Monitoring of Large Indoor Areas Using an Autonomous Dirigible

Worcester Polytechnic Institute

Depts. of: Aerospace Engineering, Robotics Engineering
Major Qualifying Project

Authors:
Arreola, Juan
Henchy, Cameron
Moise, Andy
Rollins, William
Saddler, Teresa
Sperotto, Guilherme

Advisor:
Prof. Demetriou, Michael A.

Co-Advisor:
Prof. Pinciroli, Carlo

May 6, 2021

Abstract

Blimps are lighter than air vehicles characterized by their non-rigid structure and their use of pressurized gas to maintain their shape as well as provide lift. In this project report, a system for converting an RC blimp for autonomous operation is proposed. The intended application of the autonomous blimp is for indoor surveillance, as blimps are particularly well-suited for such tasks. Their low weight allows for the use of low thrust, electric motors that reduce the power and complexity of the overall system. These low power requirements result in a safe and quiet system that can be used in close proximity with people. To modify a blimp to be operated autonomously, sensing capability must be added, in this case, computer vision and inertial sensing. Both of these sensing systems are used for localization to allow the blimp track its location and orientation within a known space. The sensing systems are used in conjunction with an extended Kalman filter (EKF) and a sensor fusion algorithm to improve the accuracy of the localization process. A Linear-Quadratic Regulator (LQR) controller is then used to create motor commands and allow the blimp to follow a path between locations. Using these control capabilities, the blimp is able to monitor a room for any indoor environmental changes or other disturbances using the existing sensing system, with the option to add additional sensors if necessary.

Acknowledgements

The Autonomous Dirigible MQP group would primarily like to thank:

Professor Michael Demetriou, Ph.D.
Professor, Aerospace Engineering Program
Department of Aerospace Engineering, Worcester Polytechnic Institute

The group would also like to give special thanks to:

Professor Carlo Pinciroli, Ph.D.
Assistant Professor, Robotics Engineering Program
Department of Robotics Engineering, Worcester Polytechnic Institute

Tina Stratis
Administrative Assistant
Department of Aerospace Engineering, Worcester Polytechnic Institute

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background on Dirigibles and Blimps

Blimps are a category of aircraft that gain their structure and buoyancy from the gas pressure contained inside an envelope. They are a type of dirigible, meaning that they are lighter-than-air, powered, and steerable, according to [1]. Zeppelins, another example of a dirigible, have a rigid or semi-rigid internal structure. Dirigibles were the precursors for modern fixed-wing airplanes both for passenger transport (such as the Hindenburg) and wartime applications such as scouting and bombing. Today, modern airplanes rule the skies, but that does not mean dirigibles have become obsolete. There are advantages and disadvantages to each type of air vehicle, and we have created a table 1 to summarize them as shown below . With their low power consumption and airspeed, and relatively low production costs, the primary purpose of dirigibles has shifted to advertising over the past decades, with new and exciting applications being researched. One of these applications, surveillance, is explored in this paper.

Table 1: Comparison of Three Air Vehicles in Nine Different Categories [2].

Category	Planes	Helicopters	Dirigibles
Operation cost	++	+	+++
Endurance	++	+	+++
Hovering capability	+	+++	+++
Maneuverability	++	+++	+
Low turbulence	+	+	+++
Take-off / Landing	+	+++	+++
Payload ratio / Weight	++	+	+++
Fuel consumption	++	+	+++
Stability	++	+	+++

1.2 Scope and Goal of Our Project

The purpose of this Major Qualifying Project (MQP) is to design a control system that can autonomously guide a single blimp or multiple blimps along a predetermined path. This project will focus on small, indoor blimps that can be used in large, indoor spaces. Autonomous control in this project represents the blimp being able to follow a path, adjust for disturbances, and continue to move on its path indefinitely. Autonomy will end when a human delivers a signal to stop operation.

1.2.1 Project Design Requirements, Constraints and Considerations

Constraints and requirements were set on the project by the team both out of practicality and to narrow the scope of the project. One of the primary anticipated applications of the blimp control system is in a network of blimps for monitoring large indoor spaces. In others words, multiple blimps would be able to fly in formation and communicate with each other to effectively and efficiently survey a designated area. For this purpose, a natural constraint is that the dirigible(s) must sustain autonomous flight for extended periods of time. Due to the buoyancy of blimps, this is easier than in many other aircraft that require power to remain airborne. However, the blimp still has a significant power requirement as it must have a large enough battery to power all electrical components for the entire mission time. The blimp also must be able to detect disturbances in the environment. The team had a budget of $2,000, which was used to purchase the dirigible and the onboard sensors. This budget also limited the number of blimps in experiments to one. Therefore this project primarily focuses on making a single blimp autonomous to allow for future research into a network of autonomous blimps. This budget dictated the choices of blimp models and the quality of the electronic payload including the sensors. Due to the testing locations available, there are also limits on the size of the blimp. The dirigible is about two meters long, which is approximately the minimum for typical commercial indoor dirigibles, and has little room to maneuver in the testing room. Other considerations for this project are the COVID-19 restrictions that have affected students throughout the school year. The team has only been able to meet via Zoom and not in person. The labs have been restricted to a set number of people at a time and the team has not had much opportunity to access the lab. This has affected the project's timeline, by requiring all theoretical work, including the controller design and derivation of the equations of motion, to occur without access to system testing during the first half of the project, while all testing had to occur in the last half of the project. These constraints together have reduced the scope of the project, and also partially defined the timeline and the tasks required.

1.3 Previous Projects Relating to Dirigibles

There have been many other projects investigating the possibility of autonomous dirigibles, with various intended applications. The University of Waterloo [3], French National Centre for Scientific Research [4], Georgia Institute of Technology [5], University of Michigan [6], MIT [7], and McGill University [8] are among a long list of institutions undertaking such projects. These projects used different methods to achieve automation. Each method used was chosen due to a specific targeted application for the autonomous blimp. Some projects used ultrasonic sensors and infrared cameras for obstacle avoidance near humans while others relied on neural networks and a constant camera feed to recognize objects in new environments. Regardless of the other applications, a successful project requires a complete understanding of the components that enable precise modeling, estimation, and control of the blimp at all times.

1.3.1 Blimp Control and Tracking

Many papers have proposed different controllers to effectively operate autonomous blimps. As in [9], [10] and [11], control is frequently split between lateral motion in the horizontal plane and direct vertical motion. This decoupling simplifies both the control and path planning, such that the blimp only ever moves in one or two dimensions at once. Beyond this, there are many techniques to define controllers for blimps. In some, controllers were created to estimate and compensate for disturbances, as in [9] and [10]. Others use PI or PID controllers, as in [11]. While this demonstrates a great diversity in complexity in blimp controllers, there is also a cost of effectiveness in some of the less complex controllers, due to the complicated nature of the nonlinear blimp dynamics.

Another common challenge encountered in blimp control projects is the low payload allowance of blimps. Blimp payloads are very limited, with the blimp used in this project having a listed maximum payload of only $200g$. This reduces options for blimp-mounted sensors to be used for position and pose tracking. This is a severe limitation, since the sensors that are available to be mounted on the blimp are significantly less accurate than would generally be desired for aircraft. In [9] and [10], this issue was mitigated by off-loading the position and pose tracking to an OptiTrack camera system, which rather than being mounted on the blimp, utilizes an array of cameras in the testing space. While this is highly effective, it is also a very expensive and inflexible system since it requires multi-camera mounting. Other researchers used GPS sensors, as in [11], but this limits the blimp to outdoor applications since GPS receivers are typically ineffective inside of buildings.

There are also challenges that come with necessary estimation and sensor filtration techniques. A standard Kalman Filter is efficient and optimal to filter sensor noise for linear systems. However, the nonlinear dynamic system of this blimp will require a Continuous-Discrete Extended Kalman Filter (CD-EKF). Using this complex variation of the Kalman Filter will lead to different implementation methods. There was a study done in 2007 at the University of Stellenbosch done by Johan Bijker that addressed different ways to use an EKF. The study used an unmanned airship, similar to the blimp in this report, equipped with a gyroscope and a GPS receiver [12]. The motion is modeled using 12 states, with three control inputs. The most significant part of the study was the testing of the EKF. Three different EKF implementations were tested and measured the sensor error, and the floating-point operations per second (FLOPS), which indicates operation power needed for basic matrix calculations. First, an EKF was studied where the state matrix was 12x12 and the control matrix was 12x3. Next, a similar EKF was used but had the addition of different sensors. A third EKF method was considered during the study. The third EKF broke the 12x12 state matrix into two 6x6 matrices. One matrix included the states describing translational motion and the other included the states describing rotational motion. The conclusion of the study was that the FLOPS greatly decreased during the use of the paired EKF estimations with a only small increase in sensor error [12]. This study on efficiency was very important as it demonstrated the actual computation power necessary to run an EKF in real time. This helped our project as we could use these finding to simplify and optimize the CD-EKF so that the processing requirements are feasible for the equipment onboard the blimp.

1.3.2 Navigation

For this project to be successful, the blimp must have a path to travel. There are numerous path planning methods that have been used for autonomous flight. One such method is the A* search algorithm, which is a path planning algorithm designed at the Stanford Research Institute in 1968 to aid in the development of autonomous robots [13]. A* attempts to find the shortest distance between start and end points given a map of obstacles between those two points. This is achieved by calculating two criteria. The first is the movement cost necessary for an object to travel from its current position to any of its closest neighboring positions. The second is the movement cost of each neighboring position to the desired end point according to a heuristic designed for the application. These costs can be in the form of time or distance, depending on the prevailing movement constraint of the intended application. The combined cost for all new path options for a given position is stored in a priority queue and sorted in order of costs with the lowest cost at the top of the queue. After calculating costs and creating this queue, A* chooses the path option at the top of the queue and uses that point as the new current point. This process of calculating cost and moving to the location with the lowest total cost is repeated until the end point is reached.

Since many previous autonomous blimp projects used vision-based navigation, additional research was done to understand different methods of vision-based navigation. The purpose of vision-based navigation is to have optical sensors give feedback to the controller about the location and orientation of the controlled body. There are an array of different methods for vision-based navigation, including tag-based methods, and more complex landmark-based methods.

One of the most common vision-based navigation methods is using generated tags as landmarks to determine the camera's position and pose with respect to the tag. A common tag-based system is AprilTag, which is a type of vision-based navigation developed by Edwin Olson, the director of the APRIL robotics lab at the University of Michigan, for the localization of objects in an image [14]. This system was designed to be computationally inexpensive and can operate with only one camera. Each AprilTag is a 2-D printed image that is very similar to QR codes. These tags can be seen by cameras in most conditions due to their black and white color scheme and simple layout, as shown in Figure 1. By knowing some characteristics about the camera used and the AprilTag printed onto paper, the translation and rotation of the camera in relation to the AprilTag can be calculated with only a photograph. This, along with some knowledge of where the AprilTag is located, can further define the current position of the camera in a user-defined reference frame. There are multiple AprilTag configurations, each with a situation where it performs best. The uramaki square, Tag36h11 in Figure 1, is the least complex of the configurations since it is defined by dots within a set of black and white borders. The circular family can also be used for localization, but its different shape allows it to be mounted on different structures or smaller objects. Recursive tags are usually used for landing because they have a blank square in the center where another, smaller tag can be placed. The advantage of AprilTags is that their simple design allows for their position to be calculated very quickly. The disadvantage of using a visual tag-based system is that the tags can fail to be recognized if any part encoding data is obscured or warped by imperfections in the environment or camera [15].

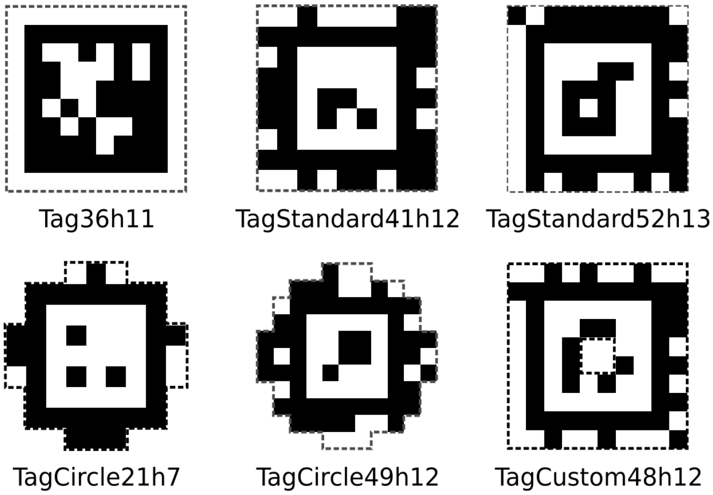


Figure 1: AprilTags example [16].

Another vision-based method is the method of intersection recognition. This method uses landmarks and boundary lines to autonomously navigate the blimp using one camera [17]. Intersection guidance markers (IGMs) are placed within the route for the camera to pick out. The location of the camera can only be determined if there are two IGMs in the camera's line of sight, as shown in Figure 2. Then, the image plane is divided into axis a and axis b. Equations (1) - (4) are then used to determine the camera's placement from the IGMs.

$$a_c = (a_{L_{max}} + a_{R_{max}}), \tag{1}$$

$$b_c = \frac{(b_L + b_R)}{2}, \tag{2}$$

$$x = \frac{az}{(f \sin\theta - b\cos\theta)}, \tag{3}$$

$$y = \frac{z(f\cos\theta + b\sin\theta)}{(f\sin\theta - b\cos\theta)}. \tag{4}$$

a_c is the center between the two markers on the a-axis, b_c is the center between the two markers on the b-axis, and the camera is located at the origin of the plane. f is the focal length of the camera, z is the

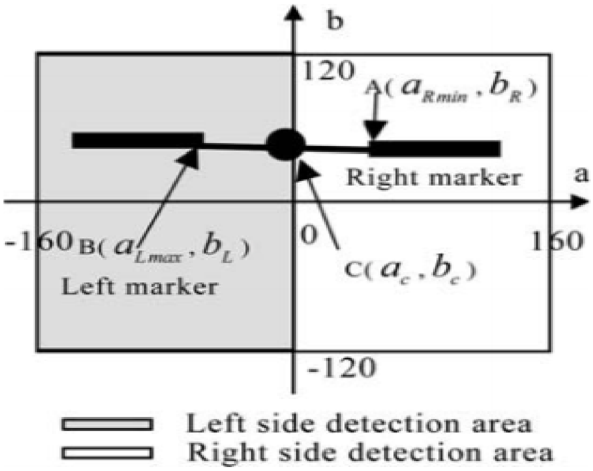


Figure 2: Schematic of the methods of intersection algorithm [17].

altitude of the camera and θ is the depression angle from the camera to the markers. The advantage of using the intersection recognition method is that IGMs can be existing landmarks and do not have to be created or placed. The disadvantage is that two markers must be within the camera's field of view and if only one is in the camera's field of view, the location cannot be determined [17].

Another method of vision-based navigation also uses one camera and relies on pre-existing maps [18]. A 3-D map of the environment would be developed and saved to the system. The camera saves an image that is processed to detect the curvature of the landscape. Then, based on the pre-made map and the features found in the processed image, the location can be determined with rapid geometric calculations. The advantage of this technique is that no landmarks need to be placed or set up in the testing environment and the camera can reliably operate anywhere within the bounds of the 3-D map. This allows the system to navigate a space of arbitrary size. The primary disadvantages are that the development of a 3-D map to a high degree of precision is extremely time consuming and may require the use of expensive equipment. Also, the geometric calculations required to use the map for localization is processor-intensive. These requirements make navigation by 3-D maps not suitable for this application.

