Autonomous Blimp Modeling: A Feasibility Study and Annotated Bibliography

Authors:
Jacob Thomas
Nick Short
Dylan Filkins
Victor Albarran
Zeming Zhou
Benjamin Vu

ECE 412
Portland State University

Revision:2.4
Date:04/07/2021

Table of contents

1 Introduction

Blimps are a class of non-rigid airships—or lighter-than-air (LTA) vehicles—consisting of an envelope containing a gas that is less dense than air. The gas provides both the lift for the blimp and the “structure” to keep the shape of the envelope constant. Being non-rigid, blimps have no internal or external structure to maintain the shape of the envelope, as in rigid and semi-rigid airships.

Blimps usually have a small section, called a gondola (usually attached to the bottom of the blimp), that is used to carry people and equipment, and serves as an attachment point for engines. The gondola is often one of only two solid parts of a blimp—the other being the tail fins.

As the blimp’s elevation increases during flight, the gas inside the envelope expands. Since blimps are generally overinflated, it is important that there be some mechanism to prevent the envelope from exploding due to gas expansion. To prevent such a catastrophe, most blimps contain internal air bags, or “ballonets” which are filled with air.

As the blimp rises, the expanding gas pushes on the ballonets, forcing air out of them and providing volume to expand into. When the blimp loses altitude, exhaust from the engines is pumped into the ballonets, expanding them to fill the volume of the contracting gas, maintaining the pressure inside, and shape of, the envelope.

From a modern perspective, blimps did not exist prior to the 1900’s. Up to that point, LTA vehicles consisted solely of hot-air balloons. Hot-air balloons were only capable of being controlled vertically, go.

Early blimps were used as passenger aircraft and military vehicles, especially from the 1920’s to the 1950’s. Modern blimps are used mainly for advertising and long-term surveying, such as whale research. The ability of a blimp to stay in the air for hours—even days—makes it a preferred vehicle for tasks that require extended presence.

The typical uses of modern blimps require envelopes that are massive in size. This allows for a payload capacity that is viable for carrying the equipment and personnel required. This project is the result of a non-standard need. The blimp designed in this project will be used entirely indoors, in an office environment, performing “surveillance”, and in a warehouse environment, playing a variation of the sport Quidditch from the Harry Potter novels. There are few examples of such small scale, unmanned blimps deployed indoors to draw upon.

The blimp will need to navigate narrow hallways and open spaces contending with air currents and obstacles. It will be required to maintain a position within a certain distance of a target location, as well as be maneuverable enough to follow a flight path with a path deviation below a specified threshold. Eventually, the blimp should be part of an autonomous system, so it will need network connectivity and the ability to identify objects. This is most likely accomplished with a camera and some image processing capabilities.

1.1 Project Constraints

Due to the COVID pandemic and constraints on face-to-face meetings, your project focus has changed from creating a physical prototype to a feasibility study on autonomous blimps in an office and competition environments. The major reason for this change is because of the difficulty in constructing a physical model and keeping safety protocols. This feasibility study hopes to be a guide for others to develop a prototype based on our simulated work.

2 Design Space

Design Space Exploration (DSE) refers to the activity of exploring design ideas or alternatives prior to implementation or application. DSE is useful for many engineering tasks since it operates on the potential candidates of design. A trade-off analysis between each of the implementation options based on a certain parameter of interest forms the basis of DSE. The parameter of interest could vary across systems, but the commonly used parameters are power, performance, and cost. Additional factors like size, shape, weight, etc. These factors don't change for a blimp. The primary application will be indoors in an office building. Factors such as size, shape, and weight now have limitations due to standard door size, narrow hallways, and stairways.

2.1 Blimp Design Space

The listed parameters define a simplistic designated design space for the blimp's performance criteria.

- Size
    - The main location of the blimp will be indoors in an office space where its maximum size is limited to ceiling height and doorway width of location.
- Shape
    - It will need to accommodate a gondola, be able to fit standard doors and narrow stairways.
- Volume

  - The total space the helium will occupy in the envelope and this will determine minimum lift capability.
- Total Mass
  - The mass of the envelope and the gondola together.
- Gondola Placement
  - How it will be mounted to the envelope. Whether by adhesive or friction mount.
- Gondola Characteristics
  - Material – High density 3D printable polymer.
  - Machinability – Ideally one continuous piece.
  - Modularity – Iterative placement of components.
- Propeller Characteristics (Ct - Pull coefficient, Cp-Power coefficient, D- propeller direct, n-propeller speed)
  - Structural characteristics- It resists stretching of uneven forces in different directions and has good toughness.
  - Pull coefficient- $C_T = \frac{T}{\rho n^2 D^4}$
  - Cp-Power coefficient- $C_P = \frac{P}{\rho n^3 D^5}$
  - Propeller efficiency- $\eta = J \frac{C_T}{C_P}$
- Battery
  - The size of the battery is limited by the size of the blimp gondola.
- Envelope – Polyester Film PVC

  Structural characteristics - PVC material elasticity, strong anti-aging ability and PVC material resistance to high temperature, shape is preserved at high temperatures.

