*Autonomous Unmanned Aerial Vehicle Development*

Bulone, Domenico

*Department of Electrical Engineering and Cyber Systems*

*United States Coast Guard Academy*

New London, United States

[Domenico.V.Bulone@uscga.edu](mailto:Domenico.V.Bulone@uscga.edu)Kim, Matthew

*Department of Electrical Engineering and Cyber Systems*

*United States Coast Guard Academy*

New London, United States

[Matthew.H.Kim@uscga.edu](mailto:Matthew.H.Kim@uscga.edu)McGahey, Gavin

*Department of Electrical Engineering and Cyber Systems*

*United States Coast Guard Academy*

New London, United States

Gavin.L.McGahey@uscga.edu

Meyers, Cody

*Department of Mechanical Engineering*

*United States Coast Guard Academy*

*New London, United States*

[Cody.L.Meyers@uscga.edu](https://cgacademy-my.sharepoint.com/personal/cody_l_meyers_uscga_edu/Documents/Cody.L.Meyers@uscga.edu)Schellman, Jacob

*Department of Electrical Engineering and Cyber Systems*

*United States Coast Guard Academy*

New London, United States

[Jacob.C.Schellman@uscga.edu](https://cgacademy-my.sharepoint.com/personal/cody_l_meyers_uscga_edu/Documents/Jacob.C.Schellman@uscga.edu)Von Brock, Ryan

*Department of Electrical Engineering and Cyber Systems*

*United States Coast Guard Academy*

New London, United States

[Ryan.D.VonBrock@uscga.edu](https://cgacademy-my.sharepoint.com/personal/cody_l_meyers_uscga_edu/Documents/Ryan.D.VonBrock@uscga.edu)

*Abstract— Autonomous unmanned aerial vehicles (UAVs) have great potential to fulfill an organization’s objectives in a safer and more timely manner, when compared to existing alternatives. Specifically, for the United States Coast Guard (USCG), UAVs can be utilized to accomplish missions including search and rescue, migrant interdiction, and drug interdiction by using image processing techniques to identify people and vessels in the water.* *This paper documents the production of an autonomous unmanned quadrotor in the scope of USCG missions. Throughout the 2021-2022 school year, the Autonomous UAV Capstone group of the US Coast Guard Academy Electrical Engineering Department, guided by the Roboboat 2022 competition requirements, developed an unmanned aerial vehicle. The unmanned aerial vehicle is capable of autonomous and manual flight, including landing, taking off, and delivering payloads autonomously. This device serves as the foundation for a product that may provide the Coast Guard with an inexpensive platform to supplement its existing assets in completing core missions. Using the current prototype, the group has completed successful test flights, demonstrated the functionality of computer vision algorithms, integrated real time kinematic positioning, and performed basic autonomous flight.*

# Introduction

Autonomous unmanned aerial vehicles (UAVs) are an invaluable tool for organizations to use in their goals while minimizing the number of resources needed. Many of the Coast Guard’s missions, including search and rescue, and migrant and drug interdiction, have already been supplemented by UAVs like the Boeing ScanEagle. Small UAVs, like the one researched in this paper, have the potential to further supplement these missions. The UAV capstone project at the United States Coast Guard Academy has designed a quadcopter-style UAV over the course of the Fall 2021 and Spring 2022 semesters. A quadcopter is the common name for a relatively small, four rotor UAV. The group’s intent was to prove such a device’s potential benefit for the service. The team cooperated with another capstone group—responsible for developing an autonomous surface vessel—while using the requirements of the Roboboat 2022 International Competition as guidelines. The quadcopter was tested in two environments: both virtual and in the real-world. *Gazebo*, an open-source high performance physics engine, allowed the group to simulate testing of the quadcopter in a no-risk environment. This ensured proper functioning prior to live testing and greatly reduced the risk associated with real-world flights. The second environment was real-world testing, primarily on the Coast Guard Academy’s Cadet Memorial Field. Testing was completed in both environments to ensure the accuracy of navigation systems, positive buoyancy of the UAV, accuracy of color detection modeling, and more. Testing was imperative to create a functional product, and in doing so, the team has constructed a quadcopter capable of both remote and autonomous flight.

# Background

Over the last decade, unmanned flights have become a dominant front-runner in the world of aerial advancements. As drone flight continues to increase in popularity among private businesses and governments, research has shifted focus to a new advancement in the industry: autonomous flight.

An autonomous aircraft can be classified as one that “does not require pilot intervention in the management of flight” [1]. This is exactly what the autonomous unmanned aerial vehicle presented in this paper is designed to do. Specifically, the UAV is designed to assist the United States Coast Guard (USCG) in missions such as alien migrant interdiction operations (AMIO) and search and rescue (SAR).

