Design and Development of an Autonomous Payload Return Vehicle

Sean Widmier[1], Thomas Salverson[2], Colin Oberthur[3], An Nguyen[4], Sean Mitchell[5], Kyle Daigle[6], Trystan May[7]
The University of Alabama in Huntsville, AL, 35899

The Ram-Air Parafoil Targeted Object Return (RAPTOR) system is an autonomous payload designed for the simplification of high-altitude balloon (HAB) payload recovery. Currently HAB payloads are returned by unguided parachutes and frequently land in areas that make retrieval difficult or dangerous. In order to avoid unwanted landing areas, RAPTOR directs payloads to one of multiple predetermined landing sites using a mechanically articulated parafoil. The system's navigational algorithms utilize a neural network to narrow down potential landing locations based on size, distance, and wind patterns. A data-driven PID controller is then used to guide the payload to the chosen landing location. The RAPTOR system is designed to be a reusable addition to any payload line that both serves as the cutdown and guidance. It consists of easily replaceable additively manufactured components and a custom PCB that reduces weight and construction time. The development of the project is split up into several iterative phases, each of which entails the design, manufacturing, and testing of individual systems with increasing capabilities. The first phase, Kestrel, was active from March to August 2018. The second phase, Falcon, was active from August to December 2018. The third phase, Phoenix, is scheduled to fly by March 2019.

Nomenclature

SHC	=	Space Hardware Club
PC	=	Polycarbonate
ABS	=	Acrylonitrile Butadiene Styrene
CD	=	Coefficient of Drag
CL	=	Coefficient of Lift
IMU	=	Inertial Measurement Unit
ANN	=	Artificial Neural Network
HAB	=	High Altitude Balloon

[1] Undergraduate, Electrical and Computer Engineering Department, smw0025@uah.edu
[2] Undergraduate, Mechanical and Aerospace Engineering Department, tms0034@uah.edu, AIAA Student Member
[3] Undergraduate, Mechanical and Aerospace Engineering Department, cjo0008@uah.edu, AIAA Student Member
[4] Undergraduate, Mechanical and Aerospace Engineering Department, ann0012@uah.edu
[5] Undergraduate, Electrical and Computer Engineering Department, sm0077@uah.edu
[6] Undergraduate, Computer Science Department, kad0011@uah.edu
[7] Undergraduate, Computer Science Department, tkm0010@uah.edu

I. Introduction

THE University of Alabama in Huntsville Space Hardware Club (SHC) is a large student-run organization that facilitates the development of undergraduate research projects. These projects provide students with valuable hands-on design and fabrication experience so that they are better prepared for a future in the engineering field. Many of these research projects utilize HABs in order to carry scientific payloads into the upper atmosphere and gather data. One of the main issues with these types of ballooning projects is the frequency in which payloads land in unfavorable locations such as in trees or water. With RAPTOR, our team intends to alleviate these concerns by improving the efficiency and reliability of HAB payload recovery. The RAPTOR system was inspired by the Joint Precision Arial Delivery System (JPADS) developed by the US Army and Air Force[1], as well as research done by Oleg Yakimenko at the Naval Postgraduate School[2].

A. Project Overview

The primary objective of RAPTOR is to return SHC HAB payload trains safely. Prior to RAPTOR, all SHC balloon payload trains would descend under an unguided parachute, which frequently resulted in the loss of payload trains to tall trees or other hazardous landing sites. On several occasions, in which high-value payloads were lost to trees, professional tree-climbers were hired to collect them. However, with the implementation of RAPTOR, these payload trains would be guided via a steerable parafoil system to safe landing zones which would drastically increase the ease and reliability of payload train recovery.

B. Concept of Operations

RAPTOR's concept of operations is very similar to that of most high-altitude balloon payloads. The entire flight is separated into three main stages: ascent, apogee/burst/cutdown, and descent. During the balloon's ascent, the RAPTOR system will collect weather data which will be used during descent to increase the accuracy of flight predictions and path corrections. Once the payload has reached apogee, and the payload has cutdown from the balloon, the system will descend to 3,100 meters, where the parafoil will be deployed, and then begin its guided descent towards a safe landing location. As it descends under 3,100 meters, the system's prediction and guidance algorithms will continuously determine the best landing location and the necessary flight path to reach it safely.