- Gas– Helium

  Physical properties - The surface tension is very small, the thermal conductivity is very strong, the gas density is 0.1786g/L. (0C, 1atm)

- Computing Placement – The geographical placement of the computing components of the blimp.
  - Electronic Control Units – A centralized Electronic Control Units can be placed on the gondola's frame, or a data transmitter can be used to send data to an off-frame computer.
  - Sensors – Need to be integrated into the chassis of the blimp.
  - Motors – Are required to be on the frame of the blimp.

    - Servos – need to be placed on the blimp’s gondola/ frame that is configured by the mechanical engineering students.

2.2 Blimp Performance Characteristics

Blimp performance is dependent on several characteristics. These characteristics can be modified to achieve desired performance.

Maximum Payload Weight

Payload weight is how much weight the blimp can carry. Greater payload weight allows for more sensors and a larger battery. Maximum Payload weight is affected by the buoyant force.

Maneuverability

Maneuverability is the bllimp’s ability to change position, direction, and speed.Blimp Maneuverability is affected by several factors:

- Mass (inversely proportional) – An increase in mass negatively impacts maneuverability by increasing the inertia needing to be overcome to effect a change in direction.
- Volume (inversely proportional) – Regardless of the shape, an increase in volume equates to an increase in the surface area, which increases overall air resistance.
- Shape – While some shapes may be great for forward motion, the surface area seen perpendicular to other axes is generally greater, increasing the air resistance in those directions. A spherical balloon shape is the most universally maneuverable shape.
- Thrust (proportional) – Different from the speed characteristic, the thrust parameter (regarding maneuverability) is related to the direction of the thrust more so than its amplitude.

Sensor Awareness

Sensor awareness is the blimp’s ability to see and navigate around objects. This is critical for the blimp to be able to support autonomy. Sensor awareness is contributed to by an array of various sensors working together:

- Cameras – A digital image can be used for object recognition.
- Ultrasonic Distance Sensors – These sensors send an ultrasonic wave and measure the time it takes to bounce back. This time is then used to calculate distance to an object.
- Accelerometers – Accelerometers measure the acceleration experienced by the object they’re attached to.

- Ammeters – Simple devices which measure current. This can be used in conjunction with the blimp’s operating voltage to calculate power draw.
- Microphones – Measure sound which can be used for speech recognition.

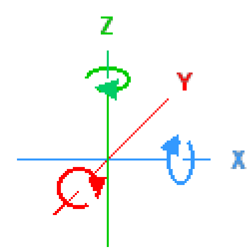
Flight Time

Flight time is how long the blimp can run and maintain an altitude. Blimp Flight time is affected by several factors:

- Passive lift (lift from helium; proportional) – Passive lift increases airtime, but only as it approaches being balanced against the mass of the blimp. As the balance shifts one way or the other, thrust must be applied in greater amounts to keep the blimp afloat.
- Battery capacity (proportional) – Larger battery capacity increases the time that thrust can be applied, at the cost of mass.
- Current draw (inversely proportional) – With no change in battery capacity, decreasing current draw will increase the airtime, as the battery will be able to supply current for a longer period of time.

2.3 Blimp Control Space

Achieving as many degrees of motion as possible will provide the greatest control of the blimp. There are six degrees of movement which are separated into three directions: forward/backward, left/right, up/down. Each direction has a translational component and a rotational component. If an aircraft points in the positive x-direction, the rotational components of the x, y, and z axes are referred to as roll, pitch, and yaw respectively.



Six Degrees of Freedom of Movement

Due to the location of the lift force relative to the center of mass, roll, and pitch are not persistent, and any movement in those axes will be reversed as the blimp returns to equilibrium.

The blimp’s control space is determined by the number of control inputs it has. These inputs will either create thrust in a certain direction or will change the direction of the thrust of a separate input. These inputs can be referred to as Force Control Inputs (FCIs) or Vectoring Control Inputs (VCIs). These inputs can be configured in several ways to achieve control within desired axes of movement.
Due to the location of the gondola below the envelope, the center of mass is located below the center of lift of the blimp. These properties cause the blimp to return to this orientation after being perturbed. Thus, roll and pitch rotation are not sustained without an active force input. This property is beneficial to a blimp, because sensors located on

the gondola may depend on a stable orientation to perform optimally. Because of this, achieving roll and pitch control is not desirable for a blimp.

The optimal control input configuration includes three FCIs. One is oriented vertically below the blimp's center of mass. This control input will be running whenever the blimp is on to counteract the buoyant force of the blimp. It will also change intensity to control altitude. The other two FCIs are oriented horizontally, parallel to each other, and facing aft. These two inputs are offset from the center of the blimp in order to create a couple of moments about the blimps Z-axis. Setting the input's intensity equal to one another moves the blimp forward or backward. Likewise, by setting the inputs to separate intensities, yaw articulation can be achieved.

2.4 Blimp Design Goals

Ideally, a blimp design would be simultaneously fast, maneuverable, and be able to stay in the air for a long time. Realistically, each of our characteristics comes with tradeoffs with respect to each other. For example, increasing the airtime by increasing the battery capacity will undoubtedly increase the mass, which would result in worse maneuverability. Increasing speed (by increasing the thrust) would lower airtime (by increasing current draw).