1.4 Broader Impact

Due to the their low cost and long flight times, the research and development of autonomous dirigibles has increased over the years. Many agencies such as NASA use airships in their research, and military forces use unmanned aerial vehicles for reconnaissance; however, the use of dirigibles is not limited to large government entities or military purposes. Many blimp applications are designed for environmental preservation and to increase human safety, and the enabling of blimps as autonomous vehicles enhances the opportunities to use blimps for these purposes.

Blimps have extensive applications in environmental monitoring for beneficial purposes such as environmental monitoring and search and rescue. For example, in [19] researchers develop an autonomous blimp for environmental monitoring, with potential applications in natural areas such as forests and national parks, as well as over farmland, cities, and industrial sites. Blimps can also be used to monitor disaster areas that may have lost the critical infrastructure necessary to support missions with greater complexity, such as cities after a major earthquake. Blimps have been used to monitor disasters because of their low altitude flying capability and reduced turbulence, which makes them safer than helicopters [20]. Between their ability to have a minimal effect on the environments they are monitoring and their unique characteristics such as low-altitude flying and long duration capabilities, autonomous blimps have many potential valuable applications.

While the potential applications for blimps is vast, the market for autonomous aircraft is currently dominated by quad-rotors or other rotorcraft. Many of the small-scale missions for these autonomous aircraft are focused on the surveillance of key systems. Small helicopters have been used by civilians to monitor agricultural fields and even automate crop dusting. Several government agencies have implemented quad-rotors in their natural disaster response as a small aerial vehicle is less likely to disturb structures in danger of collapse [21]. An autonomous blimp, much like the one developed for this project, can make surveillance jobs safer and easier. Due to their low-speed and low-power characteristics, blimps are far safer to people in close-quarters, and are less likely to cause damage to important systems in the instance of collisions. The addition of an automation system will lead to greater responsiveness and longer surveillance times, making this alternative more enticing to end users.

1.5 Project Management

The primary task was to develop a dirigible that can sustain autonomous flight along a pre-determined path in an indoor environment. The major decisions for this project, such as blimp selection and testing protocols, were made by all members of the team. For other parts of the project the six team members were split into two teams. The design team consisted of Cameron Henchy, Guilherme Sperotto, and Andy Moise. The control team consisted of Teresa Saddler, Juan Arreola, and William Rollins.

The design team's main focus was to design the blimp for flight. This included developing the equations of motion, ordering parts needed for the project, creating a computer simulation of the blimp moving, and designing mounting solutions for the added control components on the blimp. The control team's responsibilities were to select and install components on the blimp to enable autonomous flight, develop the control schema, design and test the control system, and program the controller for flight tests. The entire team met once a week via Zoom for updates, with the subteams meeting more regularly.

1.6 MQP Objectives, Methods, and Standards

Despite the reduced scope of the original project, there were still several objectives for the team to complete. The first objective was to derive and understand the equations of motion of the autonomous blimp system. To do this, previous research into the equations of motion of similar vehicles was reviewed. Using these equations of motion, test flights were used to confirm the accuracy of the modelled dynamics. Another objective was to outfit the blimp for autonomous flight, since it had been originally designed for RC-piloted flight. To do this, we selected and installed components to control the blimp autonomously, and tested any sensors that were being used to obtain their biases and covariances. The primary objective was to design a control schema to allow the blimp to fly autonomously. To do this, we had to use the sensor outputs and the equations of motion, and select an algorithm to localize the blimp. The measured state of the blimp and the equations of motion had to be used to select a controller design to define the inputs to the motors such that the blimp was able to follow the defined path. The entire control schema had to be tested in Simulink to verify its accuracy and compare different options to find the most effective control gains. The final objective is to implement this control schema on the blimp to allow it to fly autonomously. In order to do this, we had to program a development board with the algorithms from the control schema so that the blimp can fly between points on the path. We also had to run flight tests with the blimp to experimentally determine the accuracy of the controller and make adjustments as necessary.

1.7 MQP Tasks and Timetable

In order to complete the objectives of this MQP, we split the objectives up into individual tasks and made a timetable to impose deadlines to keep the project on track. Due to the restrictions pertaining to COVID-19, some of these tasks had to be postponed for later in the project. The schedule for the completed tasks for the project are listed in Table 2.

Table 2: Task Schedule.

Task	Estimated Completion
Preliminary Blimp Research	9/8
Blimp Trade Study	9/29
Wireless System Configuration	10/13
Control System Trade Study	10/20
Blimp Purchase	10/27
Control Component Purchase	10/27
Control Scheme Design	1/19
Sensor Verification	2/9
Power System Selection and Ordering	2/16
Equations of Motion Simulink Modeling	2/24
Sensor Testing	3/26
Controls Simulink Modeling	3/30
Preliminary System Testing	4/9
Controls Programming	4/19
Control System Assembly	5/3
System Testing	5/6

2 System Design and Configuration

2.1 Blimp Trade Study

This creation of a blimp is beyond the scope of the project so one had to be purchased from a blimp manufacturer. In order to make the best purchase decision possible, we decided to conduct a trade study. Several factors were important to us, including size, payload capacity, battery life, price, and shipping time. We examined four companies, eBlimp.com, BlimpGuys.com, Aero Drum Ltd., and Skyshipblimps.com. Table 3 shows the results of the trade study.

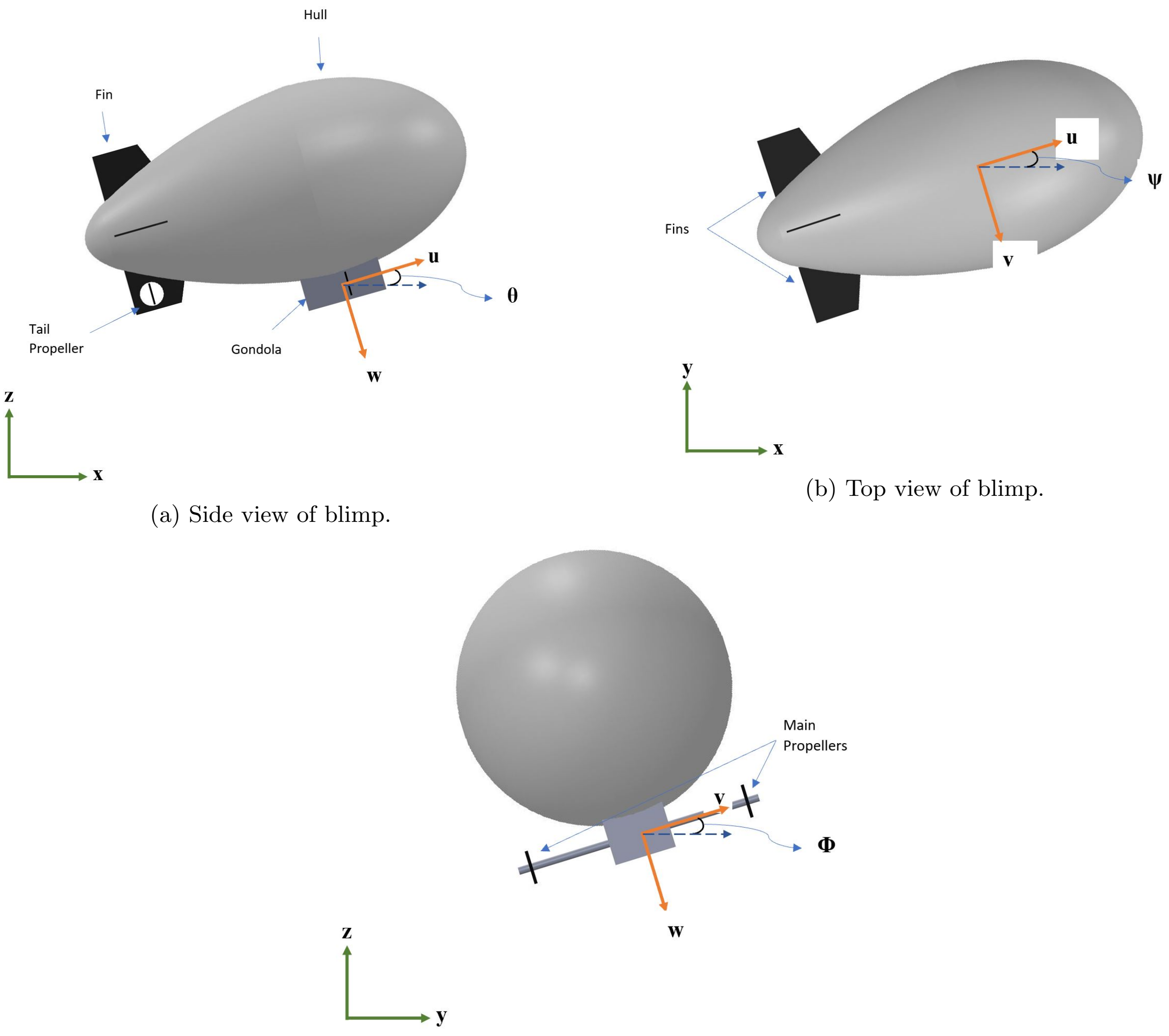
Table 3: Blimp Trade Study.

	Skyshipblimps.com	eBlimp.com	BlimpGuys.com	AeroDrum Ltd
Size (LxW) (m)	3 x 1.2	2 x 1.1	2.4 x 1.2	2.2 x 1.4
Payload Capacity (g)	400	100	100	200
Battery Life (min)	60-120	20-60	NA	40 - 60
Price ($)	$3,900	$3,500	$1,659	$1,115
Shipping From	Israel	California	Ontario, Canada	Serbia

Skyshipblimps.com was not a great choice for the project for several reasons. Although it has a great payload capacity, it is also large and expensive. Additionally, the company is based in Israel, which would result in a shipping time which would have caused delays in the project. AeroDrum Ltd appeared to be a very promising candidate, with a high payload capacity, reasonable battery life, and low price. However, they ship from Serbia, so the shipping time was potentially unpredictable. With only about 9 months to complete the project, this was too great a risk. This left two options, eBlimp.com and BlimpGuys.com. Both are based in North America, and both have similar specifications, making price the determining factor. Blimpguys.com defeats eBlimp.com easily in this category, as prices are less than half that of their competitor. Selecting Blimpguys.com as a supplier is an easy choice at the end of the trade study.

2.2 Equations of Motion

The blimp we are using for this project consists of a hull that holds the helium and a gondola with two propellers that are attached at the bottom. The blimp also has a tail fin with a third propeller and fins on the sides and top of the blimp. The two main propellers on the gondola of the blimp can tilt up and down and control the blimp's forward and vertical motion. The tail propeller is used to turn the blimp. A model of the blimp can be seen in Figure 3. Figure 3 also shows the body axes of blimp (in orange) and the orientation of the inertial axes (in green).



(a) Side view of blimp.

(b) Top view of blimp.

(c) Front view of blimp.

Figure 3: Schematic of blimp.

In order to properly control the blimp, an accurate set of equations of motion is needed. There are several pieces of existing literature on deriving the equations of motion for various types of dirigibles. One such paper [22] was found that is very similar to our project. The paper has some of the most comprehensive derivations for the equations of motion, which are summarized in the following sections.

2.2.1 Translational Motion

The translational motion of the blimp in the inertial frame can be written as:

$$\dot{r} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_{IB1}^{T} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{5}$$

where x, y, and z are the position components in the inertial frame, u, v, and w are the velocity components of the aircraft in the body frame, and R_{IB} is a rotation matrix from the inertial frame to the body frame that is based on a 3-2-1 Euler angle sequence, so that

$$R_{IB1} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi)+\sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi)+\sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi)+\cos(\phi)\sin(\theta)\cos(\psi) & -\sin(\phi)\cos(\psi)+\cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix}. \tag{6}$$

The velocity vector, V, can be derived by using Newton's second law, which states that the sum of forces, ΣF, acting on object equals the mass of the object, m multiplied by the acceleration of the object a

$$\Sigma F = ma, \tag{7}$$

$$\Sigma F = m\ddot{r}. \tag{8}$$

Substituting Equation (5) into Equation (8),

$$\Sigma F = mR_{IB1}\dot{R}_{IB1}^{T}V + m\dot{V}, \tag{9}$$

where

$$\dot{R}_{IB1} = \omega \times R_{IB1}, \tag{10}$$

ω in the Equation 10 is a 3×1 matrix, $[p\ q\ r]^T$, consisting of the roll, pitch, and yaw rates.

2.2.2 Rotational Motion

Aircraft orientations are commonly expressed in Euler angles and quaternions [23]. Euler angles are a classical method of describing rotations about a body frame. These are simple to visualize and calculate when needing to express the orientation of a body in an inertial reference frame. However, Euler angles carry the limitation that certain orientations can be the result of multiple rotation sequences and are therefore not unique to a single combination of Euler angle rotations. This ambiguity is referred to as gimbal lock, and results in the loss of a degree of freedom. In some applications, this mathematical issue makes Euler angles poorly suited to describe the orientation of a body.

Quaternions are another method of expressing a rotational transformation by decomposing the rotation into a vector of its components. This makes quaternions less intuitive to use than Euler angles, but avoids the gimbal lock issue. Many aircraft that are expected to move and rotate about all axes use quaternions to avoid the ambiguity of Euler angles. Although quaternions are useful to prevent rotation errors, these errors can also be avoided by the selection of an Euler rotation sequence with a singularity in the direction where the aircraft is unlikely to orient itself. A review of various Euler rotation sequences determined that one sequence would reach a singularity only if the blimp became fully vertical. As this would be an impossible occurrence during normal operations, and [22] uses Euler angles rather than quaternions, we have elected to use Euler angles in order to simplify calculations.