II. Payload Design

A. Parafoil Design and Flight Considerations

Since the entire goal of the project is to land payload trains safely, the success of RAPTOR depends largely upon the parafoil design and its deployment. The parafoil must be able to slow the payload train to a safe landing speed while retaining the necessary level of controllability required to reach a suitable recovery site. The two main options for accomplishing this are either commercially available parafoils or custom-made, in-house parafoils. For the purposes of this project, commercially available parafoils are the best option, since parafoils are quite elaborate and difficult to make by hand. Small hobby kites/parafoils are both cost-effective and meet the needs of RAPTOR. Commercially made parafoils are also manufactured with all of the shroud lines required for controlling the parafoil.

Selecting a parafoil is based primarily on the target descent rate of 7 meters per second, chosen to ensure the safe recovery and reusability of the payload. The calculation for descent speed can be seen in Eq. (1). the drag equation, where D is the drag force, C_D is the coefficient of drag, ρ is density, v is velocity and A is the area.

$$D=\frac{C_D}{0.5\rho * v^2 * A} \tag{1}$$

This equation assumes that the drag force of the parachute equals the weight of the payload, such that the system will be in equilibrium. From this equation, the parachute size necessary to support the given weight of the payload train can be found by rearranging the equation for area.

After deciding on a suitable parafoil size, the next important design consideration is ensuring that it deploys correctly. This issue poses a large risk for any parafoil-based system as any degree of failure can prevent the controlled descent of the payload. These types of failures are often the result of the parafoil being deployed at too high of an altitude, or being packed incorrectly. In either case, the parafoil will become entangled or fail to fully inflate. Therefore, the RAPTOR system aims to mitigate this risk by developing a semi-rigid parafoil, shown in Figure 1 and 2, using a wooden dowel cross-member as was done on a similar project, Snowflake[3]. This wooden cross-member is placed such that the parafoil is held partially open at all times and this helps prevents the shroud lines from entangling. The cross-member design also forces the parafoil to fully inflate even at lower altitudes.

