Autonomous Blimp Modeling: A Feasibility Study and Annotated Bibliography

<u>Authors</u>:
Jacob Thomas
Nick Short
Dylan Filkins
Victor Albarran
Zeming Zhou
Benjamin Vu

ECE 412
Portland State University

Revision 3.2
Date:06/08/2021

Table of contents

1. Executive Summary

This project was completed for the engineering department of Portland State University and Galois Inc for certification of a bachelor degree. The purpose of the project was to investigate the potential use cases for Galois Inc. pertaining to an autonomously operated blimp in an office environment and for a DARPA competition. The results of the project were a collection of deliverables for Galois. Including running simulations in CopelliaSim and Webots, 3D CAD work of a potential gondola, coding scripts, and an annotated bibliography. These deliverables are to assist Galois in the future development of an autonomous blimp for the stated purposes.

2. Introduction

Blimps are classified as non-rigid airships and lighter-than-air (LTA) vehicles. They consist of an envelope containing a gas that is less dense than air. The gas provides both lift and a form for the blimp. Since blimps are non-rigid, they 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, specified as a gondola (commonly attached to the bottom of the blimp), that carries people and equipment and serves as an attachment point for engines. The gondola is often one of only two fixed 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, there must be some mechanism to prevent the envelope from exploding due to gas expansion. Most blimps contain internal airbags or "ballonets" filled with air to counteract such inflation.

As the blimp rises, the expanding gas pushes on the ballonets, forcing air to escape and providing a volumetric space for the gas to expand freely. When the blimp loses altitude, exhaust from the engines is pumped into the ballonets, inflating them to fill the volume of the contracting gas, thus maintaining the pressure and shape of the envelope.

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

From the 1920s to the 1950s, LTA vehicles were used as passenger transportation and for military purposes. Modern blimps, though, are functionally used for advertising and long-term surveying, such as whale research. The ability of a blimp to stay in the air for hours, or even days, makes it a preferred vehicle for tasks that require an extended operation.

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. Additionally, the blimp may need to identify individuals that it cannot directly see. This could be done with a microphone and speech recognition algorithms.

2.1 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.

3. 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.

3.1 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 - $C_T = \frac{T}{\rho n^2 D^4}$
  - Power Coefficient - $C_P = \frac{P}{\rho n^3 D^5}$
  - Propeller Efficiency - $\eta_{prop} = J \frac{C_T}{C_p}$
- Battery
  - The size of the battery is limited by the size of the blimp gondola.

- Envelope – Polyester Film PVC; 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 so the envelope can be relatively weak. The gas density is 0.1786g/L. (0C, 1 atm)

- Computing Placement – The physical 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 blimp. Can be on the gondola.
  - Motors – Also on the blimp. Optimal placement is determined by simulation.
  - Servos – need to be placed on the blimp's gondola/ frame

3.2 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.

Sensing

The tasks that a blimp performs largely dictate the sensors required to operate. A simple radio controlled blimp needs no sensing to be controlled by the user, though some sort of object detection is handy to avoid collisions. A more complex, autonomous or semi-autonomous blimp will certainly need more sensors to operate. Tasks such as localization, object avoidance, and object detection all require disparate sensing solutions, with a small amount of overlap existing between some tasks.

It would be sensible to divide sensors into categories related to their applications, and beneficial to include sensors in multiple categories, where appropriate. This will allow an informed decider to potentially minimize the number of sensors needed to perform all the required tasks. The following is a categorical list of different sensor solutions:

- Localization
  - LiDAR – These sensors work in a similar fashion to RaDAR but using light instead of radio waves. They use "time of flight" to calculate distance. By firing a laser pulse, then measuring the time for the pulse to reach an object, be reflected, and return, the LiDAR can calculate the distance to the object (using the constant, C).
  - GPS – These sensors operate on the principle of trilateration. Each GPS satellite broadcasts their location with a timestamp. When the GPS sensor can receive signals from four satellites, it can use trilateration to calculate its precise location relative to the satellites, as well as calculate a very accurate time (GPS satellites utilize atomic clocks to generate their timestamps).
  - Accelerometers – Accelerometers measure the acceleration experienced by the accelerometer. These measurements can be used to estimate distance traveled, though not very accurately.
- Object Avoidance
  - LiDAR – Operates as above. Can be used to detect the existence of, and distance to, new objects around the blimp. Because this sensor is also used with localization, it makes a good choice, as it can simplify a sensing system and lower the impact on the control logic.
  - IR proximity – These sensors are single point detectors that often only detect the existence of an object in their range, but can sometimes communicate distance.
  - Ultrasonic Rangefinder – These sensors use a "time of flight" method, similar to LiDAR and IR proximity sensors to calculate the distance to an object. Rather

than using light, they use high frequency sound waves. The consequence of this is that they are most accurate when the density of the surrounding air is known. This necessitates the use of a barometric pressure sensor and a temperature sensor in conjunction with the ultrasonic rangefinder. This may be impractical when compared with IR proximity sensors or LiDAR modules.