The rotational motion of the blimp in the inertial frame can be written as

$$\dot{\Omega} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = R_{IB2}\omega, \tag{11}$$

where ϕ, θ, and ψ are the roll, pitch, and yaw angles, and R_{IB2} is a mapping of the angular velocity in the body frame to the inertial frame,

$$R_{IB2} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix}. \tag{12}$$

The angular velocity vector, ω, is derived using Euler's second law. Euler's second law states that the sum of moments acting on an object equals the rate of change of angular momentum of the object,

$$\Sigma M = \frac{d}{dt}\left(I_M \dot{\omega}\right), \tag{13}$$

where I_M is the moment of inertia tensor. Equation (13) can be simplified to

$$\Sigma M = I_M \dot{\omega} - \omega \times I_M \omega. \tag{14}$$

The moment of inertia tensor is a 3×3 matrix made up of the moments of inertia across each axis and the products of inertia. Since the blimp is symmetric across every axis, the products of inertia are zero and we are left with the following matrix 15).

$$I_M = \begin{bmatrix} I_{M_{xx}} & 0 & 0 \\ 0 & I_{M_{yy}} & 0 \\ 0 & 0 & I_{M_{zz}} \end{bmatrix}. \tag{15}$$

2.2.3 Forces

The forces on the blimp are due to aerodynamic/drag forces, F_0, the virtual masses, F_1 and F_2, the thrust of the propellers, T_m and T_t, and the buoyancy and weight, F_b and F_g. The equation below shows how the aerodynamic/drag forces are calculated

$$F_0 = \begin{bmatrix} F_{xb} \\ F_{yb} \\ F_{zb} \end{bmatrix} = \frac{1}{2}\rho_{air} \begin{bmatrix} |u| * u \\ |v| * v \\ |w| * w \end{bmatrix} S_{ref} \begin{bmatrix} C_T \\ C_L \\ C_N \end{bmatrix}, \tag{16}$$

where ρ_{air} is the density of air, $||V||$ is the magnitude of the velocity vector, S_{ref} is the surface area of the blimp, and C_T, C_L, and C_N are aerodynamic force/drag coefficients in the u, v, and w direction. The forces due to the virtual masses are calculated as

$$F_1 = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix}, \tag{17}$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 \\ a_{24} & 0 & a_{26} \\ 0 & a_{35} & 0 \end{bmatrix}, \tag{18}$$

where $\{a\}_{ij} = a_{ij}$ in the added mass matrix, which is defined in Appendix A. The thrust force of the main propeller, $\boldsymbol{T}_m$, is

$$\boldsymbol{T}_m = \begin{bmatrix} T_m \cos(\gamma) \\ 0 \\ -T_m \sin(\gamma) \end{bmatrix}, \tag{19}$$

where T_m is the magnitude of the thrust of the main propeller and γ is the angle the main propeller points. The thrust force of the tail propeller, $\boldsymbol{T}_t$, is

$$\boldsymbol{T}_t = \begin{bmatrix} 0 \\ T_t \\ 0 \end{bmatrix}, \tag{20}$$

where T_t is the magnitude of the thrust due to the tail propeller.

The force due to buoyancy and weight is calculated by subtracting the buoyancy from the weight of the blimp. This force only acts in the z-direction and is represented in the inertial frame as

$$F_{bg} = \begin{bmatrix} 0 \\ 0 \\ (F_g - F_b) \end{bmatrix}, \tag{21}$$

where F_g is the weight of the blimp and F_b is the buoyant force. To convert F_{bg} to the body frame, it is multiplied by the rotation matrix R_{IB1} from Equation (6).

2.2.4 Moments

The moments of the blimp are caused by the rotational drag, M_0, the virtual inertias, M_1 and M_2, and the thrust. To calculate the rotational drag moment, the following equation is used

$$M_0 = \begin{bmatrix} M_{xb} \\ M_{yb} \\ M_{zb} \end{bmatrix} = \frac{1}{2}\rho_{air} \begin{bmatrix} |p| * p \\ |q| * q \\ |r| * r \end{bmatrix} S_{ref} L_{ref} \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix}, \tag{22}$$

where L_{ref} is the length of the blimp, and C_l, C_m, and C_n are aerodynamic moment coefficients about the u, v, and w axis. The moments due to the virtual inertias, M_1 and M_2, are defined as

$$M_1 = \begin{bmatrix} a_{44} & 0 & a_{46} \\ 0 & a_{55} & 0 \\ a_{46} & 0 & a_{66} \end{bmatrix}, \tag{23}$$

$$M_2 = \begin{bmatrix} 0 & a_{42} & 0 \\ 0 & 0 & a_{53} \\ 0 & a_{62} & 0 \end{bmatrix}, \tag{24}$$

where a_{ij} are the entries of the added mass matrix defined earlier.

The moment due the tail propeller thrust can be calculated by taking the cross product of the tail propeller position vector and the tail thrust vector,

$$M_p = \rho_{T_t} \times T_t, \tag{25}$$

where ρ_{T_t} is the position vector of the tail propeller with respect to the center of gravity of the blimp.

2.2.5 Final Form of the Equations of Motion

For ease of calculations and simulations, the equations in Sections 2.2.1-2.2.4 can be combined and rearranged to form the expressions

$$\dot{V} = (mI_{3x3})^{-1}\left(F_0 + F_1\dot{V} + F_2\dot{\omega} + T_m + T_t + F_{bg} + \omega \times \dot{V}\right), \tag{26}$$

and

$$\dot{\omega} = I_M^{-1}\left(M_0 + M_1\dot{\omega} + M_2\dot{V} + M_p + \omega \times I_M\omega\right), \tag{27}$$

where M is the mass of the blimp and V is the velocity vector $[u\ v\ w]^{\mathrm{T}}$ in the body frame.

Equations (26) and (27) can be combined with Equations (5) and (11) to obtain the twelve states for the blimp. These twelve state will be the equations of motion for the blimp and will be arranged as shown in figure (28).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u\cos\phi\cos\theta + v\cos\phi\sin\psi + \cos\psi\sin\phi\sin\theta + w\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \\ u\cos\theta\sin\phi + v\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi - w\cos\psi\sin\theta - \cos\phi\sin\psi\sin\theta \\ -u\sin\theta + v\cos\theta\sin\phi + w\cos\phi\cos\theta \\ qw - rv + \frac{F_{xb}+a_{11}u+T_m\cos\gamma+\sin\theta(F_b-F_g)}{m} \\ ru - pw + \frac{F_{yb}+T_t+a_{24}\dot{p}+a_{26}\dot{r}+a_{22}v-\cos\theta\sin\phi(F_b-F_g)}{m} \\ pv - qu + \frac{F_{zb}+a_{34}\dot{q}+a_{33}w-T_m\sin\gamma-\cos\theta\cos\phi(F_b-F_g)}{m} \\ p + q\sin\phi\tan\phi + r\cos\phi\tan\phi \\ q\cos\phi - r\sin\phi \\ r\frac{\cos\phi}{\cos\theta} + q\frac{\sin\phi}{\cos\theta} \\ \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix}, \tag{28}$$

$\dot{\omega}_i$ is the i^{th} row of the $\dot{\omega}$ vector where

$$\dot{\omega} = \begin{bmatrix} \frac{M_{xb}-T_t z_{T_t}+a_{44}\dot{p}+a_{46}\dot{r}+a_{42}\dot{v}+q(I_{zx}p+I_{zy}q+I_{zz}r)-r(I_{yx}p+I_{yy}q+I_{yz}r)}{I_{M_{xx}}} \\ \frac{M_{yb}+a_{55}\dot{q}+a_{53}\dot{w}-p(I_{zx}p+I_{zy}q+I_{zz}r)+r(I_{xx}p+I_{yy}q+I_{xz}r)}{I_{M_{yy}}} \\ \frac{M_{zb}-T_t x_{T_t}+a_{46}\dot{p}+a_{66}\dot{r}+a_{62}\dot{v}+p(I_{yx}p+I_{yy}q+I_{yz}r)-q(I_{xx}p+I_{xy}q+I_{xz}r)}{I_{M_{zz}}} \end{bmatrix}. \tag{29}$$

I_{ij} is the the moment of Inertia across the ij axis.

Since the blimp doesn't roll or pitch at low speeds and it doesn't have control surfaces that can control roll or pitch, equations (28) can be simplified and reduced to 8 states. This was done by removing the states associated with roll and and pitch, ϕ, p, θ, and q The reduced equations can be seen be below in equation (30).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\psi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u\cos\psi - v\sin\psi \\ v\cos\psi + u\sin\psi \\ w \\ -rv + \frac{F_{zb}+a_{11}u+T_m\cos\gamma}{M} \\ ru + \frac{F_{yb}+T_t+a_{26}\dot{r}+a_{22}v}{M} \\ \frac{F_{zb}+a_{33}w-T_m\sin\gamma-(F_b-F_g)}{M} \\ r \\ M_{zb} - T_t x_{T_t} + a_{66}\dot{r} + a_{62}\dot{v} + I_{yz}r + I_{xz}r \end{bmatrix} \tag{30}$$

Equation (30) contributed to the creation of an accurate simulation of the blimps motion and is be used in the localization algorithm that will be discussed in Section 3.2.

2.3 Control System Design

To enable the RC blimp to function autonomously, it had to be outfitted with new electronics to supplement the components pre-installed on the blimp in order to facilitate navigation and trajectory-planning. Since small blimps have a limited payload, a primary consideration in the selection of control system components had to be weight. Weight had to be kept to a minimum so that all necessary components can be mounted on the blimp without interfering with the blimp's performance. Higher payloads can be expected to cause significant reductions in flight time and maneuverability.

2.3.1 RC Blimp Electronics

The BlimpGuys © blimp is controlled by three motors with plastic propellers. The motors are each controlled by an electronic speed controller (ESC) which has a PWM input in the range of 0-5V to command the motors. Based on the PWM signal, the ESC outputs current-controlled power to control the speed of the motors, which run at 8-13V. The ESC excepts RC servo PWM, which has a standard pulse width between 1ms and 2ms, with neutral being 1.5ms. The PWM signals are relayed to the ESC by a Spektrum 2.4GHz radio transmitter and receiver, which communicates with a controller on the ground.

2.3.2 Control System Configuration

Several component combinations have been proven to enable autonomous flight for small, indoor blimps. These options all belong to one of two primary configuration categories: fully onboard processing or processing offloaded to a ground station. Within the ground station configuration, several additional options exist pertaining to the method of communication between the blimp and the ground station.

The most simple configuration option is to have the entire control system onboard the blimp, including all processing. In this case, all processing will be done by a development board, meaning that the processing power will be limited by the performance of this board. For some navigation systems, this is sufficient, whereas with more advanced navigation systems such as some computer vision applications, latency in the processor may decrease the accuracy of the control system and cause the blimp to become unstable. However, the advantage of this system is the simplicity in the configuration as no communication between the blimp and other entities is necessary for the blimp to navigate and be controlled. This makes expanding the system to a multi-blimp cooperative system easier, as there are fewer radio signals to differentiate. Additionally, this eliminates any concern of lack of control over the blimp in-flight due to signal downage.

The alternative to onboard processing is to offload this processing to a ground station, with communication occurring between the ground station and the blimp in order to control the blimp. In this case, the blimp would transmit sensor outputs to the ground station for processing, and the ground station would transmit commands for motors back to the blimp as a response. With the processing occurring on the ground, the processing device is no longer weight-limited, meaning that it can be much more powerful. This has the potential of decreasing the processing time, especially for more advanced navigation systems, but it also results in an additional time delay for the communications to occur between the blimp and the ground station which may negate the time gained from the processing speed. Additional work is also required to ensure safety if the communication signal is lost and commands cannot be sent to the blimp.

There are multiple configurations possible to enable communication between the blimp and the ground station. Different radio signals have different frequencies, and this influences how quickly information can be transmitted. On the high end, WiFi connections are very high frequency, either 2.4 or 5 GHz, and would allow for fast data transfer between the blimp and the ground station. The primary issue with this method is that our testing location is on the WPI campus. There are regulations on what devices can use the existing WiFi and who is allowed to connect new networking equipment. While the easiest and fastest communication method would be setting up an access point exclusive to the blimp and ground station for them to communicate over, this is prohibited at the testing location and special permission would have to be obtained. It is also possible to communicate over the existing WiFi access points, which requires both the ground station and the blimp development board to be registered on the WPI network. While this adds additional setup, the WiFi access points are already high speed, allowing for efficient communication between the blimp and the ground station. In this configuration, however, static IP addresses cannot be created for the devices as they can for the personal access point, meaning that the code to support the communications has to be more complex. The regulations around WiFi communication can be circumvented by using a lower frequency radio signal, such as LoRa radio, which uses sub-gigahertz frequency bands. This method is easy to use, without the complex code required for existing WiFi access points, but is slower, which will increase the time delay presented by including a ground station. The advantages and disadvantages of these communication methods are summarized in Table 4. Depending on the communication schema selected, additional components may be required onboard the blimp in order to receive and transmit radio signals, if this transceiver is not already available on the selected development board.

Table 4: Communication Method Analysis Summary.

Communication Method	**Advantage**	**Disadvantage**
WiFi via WPI access points	High speed	Increased setup, no static IP addresses
WiFi via personal access points	High speed, easy setup	Requires special permission
Lower frequency radio signal	Easy setup, unregulated	Slower than WiFi

In order to ensure access to the necessary processing power to navigate and control the blimp, the team opted for a system with an optional ground station to handle sensor processing, which relies on WiFi communications between the blimp and the ground station. Given the uncertainty of WPI's approval to communicate over a personal access point, that option was left open while testing began with communicating over the existing WiFi network. In addition to ensuring fast processing speeds, this configuration allows operators to more closely monitor the control system and its response during flight testing. This helps ensure that the blimp remains in control and allows the team to make real-time adjustments if the blimp does not behave as expected in response to the control inputs.

2.3.3 Control System Component Trade Study

Since the blimp is limited to a 226g payload, nominally, an emphasis was placed on identifying lightweight components when selecting control system electronics in the trade study. Additionally, low power consumption was identified as a highly desirable trait since it enables the use of lighter batteries in order to fly the blimp for long periods of time. While lightweight electronics are necessary in this application due to the payload limitation, there is often a tradeoff between weight and performance which must be assessed.

Navigation Sensor

Several different navigation methods were considered throughout the trade study, each requiring different combinations of sensors. The first method considered was GPS, which utilizes signals from satellites to compute the location of a GPS receiver. This can be a very accurate system and there are many options available for lightweight GPS receivers. However, GPS signals are unreliable indoors, which is the environment in which the blimp will operate for this project, so GPS was determined to be unsuitable. Another navigation method considered was using acceleration data to navigate the blimp. This data can be gathered by an inertial measurement unit (IMU), which is common and lightweight, and is often included on development boards. Dead reckoning with acceleration data can introduce high degrees of uncertainty, but when combined with another data source using an EKF, it can result in very accurate localization. The third method that was considered was vision-based navigation using one or more cameras. There are many different localization methods using camera images, as discussed in Section 1.3.2, but in general this method is well-suited for indoor use since it is easy to create markers or tags in indoor environments, and the layout of rooms often stays constant. There are many lightweight cameras available and it is important to consider their data transmission rate, focus range, and resolution as these factors affect the efficiency and accuracy of vision-based navigation systems. The advantages and disadvantages of these navigation methods are summarized in Table 5.

Based on these options, the team selected a combination of a vision-based and inertial-based navigation system, in order to ensure that the localization of the blimp is accurate. In order to implement this navigation system, an IMU is needed to measure accelerations and angular rates, and a camera to capture the surroundings of the blimp. In order to obtain the position and pose of the blimp using the camera, the AprilTag system was be used, with the positions of AprilTags stored in the control software. This system was selected due to its ease of use and reliability. For the IMU, the Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 IMU was selected due to its advanced capabilities and low weight and the Raspberry Pi Camera v2 was selected for the vision-sensor due to its low weight and ease-of-use.

Development Board

A development board acts as the processor onboard the blimp and connects all other components. There are many different lightweight options for development boards available, and there is significant variation in the available features and ports. For this application, it was essential that the board have necessary compatibility with the selected navigation sensors, and enough pins to accommodate all other

Table 5: Navigation Method Analysis Summary.