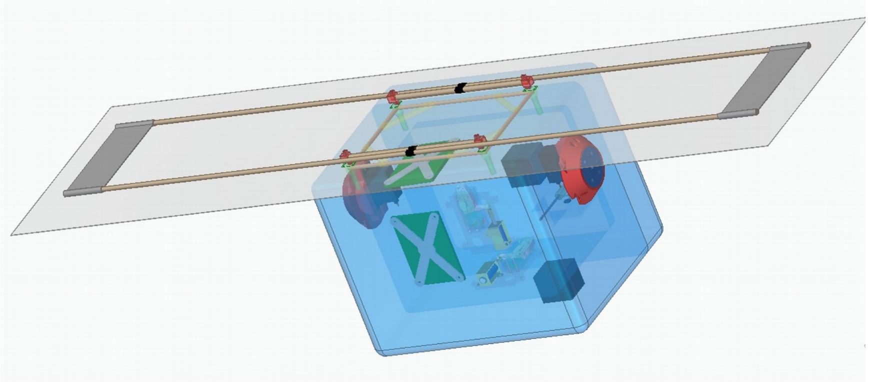

Figure 1. CAD Rendering of Parafoil Rigging

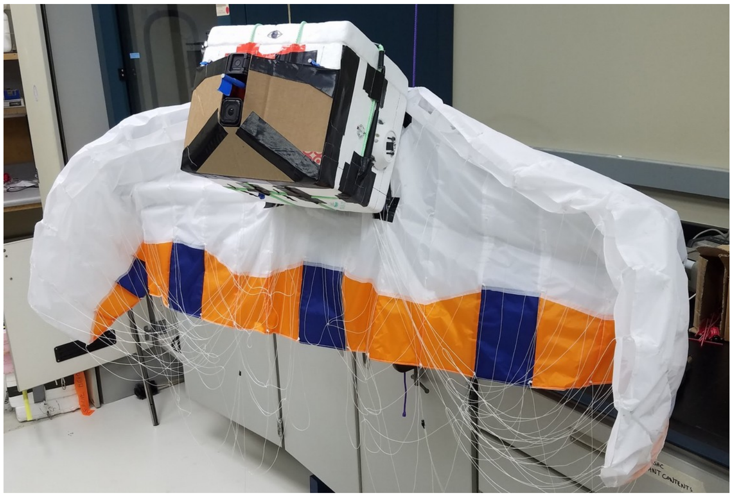

Figure 2. Falcon Parafoil

The parafoil itself will be attached to the side of the payload box and will release at an altitude of 3,100 meters. The parafoil is released at this altitude so that it has the necessary air density to inflate. By inflating the parafoil at 3,100 meters, RAPTOR also avoids most of the higher trafficked airspace. This mitigates the risk and need for special flight waivers from the FAA. While still secured to the payload box, the parafoil has the ability to act as a rear decelerator. This functionality serves as a safety feature in the event the parafoil fails to release from the payload box.

B. Brake Line Controller

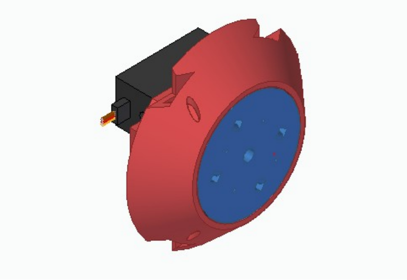

Figure 3. CAD Rendering of Brake Line Controller

In order to adjust the brake lines of the parafoil, RAPTOR is equipped with two brake line controllers. Figure 3 shows a CAD rendering of a brake line controller. Each controller consists of a commercially available Power HD AR-3606HB continuous rotation servo motor attached to a 3D printed spool (blue part in Figure 3 and housing (red part in Figure 3). A continuous rotation servo was chosen for this application since it offers more than 360 degrees of rotation, unlike conventional servos. The spool is simply used to wind-up or let-out the brake lines. The use of two motors allows for asymmetrical deflection of the parafoil. To ensure optimal motor performance at high altitudes, the motors will be inserted into the side of the main insulated payload box. As for the housing and spool, which may be subject to the severe cold of the upper atmosphere, they will be 3D printed out of Acrylonitrile Butadiene Styrene (ABS) filament. ABS provides a good balance to the strength and ductility for the conditions experienced during the flight.

C. Release Mechanism

RAPTOR will contain three different release mechanisms: one for the parafoil and two for the main balloon cutdown. As previously mentioned, the parafoil is not released at apogee; rather it remains secured to the payload until approximately 3,100 meters. Therefore, a release mechanism is required specifically for the parafoil and must be independent of the balloon release device. The system can utilize this device in the event that the payload train needs to be released prior to balloon burst.

Both the balloon and parafoil release mechanisms utilize electronic solenoids that when energized secure the parafoil to the box and the payload train to the balloon. Electronic solenoids were chosen over a standard hot-wire due to their increased reliability and brown-out resilience. Since the solenoids require power to remain open, if the electrical system were to lose power, the solenoids would close and automatically release both the balloon and the parafoil. The brown-out resilience feature of the solenoids is especially critical for the parafoil release, as the parafoil must be able to release and open even if the electronics lose power.

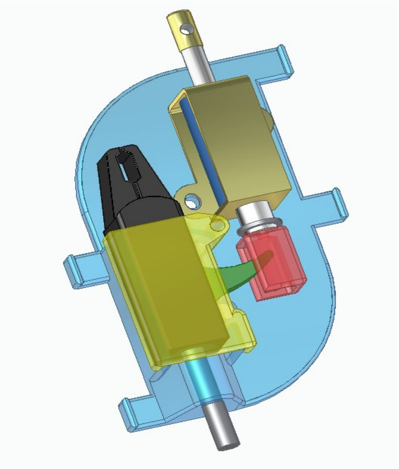

Figure 4. Release Mechanism

Due to a previous release mechanism failure during prototype testing, a secondary release mechanism was designed to ensure recovery of the payload during prototyping phases. The secondary release will consist of a hotwire enclosed in a small Styrofoam box outside the main payload box, as seen in Figure 5. So, the hotwire release does not interfere with the primary cutdown, it will be attached to the balloon line rather than the main payload. If the primary cutdown is successful, the secondary cutdown will remain attached to the balloon line while the rest of the payload is released. In the event the primary cutdown does fail, the hot wire is able to burn through the balloon line and release the payload.