Since the design characteristics are—while not exactly mutually exclusive—somewhat incompatible, it becomes important to decide which of the characteristics are most important. Perhaps a slow and long-lived, but highly maneuverable blimp is desired. It may be that the goal to be achieved is a blimp that can maintain a position, very accurately, for a long period of time.

3 Metrics

3.1 Metrics introduction

Blimp characteristics are measured and compared using several metrics: top speed, turning radius, acceleration, total mass, buoyant force, and power draw. Some of these metrics can be measured directly using onboard instruments, while others must be calculated experimentally.

3.2 Important Metrics

Top Speed is measured by flying the blimp between two points and measuring the time the blimp takes to travel. The blimp should begin traveling before reaching the first point and continue after the last point to ensure top speed is maintained between the points.

A higher top speed will allow the blimp to travel distances faster. Top speed is measured in meters per second (m/s).

Turning Radius is measured by flying the blimp in as tight a circle as possible and then measuring the radius of the circular path.blimps employing a motor couple moment for yaw articulation will have a turning radius of zero. The turning radius is measured in millimeters (mm).

Acceleration is measured with an accelerometer attached to the blimp. It affects how quickly the blimp can reach its top speed. A greater acceleration will achieve top speed faster. Acceleration is measured in meters per second squared (m/s^2).

Total Mass is measured using a digital scale or balance. A blimp with more mass will accelerate slower and require more helium to stay afloat. Every component added to the blimp will increase its total mass. The total mass measured in kilograms (kg).

Buoyant Force is calculated using the blimp's helium volume. If this force is greater than the force of gravity, the blimp will rise, and if it is less than the force of gravity, the blimp will fall. The buoyant force is measured in Newton (N).

Power Draw is measured by a digital multimeter. Amp draw will be measured and the total voltage is calculated by the total components used. The product of both will be the total wattage (W).

Since the blimp will be simulated using a physics simulation software, all of the metrics will be known by the software and will not need to be measured. Instead, the metrics can be displayed and easily viewed.

3.3 Testing Outcomes

4 Simulation Software Analysis

The main goal of a blimp simulation is to validate mathematical models of alternative blimp designs, provide visual representations of them and our sponsor can continue the research project of the blimp. The simulation software will provide control of the inputs, modification of the dimensions, and the ability to model a blimp.
The performance of the blimp will be simulated through different scenarios and by changing the values of the variables. Performance characteristics that we will include are weight load, maneuverability, sensor awareness, and flight time. This performance

characteristic will be obtained by changing the input and model of the blimp. Metrics will be obtained by comparing the performance characteristics, the metrics we would like to obtain are speed, acceleration, turning radius, power (if simulation allows it), buoyant force. It is also important for the simulation to be open source or have a free student license.

The criteria for the decision matrix have been determined based upon the importance of a particular option or characteristic that would be expected in simulation software. The criteria are either a “true/false” criteria, or a scored criteria, from one (1) to five (5). A value of two (2) or four (4) represents an option or characteristic whose performance is better than the previous value, but not quite as good as the next.

Has Student Licence

Since it is not feasible to purchase all simulation softwares for evaluation, it is important for a software to be available for experimentation. In order to choose an expensive software, it must be a clear contender for best software. Software which is free can be used and evaluated without investment so this metric will be weighted at 5.

0-Licence required

5-Student licence or open source

Blimp Physics Modeling Ability

Modeling blimp physics is crucial for a blimp modeling simulation. This metric will be weighted at 5.

1-Basic simulation only evaluates one _

3-Simulates general physics with possibility for a blimp to be simulated

5-Software tailor made for aircraft/robot simulation

Ability to Modify Blimp Dimensions

Modifying blimp dimensions allows for quick changes to components within the simulation. This allows the AI developer to quickly implement hardware changes and update AI accordingly. This metric is weighted at a 2.

1-Blimp dimensions not simulated

3-Blimp model fixed once imported from separate software

5-Blimp dimensions and density can be changed from within software

Ability to Modify Control Inputs

Modifying control inputs allows AI developers to experiment with different force input configurations. This allows more freedom in AI development if it is decided that a less efficient control configuration is easier to develop for. This metric will be weighted at 3.

1-No control modification, control inputs are fixed to model

3-Control inputs can be modified with much effort

5-Control inputs easily modified

Ability to Calculate Power Usage

Measuring power used by the blimp is in important capability of the simulation software. The blimp will only be able to run as long as its batteries will allow, so in order to choose a battery which will offer the power needs of the blimp, while not weighing the blimp too much, an accurate calculation of power usage is required. This is a simple calculation for a computer, but tedious to do by hand and important to blimp development so it will be weighted at 4.

1-software only calculates time length and users will need to calculate power usage with known average power draw of components

3-software calculates power draw of components over time but does not calculate total power usage

5-software calculates total power usage over time.

Ability to Measure Flight Path Deviation

Deviation from the intended flight path is a useful way to evaluate the effectiveness of a blimp's AI controller. A larger deviation indicates a less effective controller. Since this metric is crucial for the evaluation and development of AI, it will be weighted at 4.