The USCG is currently using one unmanned aerial vehicle flight system on a large scale: the Boeing ScanEagle. As a fixed wing drone, the ScanEagle has offered the service advanced capabilities in flying without the need for an onboard pilot. The ScanEagle, however, requires a large area for a dedicated launching system and a remote pilot. This has resulted in the ScanEagle only being deployed on larger platforms, excluding many USCG units. The USCG is certainly interested in other unmanned aerial systems that would benefit smaller platforms; however, budget constraints have prevented the service from acting. Regarding unmanned surface and air systems in March 2021, USCG Commandant ADM Schultz said “We look at commercial-off-the-shelf; following, you know, our other fellow services and seeing what they're doing. We have just got a finite amount of R&D (research and development) dollars… We just don't have a lot of that budgetary maneuver space to take high risks there” [2]. This quote signifies that the Coast Guard is not investing in its own development of an unmanned aerial system, let alone an autonomous one. This implies that a UAV designed for Coast Guard use will only come into the picture after another organization purchases and field tests one. LCDR Dom Bucciarelli, the USCG Short Range UAS Platform Manager, detailed in an email that the Coast Guard is currently looking to implement variants of the Skydio X2D and Parrot Anafi drone platforms, neither of which are autonomous. The testing of these drones is coming after their success in providing reconnaissance to Customs and Border Protection [3]. This further shows that the Coast Guard is reliant on off-the-shelf platforms that have already been tested. Thus, the USCG is not designing its own *autonomous* drone and that the service will only acquire one that has already been developed and tested. The UAV group believes that the USCG would benefit from its own production of an autonomous drone—one designed specifically for service needs and at a reduced price point.

Until recently, hardware limitations prevented all but major leaders in the drone industry from developing autonomous UAVs. Today, however, flight controllers and mission planning software that support autonomous flights are available at reasonable costs to the public: many of these products exist as open source. The UAV capstone team believes that these advancements can be used to the USCG’s advantage in the development of aerial systems. This research aims to demonstrate that a small team of undergraduate students can achieve autonomous flight at a low cost.

This year’s UAV project is a continuation of the USCG Academy’s Electrical Engineering Department’s 2021 UAV project. As a result of unforeseen challenges due to the COVID-19 pandemic, last year’s group was unable to make meaningful progress on UAV development. As a result, the UAV presented in this paper is entirely the work of the current group.

# Objective

This capstone group, in conjunction with the Autonomous Surface Vessel (ASV) capstone group, is using the 2022 RoboBoat competition requirements as guiding principles for development. As a part of this competition, the UAV will autonomously take off from the ASV, search for and fly to a dedicated target. From here, the UAV will deliver an object, provide aerial video feed for the ASV team to use while navigating their course, and return to and land aboard the ASV.

After compiling requirements from the RoboBoat Competition and other stakeholders, the group created the following requirements matrix shown in Table I.

TABLE I. REQUIREMENTS MATRIX

|  |  |  |
| --- | --- | --- |
| Operational Requirement | Functional Requirement | Non-Functional Requirement |
| The Vehicle Shall be Capable of Unassisted Flight. | The UAV must be capable of controlled flight in wind. |  |
| The Vehicle Shall Meet Competition Size Guidelines. |  | The UAV must be electronically powered: no battery voltage may exceed 60Vdc. |
| The Vehicle Shall Meet Competition Size Guidelines. |  | Together with the ASV, the dimensions must not exceed 3ft by 3ft by 6ft. |
| The Vehicle Shall Meet Competition Size Guidelines. |  | The criteria include batteries, meaning the total weight must be less than 10 pounds. |
| The Vehicle Shall be Capable of Autonomous Flight. | The UAV must be able to be controlled manually and autonomously. |  |
| The Vehicle Shall Communicate with the ASV. | The vehicles must exchange information about locations and targets of interest. | Wireless communication must utilize legal public frequencies. |
| The Vehicle Shall be Capable of Landing on an ASV Flight Deck. | The UAV must be equipped with cameras/LIDAR/RTK to land on the target. | The flight deck must be a design that is preemptively established in the computer vision module. |
| The Vehicle Shall Enable Transportation of Small Objects. | The UAV must be equipped with a claw capable of carrying and dropping objects. | UAV must be able to detect, move toward, and deliver objects to a designated target area design. |
| The Vehicle Shall Deploy from the ASV’s flight deck. | The UAV must have landing gear. |  |
| The Vehicle Shall Deploy from the ASV’s flight deck. | The UAV must be able to deploy while the ASV is moving or stationary. |  |
| The Vehicle Shall Have a Loss of Contact Instruction Set. | Must initiate slow decent within 5 seconds of losing contact with remote control or remote kill. |  |
| The Vehicle Shall Have a Loss of Contact Instruction Set. | Positively buoyant for at least 120 seconds. |  |
| The Vehicle Shall Allow for Immediate Shutdown. | The vehicle shall have a local kill switch onboard the UAV. | The override must be wirelessly capable for 1000 ft. |
| The Vehicle Shall Allow for Immediate Shutdown. | The vehicle shall have a remote kill switch at the control station. | The remote stop function must be separate from the main controller. |
| The Vehicle Shall Allow for Immediate Shutdown. | Both must turn off all motors on UAV within 1 second of being pressed. |  |