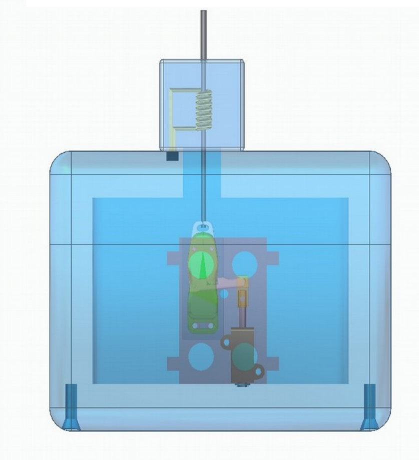

Figure 5. Secondary Release

D. Weather Collection

Although the target landing zones are relatively large, the accuracy of in-flight path predictions are still a concern for this project. Therefore, the system is designed such that it collects wind speed and direction during the balloon’s ascent that can be used by the flight algorithms during descent. Most payload trains are not stable enough to gather wind data accurately with traditional anemometers due to the movement and constant rotation of the payload. RAPTOR will use the onboard GPS system to instead estimate the wind profile. The general wind speed and direction can be calculated by assuming the lateral movement of the balloon on ascent is due to wind.

E. Payload Housing

The payload housing mounts the parafoil and maintains the systems electronics. The housing is made from a Styrofoam cooler modified to effectively and securely mount the electronics inside. The cooler provides an adequate amount of structural support to act as the connection point between the balloon and the rest of the payload train. Styrofoam was chosen over other materials since it is lightweight, cost-effective, disposable, and provides adequate insulation for the electronics.

F. Camera Mounts

Cameras are necessary to view parafoil inflation and in-flight performance characteristics. The cameras must be insulated against the freezing temperatures experienced during flight. To accomplish this, the cameras are mounted into the sidewall of the Styrofoam payload housing such that the passive heat generated by the payload electronics will warm the cameras and prevent them from shutting down. For previous versions, GoPros were selected due to weather resilience, durability, simplicity and a high picture quality.

III. Flight Operations

A. Predetermined Landing Locations

The RAPTOR guidance system relies on a set of predetermined landing locations picked prior to each flight. The main criteria for choosing a landing location is proximity to the location at which the payload will cutdown from the balloon, in order to remain within flight range. The most optimal location is a large, easily accessible field free of obstructions.

B. Cambridge University Spaceflight Landing Predictor Simulations

Cambridge University Spaceflight (CUSF) landing predictor[4], a HAB flight simulation tool, is used to predict the approximate landing location of the RAPTOR payload. The CUSF landing predictor simulation provides the predicted location of the payload during ascent, cutdown and landing. This prediction accounts for wind at varying altitudes; therefore, it becomes increasingly accurate as weather forecasts are published. With this in mind, CUSF landing predictor simulations are run several hours before the flight to acquire the most accurate prediction possible. Using these simulations allows for the selection of landing locations that will be in range during flight.

C. FAA Compliance

Prior to flying RAPTOR, it was important to verify that it complied with current FAA regulations. While RAPTOR contains an autonomous guidance system, it does not have the ability to fly against the wind. Therefore, it is not classified as an unmanned vehicle by the FAA, rather a ballooning payload capable of tacking with the wind to change its downwind landing location. Since RAPTOR is under 1.8 kilograms and is classified as a balloon payload[5], no special waivers were required to fly. Prior to flights, the local FAA is simply provided RAPTOR's flight notice and is called fifteen minutes before and after the flight. It was also suggested, not required, by an FAA representative that RAPTOR begin parafoil guidance under 3,100 meters just to ensure that the payload operated primarily outside of high-trafficked airspace.

D. Completed Flights

a. Kestrel

Kestrel was the first iteration of the RAPTOR system, developed to be a hardware testbed. The payload was built in the winter of 2018-19 and first flew in spring 2019. It contained the sensor array, solenoids and servos. The goal was to test the functionality of the electronics at high altitude, as any significant lack of insulation could cause the components to fail at freezing temperatures.

Three test flights confirmed that the components function nominally at altitudes up to 25,000 meters. Each test flight included incremental improvements to the mechanical design of the payload, and revealed major issues with flight software brown-out resiliency and wiring complexities. Overall, the Kestrel payload was successful in gathering the necessary data to move forward with the prototyping phases.