- Object Detection
  - Cameras – A digital image can be used for object recognition, as well as navigation. Images or a video feed can also be transmitted to a user to see from the blimp's perspective.
  - Microphones – Measure sound which can be used for speech recognition. This is object detection, if humans are considered objects to be detected. In the event that a person is not in the field of view of the camera, their presence in a room could still be detected.
- Diagnostics
  - Ammeters – Simple devices which measure current. This can be implemented with a precision one ohm (or less) resistor and an analog to digital converter (ADC) with a precision reference voltage (an IC voltage reference such as a 7805 may be sufficient). The resistor is put into the current path and the ADC measures the voltage across the resistor against the reference voltage. The current through the resistor–and through the circuit–can be calculated from this measurement with little impact on the circuit. Care must be taken to ensure that the current draw being measured will never exceed the capacity of the resistor.

It's important to note that oftentimes, one sensor alone is not enough to perform a task. For example, using an accelerometer to keep track of position is possible, but not very accurate, as the error compounds during the math required to calculate position. In this case, it would be prudent to include GPS or LiDAR as an occasional "sanity check".

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.

3.3 Metrics

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.

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, none of the metrics will need to be measured. Instead, some of the metrics can be calculated and displayed for easy viewing.

3.4 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.

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 one or more thrust inputs. 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.

3.5 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.

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.

4.1 Evaluation Metrics

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 an 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	4	3	2	1	5	Unique modeling 1	67
SimulationX	0	3	1	5	5	5	1	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

4.2 Software Evaluation

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 a 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 a 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 estimation.

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

4 - importable models

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 - not integrated

Ability to Visualize Simulation

5 - Full physics modeling

Has Common Scripting Language

1 - Unique Modeling

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

0 - Paid License

Blimp Physics Modeling Ability

3 - models certain aspects of blimps

Ability to Modify Blimp Dimensions

1 - No blimp model

Ability to Modify Control Inputs

5 - Control inputs added as flow chart boxes

Ability to Calculate Power Usage

5 - If simulation is designed to simulate power usage, calculation will be easy to use

Ability to Measure Flight Path Deviation

5 - good calculations if simulation is designed for this metric

Ability to Visualize Simulation

2 - Simulation IO is static flow chart and dynamic graphs

Has Common Scripting Language

2 - GUI Flowchart

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

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

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 languages C++, Python, and Matlab.

4.3 Primary Software: CopelliaSim

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.

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.

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.

4.4 Secondary Software: WeBots

Based on Table 1, it was decided that a second best choice of software for the simulation of a blimp would be Webost. In addition to having a student license, Webots allows easy importation of 3D models developed in other software, such as CAD files from Solidworks, Blender or from URDF. Webots core is based on the combination of a modern GUI (Qt), a physics engine (ODE fork) and an OpenGL 3.3 rendering engine (wren). It runs on Windows, Linux and macOS.

Simulations in Webots have local APIs in C and C++ and remote APIs in C, Java, Python, Matlab, or ROS. Model parameters are defined in nodes, but the controller is contained in a script and the software will be able to make changes to model parameters but the software needs to be compiled each time. It also has built-in modules for providing thrust and sensing, including proximity sensors and vision sensors.

In short, Webots 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.

5. Software Implementation

The main purpose of the software is to provide an environment for rapid development of a navigation AI for a physical blimp. To achieve this, the focus of the simulation will be to provide as many scenarios as possible for the AI to be tested within. The best way to provide a range of scenarios is to allow the user to easily modify the simulation from a single control board. Ideally this could be accomplished by a GUI, but may also be met with a text configuration file. The software can be used to evaluate several metrics including maneuverability, power usage, stability and minimum turning radius.

5.1 Maneuverability

Understanding the blimps maneuverability is important in order to evaluate a blimp AI's effectiveness. A more effective blimp AI will have better maneuverability which will allow it to negotiate obstacles in its environment more effectively than a poor AI.

The maneuverability also depends greatly on the configuration of the motors, and the amount of thrust they are able to generate. Different motor configurations will provide different levels of maneuverability. Some configurations will allow for better control of altitude while others will allow the blimp to turn faster. Additionally, stronger motors will always have better maneuverability because they will be able to exert force at a faster rate.

The maneuverability of the blimp is measured by calculating the sum of the deviation from an intended path and dividing by the total distance of the path. For each simulation time step, the minimum distance from the intended path is calculated, and added together. This is the area of deviation. Once the path is completed, the area of deviation is divided by the number of time steps to calculate the blimp's maneuverability.

5.2 Turning Method

Due to the blimp's inertia, the blimp will not always move in the direction the blimp is facing. This property is directly evident when the blimp makes a turn. If the blimp were to simply spin on the yaw axis it would not turn in the same way a car or airplane would. Instead, it would continue to move in the same direction, with an added motion in the direction the blimp spins. The effect of this is that the blimp will overshoot target waypoints.