1-No flight path information

3-Raw data points of flight path provided which can be compared to intended flight path

5-Flight deviation calculated and provided

Ability to Visualize Simulation

Visualizing the simulation is not critical for AI development, but it can make troubleshooting easier by allowing programmers to see decisions made by an AI as they are made. This metric will be weighted at a 2.

1-No visual information

3-Some blimp representation but no physics simulator

5-Full blimp physics simulator

Has Common Scripting Language

Having a common scripting language will make AI integration much easier and allow development without the need to learn a new programming language. Since the sponsor is technically savvy and willing to work with the simulation's back end, this metric will be weighted at 3.

1-Proprietary language

3-Uncommon language

5-Common language

Table 1: Simulation Software Decision Matrix (Higher Score is Better)

Software Name (Multiplier)	Has Free/Student Licence (5)	Blimp Physics Modeling Ability (5)	Ability to modify dimensions (2)	Ability to modify control inputs (3)	Ability to calculate power usage (4)	Ability to measure path deviation (4)	Ability to visualize simulation (2)	Has common scripting language (3)	Total
Physics Abstraction Layer	5	3	4	3	2	4	1	C++ 5	98
Matlab w/Simulink	5	1	2	3	2	4	1	Matlab 5	84
CoppeliaSim	5	4	5	5	2	4	5	C/C++, LUA 5	119
Vortex Studio	0	5	5	5	2	2	5	3D modeling GUI 1	79
RoboLogix	0	3	5	3	2	1	5	Unique modeling 1	69
SimulationX	0	3	1	5	5	5	2	Custom Flowchart 2	82
Webots: robot simulator	5	5	4	3	2	2	5	C, C++, Python, Matlab 5	108
Gazebo simulator	5	3	3	3	2	2	4	Unique modeling, run in linux 2	90
ezphysics	5	1	2	3	1	2	3	C++, Python,Matlab 5	76

Physics Abstraction Layer

PAL acts as an interface between several different physics engines and your own software. It has built-in functionality for rigid bodies, sensors, and actuators, as well as being able to. This would give us more control over things, but it would come with the added work of having to write a more heavy framework.

Has Student Licence

5-Software is open source.

Blimp Physics Modeling Ability

3-Has access to multiple physics engines, but nothing specific to blimps.

Ability to Modify Blimp Dimensions

4-Dimensions can be changed in code, but it is ungainly.

Ability to Modify Control Inputs

3-Control inputs can be modified, but need to be translated from PWM/voltage to force.

Ability to Calculate Power Usage

2-No built in functions for calculating power usage. Would require custom methods.

Ability to Measure Flight Path Deviation

4-Deviation can be calculated, but only roughly, from time-sliced flight path estimate.

Ability to Visualize Simulation

1-Has no visualization.

Has Common Scripting Language

5-Compatible with multiple common languages.

Simulink

Simulink is a semi-graphical add-in for Matlab that allows the definition of individual components of a system. The components are linked together with defined interactions to create a simulation of the system. This would be a decent option, as Matlab is a robust language for solving problems that are defined by many equations.

Has Student Licence

5-Has student license.

Blimp Physics Modeling Ability

1-No built in physics. This functionality would need to be written entirely.

Ability to Modify Blimp Dimensions

2-Model cannot be imported and dimensions are “hard-coded”; difficult to change.

Ability to Modify Control Inputs

3-Control inputs can be modified, but need to be translated from PWM/voltage to force.

Ability to Calculate Power Usage

2-No built in functions for calculating power usage. Would require custom methods.

Ability to Measure Flight Path Deviation

4-Deviation can be calculated, but only roughly, from time-sliced flight path estimate.

Ability to Visualize Simulation

1-Has no visualization.

Has Common Scripting Language

5-Uses a common and well supported language.

CoppeliaSim

CoppeliaSim is graphical simulation software that allows the user to simulate rigid body physics in a visual way. It utilizes the ODE, Bullet, Vortex, and Newton physics libraries. Provides collision detection, distance calculation, and motion planning, among other functions. Plugins are programmed in C/C++, and the behavior of the simulator can be modified with LUA. Has built-in sensors and actuators and the ability to import 3D models from external software.

Has Student Licence

5-Has student license.

Blimp Physics Modeling Ability

4-Solid physics engine with ability to import CAD models.

Ability to Modify Blimp Dimensions

5-Dimensions, mass, moments of inertia, and other parameters are all modifiable.

Ability to Modify Control Inputs

5-Can easily modify control inputs through code or GUI.

Ability to Calculate Power Usage

2-No built in functions for calculating power usage. Would require custom methods.

Ability to Measure Flight Path Deviation

4-Deviation can be calculated, but only roughly, from time-sliced flight path estimate.

Ability to Visualize Simulation

5-Simulation fully visualizable.

Has Common Scripting Language

5-Compatible with multiple common languages.