Figure 6. Kestrel Pre-Flight

b. Falcon

Falcon implemented the initial designs for active descent control using a parafoil. Falcon was designed to cutdown at a low altitude, around 300 meters, and gather data for a basic descent profile. Three flights were conducted and each had its own successes and issues. During the first flight, debris from other payloads on the flight line got entangled in the parafoil rigging lines. Although the payload could not attain controlled descent, the electrical system, balloon cutdown, and parafoil deployment all operated successfully. The second test flight had an issue with the mechanical cutdown mechanism which prevented the payload from releasing from the balloon. When the payload failed to release, a programming oversight caused the data logging function to stop. Even after modification of the cutdown mechanism, the third and final flight of Falcon also resulted in a failure of the mechanical balloon release. Falcon then landed in a tree from which recovery has been unsuccessful. A new cumulative design, Phoenix, is currently under development.

Figure 7. Falcon Pre-Flight

IV. Software Development

A. Flight Software Structure

RAPTOR software utilizes the Arduino framework to ease development and allow the use of C++ object-oriented programming. The object-oriented paradigm allows compartmentalization of the various aspects of RAPTOR flight software into classes: collections of variables and functions that can reduce complexities as abstraction levels are increased. Figure 8 shows a simplified version of the UML Diagram created for the flight software.

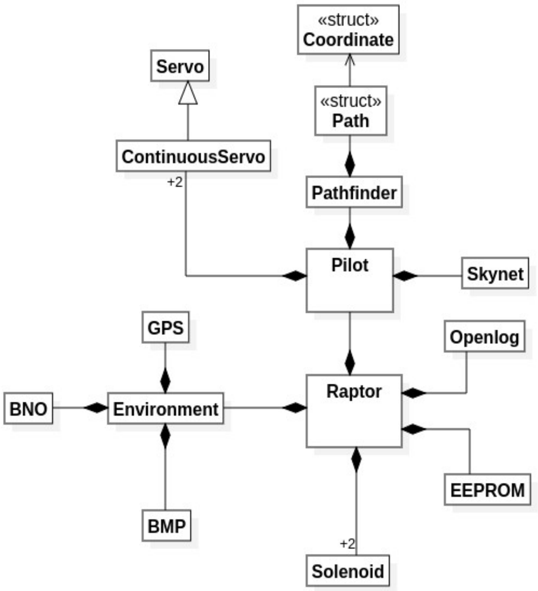

Figure 8. Simplified UML Diagram

The design contains a main class, Raptor, to be initialized at boot. This class contains functions used for each of the four flight states, as well as instances of the Environment, Pilot, Solenoid, Openlog, and EEPROM classes. Environment is a class used to contain the various sensors of RAPTOR, and will be used to update and read from all sensors at once. Pilot is the class responsible for determining and correcting the flight path of RAPTOR. In prototyping phases, Pilot utilizes a simple discrete servo control scheme, where servos rotate at a fixed rate and duration to turn either left, right, or continue straight. The modularity of the object-oriented design allows for individualized development and testing for various classes. As the system increases in sophistication, classes may be added or removed to provide differing functionalities.

B. Landing Location Determination

i. Groundnet

RAPTOR shall utilize two artificial neural networks to handle landing location selection and determination. The first of these neural networks is Groundnet. Groundnet is a convolutional neural network trained to identify open fields in satellite imagery via Google Maps API. The program is written using Google's Tensorflow framework. Groundnet is run before a flight to identify potential landing locations in a set radius around the expected landing location. The locations will be output as a list of coordinates that the team will then verify. Once the landing locations are verified, they will be entered into the RAPTOR payload to be used by Skynet during flight.

ii. Skynet

The second of these neural networks is Skynet. This is a combination of a feed forward neural network and a rule driven decision tree used in order to take all of the possible landing locations and determine the optimal one of the set. Each landing location will be run through the neural network portion of Skynet to determine if it is a viable landing location or not. Input parameters for the first stage of Skynet include altitude, velocity, distance vector, wind profile, and other environmental variables. Once the neural network has narrowed the set of potential landing locations to only the most viable for this run, the set is entered into a decision tree in order of viability. The idea is to go through the set of the viable locations and decide which of these few is the optimal one. This allows for a learning approach to narrowing down our list and a more controlled approach for deciding which one to take. This is more efficient than approaching the problem as a classification problem and choosing one location out of the list in a much larger neural network.

