe, which was used in the 2011 AUVSI SUAS competition. The airframe has a large wing area to achieve slow loiter speeds, and large fuselage hatches for easy access to the payload. Carbon-kevlar composites are used extensively in the fuselage, wings and empennage to reduce weight and increase durability. Phoenix II is shown in Figure 1.

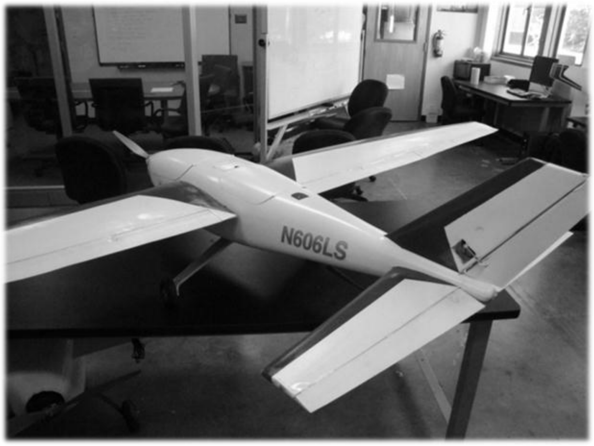


Figure 1: Phoenix II airframe

The following sections discuss Phoenix II's fuselage, wings, and propulsion system in more detail:

2.1 Fuselage

The fuselage of the Phoenix II UAS was constructed using conventional composite manufacturing techniques. Kevlar was used to ensure a lightweight and durable fuselage that would protect the payload in the event of a crash. Key structural areas such as the motor mount, wing and empennage joints, and landing gear mount were reinforced with carbon fiber. Large hatches were made to allow for easy payload access. The hatches were made from carbon fiber reinforced fiberglass to save weight. Foam-Kevlar bulkheads were added to increase rigidity and to provide mounting surfaces for the camera gimbal and avionics.

2.2 Wings

The wings were constructed from EPP foam cores. Carbon fiber spars were placed in the wing panels, and were anchored to fiberglass and carbon fiber ribs. Carbon fiber spar caps were added to the top and bottom surfaces of the wing, and were joined together via a fiberglass shear web. Each wing was sheeted in Kevlar for added torsional rigidity and durability. The wings have ailerons and flaps.

2.3 Propulsion system

Propulsion trade studies were done to select a propulsion system for the Phoenix II UAS. With a maximum power rating of 1900 W (2.1 HP), the Hacker A60-20S brushless electric motor proved most suitable for the Phoenix II propulsion requirements. The propulsion system utilizes ten 8000 mAh Li-polymer battery cells connected in series. These cells allow the propulsion system to run at peak efficiency for the required mission endurance (~35 minutes).

3 COMMUNICATIONS

The Phoenix II UAS utilizes four communication channels as shown in Figure 2. The flight computer communicates with the MOC through a bi-directional 900 MHz channel using a pair of XBees. This link exchanges critical UAS information between the two systems, including information critical for the TAC. The video feed is transmitted live from the UAV directly to the TAC via a 1,280 MHz connection. This allows for real-time reconnaissance capabilities and the collection of actionable intelligence. The 2.4 GHz remote control is used to direct the camera.

The MOC and TAC are connected over a local area network connection through LabVIEW's Data Communication Toolbox. The TAC receives updates on the aircraft state. Because no direct connection exists between the TAC and the flight computer, all camera and gimbal information is relayed through the MOC, resulting in potential synchronization errors. To mitigate this risk, the payload state is transmitted down from the flight computer, rather than inferred from the operator's inputs. This also serves as a verification of the communication link between the TAC and the flight computer.

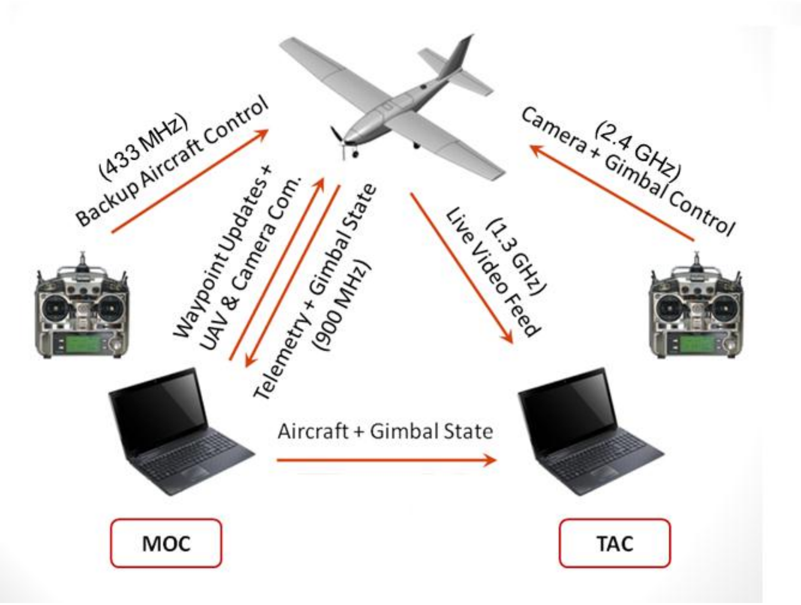


Figure 2: Communication Architecture

4 FLIGHT SYSTEMS

The Phoenix II avionics consists of three main systems: The flight computer (Ardupilot Mega board), Mission Operation Console (MOC) and the power regulation system. The flight computer is controlled from the ground control station using the Mission Operation Console (MOC).