Vortex Studio

Vortex Studio is a physics engine focusing on collision detection, and the way separate bodies interact with one another. It is primarily focused on vehicles and mobile robotics and includes a robust 3D modeling suite

Has Student Licence

0-Does not have student licence

Blimp Physics Modeling Ability

5-Physics simulator is designed to simulate mobile robots

Ability to Modify Blimp Dimensions

5-Software has native modeling ability

Ability to Modify Control Inputs

5-Control inputs are easily moved around blimp

Ability to Calculate Power Usage

2-Measures time simulation is running

Ability to Measure Flight Path Deviation

2-Tracks path of vehicle

Ability to Visualize Simulation

5-Full physics modeling suite

Has Common Scripting Language

1-Proprietary language

RoboLogix

RoboLogix is a software designed for modeling various types of robots. It has a physics engine which allows different models to interact with one another. Interaction with the robots occurs primarily with the RoboLogix Control Panel, a GUI modeled after a physical control panel. The various buttons on the panel can be tied to different actions of the robot being modeled.

Has Student Licence

0-No student license

Blimp Physics Modeling Ability

3-Some physics modeling but not intended for mobile robots

Ability to Modify Blimp Dimensions

5-

Ability to Modify Control Inputs

3-Basic control input ability

Ability to Calculate Power Usage

2-Simulation only keeps track of robot

Ability to Measure Flight Path Deviation

1-

Ability to Visualize Simulation

5-Full physics modeling

Has Common Scripting Language

SimulationX

SimulationX focuses on the inner workings of a system. It has a GUI oriented toward modeling the individual components and how they react with one another. The software does not use a model, but instead creates graphs to show various metrics set by the user.

Has Student Licence

Blimp Physics Modeling Ability

Ability to Modify Blimp Dimensions

Ability to Modify Control Inputs

Ability to Calculate Power Usage

Ability to Measure Flight Path Deviation

Ability to Visualize Simulation

Has Common Scripting Language

Webots

Webots is an open-source graphical simulation software and multi-platform desktop application used to simulate robots. It provides a complete development environment to model, program, and simulate robots. Its library includes robots, sensors, actuators, objects, and materials and uses a GUI to modify the simulation. Webots use C, C++, Python, Java, MATLAB among others to program the robots.

Has Student Licence

5-Software is open source

Blimp Physics Modeling Ability

5-Yes, it has a blimp already modeled by the community

Ability to Modify Blimp Dimensions

4-Yes, it can modify blimp dimensions

Ability to Modify Control Inputs

3-yes, some ability to control inputs

Ability to Calculate Power Usage

2-It seems to not have one

Ability to Measure Flight Path Deviation

2-it seems not to have one but it can located the position of the center of mass of object

Ability to Visualize Simulation

5-Yes, 3D simulation visualization

Has Common Scripting Language

5-Yes, common lenguajes are C, C++, java, Python, Matlab

Gazebo is an open-source software simulator with a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. It has 4 physics engines including ODE, Bullet, Simbody, and DART. Custom plugins for robot, sensor, and environmental control could be developed, Also user inputs could be entered through a GUI or hardware controls. Ability to track simulation and sensor performance.

Has Student Licence

5-Software is open source

Blimp Physics Modeling Ability

3-No, it does not have a built in blimp but it has an aerodynamics physics controller and it may have the ability to create one.

Ability to Modify Blimp Dimensions

3-It has the ability to modify dimension but unsure about the blimp

Ability to Modify Control Inputs

3-yes it has the ability to control inputs

Ability to Calculate Power Usage

2-No that I could read in the documentation

Ability to Measure Flight Path Deviation

2-Yes, It has some kind of path deviation

Ability to Visualize Simulation

4-yes, ability to visualize simulation but not totally sure with blimp

Has Common Scripting Language

1-No, seems not to have one and run only in linux

EZPhysics is a 3D simulation software and it uses Ogre 3D graphics library with ODE physics library. It builds characters by animating them using robotics control methods rather than playing pre-cooked motion sequences. EZPhysics makes it possible to create animated characters that really interact with the environment using closed-loop control techniques as do real robots. it is composed of different parts, an editor & simulator, a remote control protocol, and a C++ API.

Has Student Licence

5-Software is open source

Blimp Physics Modeling Ability

1-No, seems to not have one already built and difficult to build one

Ability to Modify Blimp Dimensions

2-Yes, It has some ability to modify model

Ability to Modify Control Inputs

3-Yes, it has the ability to control some inputs.

Ability to Calculate Power Usage

1-No, it does not have one or at least documentation does not mention it.

Ability to Measure Flight Path Deviation

2-Yes, it has a basic ability to follow the path

Ability to Visualize Simulation

3-yes it has a not polished visualization but it has one

Has Common Scripting Language

5-Yes, common lenguajes C++, Python, Matlab.

Based on weighted metrics, CoppeliaSim suited our needs the best. However, it is notable that if student licensing was not a factor, Vortex Studio and WeBots would be good contenders for simulating blimps. Both focus on vehicle design which is suitable for blimp design and testing.