Skynet is also written using the Tensorflow framework. This system is trained on simulation data and all weights will be saved to a text file for transfer learning. During training, the rule set for the decision tree will be fine-tuned for more accurate results. On launch, Skynet will load the weights off of the text file to avoid training during ascent. This has the added benefit of allowing any flight team using RAPTOR to avoid having to adjust the design to work with specific flight conditions. Figure 9 shows the current architecture of Skynet. As the project continues, the sizes and numbers of layers are subject to change for additional tuning.

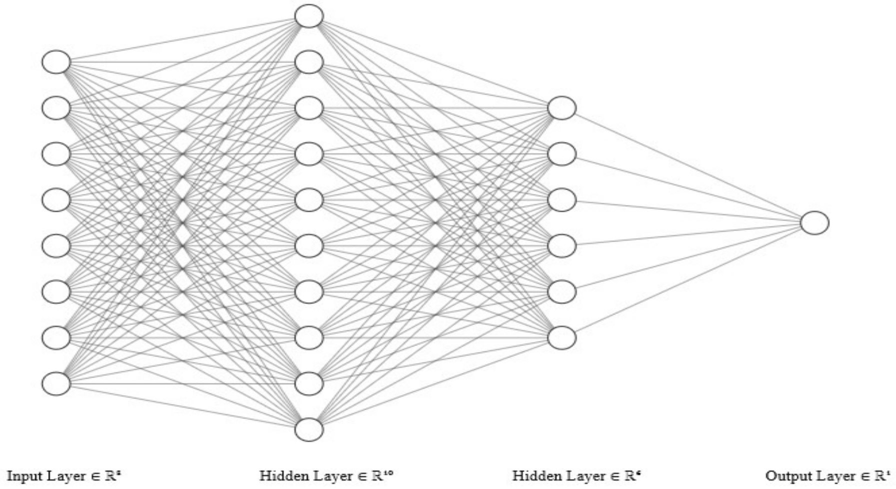
Input Layer ∈ $\mathbb{R}^8$
Hidden Layer ∈ $\mathbb{R}^{10}$
Hidden Layer ∈ $\mathbb{R}^6$
Output Layer ∈ $\mathbb{R}^1$

Figure 9. Skynet Architecture

C. Guidance Algorithms

i. Pathfinding

Pathfinder is the current iteration of the guidance algorithm, used for basic prototyping. Pathfinder takes in the current and desired heading to determine the necessary direction for the payload to turn. Figure 10 shows the graphical representation of the algorithm. Alpha and beta angles are calculated from the desired heading, forming a line perpendicular to the desired heading. The distance from the current heading to the two angles are then calculated. If the alpha angle is closer to the current heading, a left turn is necessary, if the beta angle is closer, a right turn is needed. If the desired heading within 15 degrees of the current heading the payload straightens out. This algorithm will be used while collecting data needed for implementation of a PID controller.