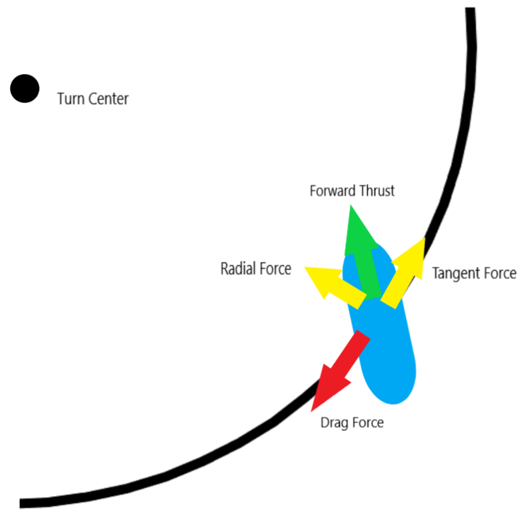


Figure 1: Turning Method

To overcome this restriction, a different method of turning must be implemented which takes into account the radial acceleration of the turn. Once a circular turn path is selected, the blimp must turn inward to face the center of the turn. If the blimp heading is directed at the center point throughout the turn, the blimp will coast through the turn, but will slow down due to air resistance. In order to maintain speed throughout the turn, the blimp heading must be adjusted toward the direction of motion such that the force tangent to the circular turn balances the drag from air resistance.

Once a desired turn is chosen, the maximum speed while navigating the turn can be calculated. The top speed of a turn depends on the maximum radial force that the blimp can exert. The required radial force for a given radius can be found with the following equation:

$$V_{max} = \sqrt{\frac{rF_r}{m}} \tag{1}$$

Where m is the mass of the blimp, r is the radius of the turn, and Fr is the radial force exerted by the blimp. The max speed of the turn is best selected after determining the desired turn. It is important to account for the reduction of maximum thrust if the blimp employs a couple moment to control yaw articulation. Since the blimp will be rotating throughout the turn, one of the propellers must rotate at a slower speed in order to create yaw torque. This means the maximum forward thrust will be lower than the sum of the forward propellers.

Since the blimp will be able to move fastest when the turn is most shallow, the most shallow turn possible should be selected. Afterward, the top speed can be calculated for that course. If a slower speed is required, the blimp will be able to make the same turn at any speed below the calculated top speed.

If the blimp is attempting to reach a known waypoint within a short distance, the fastest way to reach the waypoint will be along a circle which intersects the waypoint and is tangent to the blimp's velocity, assuming this path does not interfere with the environment. The center of this circle can be easily found by finding the intersection of two lines. The first is a line through the blimp's center of mass perpendicular to the velocity of the blimp, and the second is a line which bisects the blimp's center of mass and the target waypoint. This path will allow the blimp to travel at the maximum speed along the shortest path without coming to a stop. When targeting waypoints at a greater distance, this method can be used to redirect the blimp in the direction of the waypoint before following a straight line to the target.

When non circular paths are desired, this method can still be used. At each point along a known non circular path, there will be a curvature and associated radial distance. By recalculating the desired circular path repeatedly and adjusting the center of the arc. The maximum curvature of such a path will depend not only on the maximum force that can be applied, but also on the maximum angular acceleration and velocity. In the case of the couple moment, the relevant ρfeature of the blimp will be the distance between the horizontal props. In a design with a dedicated yaw control propeller, this will depend on the maximum force of that propeller.

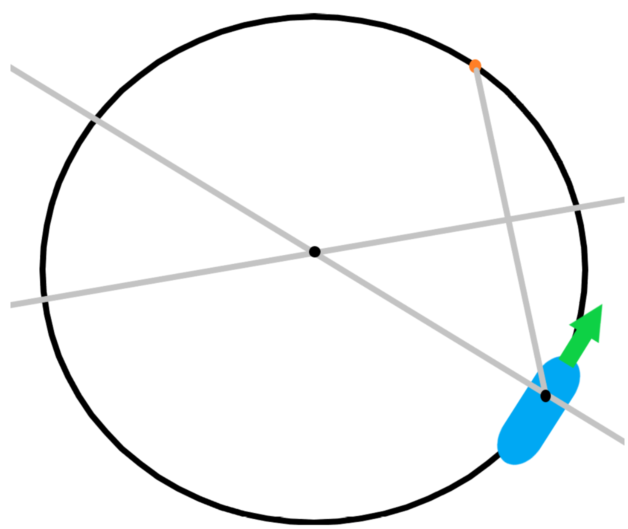


Figure 2: Finding the Turn Center

6. Component Analysis

6.1 Envelope

The envelope is the most difficult part of the blimp to obtain. Most premade balloons are not a sufficient size to avoid payload limitations. For this reason, most blimp applications will require a custom blimp envelope designed around the gondola and chosen components.

An important obstacle for the blimp is a standard doorway. In order to fit through a doorway, the maximum width for the blimp is roughly 0.5 m. For this reason, the best way to design the shape of the envelope is to start with an ellipsoid with b and c dimensions of 0.5m, and choose a dimension which will provide sufficient lift for the chosen components.