The most difficult goalof the project is to successfully take off from and landon the back of the ASV. Both tasks can be accomplished in any manner but must be fully autonomous. Once the UAV is in the air it must fly a search pattern until it finds the specific drop zone. Once detected, the UAV will hover just above the target before delivering the payload. The payload is described as a ping-pong ball sized object that must be delivered but not picked up; it will be pre-loaded on the drone before flight. This has significant design impacts as the system only needs to be designed with a release mechanism rather than a claw capable of picking up the object as well.

During the competition, the ASV will be required to autonomously navigate a course. After completing the drone-specific tasking, the UAV will position itself above the ASV to provide high altitude, aerial view of the course. These images will be sent both the ASV and UAV teams. Through coordination, the image processing may be done on the UAV, ASV, or a combination of the two. The goal is for this image to aid in the ASV’s trajectory planning through the course.

In addition to these objectives, the UAV has other niche requirements that must be met to compete. These include all pilots being certified by the FAA, the drone being able to float for 120 seconds, the drone having a remote kill-switch, and various other safety protocols in place for a loss of connectivity.

Diagram

Description automatically generated

**Fig. 1. System Block Diagram**

# System Design

The complete UAV is comprised of several independent subsystems linked through the flight controller, as depicted in Fig. 1. To best understand the design, one may view the UAV as a series of physical components supported by connecting software, frameworks, and protocols.

The hardware components of the drone include the Holybro S500 generic quadcopter frame, motors, electronic speed controllers (ESCs), a physical flight controller, and mechanisms for landing and transporting small objects. These elements are largely independent of one another with the notable exception of the flight controller. The flight controller is a physical

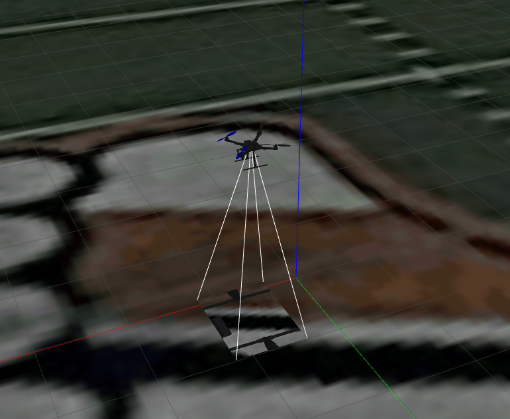
device that serves as a common link between each component and connects sensors with elements that perform real-world actions [8].

This UAV, specifically, relies on the Pixhawk 2.4.8 flight controller – a device capable of interfacing with the motors, sensors, and inputs – to enable flight. Simply put, the flight controller is the brain of the UAV. PX4 autopilot, an open-source autopilot designed to enable autonomous, manual, and other flight modes, is used in conjunction with the Pixhawk. It interfaces directly with each subsystem to include telemetry, sensors, and communications.

Each of these subsystems has distinct inputs and outputs for each stage of flight. The Pixhawk and PX4 autopilot are the cornerstones of the UAV. Each element, from the onboard communications system to the entirely separate Real-Time Kinematics (RTK) positioning system, relays its information to PX4. This allows PX4 to control flight based on all sensor input. The flight models utilized include manual control, offboard flight—commanding the drone by relative velocities, or computer vision driven control—a flight mode designed by this research group.

As part of the Roboboat competition guidelines, the UAV must be positively buoyant for 120 seconds in the event of a water landing (see Table I for a definitive requirement matrix). The group’s goal, however, was to achieve indefinite buoyancy to minimize damage to electrical components in case of an emergency landing. To meet this requirement, the UAV will be equipped with a hydrostatic float supplied by DRONE-RETRIEVER. Further analysis of this solution is provided in the results section.