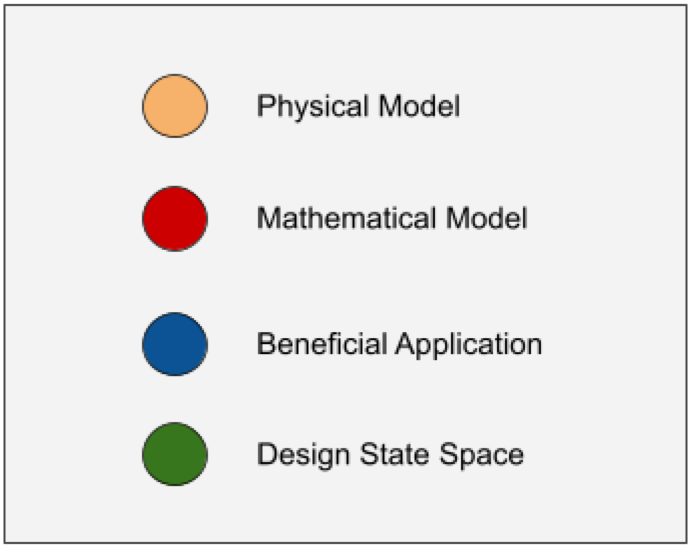
5 Software/Tool of Choice

After careful consideration, including the use of the decision matrix in Table 1, it was decided that the best choice of software for the simulation of a blimp would be using CoppeliaSim. In addition to having a student license, CoppeliaSim allows easy importation of 3D models developed in other software, such as AutoCAD and Fusion 360. CoppeliaSim utilizes four different physics engines to simulate the interaction between objects.

Simulations in CoppeliaSim have local APIs in C and LUA and remote APIs in C, Java, Python, Matlab, and LUA. With model parameters defined in scripts, the software will be able to quickly make changes to model parameters without needing to recompile code. The software can be integrated with an existing ROS or BlueZero system for autonomous control of the simulation. It also has built-in modules for providing thrust and sensing, including proximity sensors and vision sensors.

In short, CoppeliaSim will enable the model to be robust, but flexible, so that some analysis can be performed on the model's performance in response to changes in different parameters like total mass or propeller placement.

6 Annotated Bibliography



Citation:

Tao, Qiuyang & Tan, Tun & Cha, Jaeseok & Yuan, Ye & Zhang, Fumin. (2020). Modeling and Control of Swing Oscillation of Underactuated Indoor Miniature Autonomous Blimps. Unmanned Systems. 09. 10.1142/S2301385021500060.
https://www.researchgate.net/publication/340883576_Modeling_and_Control_of_Swing_Oscillation_of_Underactuated_Indoor_Miniature_Autonomous_Blimps

Summary:

This report goes into the basic setup of a circular autonomous blimp (developed at Georgia Tech). It describes their dynamic model calculations for the pitching. The paper goes into the background of linearization of their equations to come up with benchmarks of their data that they collected. As well, the paper explores the oscillation motion with graphical representation of the motion captured. The paper also goes into the calculations of buoyancy, lift, the moment of inertia. All calculations were done in Matlab.

System Specs:

Electronics Mass: 107.24 grams
PCB: Arduino Uno & XBee attachment
Flight Endurance: 2 hours
Helium Mylar Film: ~8 grams
Program: Matlab

Citation:

Nitta, Yoshihiro et al. "The Visual Inspection Methodology for Ceiling Utilizing the Blimp." Procedia Engineering. Vol. 188. Elsevier Ltd, 2017. 256–262. Procedia Engineering. Web.
https://www-sciencedirect-com.proxy.lib.pdx.edu/science/article/pii/S1877705817320349

Summary:
This report describes the advantages blimps could offer over drones for inspecting buildings for structural damage. The report offers minimal design considerations but goes into detail on which electronics are used.

The electronics used are a microcontroller that is an open-source 8-bit RISC-based platform, an Arduino Micro, two motors, and two servos. The Arduino Micro controls two servo motors with 2ch pulse-width modulation signals and two motors with two digital signals. With one of the servos in control of the wifi-enabled camera and the other to move the vertical motor arm. The camera used is a 1/9 inch CMOS sensor with 300,000 pixels and 802.11 b/g Wi-Fi interface. The two motors attached to the gondola control the horizontal and vertical motion of the propellers respectively. The system is powered by three batteries. Two 3.7V Li-Po batteries power the motors and servos and the last one is a rechargeable battery to power the wifi camera.

System Specs:
Total Mass: 300 grams
PCB: Arduino Wifi Micro
Flight Endurance: 1 hour
Helium Aluminum Film
Cost: $350

Citation:
Jinjun Rao, et al. "A Flight Control and Navigation System of a Small Size Unmanned Airship." IEEE International Conference Mechatronics and Automation, 2005, July 2005, doi:10.1109/icma.2005.1626776.
https://ieeexplore.ieee.org/document/1626776

Summary:
This report described in detail the H/ W and S/ W and flight controllers for a blimp used in a designated simulation space. The report goes into extensive detail about what hardware and software they used. It also goes into detail about the controllers used in their system with Fuzzy Controller, elevator controller, and a heading controller. The

authors also describe the design space of the blimp and use Matlab to calculate the necessary parameters. The paper also describes how they simulated the blimp in Matlab to get a better understanding graphically and spatially for their design.