Navigation Method	**How it Works**	**Advantages**	**Disadvantages**
GPS	Utilizes signals transmitted by satellites to calculate a location	Very accurate, many lightweight options	Unreliable indoors
IMU	Continuous tracking of inertial data (generally from an accelerometer, a gyroscope, and a magnetometer) can be used to track cumulative change in location and orientation	Lightweight, often provided on development boards	Position tracking with accelerometers can cause high degrees of uncertainty, gyroscopes generally have bias drift
Vision-Based	By analyzing objects in images, location and orientation can be tracked	Lightweight options, many easy to implement systems for position and pose tracking	Many lightweight cameras have very slow data transmission, many cameras require special data ports

components, including the existing components on the blimp such as the electronic speed controller. It was also preferable, though not necessary, that the board contain an IMU, since that would eliminate the need to add a separate IMU which would increase the weight of the control system. Since WiFI was selected as the ground communication method, it was also preferable for the selected board to have WiFi connectivity. The Raspberry Pi Zero W development board was selected due to its low weight and multitude of connection options.

2.3.4 Power System Component Trade Study

With the added components, consideration needed to be paid to how to power them. In order to add a power system, the team had to add several different components to provide and control the power. Attention still had to be paid to weight, but in the case of adding a battery, the weight of the battery is often directly correlated with the battery capacity, such that heavier batteries can power the same components for longer. While ultimately a higher capacity battery may be desirable to increase the flight time of the blimp, the team elected to use the battery that came with the blimp since it was compatible and there were no anticipated long-duration flight tests.

Buck Converter

A buck converter is a necessary component in order to regulate the voltage of the battery down to the appropriate level for the development board and other electronics. Specifically, the converter needs to take the 6V of the battery and convert it to 5V, the operating voltage of the Raspberry Pi development board. Since most buck converters are designed to convert 5V down to 3.3V, the only breakout board rated for 5V to 6V conversion we were able to identify was the MPM36020 5V Buck Converter Breakout. Since this component is so small, it was deemed unnecessary to find comparable chips that were not already installed on breakout boards to minimize weight.

Fuel Gauge

A fuel gauge is a beneficial addition to the power system to monitor the power level of the battery, so that the blimp can send operators a notification when the level approaches one too low for the electronics to be operated properly, which will indicate that the blimp needs to land. When searching for this component, the team found that large electronics vendors do not sell fuel gauges for 6V Ni-Cd batteries such as the one used on the blimp, only for single-cell LiPo batteries. Instead, a custom fuel gauge was designed, based conceptually on the discharge curve of a Ni-Cd battery, shown in Figure 4, which stays in a very small voltage range for the majority of the discharge, and has a sharp drop-off at the end of the discharge. While the voltage of the battery in Figure 4 has a different voltage than the battery used in the blimp, it can be extrapolated that a similar result can be found with a 6V Ni-Cd battery, where the voltage remains within a very small range for the majority of discharge and then drops off as the end of the discharge is reached. Thus, to identify when the battery is critically low, the voltage of the battery can be tracked, with a threshold for a low battery set a small amount below the nominal voltage of the battery.

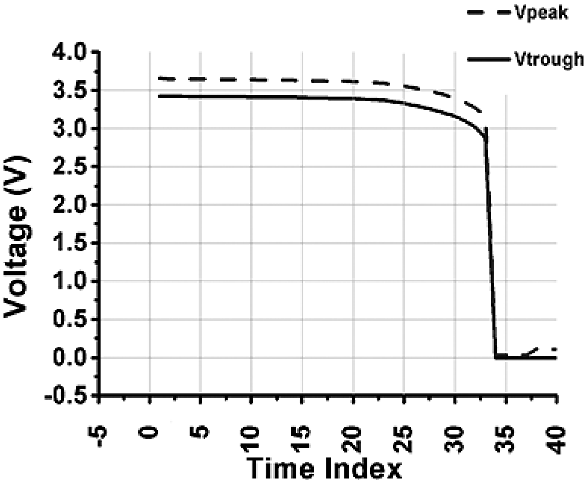


Figure 4: Discharge curve for 3.6V Ni-Cd battery [24].

.

To measure the voltage of the battery from the Raspberry Pi development board, a voltage divider and an analog-to-digital converter (ADC) must be used. The ADC converter is used to convert the analog voltage into a digital signal, since the Raspberry Pi does not have any analog inputs, only digital. The voltage divider is used to shift the voltage of the battery down to an acceptable range for the Raspberry Pi input pins, which can handle a maximum of 3.3V. A linear regulator was also added to the circuit to provide a stable low-noise voltage reference to the ADC in order to make the data received from the fuel gauge more reliable.

To select the resolution necessary for the ADC, the discharge curve in Figure 4 was used as a reference. Since the voltage level for the majority of the discharge is within 0.2 V, it is desirable to have 64 points of data within this range. To determine the number of bits required, b, was found by solving for it in the equation

$$\frac{0.2}{64} = \frac{V_{ref}}{2^b},$$

where V_{ref} is the reference voltage. The number of bits, b must be a whole number, so if it is a fraction it should be rounded up to the nearest whole number. Assuming a 4.2V reference voltage, this corresponds with a minimum requirement of 11 bits of precision for the ADC. An MCP3201 2.7V 12-Bit A/D Converter was selected based on its acceptable resolution, low price, and availability. With a reference voltage of 4.2V, the maximum input voltage must be 4.2V, meaning that at 6V input, the voltage divider must output less than or equal to 4.2V. A voltage divider is made up of two resistors, as shown in Figure 5. The relation of the output voltage, V_{out}, to the input voltage, V_{in} is then related by

$$V_{out} = V_{in}\frac{R_2}{R_1 + R_2}.$$

This allows for many combinations of the resistor values, R_1 and R_2, but based on standard resistor values, $1.47k\Omega$ was selected as R_1 and $2.94k\Omega$ was selected as R_2. This results in a maximum voltage of 4.0V.

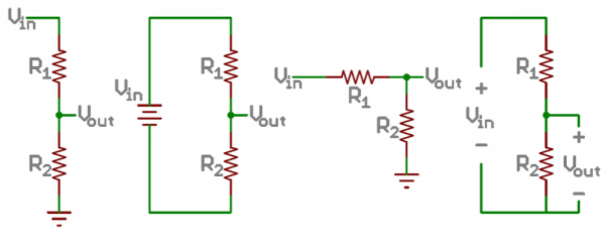


Figure 5: Generic schematic of a voltage divider [25].

For the linear regulator, an MCP1700 low drop out linear regulator was used due to its appropriate compatibility, low price, and availability. In addition to the resistors necessary for the voltage divider, several capacitors were added to help stabilize voltage and reduce noise. The entire design is shown in the schematic in Figure 6. The components were assembled on a perfboard in order to avoid heavier electronic prototyping boards as well as ordering custom-made PCBs.

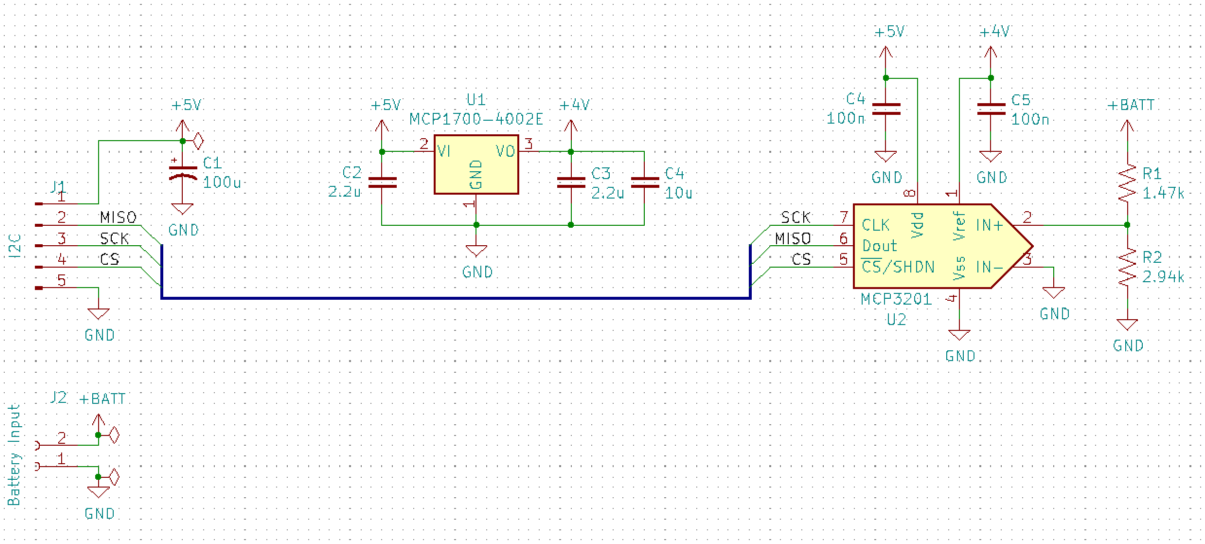


Figure 6: Custom fuel gauge circuit schematic.

Level Shifter

Since the ESCs take 0-5V PWM signals, level shifters will be used to convert the 0-3.3V PWM signals that the development board can output to 0-5V to input to the ESCs. While level shifters are not strictly necessary, since the output signal voltage of the Raspberry Pi development board is 3.3V and the ESCs require a minimum high voltage above 3V, including them will ensure that if the output voltage of the development board drops for any reason, the ESCs can continue functioning properly. Due to a lack of selection of in-stock level shifters, the SparkFun Bi-Directional Logic Level Converter was selected.

2.3.5 Complete Control System Configuration

For the onboard control system configuration the team decided to use the Raspberry Pi Zero W development board with a weight of 9.3g, the Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055 IMU that has a weight of 2.5g, and the Raspberry Pi Camera Board v2 with a weight of 3.4g. A cable was required to connect the Raspberry Pi Camera Board v2 to the Raspberry Pi Zero W development board,but the weight of this cable was negligible. For the power system, the team used a custom fuel gauge with a weight of 11.8g, a Sparkfun Bi-Directional Logic Level Converter with a weight of 1.6g and an MPM36020 5V Buck Converter Breakout with a weight of 1g. The cumulative weight not including the wiring and mounting hardware is 28.8g. This weight breakdown as well as the price breakdown is shown in detail in Table 6.

Table 6: Weight and Price Budget.

Item	Product Price	Product Weight (g)
Raspberry Pi Zero W development board	$10.00	9.3
Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout -BNO055	$19.95	2.5
Raspberry Pi Camera Board v2	$29.95	3.4
Sparkfun Bi-Directional Logic Level Converter	$2.95	1.6
MPM36020 5V Buck Converter Breakout	$5.95	1
Custom Fuel Gauge	$6.05	11.8
Total	$74.85	29.6

3 Blimp Control

3.1 Overall Control Schema

The inputs for the control logic come from the IMU, which has an accelerometer, gyroscope, and magnetometer, and the Raspberry Pi camera. Since the experiment is being conducted indoors, magnetometer readings will not be used. This choice was made because magnetometers are sensitive to the magnetic fields created by electrical wiring or devices found indoors and would likely report inaccurate measurements. Once the sensor inputs have been recorded, the accelerometer readings will be integrated to form velocity measurements. Additionally, the IMU automatically derives the Euler angles using the gyroscope measurements. Therefore, measurements will be available for the acceleration vector, $a = [\dot{u}\ \dot{v}\ \dot{w}]^T$ and the velocity vector, $V = [u\ v\ w]^T$ from the accelerometer; and the Euler rates, $[p\ q\ r]^T$, and the Euler angles $[\phi\ \theta\ \psi]^T$ from the gyroscope. By using AprilTags, the position vector, $P = [x\ y\ z]^T$ and the Euler angles can be computed using images captured from the camera. These measurements will then be used for a process of localization, as shown in Figure 7 which will be connected to a process of path planning and following, and an LQR controller in order to send signals to the blimp motors.

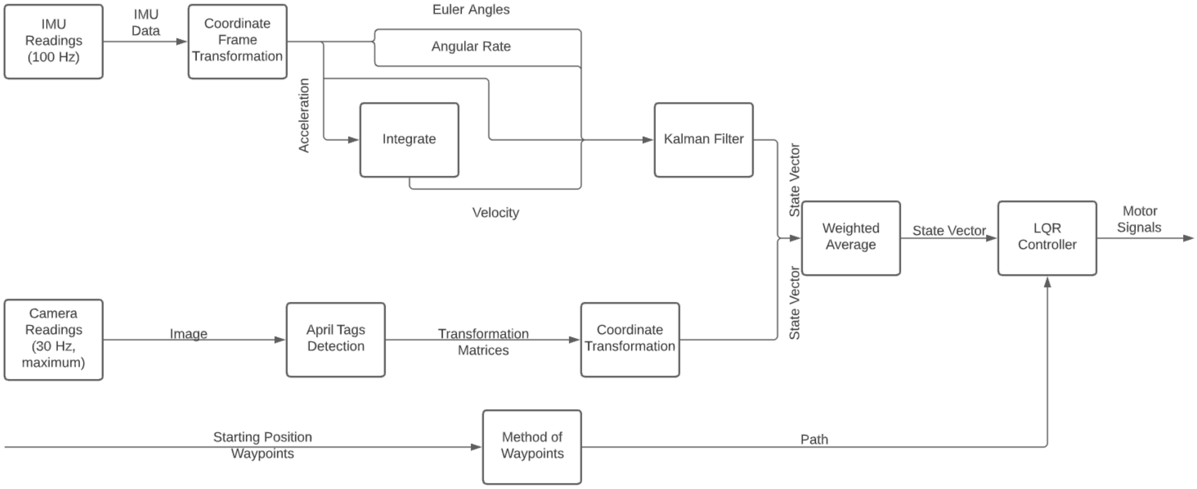


Figure 7: Control logic diagram.

3.2 Localization

The control of the blimp is dependent on the correct determination of the blimp's state at any point in time. In order to localize the blimp, the IMU data will be processed through the continuous-discrete extended Kalman filter (CD-EKF) algorithm described in more detail later in this Section. Since the BNO055 IMU can automatically compensate for gyroscope drift, the CD-EKF has been deemed sufficient to filter the data without any additional algorithms. Concurrently, the Raspberry Pi will process position and orientation data calculated from any AprilTags that are in the camera's field of view. By using the known positions of the AprilTags in the testing area, the transformation matrices included within the AprilTags software library can be used to obtain the world coordinates of the blimp. The translational and rotational data constructed from the IMU and Raspberry Pi camera are used as state vectors and a weighted average between the two will be used as an input for the LQR controller. The orientation data generated from the method of waypoints will also be used to tune the controller during flight and properly navigate the blimp along the desired path.