Unfortunately, very few companies are willing to make a blimp envelope at this scale. At a larger scale, premade blimp envelopes are available, in addition to several companies willing to fabricate custom envelopes. Likewise, if a smaller scale is achieved, a simple 18” mylar balloon can be used as the envelope. These simple party balloons have the advantage of being relatively cheap and easily accessible.

Because of its relative size, the envelope of the blimp will have the biggest impact on air resistance of the blimp as a whole. The air resistance of an ellipsoid can be calculated with the following equation:

$$D = \frac{C_d A \rho v^2}{2} \tag{2}$$

The coefficient of drag (C_d) is dependent on the ratio of the major and minor diameters of the ellipsoid; the longer the ellipsoid, the lower the coefficient of drag. Since the dimensions of our blimp is dependent on the weight of the gondola, this value will change depending on which components are chosen for the blimp.

Considering a blimp a worst case Cd value of 0.13, we can calculate the worst case air resistance for our blimp’s envelope. For a blimp with a minor diameter of 0.5m and a top speed of 4 mph, the maximum air resistance will be 0.05 N. A more likely coefficient of drag for a blimp is 0.06, which will yield an air resistance of 0.02 N at 4 mph.

6.2 Chassis

The chassis gondola design is modeled for the criteria that the sponsors desire. These criteria were to allow different use cases to be developed over the course of the capstone project. As well, they could be verified through quantitative analysis and later on, simulations. Due to varying degrees of interest in the design of the chassis throughout the capstone project from the sponsor, only one design was built through to completion. This design was analyzed with SolidWorks to determine the potential aerodynamics of the model and interactions to the environment. Below are the categories of which the chassis design had considerations.

6.2.1 Overall Design

Figure 3 represents the gondola and blimp together as an illustration of how they would be seen in a physical model. The blimp envelope is shown to be 1.4 m while the gondola is ~22 x 10 x 8.5 cm, and weighs 80 grams without electrical components. The material that we chose to model for is PLA but have considered other materials like expanding foam. Within Figure 4, it shows the gondola design more precisely. It is also the design that was used to evaluate the aerodynamics.

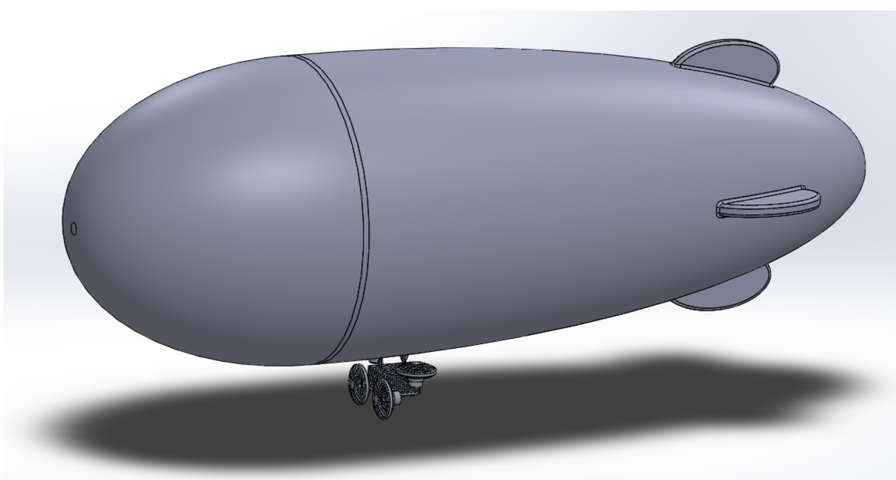


Figure 3: Overall Design with Blimp Envelope and Gondola

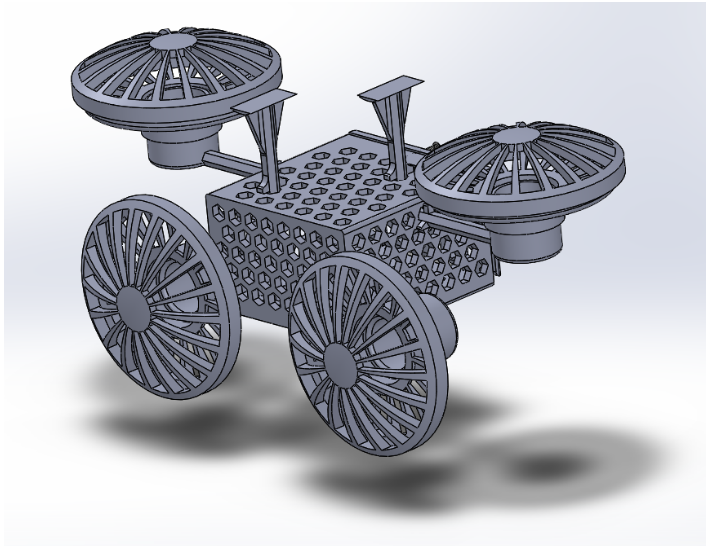


Figure 4: Model of the Gondola Design that is Expected to be 3D Printed

6.2.2 Modularity

The use case of modularity was important for the sponsor and user as they could change the position of the motors on the frame of the gondola. This could ensure that the frame would suit most cases in which parts changed or possibly the center of gravity would shift. Modulary was implemented through designing a hexagonal hole structure within the frame.