4.1 Flight Computer

The flight computer is a combination of an IMU shield and UAV controller; both are commercial components. The flight computer is connected to the backup controller's receiver, and the outputs signals actuate the control surfaces. The IMU is capable of measuring linear and angular accelerations, and the air speed is sensed with a pitot tube sensor connected to the IMU shield. The altitude is determined using a Bosch pressure sensor (temperature adjusted), and the GPS location is provided to the flight computer through the appropriate port. A picture of the shield/controller combination is shown in Figure 3.

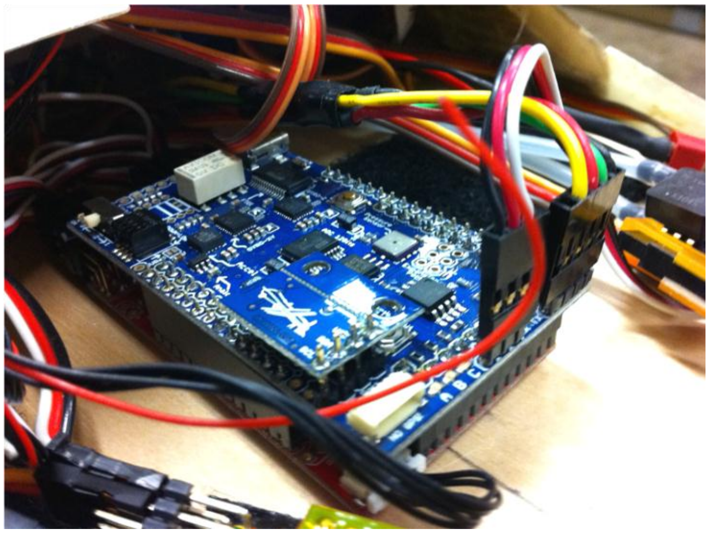
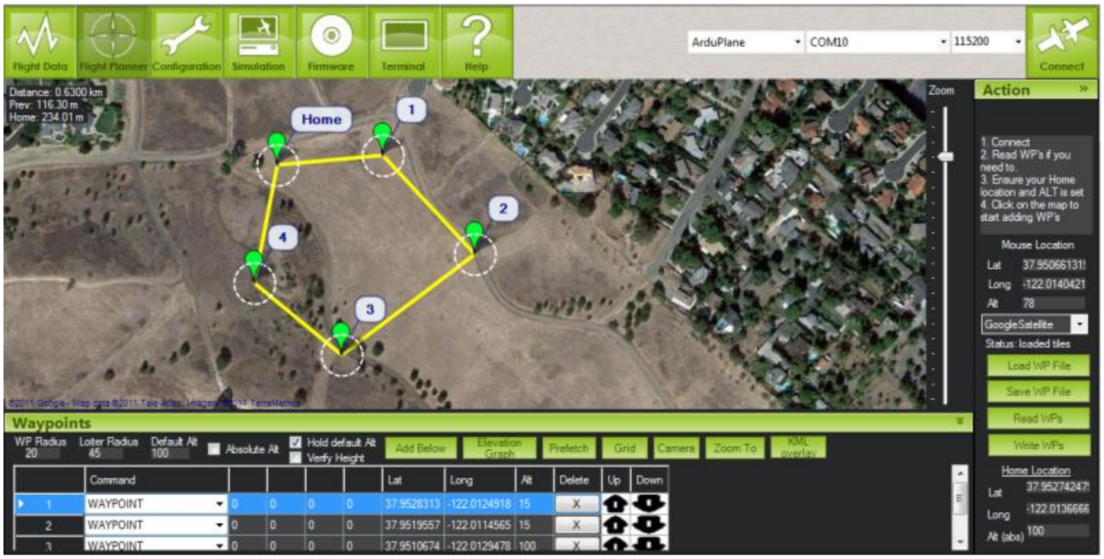


Figure 3: Ardupilot Mega flight computer

4.2 Mission Operation Console

The Mission Operation Console (MOC) is the central component of the ground control station and serves as the UAV's sole link to the ground. The MOC performs a number of tasks, including relaying information and commands to and from the UAV, controlling waypoint navigation, and providing the operators with situational awareness. The map display on the MOC shows satellite imagery of the mission area with a number of overlays to increase situational awareness. These overlays include a UAV location and heading marker, UAV ground track, flight plan waypoints, and the no-fly zone. The MOC has the capability of re-tasking the UAV during the mission by allowing the operator to add, remove, and modify individual waypoints. A picture of the Mission Planner is shown in Figure 4.