Testing of all flight software is done in the Gazebo simulation environment in followed by subsequent real-world testing. Gazebo is a physics engine that is used primarily to model robots. When paired with PX4, Gazebo creates a software-in-the-loop platform in which a virtual drone interacts with the environmental forces of a virtual environment, including drag, lift and gravity. Simulation was used to test every flight prior to the real world. This allowed rapid troubleshooting to occur in a low-risk environment. The UAV presented in this paper can be seen modeled in Gazebo in Fig. 2.



**Fig. 2. Model of UAV in Gazebo Environment**

The group used Real-Time Kinematics Global Positioning System (RTK-GPS) as the UAV’s primary source of positioning information. Much like standard GPS, RTK-GPS takes signals from Global Navigation Satellite System (GNSS) to provide a fix. RTK-GPS, specifically, use a corrections stream of Radio Technical Commission for Maritime (RTCM) corrections to better estimate location in real-time. This is crucial to allow the drone to navigate along predetermined routes. Additionally, a GPS-RTK fix is used to maintain the UAV’s current position; a more accurate position allows for more accurate navigation.

To accomplish this, the drone’s onboard receiver—often referred to as the rover receiver—must communicate with a base station, a stationary antenna and receiver located nearby. The base station receives GNSS data from several satellites and sends RTCM corrections to the rover receiver. As the drone starts receiving these RTCM corrections, the rover receiver will process that information along with GNSS data from satellites to achieve better positional accuracy. The SparkFun RTK Surveyor was chosen as the base station, and the SparkFun ZED-F9R module was utilized as the rover receiver.

Computer vision, simply put, is a method of using image processing techniques to retrieve useful information from a video stream; it allows a system to “see” the environment through a camera. In alignment with the goal of autonomously taking off and landing, an ideal computer vision test scenario was formulated to evaluate the design. From start to finish, the quadcopter would take off and execute a search pattern to discover a distinctive marker. If one is found, the drone would land on top of it. If not, the drone would return to start after exhausting the search pattern. A wrapper of the MAVLINK protocol was created to facilitate this task and manage the drone’s flight modes safely in the event of an error condition [4].

As mentioned previously, real-world testing was conducted to verify the results obtained from software simulation. Preliminary testing has proven that the methodology described in simulation translates well to real-world conditions, and successful implementation of those algorithms was completed [7].

# Results

*A. Physical Platform*

The quadcopter was built according to preliminary design requirements, outlined in Table 1, to establish a testing platform for subsequent research. The initial analysis of the physical characteristics demonstrated that the drone’s weight of 3.4 pounds meets the Roboboat competition requirements of less than 10 pounds. Foam outriggers designed to provide buoyancy were tested in a pool. In Fig. 3 below, the drone is seen floating in the pool while supporting 3000 g (6.61 lb.) of weight. This indicates the floats can support the weight of all hardware; however, test flights demonstrated that the current flight controller would have to be modified to achieve stable flight with the floats attached.



**Fig. 3. Test of Drone Floatation**

Because of this, an alternative floatation method has been chosen for flights over water. The DR9 Drone Retriever is a commercially available hydrostatic floatation device designed for recovery in water crashes. According to the product’s documentation, Weighing in at only 136 grams, the DR9 provides 10 pounds of buoyant force once it is activated by being fully submerged [6]. This alternative will not allow for the drone to remain upright in a water landing but will still allow for recovery. Silicon coating can be used to waterproof the electrical components onboard the drone to prevent damage. The DR9 is intended to be a short-term solution to allow the conduction of flights over water while a more permanent solution is established. Testing of the DR9 has yet to occur but would be completed prior to a flight over water. A picture of the DR9 drone retriever is shown below in Fig. 4.

A picture containing light

Description automatically generated

**Fig. 4. DR9 Hydrostatic Drone Recovery System**

The quadcopter’s proportional-integral-derivative controllers must be stable to maintain both manually controlled and autonomous flight. Following the initial configuration, the controllers were iteratively tuned to balance stability with responsiveness. In response to a significant disturbance input—a strong pull on the drone’s tether—the drone’s flight controller maintained stable flight and returned to its previous holding position [5]. Plots demonstrating the effectiveness of the flight controller with yaw, pitch and roll can be seen in Appendix I.

**A picture containing text, grass, sky, outdoor

Description automatically generated**

**Fig. 5. The drone’s maintenance of level flight.**

*B. Safety Criteria*

The safety of people and property is a priority of this research group, as well as the Roboboat competition guidelines. Thus, the remote kill-switch and loss of radio control (RC) signal procedures were analyzed during the first test flight. The initial test of the kill switch proved successful, immediately stopping all rotors regardless of the drone’s current position. The first test of the loss of signal failsafe appeared successful but, after further review, the flight controller failed to recognize the loss of RC signal. Instead, the most recent throttle input was maintained indefinitely; by coincidence, this was sufficiently low to result in a gradual descent that resembled a landing. In response, the RC receiver was configured to provide radio signal strength indicator readings to detect a loss of signal. Subsequent testing demonstrated that a loss of RC signal for two continuous seconds prompts the drone to land, as seen in Fig. 6.