System Specs:
Helium Volume: 50 m^3
Payload: 15 kg
Max. Velocity: 60 km/h
Program: Matlab

Citation:

Zufferey, Jean Christophe et al. "Flying over the Reality Gap: From Simulated to Real Indoor Airships." Autonomous Robots 21.3 (2006): 243–254. Autonomous Robots. Web.
https://doi-org.proxy.lib.pdx.edu/10.1007/s10514-006-9718-8

Summary:
The report goes into extensive detail about mechanics theory. It goes through the detailed proof of the dynamic modeling of the blimp so that it could be employed later in a simulation. The Newton-Euler equation is used to link the acceleration to the moments and forces applied to the blimp. The simulation model is done through Webots and a targeted path coordination was optimized through it. The team also looked at image recognition with the usage of simulations.

The electronics and hardware are detailed in the following. Three thrusters (8 mm DC motors, gear, and propellers), a forward-looking 1D camera (Taos inc. TSL3301) with 50 active pixels covering a horizontal field of view of 70◦, an anemometer, a vertical distance sensor (SharpTM GP2Y0A02YK), a MEMS piezoelectric rate gyro (AnalogDevicesTM ADXRS300) measuring yaw rotation, and an electronic board featuring an 8-bit microcontroller running at 20 MHz (MicrochipTM PIC18F452) together with a Bluetooth module (MitsumiTM WML-C10-AHR) for bidirectional wireless communication with a ground station. On-board energy is provided by a 1200mAh Lithium-polymer battery, which is sufficient for 2–3 hours of autonomy. Further details on the microcontroller board, wireless link, and camera.

System Specs:
The envelope measures: 110 × 60 × 60 cm
Lift Capacity: 200g
Parts: 3 DC rotors, ID Camera, vertical distance sensor, gyro, 1200mh lithium battery, and 8bit microcontroller running at 20 Mhz

Citation:

Palossi, Daniele et al. "Self-Sustainability in Nano Unmanned Aerial Vehicles: A Blimp Case Study." ACM International Conference on Computing Frontiers 2017, CF 2017. Association for Computing Machinery, Inc, 2017. 79–88. ACM International Conference on Computing Frontiers 2017, CF 2017. Web.
https://dl-acm-org.proxy.lib.pdx.edu/doi/abs/10.1145/3075564.3075580

Summary:

The report goes into detail about how to achieve low power consumption using the duty cycle, as the blimp rotor was turned on and off. It does not go into detail hardware but has a list of materials. Also, they used a solar cell attached to the blimp to extend the flight time, and they argue that it can stay several hours flying.

Basic mathematical equations are employed as a way to describe a blimp, but not of significant importance towards our use case.

The journal does describe the power usage management with software and NRF51 firmware to optimize the system as a whole.

System Specs:

The blimp measures: 0.4m3
Weight: 55g
Parts: solar panel, MCU, battery, harvester, and rotor.

Citation:

Zheng, Zewei, Huo, Wei, & Wu, Zhe. (2013). Autonomous airship path following control: Theory and experiments. Control Engineering Practice, 21(6), 769–788.
https://doi.org/10.1016/j.conengprac.2013.02.002

Summary:

The report looks into the mathematical model and simulation framework for the movement of a blimp 3D space. This is done through its simulation data and practical testing afterward. Goes into significant detail about how the mathematical model is achieved. The students used the guidance-based path following (GBPF) principle and trajectory linearization control (TLC) theory to explain their dynamic modeling.

As well goes into controller diagrams and explanations of it.

Specs:
Length: 13.2 m
Diameter: 3.38 m
Volume: 80m^3

Citation:

Ko, Jonathan et al. “Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp.” Proceedings - IEEE International Conference on Robotics and Automation. N.p., 2007. 742–747. Proceedings - IEEE International Conference on Robotics and Automation. Web.
https://ieeexplore-ieee-org.proxy.lib.pdx.edu/document/4209179

Summary:
The academic goes into the basic knowledge of the below topics and summarizes other academic articles. The article does a good job looking at the topics in a holistic and simplistic manner. The topics can be easily understood due to the summarization of other academic articles and the high arching overview of them. A benefit of this article is that it has links and references to all the articles that it reviews. The article also goes into a simplistic overview of control systems and autonomous control systems.

Topics:
-Motion Control
-Navigation Control
-Control Methods (PID)
-Controlling Algorithms
    -Backstepping Controls
    -Model-Predictive Control
-Autonomous Control Systems

Citation:

Wang, Yue et al. “Altitude Control for an Indoor Blimp Robot.” IFAC-PapersOnLine 50.1 (2017): 15990–15995. IFAC-PapersOnLine. Web.
https://www-sciencedirect-com.proxy.lib.pdx.edu/science/article/pii/S2405896317325363

Summary:
This report goes into detail briefly the parameters for the altitude control model, create simulation and apply it to the physical blimp. The end goal is for an autonomous blimp with a feedback loop of position of the environment to navigate.