Distance: 0.6300 km
Prev: 116.30 m
Home: 234.01 m

1. Connect
2. Read WP's if you need to.
3. Ensure your Home location and ALT is set
4. Click on the map to start adding WP's

Mouse Location
Lat 37.95066131
Long -122.0140421
Alt 78

Status: loaded tiles

Home Location
Lat 37.95274247
Long -122.0136666
Alt (abs) 100

WP Radius 20   Loiter Radius 45   Default Alt 100   ☐ Absolute Alt   ☑ Hold default Alt   ☐ Verify Height

	Command					Lat	Long	Alt	Delete	Up	Down
1	WAYPOINT	0	0	0	0	37.9528313	-122.0124918	15	X	⬆	⬇
2	WAYPOINT	0	0	0	0	37.9519557	-122.0114565	15	X	⬆	⬇
3	WAYPOINT	0	0	0	0	37.9510674	-122.0129478	100	X	⬆	⬇

Figure 4: Ardupilot Mega Mission Planner

4.3 Power Regulation System

The power distribution circuit provides regulated power at 5, 12, and 40 Volts. The avionics Access Panel includes switches to power the system. There are two switches to activate the main power source to the flight computer and receiver (primary and backup). A composite video jack is also included in the access panel for testing the camera's video feed. Lastly, the access panel contains a motor circuit breaker (rated for 50A). This allows the operator to arm and disarm the motor from behind the propeller.

5 IMAGERY

The imagery system was designed and built in-house and includes a real-time target analysis suite which employs several sophisticated algorithms to accomplish the high demands of contemporary surveillance and reconnaissance missions.

5.1 Requirements and Specifications

The competition rules outline several performance metrics for scoring the imagery system [1]. Other requirements and specifications are derived from the competition requirements in order to design an effective system. Table 1 lists

[1] "The Association for Unmanned Vehicle Systems International Student Unmanned Air Systems Competition Rules," AUVSI Foundation, http://65.210.16.57/studentcomp2010/ rules/2010RFP20090824_updated.pdf (current Nov. 2010).

these specifications. In particular, the system should be able to capture images of targets and report their locations and features to the judges. The team aimed for 80% target detection accuracy at a real-time rate of 10 frames per second - specifications that are in line with conventional detection algorithms[2]. The imagery system uses a combination of modular hardware components and novel software algorithms to meet and exceed the design specifications. The following sections discuss the imagery hardware, stabilization capabilities, target detection, and target analysis.

Table 1: Design Specifications

Specification	Constraint
Field of Regard	60° in all directions
Detection Precision	Within 250 feet
Detection Accuracy	80% correct detection
Target Feature Recognition	At least two characteristics
System Speed	Real time (10 frames per second)

5.2 Imagery Hardware

A Sony FCB EX-980S optical video camera is mounted on a dual-axis gimbal. The camera supports up to 26X optical zoom and outputs analog S-video with a resolution of 480x720 pixels[3]. The gimbal provides an enhanced field of regard and stabilizes the camera against aircraft movement. Two high torque servos controlled through the aircraft flight computer actuate the gimbal and two absolute encoders provide gimbal position feedback. Figure 5 shows the camera-gimbal system and the hardware enclosure, which houses the imagery circuitry. The system uses a 2.4 GHz receiver for control and a 1280 MHz transmitter to stream the live video to the ground station. The ground station includes an analog to digital video converter and a dedicated laptop called Target Acquisition Console (TAC) to run the imagery software.

[2] P. Viola, and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Comp. Vision and Pattern Recognition (CVPR), IEEE SC Press, 2001, pp. 3-22.

[3] "Color Block Cameras FCB-EX Series," *Sony Corporation,* 2005.