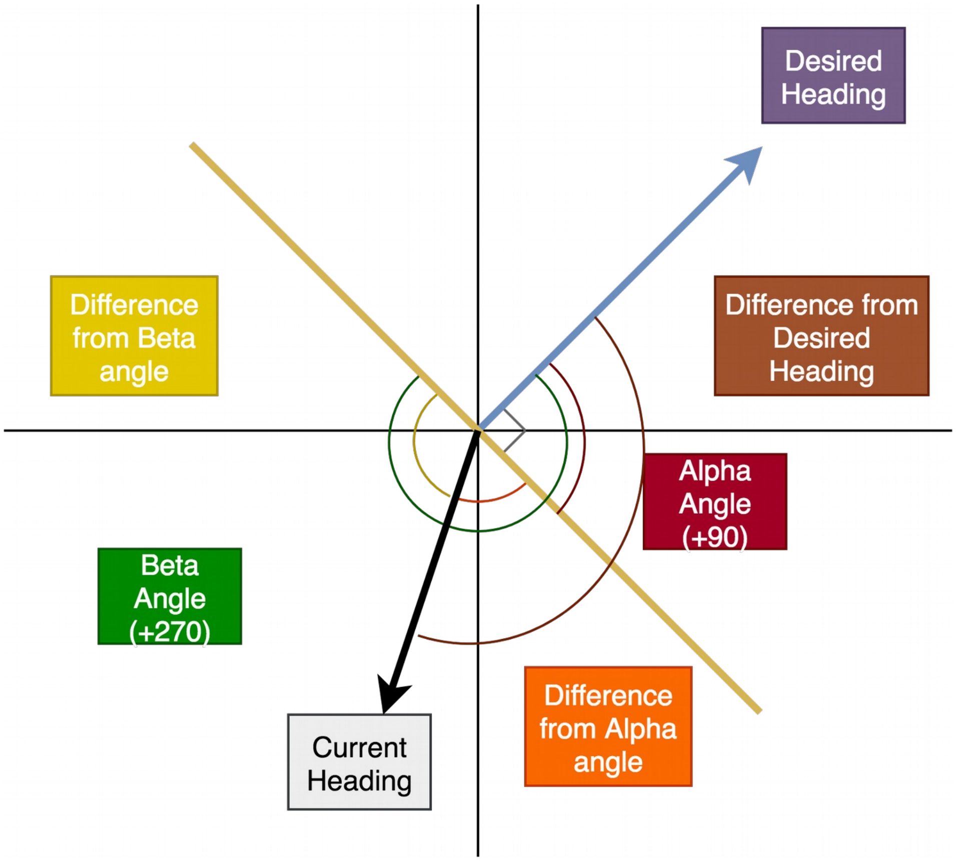
Desired Heading
Difference from Beta angle
Difference from Desired Heading
Alpha Angle (+90)
Beta Angle (+270)
Current Heading
Difference from Alpha angle

Figure 10. Pathfinding Algorithm Visualization

ii. PID Controller

Despite the robust simplicity of Pilot, a more efficient solution is desired. This solution must be able to compensate for wind gusts or extreme changes in vehicle orientation. Pilot controls the servos in a binary fashion, meaning the servos do not have proportional control and can only turn full-left or full-right. A proposed solution is to replace Pathfinder with a Proportional-Integral-Derivative controller, otherwise known as a PID controller.

The upgrade to a PID controller would solve the weaknesses identified in Pilot. A PID controller would be able to remain stable even with disturbances to the vehicle. A PID controller would also be able to dynamically operate the servos in a manner that is more efficient and precise.

A PID controller is an industry standard control scheme, utilizing three gain values to achieve stability. A PID controller can be described by the follow function.

$$u(t)=K_p e(t)+K_i\int_o^t e(t')dt'+K_d\frac{de(t)}{dt} \quad (2)$$

Where K_p, K_i, and K_d are non-negative gain coefficients.

Rather than using a system model to derive an analytical controller using classical control theory, RAPTOR aims to apply data driven control to create its PID controller. Data driven control is a subfield of control theory that utilizes sets of empirical test data to design a controller based on measured system dynamics. This test data can be processed by a computing suite, such as MATLAB, to generate a controller. The goal of a data driven control system is to synthesize a controller C that achieves closed-loop stability and meets the performance specifications of $\widehat{G}$. Where $\widehat{G}=G(\widehat{\Theta}_N)$ and $\widehat{\Theta}$is a vector of parameters of Gidentified on a set of Ndata.

RAPTOR is going to extend its current flight data recording to develop a set of real time flight data. Using this data, MATLAB's Control System Toolbox can be used to synthesize a controller. With each successful flight, a more accurate controller will be developed.

The data driven control system will be designed using direct off-line iterative methodology. This allows for design parameters to be updated over time through normal vehicle operations. For a data driven control system using iterative feedback tuning, the control design objective can be described as the minimization of the objective function.

$$J(\rho)=\frac{1}{2N}\sum_{t=1}^{N} E\,\text{¿¿}\,(3)$$

Where $y^d$is the output of a reference signal r and $\underset{\sim}{y}= y(\rho)- y^d$is the error between the desired response and the achieved response.

As part of the unified guidance scheme the PID controller enacts the physical results of Skynet. Once Skynet has computed a landing location, that location will be used as the reference for the controller. Using this reference, the controller will apply the control signals to the servos. To improve power efficiency of the controller, reference error, $e_a(t)$, checking will vary with respect to current flight state. For example, the controller should be more responsive in a landing flight state.