The hexagon shaped holes around the body are to allow protection of the electrical components within the housing. The holes also allow air flow through the housing to allow the electrical components to properly cool when the blimp is operating. The shape of the holes are to offer a modular design

6.2.3 Aerodynamics:

The gondola design was evaluated within the Flow Simulation in Solidworks to determine the change in pressure and velocity while interacting with the CAD model. The analysis of the design is elementary and was to show a representation of which parts of the design could impact the aerodynamics the most. Though, in the applications of this blimp, aerodynamics shouldn't affect the movement of the whole system. This is evident with the results found in figures one and two as they show the pressure and velocity drop as it passes across the design. Figures three and four show how the flow simulations were constructed to gather the data and calculations.

The results show that the model of the gondola causes a slight drop in pressure and a greater drop in velocity. It would be recommended that more analysis be done to have a verified numerical analysis on the design. As well, use these results to iterate a more aerodynamic model.

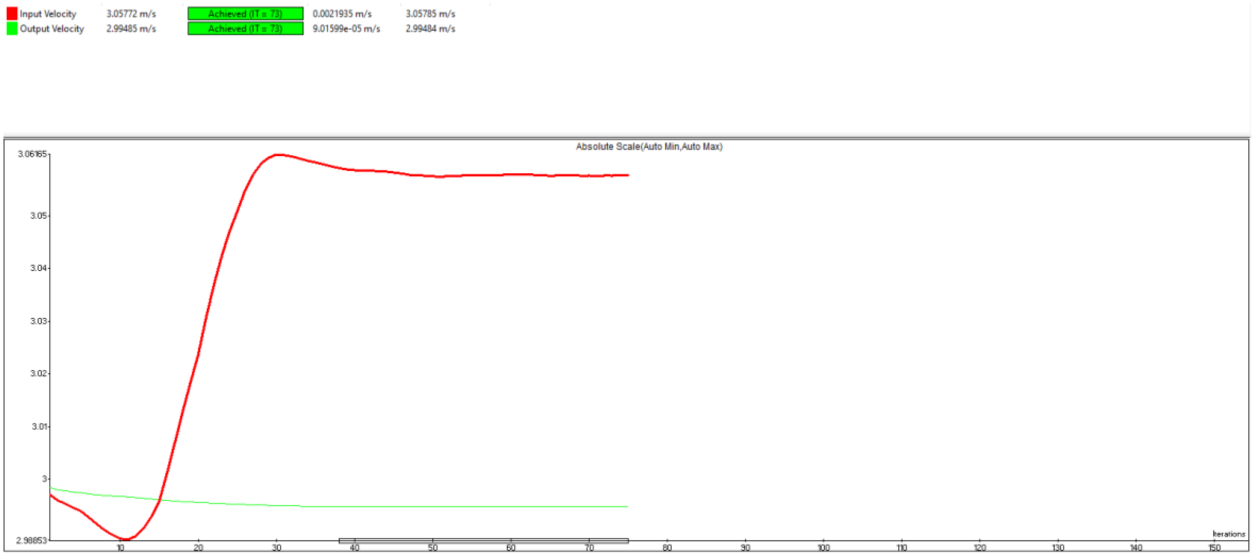


Figure 5: Velocity Drop Across the System.