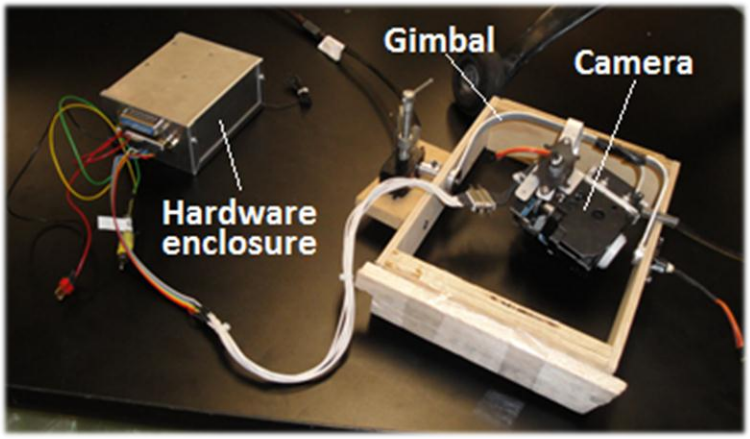


Figure 5: On-board Imagery Hardware

5.3 Target Acquisition Console

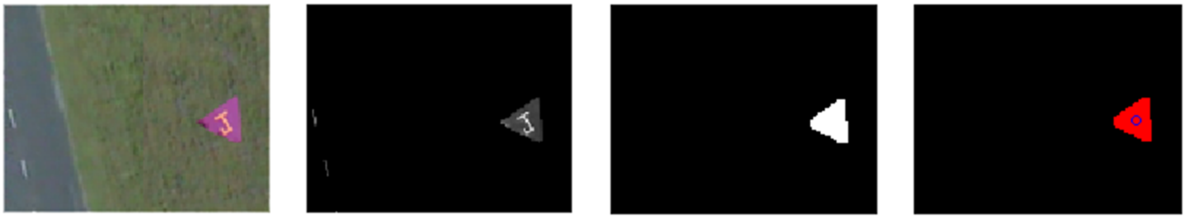
The Target Acquisition Console (TAC), houses all imagery software, which is written in LabVIEW. The TAC program includes a user interface panel and algorithms that detect targets, calculate target coordinates, and identify target features. A single operator controls all aspects of the imagery system using a dual-joystick gamepad. In the case of false detections, the operator can manually override the detection. This provides an additional layer of safety against false recognition and identification, mitigating errors that may have catastrophic consequences in real-world situations. The gamepad provides TAC operator the ability to manually override and direct the target detection and analysis, and in the future, the same gamepad will be used to control camera gimbal and zoom. The camera gimbal is currently configured to receive commands from a 2.4 GHz controller.

5.4 Target Detection

Detection is the first stage in target acquisition. As per the rules, a target is a plywood cutout of a geometric shape and has two unique colors. Additionally, all targets are between four and eight feet in length, which effectively limits their aspect ratios between 0.5 and 2. Runways, trees, and other occlusions, make shape and edge detection very difficult. However due to the nature of the targets, a color-based approach is feasible.

The target detection algorithm operates on a single image frame from the video stream, which is identified in real time by the TAC operator and then sent to the TAC system to retrieve the target from the image frame and process the target for analysis. Figure 6 shows an example image processed through the detection method. The image is represented as the union of a background (the airfield) and foreground (the targets). Experimental work performed on last year’s flight video shows that the airfield is normally distributed in color. Thus, each pixel is ordered by a distance metric related to the background’s mean and covariance. Using this ordering, an *outlier image* is constructed that contains the foreground pixels (plus noise) while preserving all spatial information.

The outlier image is further reduced based on a threshold and is noise-filtered to create a smoothed binary image. The binary image is segmented into objects using connected component labeling, and each object's spatial centroid and bounding rectangle is computed. To reduce the rate of false positives, objects are filtered by size and aspect ratio; objects that do not fall within the bounds of mission target specifications are discarded. The final result is a list of detected target centroids and bounding rectangles.



Original Image Outlier Image Smoothed Binary Image Label Image

Figure 6: Target Detection Progression

During the early testing stages, the team tested the target detection algorithm in MATLAB using synthetic data and previous flight video. Later, the team tested the algorithm's performance in embedded C on a digital signal processor using a scaled airfield. The airfield contained targets, trees and a large runway. The camera was swept along the field and the ratio of detected targets to the number of targets seen by the camera was calculated. This was done 10 times and the results were averaged. Table 2 summarizes the detection results. The algorithm operated at 12 frames per second with 85% correct detection. Notably, the algorithm also had a low false detection rate of about 10%. These false positives were mainly from the lane markers on the runway. The algorithm rarely marked the entire runway or the trees as targets. The test showed that certain colors, such as black were not detected as easily as other colors, such as red. This is because black is not an outlier in a grassy-green background. However, these targets were usually still detected because the color of the letter inside the target deviated sufficiently from grass. A full scale outdoor static test was conducted with similar results, exhibiting the method's robustness to vegetation and other obstructions. The team plans to test the algorithm in flight several times over the next few weeks.

Table 2: Target Detection Performance

Specification	**Result**
Correct Detection	85.00%
False Positives	10.00%
Detection Speed	12 frames per second

The target detection algorithm records target coordinates in an image. Although this is useful for comparing targets located in the same frame, it does not afford a basis for comparison throughout the mission. A number of practices exist for converting the image coordinates to an absolute position. When the camera is pointed straight down, it is sufficient to assign the target the location of the aircraft at the time of detection. However, such an approach can cause an error of up to 900 feet, which is more than three times higher than the minimum constraint listed in Table 1. To increase position accuracy, the imagery system uses a target position determination algorithm that converts a target's position from a camera-based reference frame to an absolute position by factoring in the aircraft's attitude and the gimbal's state. This enhanced method also employs the reverse functionality, enabling the imagery system to perform reverse targeting with *a priori* position information, such as the general location of a "pop-up" target.