Graphical user interface

Description automatically generated

**Fig. 6. The drone’s response to a loss of RC signal.**

If the drone was receiving GPS data, the drone returned to its original point of takeoff. If not, the drone slowly landed at its current position.

*C. Computer Vision*

The computer vision algorithms designed to detect the Roboboat competition landing platform were tested extensively. Initially, these yielded a 72% detection rate of the platform under indoor lighting, as seen in Fig. 7.

Graphical user interface, application

Description automatically generated

**Fig. 7. Indoor recognition testing of the landing platform.**

However, the algorithms failed to properly recognize the landing platform during outdoor test flights under non-ideal lighting conditions, Fig. 8.

A picture containing text

Description automatically generated

**Fig. 8. Outdoor recognition testing of the landing platform.**

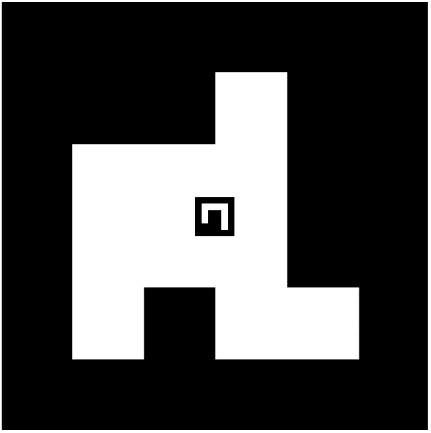
In response to these tests, the proposed landing platform of the ASV was changed to an ArUco marker, a black and white square marker that resembles a matrix. These markers are designed for fast and reliable detection. In Fig. 8 below, one can see the OpenCV script detecting the ArUco marker and outlining it in green. The script alongside the Raspberry Pi Cam can detect the marker out to approximately 35 feet before detection becomes inconsistent.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig. 9. The detection of an ArUco marker from in-flight video.**

The AruCo marker system was revised again; the ArUco marker itself was reduced to 4X4 blocks rather than 7X7. The was done to make it detectable at further distances. Testing revealed a second issue, where during the landing sequence the camera would lose detection when the drone got too close. This is due to the field of view (FOV) of the camera not being wide enough to see the entire marker when the drone is directly above it. To combat this, a smaller AruCo marker was inscribed in the middle of a larger one. This way, at least one marker is detectable when the drone is on the platform, Fig 10. The larger outside marker is detectable at up to 150 ft, and the smaller one is detectable up to 15 ft, which allows for a smooth transition going into the landing. AruCo markers work by finding the majority color (black or white) in each of the 16 squares in the 4X4 matrix. This left room to insert the smaller AruCo marker in the corner of the four middle most squares of the larger marker. This did not impact the detection of the larger marker and allowed the drone to have an AruCo marker insight throughout the entirety of the landing process.



**Fig. 10. Dual AruCo Marker Solution**

*D. Autonomous Flight*

Autonomous flight is a key requirement of the proposed autonomous UAV. The Gazebo simulation environment was utilized to test autonomous takeoff, landing, waypoint flight, as well as various search patterns. Following successful simulated results, the same commands were sent to the actual quadcopter. These test flights demonstrated successful autonomous flight, as well as the transfer of commands from the simulated environment to real-world results. A waypoint flight log is shown in Appendix II.

*E.* *GNSS and RTK Measurements*

Several tests were conducted to provide measurements of observed GNSS and GPS-RTK accuracy. Standard GNSS signals—a combination of GPS, Galileo, GLONASS, and BeiDou—yielded an average deviation of +/- 60 cm over the course of 15 minutes, as seen in Fig. 11.

A picture containing light, night sky

Description automatically generated

**Fig. 11. Deviation Map of a 15-minute GNSS fix.**

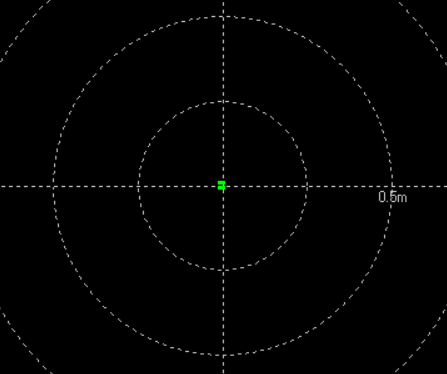
This level of accuracy from standard GNSS navigation was insufficient for the precise adjustments needed to perform a computer vision enabled landing onboard the ASV. An inaccuracy of 60 cm could place the drone beside the ASV rather than above. Thus, an RTK base-station position was measured to an accuracy of 17 mm, as given in Appendix III. When configured, the base station is capable of calculated and sending RTCM corrections to the drone. A second test was performed using the same antenna and receiver, as well as under the same satellite constellation, weather conditions, and location. The observed accuracy of the GNSS-RTK enabled fix was +/- 6 cm, Fig. 12.