System Specs:
Total Payload: 250 grams
PCB: Arduino FIO
Flight Endurance: 1 hour
Envelope: nylon 132cm length, 94cm diameter
Sensors: ultrasonic sensor, IMU, Camera,
Xbee unit, 3 mini rotors.
Simulation: matlab Simulink

Citation:
Alsayed, Ahmad, and Eric Lanteigne. "Experimental Pitch Control of an Unmanned Airship with Sliding Ballast." 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, doi:10.1109/icuas.2017.7991326.
http://hdl.handle.net/10393/36594

Summary:
The report describes the mathematical model, simulation and experimental results of a blimp with a gondola with and without wind disturbances. Author wanted to develop a close-loop control for a blimp using altitude and pitch using the gondola on a rail that can move as one of the inputs. Blimp was simulated in Matlab/simulink and control used a matlab GUI. simulation results were compared to experimental data/tests.

Topics:
-Motion Control
-Navigation Control
-Control Methods (PID)
-Controlling Algorithms (matlab GUI)
-Autonomous Control Systems

System Specs:
Total Payload: 350 grams
PCB: Nano Wii board
Flight Endurance: 1 hour
Envelope: 2 Mil polyester
polyurethane lm with 90 Shore A hardness
length is 1.83 m, volume 0.39 m3, and diameter is 0.6 m.
Sensors: bluetooth modules, IMU, 2 mini rotors.
Simulation and control: matlab Simulink / GUI

Battery: 2 cell 7.4V

Simulation Software

Physics Abstraction Layer
The Physics Abstraction Layer (PAL) provides a unified interface to a number of different physics engines. This enables the use of multiple physics engines within one application. It is not just a simple physics wrapper, but provides an extensible plug-in architecture for the physics system, as well as extended functionality for common simulation components.
Website: http://www.adrianboeing.com/pal/index.html
Reference Manual: http://www.adrianboeing.com/pal/documentation.html
User Guide: http://www.adrianboeing.com/pal/benchmark.html

Simulink
Simulink® is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.
Website: https://www.mathworks.com/help/simulink/
Reference Manual: ttps://www.mathworks.com/help/simulink/simulink-environment.html
User Guide: https://www.mathworks.com/help/simulink/getting-started-with-simulink.html

CoppeliaSim
The robot simulator CoppeliaSim, with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution. This makes CoppeliaSim very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave
Website: https://www.coppeliarobotics.com/
Reference Manual: https://www.coppeliarobotics.com/features
User Guide: https://www.coppeliarobotics.com/helpFiles/index.html

Vortex Studio
Vortex Studio is CM Labs' advanced suite of real-time simulation and visualization software, a high-fidelity platform for fast-paced, user-centric mechanical prototyping,

streamlined product design and deployment of immersive virtual experiences for human-in-the-loop testing, immersive training, and enhanced marketing experiences.
Website: https://www.cm-labs.com/vortex-studio/
Reference Manual:
User Guide:

RoboLogix
RoboLogix is a state-of-the-art robotics simulation software package that is designed to emulate real-world robotics applications. With RoboLogix, you teach, test, run, and debug programs that you have written yourself using a five-axis industrial robot in a wide range of practical applications. These applications include pick-and-place, palletizing, welding, painting and allow for customized environments so that you can design your own robotics application. With RoboLogix, the user can run the simulator to test and visually examine the execution of robot programs and control algorithms.
Website: https://robologix.com/
Reference Manual: https://www.robologix.com/robologix_overview.php
User Guide: https://www.robologix.com/programming_robologix.php

SimulationX
With SimulationX, you have a single platform for modeling, simulating, and analyzing technical systems, including mechanics, hydraulics, pneumatics, electronics, and controls, as well as thermal, magnetic, and other physical behavior. Comprehensive component libraries with application-oriented model elements ensure you have the right tools available for your task.
Website: https://www.esi-group.com/products/system-simulation
Reference Manual:
User Guide:

Webots
Webots is an open-source and multi-platform desktop application used to simulate robots. It provides a complete development environment to model, program, and simulates robots.
It has been designed for professional use, and it is widely used in industry, education, and research. Cyberbotics Ltd. maintains Webots as its main product continuously since 1998.
Website: https://www.cyberbotics.com/#features
Reference Manual: https://www.cyberbotics.com/doc/reference/thanks
User guide: https://www.cyberbotics.com/doc/guide/index

Gazebo

Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. At your fingertips is a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Best of all, Gazebo is free with a vibrant community.
Website: http://gazebosim.org/
Reference Manual(API):
http://osrf-distributions.s3.amazonaws.com/gazebo/api/11.0.0/index.html
User guide: http://gazebosim.org/tutorials

EZPhysics
EZPhysics integrates Ogre 3D graphics library with ODE physics library. It aims to breathe life into 3D game characters by animating them using robotics control methods rather than playing pre-cooked motion sequences.
Website: http://ezphysics.org/joomla/
User guide: http://ezphysics.org/joomla/index.php?option=com_jfusion&Itemid=83

Revision

Revision 2.0: 03/31/2021: Initial document
Revision 2.1: 04/02/2021: Added sections, Formatting
Revision 2.2: 04/04/2021: Added info and software research, Formatting
Revision 2.3: 04/07/2021: Added info to Ch1, Ch3, Ch4
Revision 2.4: 04/08/2021: Added info to Ch5, Formatting
Revision 2.5: 04/

Formal Sign Off