The team performed Monte Carlo error analysis on the target position determination algorithm by introducing typical errors to the sensor measurements. Figure 7 shows the standard deviation of error at 500 feet altitude for 60 degrees in all directions from vertically below the aircraft. The results show that the expected error is lower than the 250 feet requirement with 99% confidence. Furthermore, the analysis shows that limiting the view to at most 26 degrees from the vertical keeps the expected error below the more stringent requirement of 50 feet with 99% confidence. This corresponds to the red circle shown in Figure 7. Flying the aircraft at lower altitudes further reduces the error.

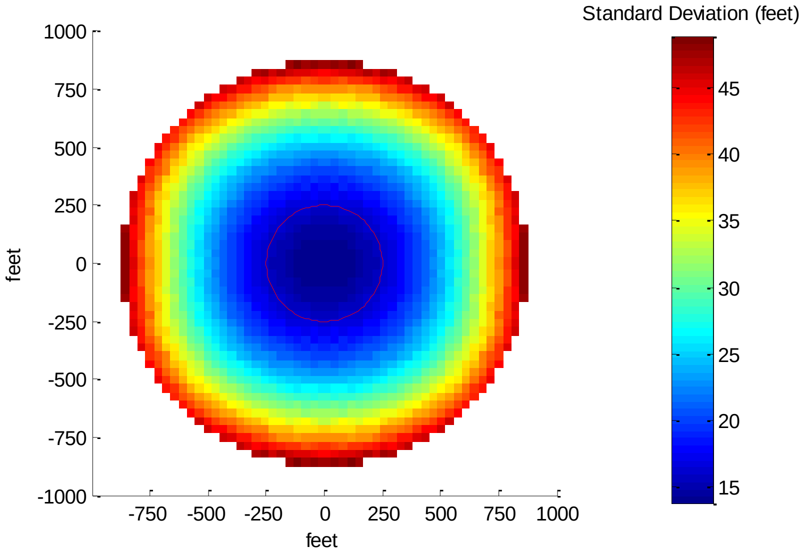


Figure 7: Target Position Determination Error Analysis

5.5 Target Analysis

The analysis portion of the imagery software identifies five target features from a cropped image containing the target. Figure 8 shows the analysis process.

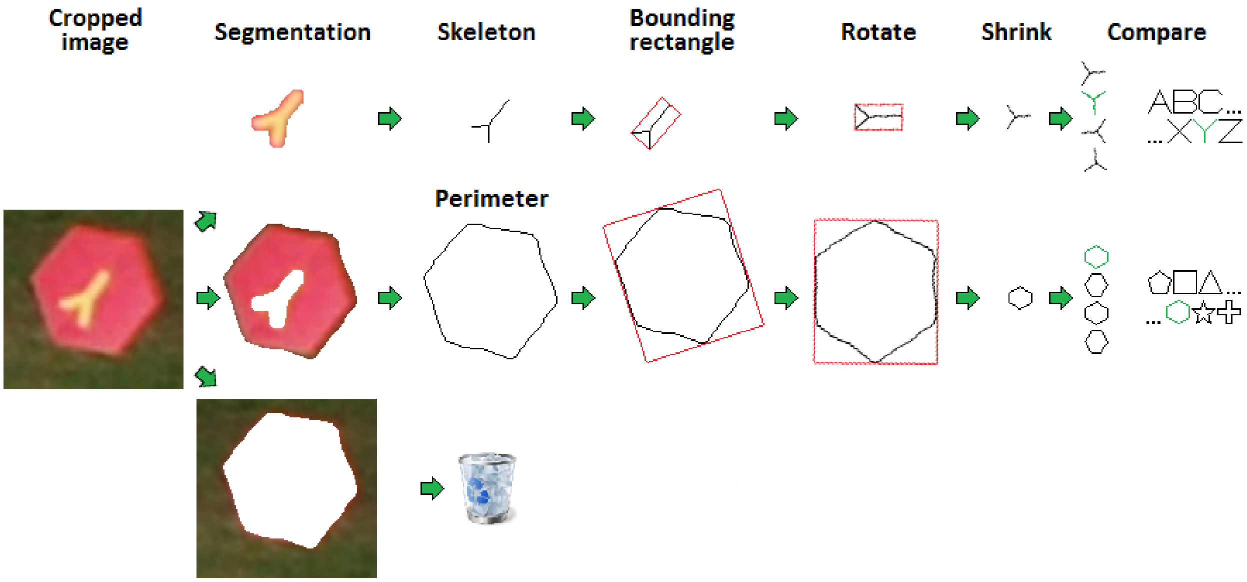


Figure 8: Sequence diagram for target analysis[4]

Analysis begins with segmentation where the algorithm finds and separates the image into three dominant color segments. The largest segment represents background features (grass or dirt), and is discarded. The smallest segment is the letter while the medium segment is the shape. Segmentation usually gives poor results for noisy, blurred, or low-quality images, therefore the airplane must fly at low speeds with gimbal stabilization turned on for target analysis to work reliably. Segmentation also gives bad results when two of three colors are similar (for example, a green target surrounded by grass) or when background features have two distinct colors (for example, grass mixed with dirt). The team is currently developing algorithms to minimize these problems. For most other cases, segmentation correctly identifies shape color and letter color and accurately isolates important target segments for further analysis.