Radar chart

Description automatically generated

**Fig. 12. Deviation Map of a 15-minute GNSS-RTK fix using QGroundControl.**

Streaming RTK corrections directly to the GNSS receiver has presented several technical challenges. Troubleshooting is a primary concern prior to a computer vision enabled landing, however, the observed accuracy of the GNSS-RTK equipment is well below our objective requirement of 10 cm.

****

**Fig. 13. Deviation Map of a 10-minute GNSS-RTK fix for the XBee 3 modules.**

Significant challenges were encountered when attempting to send RTCM corrections directly between the rover receiver and base station. In Fig. 11, QGroundControl—a program used to remotely monitor the UAV—was used to send corrections from the base station to the rover. However, the accuracy of RTK-GPS was hampered. This disparity was attributed to using QGroundControl as an intermediate program. Thus, a direct wireless connection was established for the test shown in Fig. 13. These tests yielded an accuracy 17.3 millimeters. It is important to note that this test was conducted with a stationary rover receiver and was not conducted while the UAV was in flight.

*F. Computer Vision Enabled Landing*

The Gazebo simulation environment facilitated preliminary testing of a computer vision-controlled landing. The preliminary controller, which calculated the distance to the center of the marker in pixels, centered the drone over the landing platform and slowly lowered before touching the platform. These distance estimates did not change with altitude, though, and revealed characteristics of an unstable controller. After several iterations, the controller was updated to utilize the drone’s current altitude and attitude to estimate the distance between the point directly beneath the drone and the center of the marker. The resulting computer vision-controlled landing followed a stable path, as shown in Fig. 14.

A screenshot of a video game

Description automatically generated

**Fig. 14. A simulated computer-vision enabled landing.**

Following this initial success, the Gazebo-SITL simulation environment was used to test the computer vision enabled landing sequence contained within the MAVLINK wrapper. When the ArUco marker was within the camera’s field of view, the simulated quadcopter recognized the marker, centered above it, and began descending. Below a specified altitude, and when the smaller inscribed marker was identified, the descent rate slowed until a landing was detected.

Separately, the wrapper can execute take off and a corresponding search pattern based on a starting coordinate, desired path spacing, and desired altitude. During this time, the drone monitors for ArUco detections; however, the wrapper is not yet capable of transitioning from search to landing states. All testing has been conducted within SITL (Software in the Loop). If the ArUco marker detection falls below a specified threshold, the quadcopter holds position until it reestablishes positive identification or determines the marker to be lost, at which point it returns to its takeoff position.

Following the successful SITL test of a computer vision enabled landing, an outline of tests was designed to facilitate the transition to real-world flight. Foremost, the script’s performance when running on the Raspberry Pi was tested. It yielded a range of iterations, or camera frames viewed, with a mean of approximately 20 per second, as seen in Appendix IV. While 20 frames per second is likely sufficient for this project, the controller is set to provide 30 fps and this variability hinders optimally tuning the landing controller. Nevertheless, the Raspberry Pi’s performance was deemed sufficient and real-world testing continued.

Each computer vision enabled landing was conducted with the drone positioned above the landing platform and the ArUco marker in view. A series of test flights demonstrated that the controller was capable of centering the drone overhead a stationary ArUco marker and descending to perform a landing. While the controller appears slow, and likely over dampened, these landings were repeatable [7].

*G.*  *Landing Platform*

The final landing platform design consists of a 30x30 inch piece of plywood. Attached to the plywood is a hook-side Velcro model of the dual ArUco marker mentioned previously. Attached to the landing gear of the drone is loop-side Velcro. Upon landing on the platform, the Velcro is intended to provide sufficient friction to prevent the drone from falling off due to disturbances while still allowing for takeoff. Real world testing has supported this designs viability. Additional tests were performed to simulate a botched landing where only one leg of the drone was affixed to the platform. In this case, the drone was still able to take off in a stable manner. The drone and Velcro dual AruCo marker platform are shown below in Fig. 15.