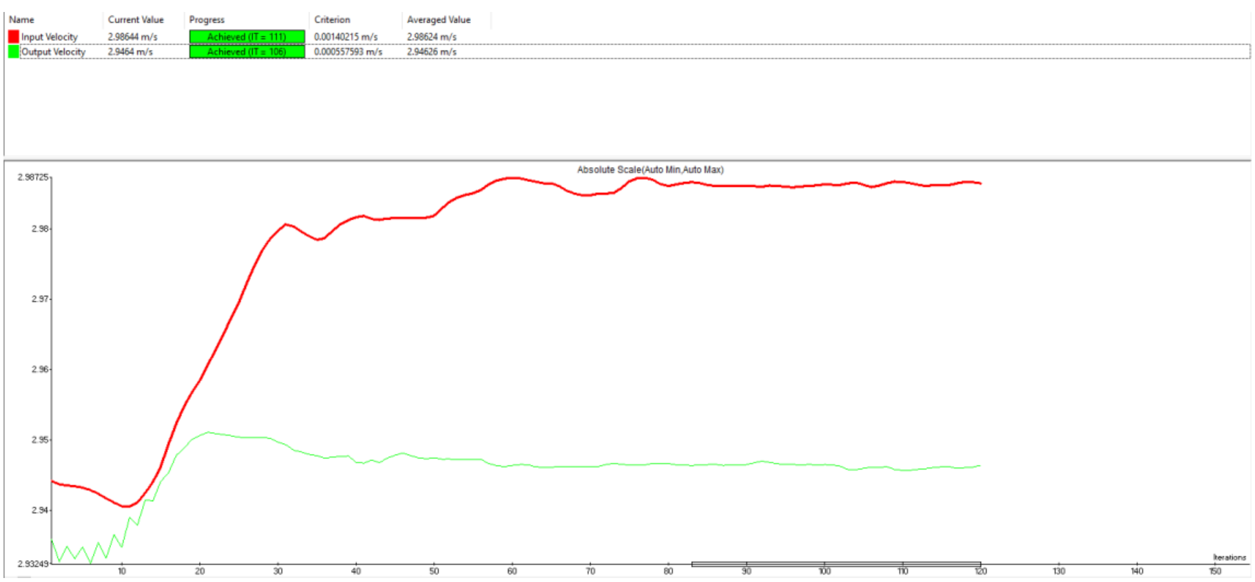


Figure 6: Pressure Drop Across the System.

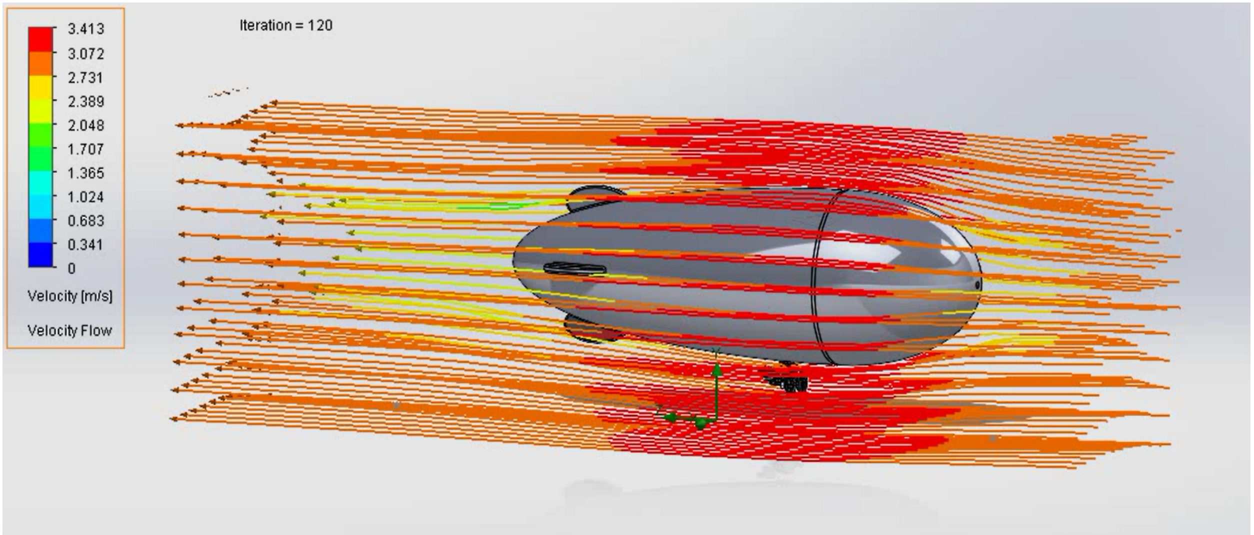


Figure 7: Flow of Air Interacting with the Complete Blimp Design.