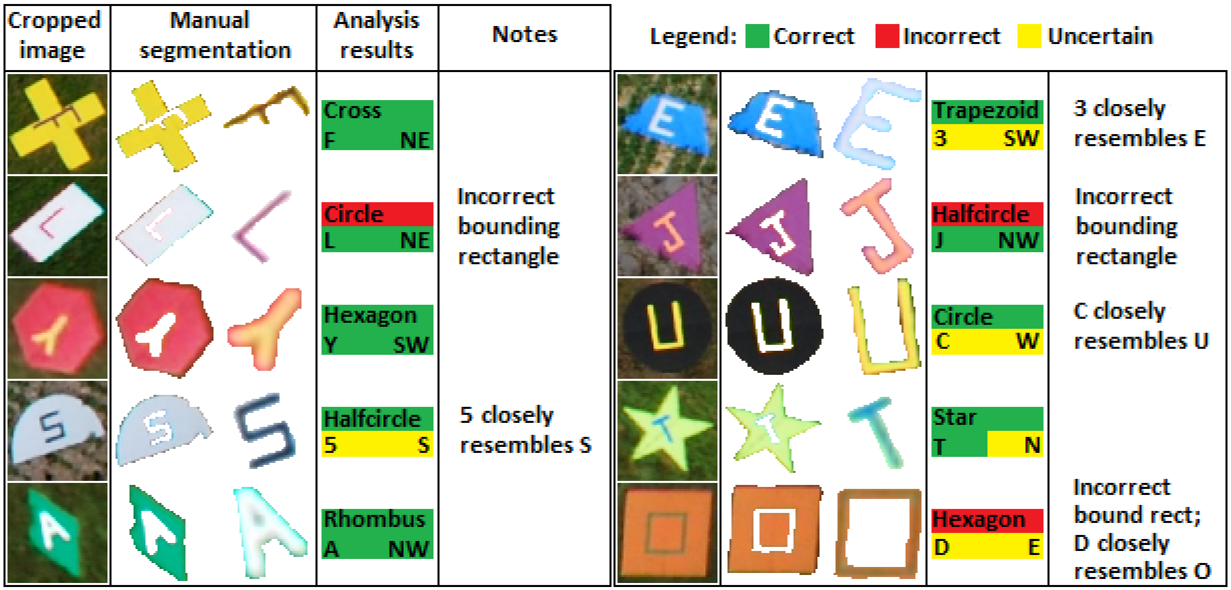
Before performing further analysis, the software applies a noise removal algorithm to the segments. This process discards any rogue pixels that are not consistent with the overall segment color. To analyze a shape, a perimeter is extracted out of the shape. For letters, each letter is eroded until it becomes one pixel thick and resembles a skeleton. Since letters with different fonts usually have similar skeletons and only embellishments such as serifs distinguish one font from another, this step ensures that the algorithm works reliably on all fonts.

Next, each letter is rotated and shrunk. A letter rotation is done to re-orient the letter into either upright, resting on its side, or inverted orientation, and prepares each letter for the identification stage. The angle of rotation, after some modifications, yields the target's cardinal orientation. After rotation, each letter is shrunk until its width and length becomes one unit. Shrinkage allows the identification algorithm to identify letters of any size and aspect ratio.

[4] “NCSU Aerial Robotics Club 2010 Results”, [Online]. http://uavs.us/?cat=3 (current May 2010)

The identification algorithm compares each letter with a library of pre-existing letter templates and finds the best match based on the amount of overlap. The program minimizes mistakes by checking overlap in both directions (for example, a P overlaps very well with an R, but not the other way around). Because the code normalizes letter dimensions, aspect ratio, thickness, font, and orientation, the library needs only 35 templates to take care of all possible cases. This allows the entire analysis algorithm to run in less than ten seconds for each target. This advantage allows the algorithm to autonomously provide target data during flight without any post-processing. This allows the system to provide the judges with actionable intelligence during the flight. In addition, the library approach makes the program versatile and robust. Shapes also undergo a similar process (using a different library with only 11 templates).

The current analysis program identifies target features with roughly 15% accuracy. Accuracy is low because image segmentation frequently makes mistakes. To test the method's accuracy after segmentation, each image was manually segmented and run through a partial analysis program. Figure 9 shows the results. Shape identification accuracy increased to 70%, letter identification accuracy increased to 60-100%, and orientation accuracy increased to 50-100%. Given proper image segmentation, the program will autonomously and correctly identify most target features.