**A picture containing text

Description automatically generated**

**Fig. 15. Dual AruCo Velcro Landing Platform**

*H. Payload Delivery System*

From Table 1, the drone shall also be capable of transporting small objects. This task has been accomplished through the creation of a scoop mechanism for payload delivery. The mechanism is powered by an SG90 continuous micro-servo motor. This servo motor provides transmission to a gear system that guides one of the two scoops along a rail, creating an opening for objects to be dropped at the bottom. The mechanism can open to a maximum of 1.8 inches, allowing for the delivery of table-tennis ball sized objects. This mechanism has been tested in the real world and has performed as designed. The payload delivery system can be seen in Figs. 16 and 17.

Diagram, engineering drawing

Description automatically generated

**Fig. 16. SolidWorks Rendering of Payload Delivery System**

A picture containing indoor

Description automatically generated

**Fig. 17. Payload Delivery System Attached to Drone.**

# Conclusions

The UAV capstone group has made great strides in meeting the given requirements of this project as outlined previously. The group created a quadrotor UAV capable of remote and autonomous flight. This platform satisfied the physical requirements of the RoboBoat competition and was tuned to provide stable flight, even in response to significant external forces [5].

A preliminary buoyancy system was developed and constructed for the UAV. While it provided sufficient flotation and would allow the drone to remain right-side up, real-world testing determined it to detrimentally impact the UAV’s ability to maintain stable flight. In response, a new buoyancy solution was proposed that would not significantly impact the UAV’s flight characteristics.

A payload delivery mechanism was designed and constructed. This mechanism proved successful by remotely dropping a golf ball, midflight. A landing platform was developed, with velco, that proved capable of securing the drone while still allowing takeoff capabilities.

To integrate RTK-GPS as a primary positioning source, a base station position was surveyed to an accuracy of 17 millimeters. Despite significant obstacles in its implementation, static tests demonstrated the successful use of RTCM corrections, sent over wireless communications, when developing an RTK fix. While not implemented in flight, the stationary receiver demonstrated an accuracy of 17.3 millimeters.

The group developed and tested computer vision algorithms that could identify a predetermined ArUco marker. The group integrated this computer vision component with the drone’s controller, resulting in successful computer vision enabled landings in simulated and real-world tests. has completed several test flights to assess the quality of both.

In summary, the group used the objectives of the Roboboat competition as guiding requirements when developing a low-cost quadrotor UAV that is capable of autonomous flight. The UAV implemented various technologies, including onboard computer vision, payload delivery, and RTK-GPS positioning that leave room for future development and are applicable to many Coast Guard missions. The capstone experience has been a significant learning experience for all group members and demonstrated the ability for a small group of undergraduate students to research autonomous UAV technologies with relevance to USCG operations.

REFERENCES

1. National Research Council. 2014. Autonomy Research for Civil Aviation: Toward a New Era of Flight. Washington, DC: The National Academies Press. <https://doi.org/10.17226/18815>
2. J. Harper, “New Coast Guard Team to Flesh Out Unmanned Systems Requirements,” *National Defense Magazine | NDIA’s Business & Technology Magazine*, Mar. 11, 2021. <https://www.nationaldefensemagazine.org/articles/2021/3/11/just-in-new-coast-guard-team-to-flesh-out-unmanned-systems-requirements> (accessed Dec. 05, 2021)
3. D. Bucciarelli, Private Communication, Jan. 2022
4. J Schellman. 2022. *Stable Controller?* New London, CT. United States Coast Guard Academy <https://www.youtube.com/watch?v=RpAQXoV6FPE>
5. J Schellman. 2022. *Drone Test*. New London, CT: United States Coast Guard Academy <https://www.youtube.com/watch?v=hw5l6TqirGs>
6. “DRONE-RETRIEVER: How It Works.” *Drone-Retriever.Com*, <https://img1.wsimg.com/blobby/go/ddbd1905-b0d8-40ee-a213-77ec946f7731/downloads/HOW%20IT%20WORKS-r8.pdf?ver=1641251807366>. Accessed 5 Apr. 2022.
7. J Schellman. 2022. *Autonomous Landing 2.* New London, CT. United States Coast Guard Academy <https://www.youtube.com/watch?v=RymxTethCiw>
8. Holybro. 2019. *S500 V2 Kit*. RM 1902 EASEY COMM BLDG253-261 HENNESSY RD WANCHAI. Hongkong, China. <http://www.holybro.com/product/pixhawk4-s500-v2-kit/>

##### Appendecies

##### Appendix I

##### Tuning

Chart, line chart, histogram

Description automatically generated

Time

Time

Time

Fig. I.1: Roll, Pitch, and Yaw Angle.