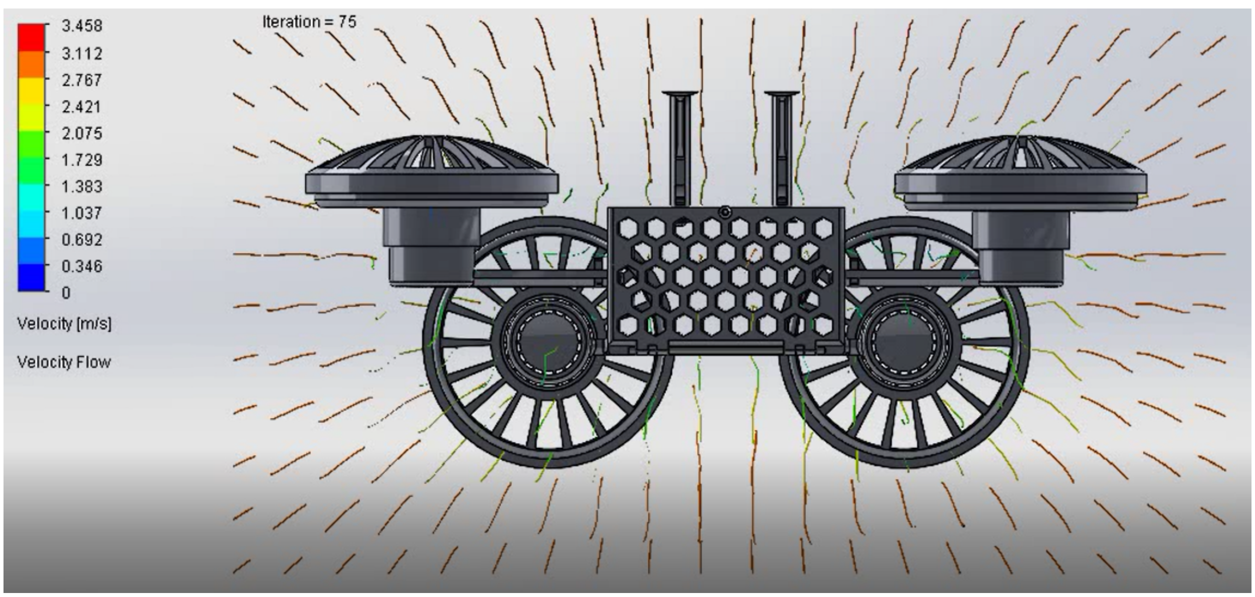


Figure 8: Movement of Air Passing through the Gondola

6.2.4 Mass

As the blimp is a lighter than air vehicle, the mass plays an important role in determining the amount of payload allowed. This was a design consideration when looking at the design of a structural and functional chassis. For this, it was considered how the parts would be manufactured on a 3D printer. To reduce weight, it was discussed changing the infill or the material used for the gondola. As well, looking at the amount of holes in the design, which requires a 3D printer to outline with a thicker wall.

6.2.5 Future Design Considerations

These future design considerations are to assist in the future possibility of bringing this capstone project into a physical model.

Due to the time constraints with the project, it would be advantageous to look into other considerations of the designed chassis. One design criteria should be looked into the implemented velcro design life. How many cycles could a velcro system be sustained before the velcro no longer performs at an acceptable level. Another criteria that would be significant to consider is overall strength of the gondola parts. It would be valuable to investigate the bending strength of the prototyped parts and material. Finally, a more in depth analysis of the aerodynamics of how the gondola affects the movement of the blimp could be significant.

6.3 Motors

The type of motor to be used depends heavily on the size of the blimp being made. For a very small blimp the obvious choice is a brushed DC motor. These motors are electrically simple and do not require separate ICs to drive them. Since the motors are not intended to be spinning for the entire duration of the run time, the life of the motors will also not be a limitation. Brushed DC motors can be made very small for the smallest blimps, but also scale up to medium sized indoor blimps. These motors can range from $2 to $100, although motors which cost more than $20 would likely be better replaced with brushless AC motors.

Brushless DC motors are not a good choice for blimp propulsion. They are usually designed to have high torque but relatively low power output. They are also very heavy for their power output as well.

At larger scales, a brushless AC motor is likely a better choice. These motors have a high power output, but are relatively heavy compared to brushed DC motors. The larger scale means the AC motor driver will not be a meaningful weight addition to the blimp. Additionally, non neutral buoyancy will need to be counteracted with a vertically oriented motor, which will need to be running any time the blimp isn’t tethered. Brushless AC motors are ideal for this situation because they do not have brushes which can wear out over time. These motors cost $10 and up depending on the power output needed.

At very large scales an internal combustion engine may be the best choice. Electric motors are limited by the batteries that power them because batteries are very heavy for the amount of power they can hold. Compared to batteries, gasoline has a very high energy density and will be able to provide propulsion with a lower impact on payload weight.

Motor power depends heavily on the desired characteristics of the blimp. However, a small blimp weighing around 2 lbs will be able to maneuver at a reasonable speed with motors rated between 1 and 5 watts. A larger blimp of 10 lbs will require more power, between 10 and 15 watts. In general, for an acceleration of 1 m/s^2 a blimp should have a motor which produces 1W for every Kg the blimp weighs.

5.4 Props

Motor thrust will depend on the dynamics of the propeller. Fortunately, in most cases the blimp will produce more than enough power to move the blimp. For this reason, a propeller should be chosen to maximize the motor’s efficiency. Each motor will have varying efficiency throughout its rpm range. The propeller should be chosen such that the desired maximum force from the motor will occur near the motor's peak efficiency. This will reduce the power being used by the blimp, and extend the blimp’s life between charges.

If blimp maneuverability is desired above efficiency, the propeller should instead be chosen which maximizes force at full power. This will be less efficient, but will allow for greater blimp maneuverability and top speed.

6.5 Batteries

Since the weight of the blimp is such an important metric, it is important to pick a battery which has a very high power density. One good choice to meet this requirement is the LiPo battery. These batteries have good power density and come in a wide range of sizes and power ratings. However, it is important that these types of batteries be charged with a LiPo specific charger to avoid starting a fire.

NiMh batteries are another good choice. While they do not offer the same power density as LiPo batteries, they are much more stable and do not require a special charger. They are also cheaper than LiPo batteries for the same power volume. NiMh batteries take longer to charge as well but this will not be important if the blimp has periods of down time.