Legend: 🟩 Correct 🟥 Incorrect 🟨 Uncertain

Cropped image	Manual segmentation	Analysis results	Notes
		🟩 Cross 🟩 F 🟩 NE	
		🟥 Circle 🟩 L 🟩 NE	Incorrect bounding rectangle
		🟩 Hexagon 🟩 Y 🟩 SW	
		🟩 Halfcircle 🟨 5 🟨 S	5 closely resembles S
		🟩 Rhombus 🟩 A 🟩 NW	
		🟩 Trapezoid 🟨 3 🟨 SW	3 closely resembles E
		🟥 Halfcircle 🟩 J 🟩 NW	Incorrect bounding rectangle
		🟩 Circle 🟨 C 🟨 W	C closely resembles U
		🟩 Star 🟩 T 🟨 N	
		🟥 Hexagon 🟨 D 🟨 E	Incorrect bound rect; D closely resembles O

Figure 9: Test results for target analysis[5].

[5] (2010, June 21). NCSU Aerial Robotics Club 2010 Results[Online]. Available: http://uavs.us/?cat=3

6 SEARCH TRAJECTORY

Once the aircraft enters the search area, it is free to fly any trajectory that stays within the no-fly zone boundaries, between the altitudes of 100 and 750 feet MSL. Since the set of possible trajectories is infinite, a number of practical trajectories were created. Figure 10 shows three types of trajectories that were considered.

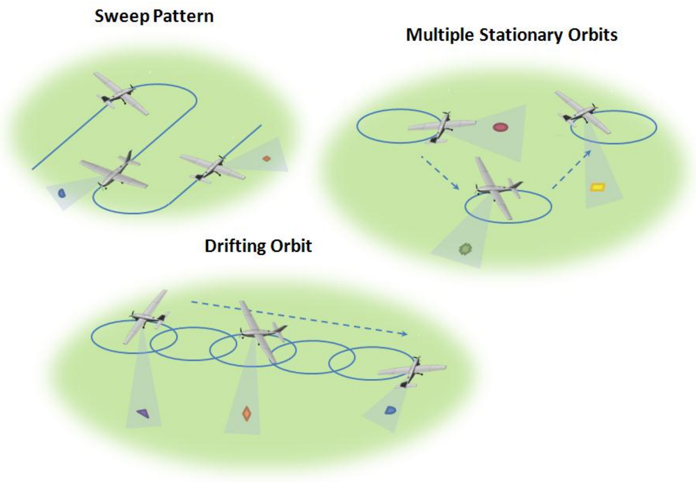


Figure 10: Types of trajectories considered for search area trajectory

Using last year's search area, a number of flight plans were constructed from the trajectories shown above. Each flight plan consisted of a set of waypoints. A selection method was then devised to compare the performances of the candidates. Generally, trajectories can be optimized with respect to a multitude of parameters. In this case, the probability of target detection was chosen as the best metric for selection.

A Monte Carlo MATLAB simulation was used to evaluate this metric for each trajectory. Using a simplified aircraft dynamics model and the autopilot's guidance and control code, the simulation generated a trajectory from a proposed flight plan. Two sample trajectories generated using last year's search area is provided in Figure 10 and Figure 11. The differences between the input flight plans (dotted blue lines) and the output trajectories (solid red lines) are primarily a consequence of the lag in the aircraft's control dynamics, and of randomly changing winds.

A large number of trials were conducted for every trajectory. Each trial used a different set of 10 targets randomly scattered about the search area (indicated by the dashed red bounding polygon). Accounting for gimbal lag, camera optics, and aircraft position and attitude, the number of targets visible on the screen was recorded at every point along the trajectory. Additionally, apparent target pixel sizes were used to exclude instances where the targets were too small for detection.

The simulation results for the trajectories shown in Figure 11 and Figure 12 are provided in Figure 13 and Figure 14, respectively. The color at a point (x, y) represents the probability of seeing x or more targets for at least y amount of seconds throughout the entire flight. For example, the top right corners show that there is a low probability that 10 or more targets will be seen for long time. Conversely, the bottom left corners suggest that there is a high probability that at least one target will be seen for one second. Lastly, a trajectory is considered superior to another if its color transition line is shifted up along the vertical axis. Thus, one would conclude that the trajectory in Figure 10 is superior to the one in Figure 11. This conclusion is supported by other practical considerations, such as the perpetual unsteadiness of the aircraft's attitude throughout the sweep trajectory.