3.2.1 IMU Transformation

The IMU cannot be located at the center of mass of the blimp, since this is located inside the envelope, whereas the IMU must be located in the gondola. Thus, in order to obtain the accelerations, angular rates and Euler angles of the center of mass of the blimp, the sensor measurements must undergo a transformation. From [26], the transformation for the gyroscope readings are defined as

$$\begin{aligned} p &= p_i - \psi_{IMU} q_i + \theta_{IMU} r_i, && (31) \\ q &= \psi_{IMU} p_i + q_i - \phi_{IMU} r_i, && (32) \\ r &= -\theta_{IMU} p_i + \phi_{IMU} q_i + r_i, && (33) \end{aligned}$$

where p_i, q_i, r_i are the components of angular velocity indicated by the rate gyros about the IMU's axes, ψ_{IMU}, θ_{IMU}, ϕ_{IMU} are the Euler angles of the IMU measured with respect to the body axes of the blimp. Likewise, from [26], the transformation for the accelerometer readings are defined as:

$$\begin{aligned} gA_{X,cg} &= g(A_{X,i} - \psi_{IMU} A_{Y,i} + \theta_{IMU} A_{Z,i}) + (q^2 + r^2)\bar{x}_{IMU} - (pq - \dot{r})\bar{y}_{IMU} - (pr + \dot{q})\bar{z}_{IMU}, && (34) \\ gA_{Y,cg} &= g(\psi_{IMU} A_{X,i} + A_{Y,i} - \phi_{IMU} A_{Z,i}) - (pq + \dot{r})\bar{x}_{IMU} + (p^2 + r^2)\bar{y}_{IMU} - (qr - \dot{p})\bar{z}_{IMU}, && (35) \\ gA_{Z,cg} &= g(-\theta_{IMU} A_{X,i} + \phi_{IMU} A_{Y,i} + A_{Z,i}) - (pr - \dot{q})\bar{x}_{IMU} - (qr + \dot{p})\bar{y}_{IMU} + (q^2 + p^2)\bar{z}_{IMU}, && (36) \end{aligned}$$

where $A_{X,i}, A_{Y,i}, A_{Z,i}$ are the components of acceleration indicated by the accelerometers about the IMU's axes, $\bar{x}_{IMU}$, $\bar{y}_{IMU}$, $\bar{z}_{IMU}$ are the coordinates of the IMU measured from the body axes of the blimp, and $\dot{p}$, $\dot{q}$, $\dot{r}$ are the angular accelerations about the body axes of the blimp. With the inclusion of the derivatives of the angular rates, an additional estimation is incorporated into the system, since this must be calculated using the equations of motion derived for the system, as this value is not being directly measured.

3.2.2 The Kalman Filter

The Kalman Filter is an algorithm developed to produce the most accurate estimate of a system at any point in time. This algorithm makes the assumption that the system being studied is linear and any measurement noise follows a Gaussian distribution. Sensor imperfections and external disturbances are common effects that can cause inaccurate data to be collected. The Kalman Filter eliminates this problem by combining many measurements into a weighted average. This creates a best estimate of a system by assigning greater weights to measurements that come from more accurate sensors but still includes all available measurements. The Kalman Filter can be modified to include a model of the system dynamics and sensors with non-Gaussian noise distributions to enable the estimation of more complex, nonlinear systems or systems that can only use indirect measurements of a desired system property. A common application of this type of filter is in the guidance, navigation, and control of aircraft through the creation of precise position and heading estimates[27].

The structure of a Kalman Filter can be divided into two components: prediction and update. The prediction uses a measurement and its accompanying noise characteristics to create an estimate of the system. The filter then assigns a weight to this estimate representing the probability of it being the true system state. The update occurs when a new measurement is received and must be incorporated into the estimate generated previously. The filter weighs the confidence in the estimate and measurement and fuses the two to create the next estimate. The goal of the Kalman Filter is to arrive at a highly accurate estimate of the system.

For this project, the Kalman Filter will use noisy sensor data and noise characteristics to create an estimate of the blimp's position, velocity, and orientation. These states must be accurate as they are necessary for the navigation of the blimp along a chosen path. The governing equations of the blimp system are comprised of nonlinear equations of motion. Since the Kalman Filter is only designed for linear systems, an Extended Kalman Filter must be used instead.

3.2.3 Extended Kalman Filter

The Extended Kalman Filter (EKF) uses the same concept of estimation and sensor fusion as the Kalman filter, but is applied to nonlinear systems. The EKF linearizes the equations for the system dynamics model by creating Jacobian matrices from the system model equations. Jacobian matrices are matrices made up of the partial derivatives of each equation with respect to the variable they are a function of. This process creates a linear approximation of the system at a point in time so that the Kalman Filter process can continue as designed. The EKF process uses state measurements and sensor error to provides new estimates about the system state at each time step[27]. The state and measurement are defined as:

$$X_k = f(X_{k-1}, W_k), \tag{37}$$

$$z_k = h(X_k) + V_k. \tag{38}$$

In this case, the state vector X_k contains the states defined in Equation (28). The system state at a given time, X_k, is a function of its previous state with the addition of some processing noise. The observation, z_k, is a sensor measurement of the state, X_k, that also includes some sensor noise. The sensor noise is assumed to be of a Gaussian distribution. The first step of the Kalman Filter is to predict the next state. The *prediction step* is defined as the following:

$$\begin{aligned} \hat{X}_{k|k-1} &= f(\hat{X}_{k-1|k-1}, 0), \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + G_k Q_v G_k^T. \end{aligned}$$

The prediction consists of the evaluation of the nonlinear system model with the previous state as the initial conditions. This provides the prediction for the system state at the current time, $\hat{X}_{k|k-1}$, prior to the inclusion of any measurements. The error in this estimate, $P_{k|k-1}$, is the sum of the error in the previous estimate and the error intrinsic to the sensors' ability to accurately respond to any inputs to the system. Here, F_k and G_k are the Jacobian matrices of the system state equations and system input equations, respectively. These matrices are also commonly known as A and B in other texts [28].

Next, using the measured state data, the Kalman filter has the following *correction step*:

$$\begin{aligned} K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + Q_v)^{-1}, \\ \hat{X}_{k|k} &= \hat{X}_{k|k-1} + K_k (Y_k - h(\hat{X}_{k|k-1})), \\ P_{k|k} &= P_{k|k-1} - K_k H_k P_{k|k-1}. \end{aligned}$$

The correction step of the filter is the reconciliation of the predicted state as governed by the system model, and the measurements received from the sensors about the current state. The Kalman gain, K_k, is the weight attributed to new measurements as decided by the filter's error in its prediction and the sensor error. Weighing the prediction and new measurement with the scaling provided by the Kalman gain results in the optimal estimate of the current state, $\hat{X}_{k|k}$. The error of this optimal estimate, $P_{k|k}$, is recalculated with the Kalman gain and changes according to how accurate the prediction and sensors were in reporting the system state. The goal is for $P_{k|k}$ to become very small over time as the filter becomes more accurate at providing an optimal estimate of the system. The definitions of the matrices used for the Kalman filter with this project's specific model can be found in Appendix B.

3.2.4 Continuous-Discrete Extended Kalman Filter

The CD-EKF can be used instead of the EKF to allow estimation for continuous-time nonlinear models. This modification enables greater flexibility as the EKF assumes a discrete-time model. The CD-EKF replaces the prediction step of the EKF with the following from [29]:

$$\begin{aligned}\frac{dm}{dt} &= f(m,t),\\ \frac{dP}{dt} &= R(P,t),\\ &= F(m,t)P + PF(m,t)^T + G(t)Q(t)G(t)^T,\end{aligned}$$

where

$$\begin{aligned}m(t_{k-1}) &= \hat{X}_{k-1|k-1},\\ P(t_{k-1}) &= P_{k-1|k-1},\\ F(m,t) &= \frac{\delta f}{\delta X}(m,t),\end{aligned}$$

such that the mean and covariance are ordinary differential equations that are integrated from t_{k-1} to t_k. Then, the state estimate and the covariance of this estimate become

$$\begin{aligned}\hat{X}_{k|k-1} &= m(t_k),\\ P_{k|k-1} &= P(t_k).\end{aligned}$$

The update and correction components remain the same as in the EKF.

3.2.5 Sensor Fusion

The blimp state measurements will be received from two different devices, the IMU and Raspberry Pi camera. These devices both measure different system states with different errors. To resolve this, a weighted average fusion of the estimates, based on the value of the confidence in the estimate, will be used. This can be done with the equation from [30]:

$$\hat{x}_j = \sum_{i=1}^{n} \alpha_i W_k^i v_j^i, \tag{39}$$

where α is the fusion weight, defined by

$$\alpha_i = \frac{1}{\sigma_i^2 \sum_{i=1}^{n} \frac{1}{\sigma_i^2}}. \tag{40}$$

σ_i is the covariance of the measurement, W_k is the optimal proportion of weight defined by

$$W_k = \frac{\hat{x}_k}{(z + k + \hat{x}_k)}, \tag{41}$$

and v_j is the sum of the observational values and the state estimated values. This fusion of measurements will help offset errors from either sensor and make the measurements, and therefore the controller, more robust.

3.3 Linear-Quadratic Regulator Controller Design

Once the IMU and camera data has been collected and filtered, a controller must be designed to use this information. The act of controlling a system's actuation to reach a desired state is the core of any autonomous vehicle. These controllers range from simple algorithms, such as Bang-Bang control, to more complex forms. The choice of control scheme must be carefully made by weighing each controller's strengths with the preparations and models necessary for their implementation. The Linear-Quadratic Regulator (LQR) is a control method designed to balance two criteria: the error between current and desired states and the work necessary to achieve the desired state. LQR is intended to minimize the cost function defined in Equation (42) [31].

$$J = \int_0^\infty (X^T Q X + U^T R U) dt. \tag{42}$$

The design of this controller can be summarized by the Q and R matrices. The entries in these matrix are penalty weights assigned to to the presence of non-zero states and non-zero inputs, respectively. Each entry is to be designed for the application with regard to the constraints of the real world system. High values can be used as entries in the R matrix to ensure that physical limits in the real world system are adhered to such as the maximum propulsion allowable by a motor. Similarly, entries in the Q matrix can be selected so that certain states are corrected faster than others. This is critically important for applications that involve devices operating near humans as safety requirements more important than device performance. The weights in these matrices are relative to other entries within each matrix. That is to say that the magnitude of the penalty weights are related to their relative importance and not necessarily the absolute limit of its associated state or input. The strength of LQR is that it is by definition the best controller for minimizing error while also minimizing the work necessary to achieve a state without error. This controller is more limited than other control laws in that it is assumed that the system dynamics are linear and all system states and inputs can be accurately determined at all times. These are represented in Equation (42) by X, the state matrix, and U, the input matrix. Both matrices for this project can be found in Appendix B. For some systems, exact knowledge of the current state is not feasible but the implementation of a Kalman Filter can create an optimal estimate of the system and often provides the information necessary for LQR to function. LQR also faces another limitation in that the Q and R choices must be used to find a unique, positive, semi-definite solution to the Ricatti equation shown in Equation 43.

$$0 = A^T P + PA + Q - PBR^{-1}B^T P. \tag{43}$$

Due to the matrix inversions necessary to solve the Ricatti equation, not all Q and R matrices will provide a valid stabilization matrix. This results in no gain matrix being found for the chosen values of

Q and R. Tuning the Q and R matrices is an iterative process to find the correct balance of weights that results in a desirable system response.

For the purposes of the blimp, pitch and roll are not controllable, and as a result they are not included in the LQR control calculations. This reduces the A and B matrices in the Ricatti equation to an 8x8 matrix and an 8×3 matrix respectively. Q becomes an 8×8 matrix and R is a 3×3 matrix. The computations for these matrices and their resultant gain matrices can be found in Appendix C. Since the blimp stays at a fixed height, the two control gains that were computed were for a straight path and a level turn. Where a level turn is when the blimp rotates on the x-y plane to change the current yaw angle of the blimp.

3.4 Gain Scheduling Law

Two separate gain matrices have been calculated and a gain scheduling law will be used to dictate which control gain to use. The gain matrices will be annotated as such, K_{sp} for straight path and K_{lt} for level turn. K_{sp} is designed to propel the blimp forward while K_{lt} changes the yaw or heading angle. Using these two gains we can write the scheduling law for matrix K_{gs} [22],

$$K_{gs} = (1 - \sigma)K_{sp} + \sigma K_{lt}. \tag{44}$$

where σ is a scheduling parameter and is between the values of one and zero. When σ is zero the blimp flies in a straight path, whereas when σ is one the blimp turns at its maximum yaw turn radius. To calculate the scheduling parameters, two new constants will be introduced, a_1 and a_2. The calculation of the scheduling parameters are derived from the equation

$$\sigma = a_1|\dot{\psi}| + a_2. \tag{45}$$

Since the end conditions for σ are bounded to either 0 or 1, we will create two new equations from Equation (45) representing these conditions.

$$a_1\dot{\psi}_{sp} + a_2 = 0, \tag{46}$$

$$a_1\dot{\psi}_{lt} + a_2 = 1. \tag{47}$$

After isolating each term and then solving Equations (46) and (47), the variables a_1 and a_2 can be written in terms of $\dot{\psi}_{sp}$ and $\dot{\psi}_{lt}$,

$$a_2 = \frac{\dot{\psi}_{sp}}{\dot{\psi}_{sp} - \dot{\psi}_{lt}}, \tag{48}$$

$$a_1 = -\frac{1}{\dot{\psi}_{sp} - \dot{\psi}_{lt}}. \tag{49}$$

Replacing a_1 and a_2 in Equation (45) by using Equations (48) and (49),

$$\sigma = -\frac{1}{\dot{\psi}_{sp} - \dot{\psi}_{lt}}|\dot{\psi}| + \frac{\dot{\psi}_{sp}}{\dot{\psi}_{sp} - \dot{\psi}_{lt}}. \tag{50}$$

The yaw command rate for a straight path, $\dot{\psi}_{sp}$, is zero, therefore

$$\sigma = \frac{|\dot{\psi}|}{\dot{\psi}_{lt}}. \tag{51}$$

Since $\dot{\psi}_{lt}$ is the maximum command yaw rate for a level turn, a saturation limit will be imposed. This forces σ to never grow larger than one. The saturation limit is represented by

$$\sigma \equiv \text{sat}\left(\frac{|\dot{\psi}|}{\dot{\psi}_{lt}}\right) \equiv \begin{cases} 1 & |\dot{\psi}| > \dot{\psi}_{lt}, \\ |\dot{\psi}|/\dot{\psi}_{lt} & |\dot{\psi}| \leq \dot{\psi}_{lt}. \end{cases} \tag{52}$$

3.5 Path Planning

A controller has been designed but a path must still be generated. The A* method was discussed in Section 1.3.2 and provides an efficient path through an obstacle field [13]. Since the goal of this project is to have the blimp monitor a room, navigation through an obstacle field can be simplified to navigation between waypoints with clear paths between each waypoint. To enable long term monitoring of the room, the path must create a loop. For this experiment, Alden Hall has been chosen as the testing facility and waypoints are used to set the path of the blimp. In order to simplify calculations, path planning is split into two domains: the horizontal plane and direct vertical plane. The blimp will be set at a certain altitude and is neutrally buoyant. Therefore, there will be no need to control the movements in the vertical plane. To navigate the horizontal plane, a track-specific guidance law will be used to navigate the blimp between each waypoint.

3.5.1 Testing Path

Six waypoints are set within Alden Hall, and will be spread out as indicated in Figure 8. The waypoints were set so the entire room can be monitored. The blimp starts at waypoint A. It then travels in a straight path to waypoint B. When the blimp is in proximity of waypoint B, it initiates a level turn. Once it completes the level turn it then flies straight toward waypoint C. The blimp will continue the path alphabetically until waypoint F is reached. Once the blimp has reached waypoint F, the blimp will make its way back to waypoint A by revisiting points C and D. Since the blimp is monitoring the room the path will be repeated until stopped.