While the primary purpose of data driven control is to design an efficient controller, this design process will also help generate a model of RAPTOR's flight dynamics. These dynamics could be used in simulations or for further parafoil research. As the model accuracy grows, there is potential to apply additional control techniques. Additions of feed forward loops or open loop control might allow for a more responsive controller.

V. Electrical Development

A. Single Power Source

Previous iterations of RAPTOR have utilized two independent power supplies. This was due to demands of both the servos and solenoids. While this approach is effective for meeting the power demands of the vehicle, it presented several problems.

The use of disposable batteries meant that each flight had to use a new set of nine batteries. This added an unnecessary layer of complexity cost, and weight. To determine the power demands of RAPTOR, measurement of the battery voltages was conducted. Using this information, Phoenix will replace the disposable power supply with a single rechargeable LiPo battery power source. One of the benefits offered by a LiPo battery is that it acts as a single source, instead of two unique sources, reducing the number of failure points in the power system. A LiPo battery is also rechargeable, eliminating the cost of replacing disposable batteries after each flight. Due to reduced PCB and wire weight, this new power source is also significantly lighter. Additionally, the LiPo is capable of handling higher power demands.

B. Unified Sensor PCB

For ease of development, previous iterations of RAPTOR had sensors on several isolated breakout boards. This allowed for rapid design and testing of sensors between flights. As the prototyping phase continues, additional sensor boards only act as unnecessary weight. In addition to an improved power supply, Phoenix will utilize a custom-made PCB shield for an Arduino Uno containing the vehicle's sensor suite. This allows for a more compact design and neater internal wiring, eliminating undue complexity and room for error.

C. Future Plans

Phoenix represents a stepping stone in the electrical development of RAPTOR. Future iterations of RAPTOR will take lessons learned on Phoenix to further simplify the electrical design. This approach will allow RAPTOR to have increasing electrical capabilities while reducing weight. Additional improvements may include active heating of the vehicle interior and enhanced telemetry.

VI. Project Development

RAPTOR uses the Agile approach to project development. Incremental changes are made to the payload to increase capabilities between flights, where major versions indicate significant progress made. Previous iterations of the payload included Kestrel and Falcon, both meant to act as testing platforms for the mechanical design, electronics, flight software, and parafoil control. Future versions will introduce enhanced electronics, improved mechanical elements, and more sophisticated flight software. Phoenix will be the final version where the RAPTOR payload will fly only itself. Hawk will introduce the ability to carry payload trains, while Eagle will be the final iteration, incorporating the culmination of all of RAPTORs design improvements.

VII. Conclusion

Development of RAPTOR is well underway, having flown two prototype payloads, Kestrel and Falcon. The project still aims to be a valuable asset to HAB payload flights. The authors look forward to continuing the development of RAPTOR with the next version, Phoenix. Phoenix is currently being developed, and has several test flights planned during the summer of 2019. Additional applications of the RAPTOR system are being explored, such as an adaptation for high power rockets.

Acknowledgments

The authors would like the thank the University of Alabama in Huntsville Space Hardware Club for providing resources for this project, and Dr. Francis Wessling, Dr. Daniel Armentrout, Dr. Richard Tantaris, Dr. Brian Landrum, Dave Arterburn, Eric Salverson, Chris Sallis, and Dr. Scott Widmier for assistance and advisement in the development of the project and writing of this paper.

References

[1] Miles, Donna., “New Airdrop System Offers More Precision from Higher Altitudes,” *American Forces Press Service* [online news article], URL: http://archive.defense.gov/news/NewsArticle.aspx?ID=1815 [cited 28 February 2019]

[2] Yakimenko, Oleg A., *Precision Aerial Delivery Systems: Modeling, Dynamics, and Control*, Progress in Astronautics and Aeronautics, AIAA, Reston, Virginia, 2015.

[3] Benton, J. E., and Yakimenko, O. A., “On Development of Autonomous HAHO Parafoil System for Target Payload Return,” AIAA paper.

[4] “Cambridge University Spaceflight Landing Predictor,” Cambridge University [online simulation tool], URL: http://predict.habhub.org/ [cited 28 February 2019]

[5] “14 CFR 101.33 – Operating Limitations,” Cornell Law School [online database], URL: https://www.law.cornell.edu/cfr/text/14/101.33 [cited 28 February 2019]