##### Appendix II

##### Observed In-Flight GPS Accuracy

Chart, line chart

Description automatically generated

0m 5m 10m 15m 20m

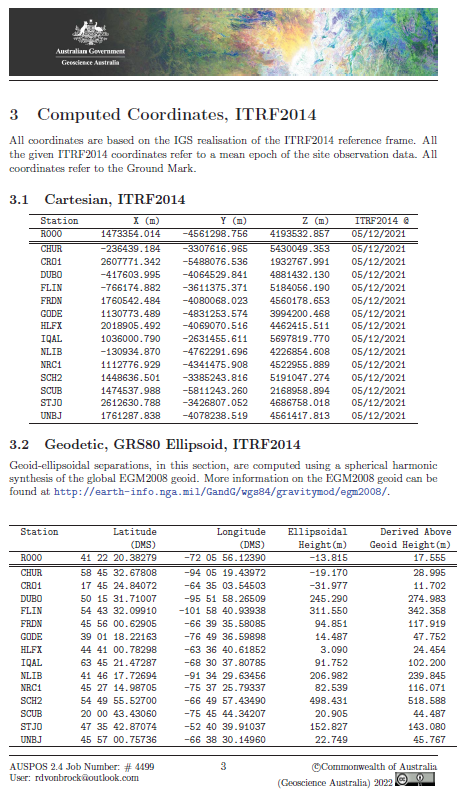
Fig. II.1: GPS Accuracy

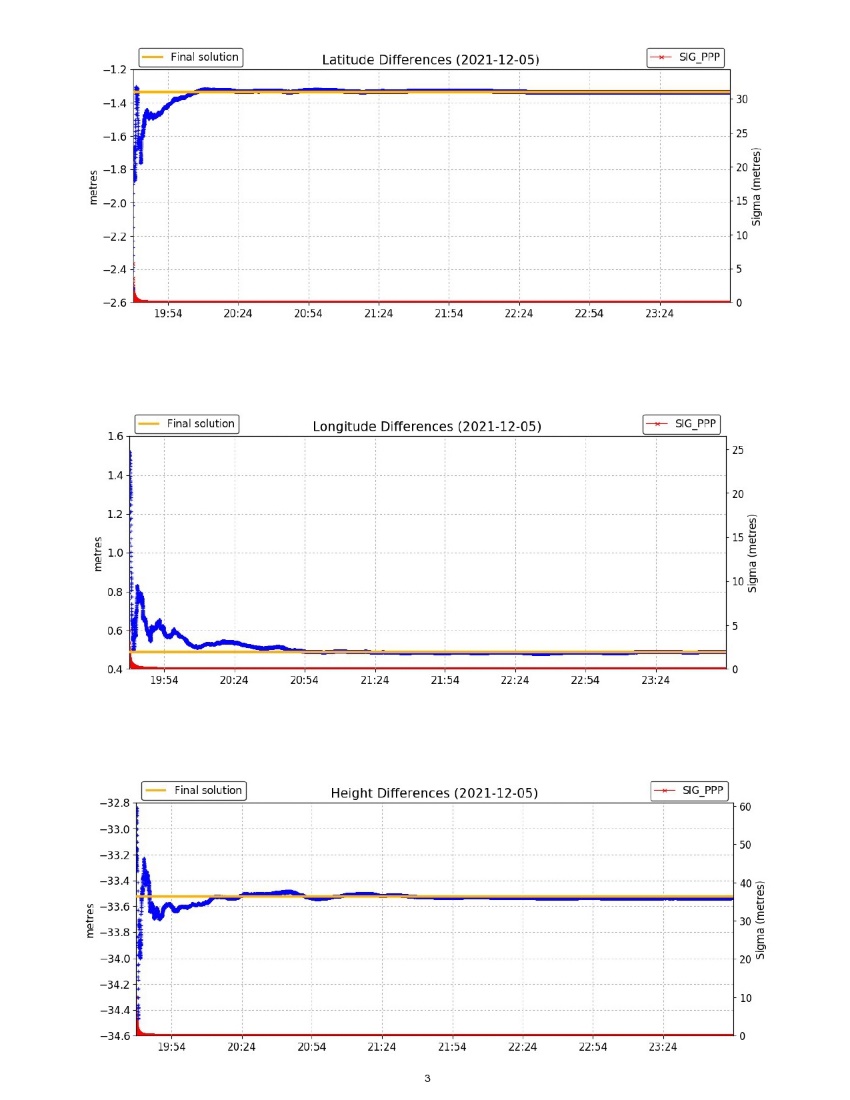
##### Appendix III

##### RTK Auspos Post Processing Report

Text

Description automatically generated





##### Appendix IV

##### 60 Second Log of Camera FPS

A picture containing text, outdoor

Description automatically generated