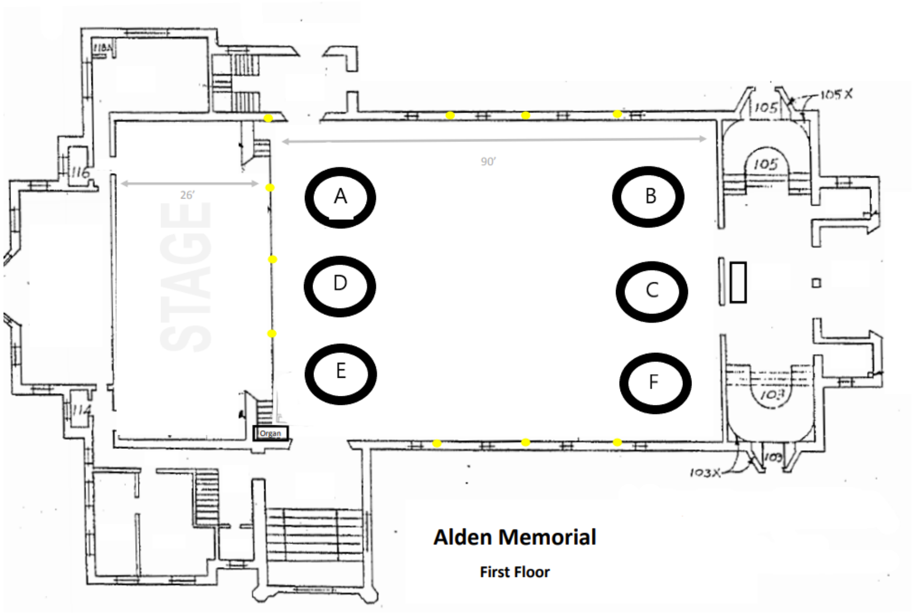


Figure 8: Arrangement of waypoints in Alden Hall.

3.5.2 Track-Specific Guidance Law

The track-specific guidance law is used to determine the yaw angle necessary for the blimp to turn toward the next waypoint. In Equation (53), B_y and A_y represent the respective waypoint's y-position and B_x and A_x represent the x-position.

$$X_{geo} = \arctan\left(\frac{B_y - A_y}{B_x - A_x}\right). \tag{53}$$

The geometric heading remains constant for each pair of waypoints since they are stationary. Using the airship's ground speed, the heading, X_a is calculated as

$$X_a = \arctan\left(\frac{\dot{y}}{\dot{x}}\right). \tag{54}$$

In Equation (57), d is the normal distance from the imaginary straight line connecting waypoint A and waypoint B to the blimp.

$$d = ||\boldsymbol{R}_t|| \sin\left(X_{geo} - X_a\right). \tag{55}$$

$||\boldsymbol{R}_t||$ is the distance the blimp has traveled from the previous waypoint. L_{des}, a function of the airship speed, is calculated using the equation

$$L_{des} = ||\boldsymbol{V}||\tau. \tag{56}$$

L_{des} can be thought of as a maximum limit on the error distance, d. The L_{des} term is designed through the use of τ. τ is a scaling factor that is multiplied by the magnitude of the velocity vector to create the desired limit. Although τ is a constant, L_{des} is not since the limit imposed on d must vary to account for different error tolerances during low speed turns or higher speed travel between waypoints. τ is the design factor that decides the aggressiveness of this control law as small values of τ result in small error tolerances and therefore a higher commanded heading. After both headings and error distances are calculated, a desired heading, X_d can be obtained using the equation

$$X_d = \frac{\pi}{2} \tanh \left(\frac{d}{L_{des}} \right). \tag{57}$$

Once X_d is determined, the command yaw angle can be calculated using Equation (58).

$$\psi_{comm} = X_d \tag{58}$$

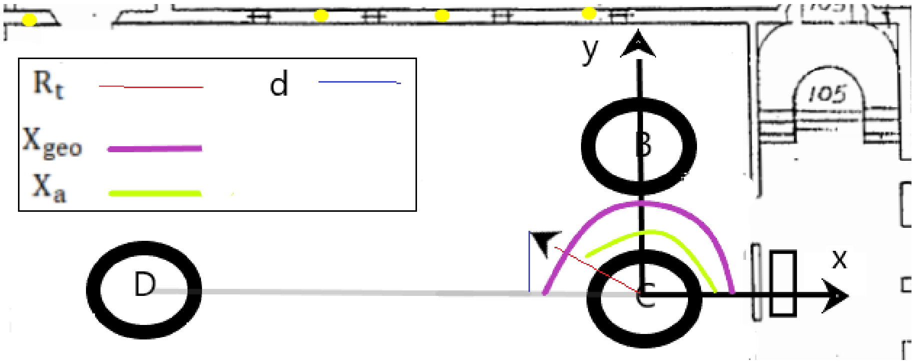


Figure 9: Blimp tracking from waypoint C to D.

Figure 9 is a 2-D representation of the track-specific guidance law from waypoint C to D. This will complete the planned path section.

4 Testing and Verification

The control system is dependent on knowledge of the sensors and their noise characteristics. Incorrect values for these statistics can lead to significant degradation in the performance of the Kalman Filter. Poor filtering and estimation increases the likelihood of the state estimate converging to an incorrect value or the blimp exhibiting unexpected behavior in response to inputs.

4.1 Sensor Covariance and Bias Calculations

In order to implement the Kalman filter and the weighted average sensor fusion, the covariance and bias of the accelerometer, gyroscope, and camera had to be experimentally derived. In the case of the accelerometer, the IMU was aligned such that each axis pointed parallel to the acceleration of gravity. This acceleration measurements were collected with the IMU remaining stationary for an extended period of time. From this data, the bias was calculated by subtracting the expected value from the mean value. The covariance was calculated from this same data. Similarly, the gyroscope was placed on a turntable rotating at a known rate for an extended period of time while collecting data. This was done such that each axis had measurements taken at multiple rates. In the same manner as with the accelerometer, the gyroscope bias was be calculated by subtracting the expected value from the mean, and the covariance was also calculated.

When analyzing the accelerometer and gyroscope data, it was noted that there were several measurements that were extreme outliers, at a consistent value, as shown in Figures 10 and 11. Using the large amount of data, it could be noted that the value of these outliers were very consistent, allowing them to be easily filtered out. Thus, in order to obtain accurate values for bias and covariance, these measurements were filtered out with the assumption that this will also be done when processing IMU data during flight testing, and therefore they do not need to be modelled as part of the bias or noise. For the accelerometer, the value for the outlier measurements was approximately 318 $\frac{m}{s^2}$. The resultant bias and noise data for the accelerometer is displayed in Table 7.

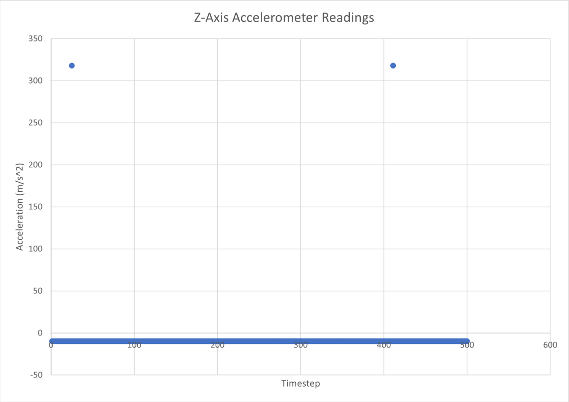


Figure 10: Unfiltered z-axis accelerometer data.

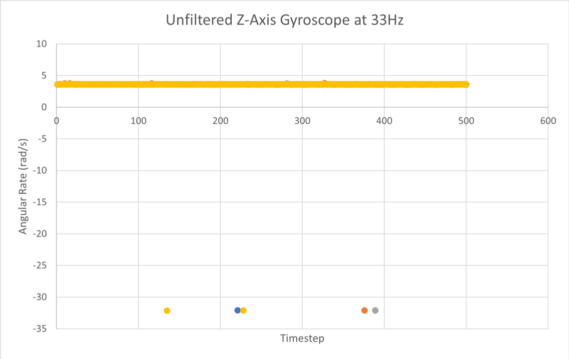


Figure 11: Unfiltered z-axis gyroscope data at 33Hz.

Table 7: Accelerometer Experimental Bias and Noise Measurements.

Axis	**Bias** $\left(\frac{m}{s^2}\right)$	**Variance** $\left(\frac{m^2}{s^4}\right)$
X-Axis	0.1198	0.0029
Y-Axis	0.1836	0.0001
Z-Axis	0.1049	0.0074

The outlier values for the gyroscope were consistent at each different frequency tested, at $33Hz$, the outlier measurements had values of approximately $-32\frac{rad}{s}$, at $45Hz$ the outlier measurements had values of approximately $-30\frac{rad}{s}$, and at $78Hz$ the outlier measurements had values of approximately $-27\frac{rad}{s}$. While this indicates that this is not a constant value but varies based on the actual angular velocity, it can be expected that the blimp won't ever rotate at a speed greater than the about $8.2\frac{rad}{s}$ demonstrated in the $78Hz$ test, so gyroscope measurements with magnitude of greater than $25\frac{rad}{s}$ can safely be filtered out. The resultant bias and noise data for the accelerometer is displayed in Table 8.

Table 8: Gyroscope Experimental Bias and Noise Measurements.

Axis	**Bias** $\left(\frac{rad}{s}\right)$	**Variance** $\left(\frac{rad^2}{s^2}\right)$
X-Axis	0.4316	0.0031
Y-Axis	0.3825	0.0008
Z-Axis	0.2503	0.0381

For the camera, the methodology is less straightforward as there are no common, established practices to calibrate a camera with AprilTags. As a result, the team devised a simple calibration method. The camera would be placed at various poses with respect to an AprilTag and the AprilTag software would then calculate translation and rotation data. This data would then be compared to the known poses to provide the bias and covariance necessary for the sensor fusion calculations. However, this testing could not occur. The Raspberry Pi Zero used for this project is based on an ARMv6 instruction set architecture while the AprilTags software library is written for the more common x86 architecture. These are not compatible and the AprilTags software could not be installed on the processor. There exist emulation techniques that can enable the different architectures to interact but these require higher processing power beyond that of the Raspberry Pi Zero. To resolve this issue, a different processor would have to be sourced and placed on the blimp. The project trade study had already determined that any other processor would exceed the payload capacity of the blimp. Therefore, AprilTags were not used for the blimp in flight.

4.2 Propeller Thrust

To test the thrust of each propeller a set of simple experiments was done. To measure the total thrust produced by the two propellers attached to the gondola, the gondola was secured onto a scale and the propellers were rotated so that the force produced by them would point directly into the scale. To measure the thrust produced by the back propeller, the housing was secured, and a scale was placed under the propeller, as can be seen in Figure 12. The force of the air on the scale when the propeller is turned on is then approximately equivalent to the thrust produced by the propeller. This was experimentally tested by measuring the thrust of the main propellers using the first method mentioned above along with this method. Both resulted in approximately the same values. The scale measures mass, so the measurements had to be converted to force by multiplying by gravity. The tests needed to be conducted for both forwards and reverse thrust for both propellers, as they differed due to the configuration of the motors. The results of the thrust tests are summarized in Table 9.

Table 9: Propeller Thrust.

Propeller	Scale Measurement (g)	Force (N)
Gondola Propellers Forward	68	0.667
Gondola Propellers Reverse	36	0.353
Tail Propeller Right	46	0.451
Tail Propellers Left	24	0.235

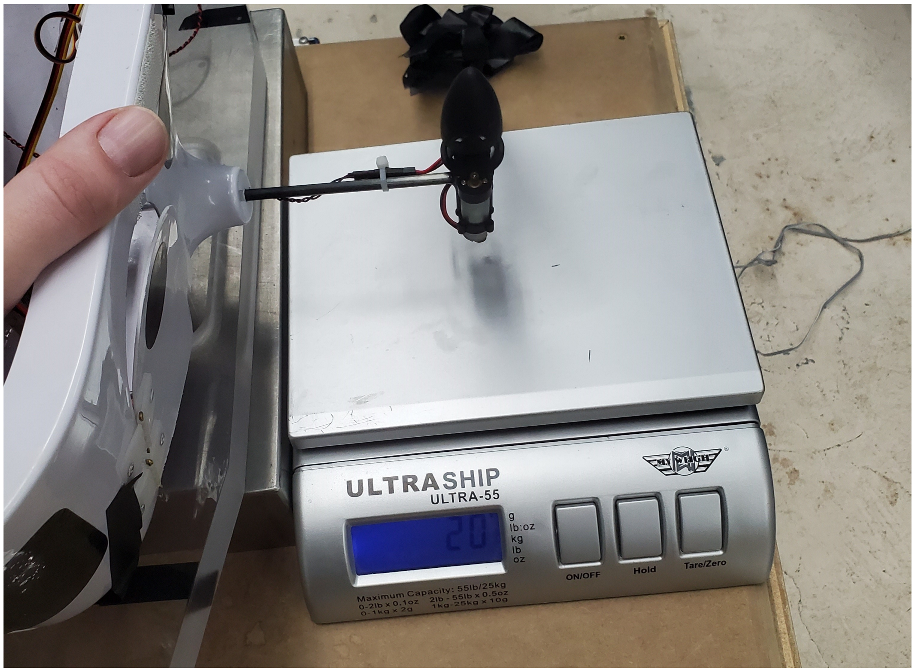


Figure 12: Propeller thrust test.

In the original configuration of the blimp, all of the thrust motors are controlled by the RC controller through an ESC. RC controllers transmit servo PWM, which operates with a minimum pulse width of 1ms and a maximum pulse width of 2ms, where a 1.5ms pulse width is neutral. This is converted by the ESC such that the 2ms pulse width corresponds with the maximum thrust, the 1ms pulse width corresponds with the minimum (negative) thrust and the 1.5ms pulse width corresponds with zero thrust. Using the assumption that this relationship is linearly related in both positive and negative directions, the pulse width, PW can be correlated with the thrusts forward thrust, T_m and T_t:

$$T_m = \begin{cases} -0.706(PW - 1.5) & 1.0ms \leq PW \leq 1.5ms, \\ 1.334(PW - 1.5) & 1.5ms \leq PW \leq 2.0ms. \end{cases}$$

$$T_t = \begin{cases} -0.903(PW - 1.5) & 1.0ms \leq PW \leq 1.5ms, \\ 0.471(PW - 1.5) & 1.5ms \leq PW \leq 2.0ms. \end{cases}$$

4.3 Preliminary Movement Testing

Blimp aerodynamics is not taught in a usual aerodynamics class, so a preliminary flight test with an RC controller was needed to confirm many assumptions made about blimp dynamics. The first tests were done with using the included RC controller. The maximum speed and cruise speed were measured by timing how long the blimp took to travel a distance of 10 feet. The turning radius at the cruise speed was measured by making a 180 degree turn at cruise speed, and measuring the radius of that turn. Both sets of tested were repeated a minimum of 5 times each, and the average values of our test results were used. The turning at maximum speed was too inconsistent when controlling with the RC remote, so no value was measured, but values ranged from about 10 ft to 26 feet. The average results for cruise speed were 3.3 ft/s, or 1.0 m/s.