7. Conclusion

Through the use of simulation software this project intended to learn about the dynamics of blimp movement in order to choose effective propeller configuration, important components, and design a gondola chassis.The plan was to simultaneously develop a simulation program and a CAD model. During this process both scenarios would be optimized.

The main purpose of the software was to provide an environment for rapid development of a navigation AI for a physical blimp. The software was used to evaluate some metrics including maneuverability, power usage, stability and minimum turning radius. The use of simulation software was important to our understanding of blimps, like their physics and how to apply forces and torque to the motors. One of the challenges was to adapt a turning method to and apply it to the blimp, we had some success implementing it.

The team constructed a 3D computer aided design using SolidWorks. The design integrated several different design criteria based on the desires from Galois Inc. The model specifically took into account modularity, airflow over electrical components, protection, and manufacturability. The development of the CAD model investigated the performance of aerodynamics and the influence it had on velocity and pressure. The design's function is to be used concurrently with the simulations to repetitively design an optimal design.

The team's recommendation is to use both of the simulations (CopelliaSim and Webots) to develop a more robust analytical tool for future use. The function of using both simulations is to construct different environmental constraints and situations and have the software eventually be refined into an AI system. Another recommendation is for the team successor to use SolidWorks to construct further analysis on the gondola and blimp design. The reasoning is because the original construction of the gondola was designed in SolidWorks, so making changes would be uncomplicated. SolidWorks also has a set of rigorous simulation functions that can be used to evaluate the aerodynamics, finite element analysis, and mass properties of the whole system.

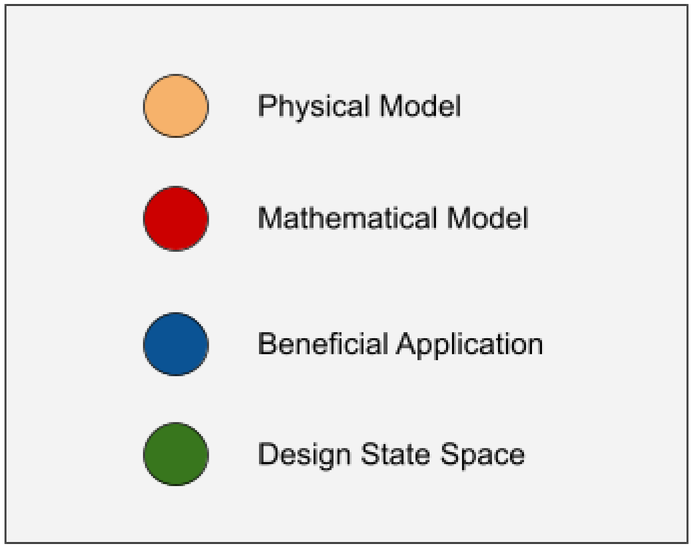
The work that has been done through this capstone project is expected to be developed further through Galois. The deliverables allow an introduction and a path for future investigators to implement a physical model. With the usage of a simulation that has been completed and a gondola design, it is expected that the future team will work iteratively through different blimp designs to find an optimal solution for Galois and the DARPA competition.

8. Works Cited

1.
P. Burgers, “A thrust equation treats propellers and rotors as aerodynamic cycles and calculates their thrust without resorting to the blade element method,” *Scholarly Commons*, 2019. [Online]. Available: https://commons.erau.edu/cgi/viewcontent.cgi?article=1427&context=ijaaa. [Accessed: 08-Jun-2021].
2.
“Ellipsoid Surface Drag, Drag Coefficient Equation and Calculator,” *Engineers Edge*. [Online]. Available: https://www.engineersedge.com/fluid_flow/ellipsoid_surface_drag_14047.htm. [Accessed: 08-Jun-2021].

9. Table of Figures

10. 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 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, and 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 about 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

Citation:
Boardman, Beth, and Linh Bui. "AA 449 Final Milestone Report Blimp Project." (2010): 1–55. Print.
https://www.washington.edu/search/?q=Final+Milestone+Report+Blimp+Project

Summary:
The report describes the mathematical model, simulation and experimental results of a blimp with a gondola. Author wanted to develop and control the blimp using 5 propellers. 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:
Motor: bidirectional propellers
Total Payload: 350 grams
PCB: Arduino Duemilanove
Envelope: 4 foot diameter

10. Simulation Softwares

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 multi domain 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 simulate robots.
It has been designed for professional use, and it is widely used in industry, education, and research. Cyberbotics Ltd. has maintained 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

Revisions

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: 04/16/2021: Added items table of Contents, Added info various.
Revision 2.6: 04/23/2021: Added items Table of contents, Added info to Ch4.
Revision 2.7: 04/30/2021: Added Info to Ch4.
Revision 2.8: 05/07/2021: Added Ch5.
Revision 2.9: 05/14/2021: Added info to Ch5.
Revision 3.0: 02/21/2021: Added info to Ch4.
Revision 3.1: 05/28/2021: Added info to Ch3, CH5.
Revision 3.2: 06/08/2021: Formatting.