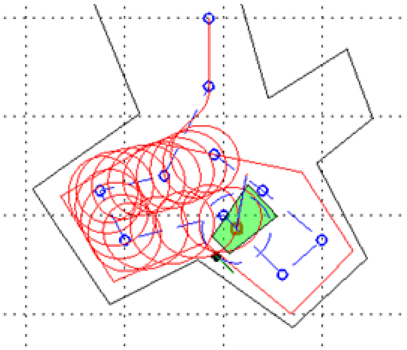


Figure 11: Drifting Orbit Trajectory

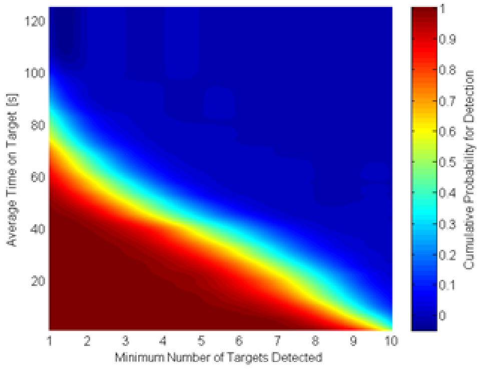


Figure 13: Cumulative probability distribution for target detection - drifting orbit trajectory

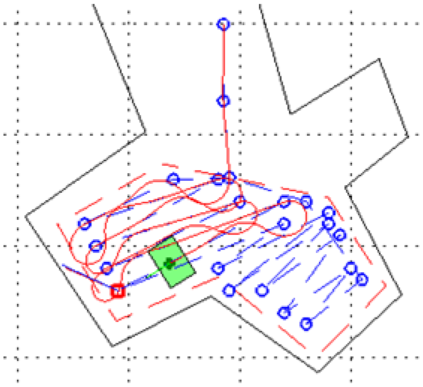


Figure 12: Sweep Trajectory

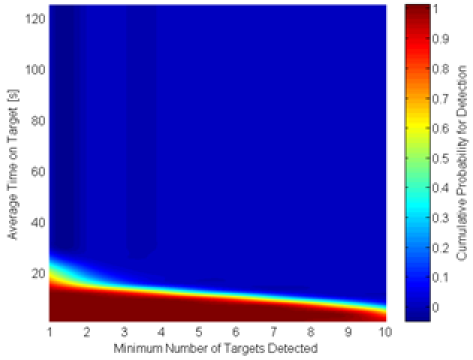


Figure 14: Cumulative probability distribution for target detection - sweep trajectory

7 Safety and Testing

The occurrence of accidents can never be completely eliminated, but the following measures taken by the UT UAV Group have prevented any damage to the aircraft, personnel, and property during testing this year:

- Two independent batteries power the flight computer. The primary battery can fail without risking the mission viability.
- Phoenix II is programmed to enter failsafe mode when signal to the backup controller is lost. This feature has been thoroughly tested on the ground.
- The circuit breaker prevents the propeller motor from overdrawing current, and the switch on the access panel prevents the motor from being activated accidentally.
- Servo safety clips are used on all servo connection to prevent disconnections from occurring in flight.
- The fuselage of Phoenix II is extremely durable which protects the internal components in case of a mishap.
- The preflight check ensures that all systems are wired correctly, and all items within the aircraft are completely secure.

The Ardupilot Mega autopilot system has been ground tested and flight tested. A ground test preceded every flight test to ensure the control surfaces were responding correctly to manual inputs. The aircraft was perturbed while in stabilize mode to check for the correct responses. Also, a model of Phoenix II was developed in X-Plane software in order to simulate flight. This program gave us the experience working in the Mission Planner software to become comfortable flight-testing in full auto mode.

The flight tests conducted this year have proven the viability of this system at competition, but the team is currently uncomfortable with the signal strength being achieved by the 900 MHz link at large distances. Many more flight tests are planned before the competition to verify measures being taken to increase the signal strength are effective. The automatic take off and landing features of the autopilot have yet to be flight-tested, but it is our goal to have them ready for the competition. All other flight systems we plan to use at competition are flight-tested and perform as intended with all active hardware running.

8 MISSION RISKS AND CONTINGENCY PLANS

The following is concatenated list on contingency plans in the case of component malfunctions:

Problem	Possible causes	Symptom	Response	Effect
Autopilot fails	Software crashes; Failed sensor (GPS, IMU, or airspeed)	Erratic aircraft movement	Safety pilot takes manual control	Mission continues, but becomes less autonomous
Structural failure on aircraft	Improperly secured latches; fatigue stress failure	Visual evidence of structural damage, uncontrollable airplane	Safety pilot takes manual control and attempts safe landing	Mission failure
One aileron or rudder servo fails	Poor servo wire connections	Erratic aircraft movement	Autopilot no longer effective; Safety Pilot attempts to take control	Mission continues, but becomes less autonomous
Single elevator servo fails	Poor servo wire connections	Erratic aircraft movement	Safety pilot takes manual control and attempts safe landing	Mission failure
Image transmission becomes problematic	Interference on 1.3 GHz band, Antenna fails	Low-quality video stream at ground station	Return to initial waypoint and orbit for a minute; if problem persists, land airplane	Mission continues if communication link is re-established, otherwise airplane is manually landed
Telemetry signal becomes problematic	Interference on 900 MHz band	Dropped packets seen on MOC	Return to initial waypoint and orbit for a minute; if problem persists, land airplane	Mission continues if communication link is re-established, otherwise airplane is manually landed
Engine Failure	Depleted battery packs; electric connection disconnected; motor fails	Aircraft slows down, loses altitude	Safety pilot takes manual control and attempts safe landing	Mission failure