5 Simulation

Before we could use our designed controller and EKF on the blimp, a Simulink model was made to ensure their efficacy. In the simulation, the accelerometer and gyroscope are modeled using our equations of motion. Since we have the equations for translational acceleration in the body frame, $\dot{u}$, $\dot{v}$, $\dot{w}$, we are able to use those outputs and add noise based on our accelerometer covariance measurements. This allows us to have an accurate representation of the accelerometer in real flight. To model the gyroscope, the equations for the rotational acceleration in the body frame are integrated and noise is added based on our covariance measurements for the gyroscope. We are then able to take the outputs of the modeled accelerometer and gyroscope and integrate the values to get velocity in the body frame and the Euler angles. Using the Euler angles, we rotate the velocity in the body frame to get the velocity in the inertial frame. This value is then integrated to get the position of the blimp in the inertial frame. This process allows us to obtain numeric values for all 8 observed states. These states are then passed through an EKF to remove the noise. The filtered states are used as the initial conditions for the equations of motion to calculate the new translational and rotational accelerations and the state at the next time step. The new states and sent to the LQR controller to find the desired inputs to the motors. To check for any errors in the simulation, a series of MATLAB scripts were created that used the same process as mentioned above. The outputs of the MATLAB script and Simulink model were compared to see if there were any differences. If there were difference, then there was something wrong with either implementation of the simulation or flaws in the definition of the equations of motion or constants used in their calculations. The simulation also allowed a more intuitive viewing of the blimp's movement using Simulink's 3-D animation software. The animation showed the blimp's orientation and speed at all times as well as the path the blimp was attempting to follow.

5.1 Low-pass Filter

Due to time constraints, we weren't able to complete and verify the EKF design for the blimp. Instead, we decided to use a low-pass filter to clean noisy data. A low-pass filter works by removing measurements occurring at frequencies above a certain threshold. To determine the right cutoff frequency for each state, multiple simulations we run and compared to data from an ideal simulation without noise. After multiple simulations, it was determined that the best cutoff frequencies were 10 Hz, 15 Hz, 100 Hz, and 10 Hz for $\dot{u}$, $\dot{v}$, $\dot{w}$, and $\dot{r}$ respectively. These cutoff frequencies had the lowest error. Figure 13 shows the percent error of the X, Y, and Z position.

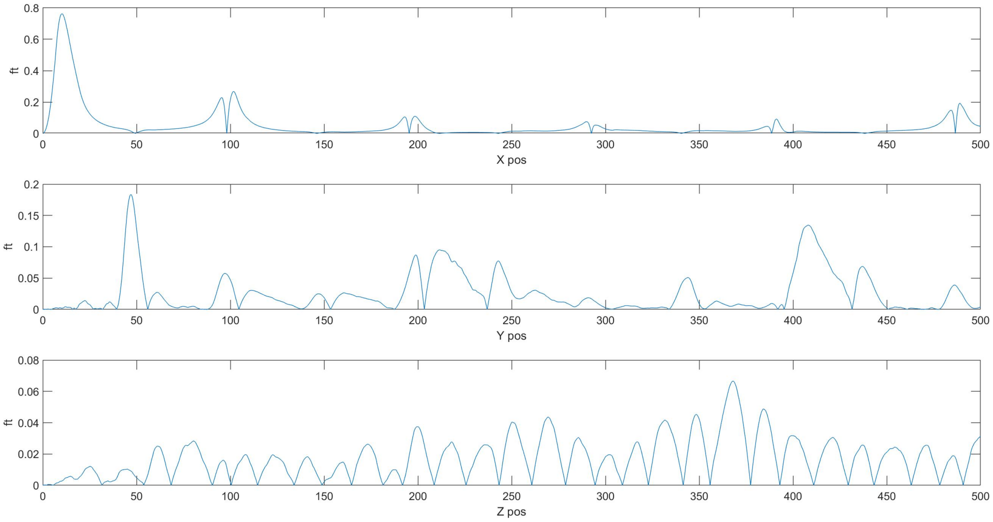


Figure 13: Percent error of the filtered X, Y, and Z states.

As you see there isn't too much of a difference between the filtered position states and the actual position states. There are a few spikes which is caused when the blimp turns but they settle down once the blimps stops turning. To further help filter out noise, a weighted average can be applied to the the real and filtered outputs to get closer to the actual position of the blimp. Once the low pass filter was added to the Simulink model, the simulation was run and the figure below, figure 14, shows the output.

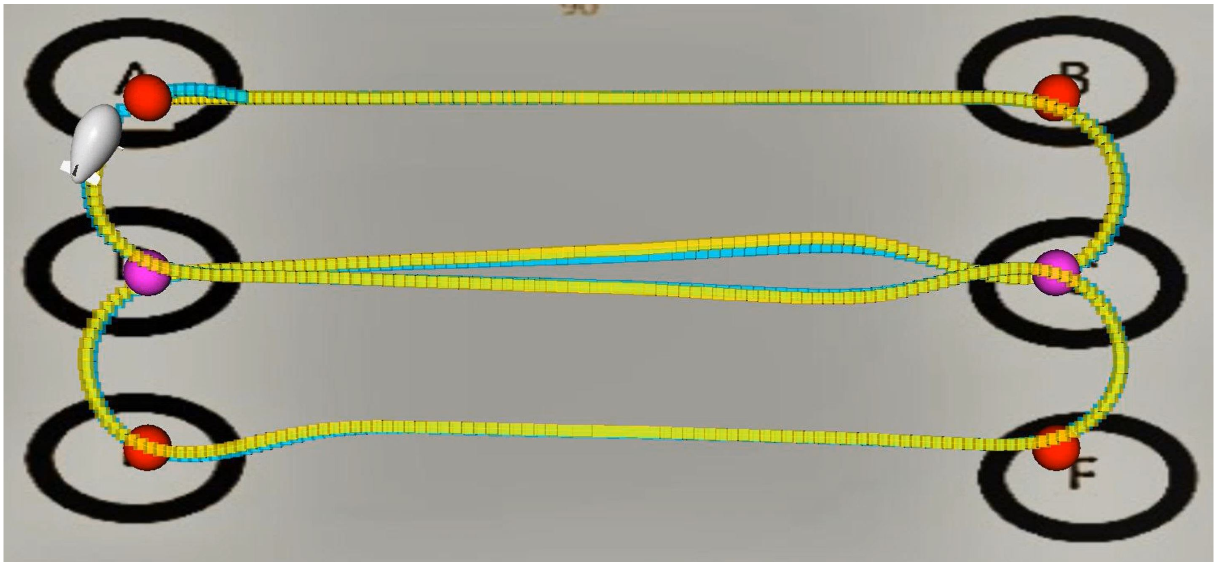


Figure 14: Final Simulink Simulation.

The blue line in the figure represents the real states and the yellow line represents the filtered states. As you can see the filtered states are very close to to the actual states with a few errors when the blimp turns. The errors lower as time goes on and the filtered states eventually line up with the real states.

6 Flight

Due to COVID-19 restrictions, there was only time for one flight after the controller and filter had been completed. All wires and motors had been checked for defects in the previous days so that the flight test would continue without issue. A pre-flight check of the motors showed that all motors could operate with forward or backward thrust. The only failure was the pitching motor that allowed for altitude control. This motor was stuck in an upward pitch and many attempts to dislodge it were unsuccessful. The motor failure eventually caused the rest of the motors to also become unresponsive to controller signals. Unfortunately, the blimp never flew without the original RC controller. With additional effort and time to repair the motors, it is expected that autonomous flight will easily be achieved.

7 Conclusion

This project is centered on the creation of an autonomous blimp. The team successfully developed a set of equations describing the dynamics of the blimp and using these equations, designed a control system to allow the blimp to autonomously travel through a set of pre-defined waypoints. This control system was tested in simulation in Simulink with positive results. However, there were significant delays in obtaining access to the blimp due to the COVID-19 pandemic, which resulted in the lab being closed for more than half of the project duration. As a result, the intended steps of the project which required the blimp, including preliminary flight testing, sensor testing, and basic characteristic determination occurred much later in the project than they would have otherwise. This culminated in significantly reduced time to develop the control software to test onboard the physical blimp. Near the end of the project, a hardware compatibility issue was encountered which resulted in being unable to implement part of the localization scheme due to the lack of time to replace components. While there exists a partial control implementation using the Simulink Support Package for Raspberry Pi, the full control system was never implemented. The team did not have the opportunity to fully assess the accuracy of the blimp dynamics model used for the control system. During the only flight test of the control system, a motor defect was found that eventually led to the failure of all other motors. This prevented the control system from being fully tested onboard the blimp. With additional time, the motors could be repaired and a different Raspberry Pi implementation can be created that would allow for the full control system to be used. Another flight test can be made to test all these components but time constraints prevent that from occurring within this MQP.

8 Future Work

In this project, a control system for an autonomous blimp was partially implemented. There is plenty of work left to build onto this MQP both in what was left unimplemented from the original control system design and future steps to make the blimp ready for different applications. In order to complete the control system design, the Raspberry Pi Zero W development board has to be replaced with a full-sized Raspberry Pi development board which has a CPU architecture compatible with the AprilTags software. This enables the use of AprilTags which will reduce the position and orientation error accumulated over time. Additionally, the fuel gauge should be added to the physical blimp in order to enable battery-level tracking during flight. The CD-EKF should be implemented, along with sensor fusion, in order to be able to run full blimp tests and verify the control system's performance and the dynamics model already created in Simulink.

In order to expand on the development of the autonomous blimp control system, future work could expand to the multi-blimp system originally envisioned for this project. In addition to acquiring a new blimp, this would require implementing an algorithm to coordinate and deconflict paths, as well as enabling inter-blimp communication via a WiFi network or radio signals. In order to enable the blimp to complete the envisioned task of monitoring large indoor systems, monitoring capabilities should also be added to the blimp which would then require expanding the sensor payload. A new software implementation is needed to poll those sensors and then identify and communicate any irregularities found. In addition, this may require increasing processing capabilities with more work offloaded to the ground station. Finally, this system could be applied to outdoor applications, which would require making the control system more robust to handle the greater disturbances associated with the outdoors such as wind and electronic interference. The localization scheme would have to be changed to use GPS since AprilTags must be pre-placed and in view of a camera which is prohibitive with outdoor environments.

References

[1] E. Bleier. (Mar. 2017). "What's the difference between a blimp, zeppelin, airship, and dirigible?" [Online]. Available: `https://www.insidehook.com/article/vehicles/blimps-zeppelins-and-dirigibles`.

[2] Y. Meddahi, *Modélisation; Estimation et Commande d'un Ballon Dirigeable Autonome*. Université des Sciences et de la Technologie d'Oran, 2015.

[3] K. C. J. J. Dworakowski Park, 2018. [Online]. Available: `https://uwaterloo.ca/capstone-design/2018-mechatronics-engineering-capstone-design-projects`.

[4] E. Hygounenc, I.-K. Jung, P. Souères, and S. Lacroix, "The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 473–511, 2004. DOI: `10.1177/0278364904042200`.

[5] N. Yao, Q. Tao, W. Liu, Z. Liu, Y. Tian, P.-y. Wang, T. Li, and F. Zhang, "Autonomous flying blimp interaction with human in an indoor space," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, pp. 45–59, Jan. 2019. DOI: `10.1631/FITEE.1800587`.

[6] S. U. Ferdous, A. Mohammadi, and S. Lakshmanan, "Developing a low-cost autonomous blimp with a reduced number of actuators," in *Unmanned Systems Technology XXI*, C. M. Shoemaker, H. G. Nguyen, and P. L. Muench, Eds., International Society for Optics and Photonics, vol. 11021, SPIE, 2019, pp. 73–80. DOI: `10.1117/12.2519252`. [Online]. Available: `https://doi.org/10.1117/12.2519252`.

[7] S. Sudhakar, S. Karaman, and V. Sze, "Balancing actuation and computing energy in motion planning," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4259–4265, 2020.

[8] T. Liesk, M. Nahon, and B. Boulet, "Design and experimental validation of a nonlinear low-level controller for an unmanned fin-less airship," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 149–161, 2013. DOI: `10.1109/TCST.2011.2178415`.

[9] Y. Wang, G. Zheng, D. Efimov, and W. Perruquetti, "Disturbance compensation based control for an indoor blimp robot," in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2040–2046. DOI: `10.1109/ICRA.2019.8793535`.

[10] ——, "Improved altitude control method with disturbance compensation for an indoor blimp robot," in *Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 3902–3907. DOI: `10.1109/CDC.2017.8264233`.

[11] E. Hygounenc and P. Soueres, "Lateral path following gps-based control of a small-size unmanned blimp," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 1, 2003, pp. 540–545. DOI: `10.1109/ROBOT.2003.1241650`.

[12] J. Bijker and W. Steyn, "Kalman filter configurations for a low-cost loosely integrated inertial navigation system on an airship," *Control Engineering Practice*, vol. 16, pp. 1509–1518, Apr. 2008. DOI: `10.1016/j.conengprac.2008.04.011`.

[13] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. DOI: `10.1109/TSSC.1968.300136`.

[14] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2011, pp. 3400–3407.

[15] J. Courbon, Y. Mezouar, N. Guénard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice*, vol. 18, pp. 789–799, 2010. DOI: 10.1016/j.conengprac.2010.03.004.

[16] april robotics laboratory. (2010). "Apriltag," [Online]. Available: https://april.eecs.umich.edu/software/apriltag.

[17] E. Morimoto, M. Suguri, and M. Umeda, "Vision-based navigation system for autonomous transportation vehicle," *Precision Agriculture*, vol. 6, pp. 239–254, 2005. DOI: 10.1007/s11119-005-1384-x.

[18] K. D. Li Wang, "Vision-based positioning with a single camera and 3d maps: Accuracy and reliability analysis," *Journal of Global Positioning Systems*, 2011.

[19] A. Elfes, S. Siqueira Bueno, M. Bergerman, and J. G. Ramos, "A semi-autonomous robotic airship for environmental monitoring missions," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 4, 1998, pp. 3449–3455. DOI: 10.1109/ROBOT.1998.680971.

[20] T. Fukao, K. Fugijani, and T. Kanade, "Image-based tracking control of a blimp," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 5, Dec. 2003, 5414–5419 Vol.5. DOI: 10.1109/CDC.2003.1272498.

[21] N. R. Council, "Autonomy research for civil aviation: Toward a new era of flight," 2014. DOI: 10.17226/18815.

[22] G. Atmeh and K. Subbarao, "Guidance, navigation, and control of unmanned airships under time-varying wind for extended surveillance," *Aerospace*, vol. 8, p. 3, 2016.

[23] H. Curtis, *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, 2019.

[24] R. Islam, S. R. M. Anwar, A. T. Kamal, S. N. Rahman, H. M. Faraby, H. Imtiaz, M. S. Jahan, and F. S. Tulip, "Battery chemistry detection algorithm implementable with intelligent systems: A step towards the development of a novel charger applicable for multi-chemistry environment," in *Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application*, vol. 3, 2008, pp. 805–810. DOI: 10.1109/IITA.2008.441.

[25] J. Lindblom. (n.d.). "Voltage dividers," [Online]. Available: https://learn.sparkfun.com/tutorials/voltage-dividers/all.

[26] T. G. Gainer and S. Hoffman, "Summary of transformation equations and equations of motion used in free-flight and wind-tunnel data reduction and analysis," National Aeronautics, Space Administration Scientific, and Technical Information Office, Tech. Rep., Jan. 1972.

[27] Y. Kim and H. Bang. (Nov. 2018). "Introduction to kalman filter and its applications."

[28] F. L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*. CRC Press, 2008. DOI: 10.1201/9781315221656.

[29] P. Frogerais, J.-J. Bellanger, and L. Senhadji, "Various ways to compute the continuous-discrete extended kalman filter," *IEEE Transactions on Autonomous Control*, vol. 57, pp. 1000–1004, 2012. DOI: 10.1109/TAC.2011.2168129.

[30] H. Liu, S. Fang, and J. Ji, "An improved weighted fusion algorithm of multi-sensor," *Journal of Physics*, 2019. DOI: 10.1088/1742-6596/1453/1/012009.

[31] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Wiley-Blackwell, 2015.

[32] J. B. Mueller, M. A. Paluszek, and Y. Zhao, “Development of an aerodynamic model and control law design for a high altitude airship,” *American Institute of Aeronautics and Astronautics*, 2004.

[33] O. M. McGrath, M. Kontopyrgos, S. Kasapis, S. Halama, and B. Preston, “Estimation of a plume with an unmanned terrestrial vehicle,” Worcester Polytechnic Institute, Tech. Rep., Mar. 2018.

[34] A. Bryson, Y.-C. Ho, and G. Siouris, “Applied optimal control: Optimization, estimation, and control,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, pp. 366–367, Jul. 1979. DOI: 10.1109/TSMC.1979.4310229.

Appendices

A Added Mass/Inertia

Since added mass/inertia is dependent on the shape of the body, a double ellipsoid model is used to determine the values.

$$\text{Added Mass/Inertia} = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{bmatrix}.$$

$$a_{11} = -\frac{\alpha}{2-\alpha} * \rho_{air}, \tag{A.1}$$

$$a_{22} = a_{33} = -\frac{\beta}{2-\beta} * \rho_{air}, \tag{A.2}$$

$$a_{55} = a_{66} = -\frac{1}{5} * \frac{(b^2 - a^2)(\alpha - \beta)}{2(b^2 - a^2) + (b^2 + a^2)(\beta - \alpha)} * \rho_{air}. \tag{A.3}$$

$$\alpha = 2g\left(\frac{f}{2} - e\right), \tag{A.4}$$

$$\beta = \frac{1}{e^2} - \frac{gf}{2}, \tag{A.5}$$

$$g = \frac{1-e^2}{e^3}, \tag{A.6}$$

$$f = log\left(\frac{1+e}{1-e}\right), \tag{A.7}$$

$$e = \sqrt{1 - \frac{b^2}{a}}, \tag{A.8}$$

where a and b are the semi major and semi minor axis of the blimp, respectively. Equations (A.1) - (A.8) are from [32].

B Kalman Filter Calculations

The state vector is defined as

$$X = \begin{bmatrix} x \\ y \\ z \\ u \\ v \\ w \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix},$$

with the derivative of the state defined by the equations of motion that were derived in section 2.2.5. The equations of motion are significantly simplified by substituting the all the values from Appendix A and solving so that the equations are not dependent on one another,

$$\dot{X} = f(X(k), U(k))$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u\cos\psi\cos\theta - v(\cos\phi\sin\psi - \cos\psi\sin\phi\sin\theta) + w(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta) \\ u\cos\theta\sin\psi + v(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) - w(\cos\psi\sin\theta - \cos\phi\sin\psi\sin\theta) \\ -u\sin\theta + v\cos\theta\sin\phi + w\cos\phi\cos\theta \\ -\frac{qw - rv + \frac{T_m\cos\gamma - F_{bg}\sin\theta + \frac{CTS_{ref}\rho_{air}(u^2+v^2+w^2)}{2}}{M}}{\frac{a_{11}}{M} - 1} \\ -\frac{ru - pw + \frac{T_t - F_{bg}\cos\theta\sin\phi + \frac{CLS_{ref}\rho_{air}(u^2+v^2+w^2)}{2}}{M}}{\frac{a_{22}}{M} - 1} \\ -\frac{pv - qu + \frac{F_{bg}\cos\phi\cos\theta - T_m\sin\gamma + \frac{CNS_{ref}\rho_{air}(u^2+v^2+w^2)}{2}}{M}}{\frac{a_{33}}{M} - 1} \\ p + q\sin\phi\tan\phi + r\cos\phi\tan\phi \\ q\cos\phi - r\sin\phi \\ r\frac{\cos\phi}{\cos\theta} + q\frac{\sin\phi}{\cos\theta} \\ -\frac{T_t z_{Tt} - \frac{ClS_{ref}\rho_{air}(u^2+v^2+w^2)}{2}}{I_{M_{xx}}} \\ -\frac{CmS_{ref}\rho_{air}(u^2+v^2+w^2)}{2I_{M_{yy}}(\frac{a_{55}}{I_{M_{yy}}} - 1)} \\ -\frac{T_t x_{Tt} + \frac{CmS_{ref}\rho_{air}(u^2+v^2+w^2)}{2}}{I_{M_{zz}}(\frac{a_{66}}{I_{M_{zz}}} - 1)} \end{bmatrix}.$$

The system inputs vector are composed of the three signals, γ, T_t and T_m:

$$U = \begin{bmatrix} T_t \\ T_m \\ \gamma \end{bmatrix},$$

The measurement vector is made up of the single integration of the accelerometer data to obtain the velocity data, and the rate gyro measurements to obtain the Euler angle measurements from the IMU. A pseudo-integration method was that was found in [33] used to approximate these measurements.

$$Z = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}.$$

The measurement vector can be related to the state estimation vector by the matrix C,

$$Z = C\hat{X}$$

$$C = \begin{bmatrix} \frac{1}{\delta T} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\delta T} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\delta T} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\delta T} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\delta T} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\delta T} \end{bmatrix} . \hat{X} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$$

The pseudo-integration method is able to be used because δT is the change in time from each measurement gathered from the IMU. In this case, δT is less than 1 second. In order to obtain the F and G matrices used in the EKF, the A and B matrices must be found by taking the partial derivative of the state equations with respect to the state vector variables and the input vector variables,

$$\begin{aligned}
A &= \frac{\delta f}{\delta X}\\
&= \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix},
\end{aligned}$$

$$A_1 = \begin{bmatrix}
0 & 0 & 0 & \cos\psi\cos\theta & \sigma_3 - \sigma_6 & \sigma_1 \\
0 & 0 & 0 & \cos\theta\sin\psi & \sigma_2 & \sigma_4 - \sigma_5 \\
0 & 0 & 0 & -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \\
0 & 0 & 0 & -\frac{C_T S_{ref} \rho_{air} u}{a_{11} - M} & \frac{r}{\frac{a_{11}}{M} - 1} - \frac{C_T S_{ref} \rho_{air} v}{a_{11} - M} & -\frac{q}{\frac{a_{11}}{M} - 1} - \frac{C_T S_{ref} \rho_{air} w}{a_{11} - M} \\
0 & 0 & 0 & -\frac{r + \frac{C_L S_{ref} \rho_{air} u}{M}}{\frac{a_{22}}{M} - 1} & -\frac{C_L S_{ref} \rho_{air} v}{a_{22} - M} & \frac{p - \frac{C_L S_{ref} \rho_{air} w}{M}}{\frac{a_{22}}{M} - 1} \\
0 & 0 & 0 & \frac{q - \frac{C_N S_{ref} \rho_{air} u}{M}}{\frac{a_{33}}{M} - 1} & -\frac{p + \frac{C_N S_{ref} \rho_{air} v}{M}}{\frac{a_{33}}{M} - 1} & -\frac{C_N S_{ref} \rho_{air} w}{a_{33} - M}
\end{bmatrix},$$

$$A_2 = \begin{bmatrix}
v\sigma_1 + w(\sigma_6 - \sigma_3) & w\sigma_7 - \sigma_{17} + v\sigma_9 & w(\sigma_5 - \sigma_4) - v\sigma_2 - \sigma_{15} & 0 & 0 & 0 \\
-v(\sigma_5 - \sigma_4) - w\sigma_2 & w\sigma_8 - \sigma_{18} + v\sigma_{10} & w\sigma_1 - v(\sigma_6 - \sigma_3) + \sigma_{16} & 0 & 0 & 0 \\
v\cos\phi\cos\theta - w\sigma_{11} & -u\cos\theta - w\sigma_{13} - v\sigma_{12} & 0 & 0 & 0 & 0 \\
0 & \frac{F_{bg}\cos\theta}{a_{11} - M} & 0 & 0 & -\frac{w}{\frac{a_{11}}{M} - 1} & \frac{v}{\frac{a_{11}}{M} - 1} \\
-\frac{F_{bg}\cos\theta\cos\phi}{a_{22} - M} & \frac{F_{bg}\sin\theta\sin\phi}{a_{22} - M} & 0 & \frac{w}{\frac{a_{22}}{M} - 1} & 0 & -\frac{u}{\frac{a_{22}}{M} - 1} \\
\frac{F_{bg}\cos\theta\sin\phi}{a_{33} - M} & \frac{F_{bg}\sin\theta\cos\phi}{a_{33} - M} & 0 & -\frac{v}{\frac{a_{33}}{M} - 1} & \frac{u}{\frac{a_{33}}{M} - 1} & 0
\end{bmatrix},$$

$$A_3 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{C_l S_{ref} \rho_{air} u}{I_{Mxx}} & \frac{C_l S_{ref} \rho_{air} v}{I_{Mxx}} & \frac{C_l S_{ref} \rho_{air} w}{I_{Mxx}} \\
0 & 0 & 0 & -\frac{C_m S_{ref} \rho_{air} u}{I_{Myy}(\frac{a_{55}}{I_{Myy}} - 1)} & -\frac{C_m S_{ref} \rho_{air} v}{I_{Myy}(\frac{a_{55}}{I_{Myy}} - 1)} & -\frac{C_m S_{ref} \rho_{air} w}{I_{Myy}(\frac{a_{55}}{I_{Myy}} - 1)} \\
0 & 0 & 0 & -\frac{C_n S_{ref} \rho_{air} u}{I_{Mzz}(\frac{a_{66}}{I_{Mzz}} - 1)} & -\frac{C_n S_{ref} \rho_{air} v}{I_{Mzz}(\frac{a_{66}}{I_{Mzz}} - 1)} & -\frac{C_n S_{ref} \rho_{air} w}{I_{Mzz}(\frac{a_{66}}{I_{Mzz}} - 1)}
\end{bmatrix},$$

$$A_4 = \begin{bmatrix}
q\cos\phi\tan\theta - r\sin\phi\tan\theta & r\sigma_{14}\cos\phi + q\sigma_{14}\sin\phi & 0 & 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\
-r\cos\phi - q\sin\phi & 0 & 0 & 0 & \cos\phi & -sin\phi \\
\frac{q\cos\phi}{\cos\theta} - \frac{r\sin\phi}{\cos\theta} & \frac{r\cos\phi\sin\theta}{\cos^2\theta} + \frac{q\sin\phi\sin\theta}{\cos^2\theta} & 0 & 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

$\sigma_1 = \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \qquad \sigma_2 = \cos\psi\cos\phi + \sin\phi\sin\theta\sin\psi \qquad \sigma_3 = \cos\psi\sin\theta\sin\phi$
$\sigma_4 = \cos\phi\sin\psi\sin\theta \qquad \sigma_5 = \cos\psi\sin\phi \qquad \sigma_6 = \cos\phi\sin\psi \qquad \sigma_7 = \cos\psi\cos\theta\cos\phi$
$\sigma_8 = \cos\psi\cos\theta\sin\psi \qquad \sigma_9 = \cos\psi\cos\theta\sin\phi \qquad \sigma_{10} = \cos\theta\sin\phi\sin\psi \qquad \sigma_{11} = \cos\theta\sin\phi$
$\sigma_{12} = \sin\phi\sin\theta \qquad \sigma_{13} = \cos\phi\sin\theta \qquad \sigma_{14} = \tan^2\phi + 1 \qquad \sigma_{15} = u\cos\theta\sin\psi$
$\sigma_{16} = u\cos\psi\cos\theta \qquad \sigma_{17} = u\cos\psi\sin\theta \qquad \sigma_{18} = u\sin\psi\sin\theta$

$$\begin{aligned} B &= \frac{\delta f}{\delta U} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{\cos\gamma}{a_{11}-M} & 0 & \frac{T_m \sin\gamma}{a_{11}-M} \\ 0 & -\frac{1}{a_{22}-M} & 0 \\ \frac{\sin\gamma}{a_{33}-M} & 0 & \frac{T_m \cos\gamma}{a_{33}-M} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{z_{T_t}}{I_{M_{xx}}} & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{x_{T_t}}{I_{M_{zz}}(\frac{a_{66}}{I_{M_{zz}}}-1)} & 0 \end{bmatrix}. \end{aligned}$$

These matrices will need to be recalculated on each iteration of the Kalman filter with the time-based state data. By doing all of these calculations in the loop, the EKF will be able to converge on an accurate estimate of the state vector.

C LQR Design

The LQR design is based around the techniques developed by Arthur E. Bryson and Yu-Chi Ho in [34]. The primary technique utilized is the use of the maximum allowable state and input for a system as the diagonal elements of the Q and R matrices, respectively. With this technique as a starting point, different values within the Q and R matrices were substituted based on their simulated performance.

The first three diagonal elements of the Q matrix denote the maximum values allowable for the x, y, and z blimp states. The x, and y states cannot be exactly zero due to later algebraic manipulation so are instead designed to be nearly zero. The next three diagonal elements denote the maximum velocities in the body-frame corresponding to the u, v, and w blimp states. These u and w maximum allowable states are large since there should be little penalty to moving the blimp throughout the testing space. The v maximum allowable state is very small as this corresponds to the lateral velocity and research has indicated that blimps become uncontrollable as the lateral velocity becomes large. The last two diagonal elements denote the yaw angle ψ and its corresponding angular velocity, r. These are left large as there should be little penalty to the blimp spinning about its z-axis to align itself with its next target location.

The final forms of the Q and R matrices are shown below.

$$Q = \begin{bmatrix} 1\times10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1\times10^{-6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}.$$

$$R = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1008.1 & 0 \\ 0 & 0 & 700 \end{bmatrix}.$$

The selected Q and R matrices are used to solve the Ricatti Equation (43) while also minimizing the performance index in (42). The system is evaluated at two conditions, straight path and level turn. Each of these conditions yields a different numerical state space representation of the system and therefore two unique gain matrices are computed.

$$K_{sp} = \begin{bmatrix} 0 & 0 & -0.0687 & -0.8433 & 0 & -0.2981 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1377 & 0 & 0.4461 & 1.3342 \\ 0 & 0 & 0.2543 & -0.4208 & 0 & 1.4069 & 0 & 0 \end{bmatrix}.$$

$$K_{lt} = \begin{bmatrix} -0.0000 & -0.0000 & -0.1690 & -0.5939 & -0.0000 & -0.7857 & -0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.1377 & -0.0000 & 0.4461 & 1.3342 \\ -0.0000 & -0.0000 & 0.1750 & -1.0234 & -0.0000 & 0.9766 & -0.0000 & -0.0000 \end{bmatrix}.$$