

Formation Flying Blimps Project

AA449 Milestone Report 4

Beth Boardman

Linh Bui

Kyle Odland

Matthew Walker

Maggie Wintermute

Friday, May 7 2010

Introduction

This report will discuss the current state of the Formation Flying Blimps Project. The primary emphasis of this report will be on the blimp structural and power systems hardware, and the software implementation of the control law. Prior to the discussion of these focus points, the modified project goals are reiterated. An updated task list detailing the work still remaining for the project is also presented in this section of the report.

Project Update

The objective of the Formation Flying Blimps project is to design and implement two autonomously controlled blimp vehicles into the University of Washington Distributed Space System Laboratory. The project goals detailed in MS3 are reiterated below:

1. Construct one working blimp vehicle that can complete simple waypoint tracking within the confines of the Distributed Space System Laboratory testbed facility, robust enough to be used for the future plans of the lab.

2. Incorporate the blimp vehicle into the existing lead-follow control algorithm employed by ground vehicles in the Distributed Space Systems Laboratory.
3. Construct a second working blimp vehicle that can complete simple waypoint tracking within the confines of the Distributed Space Systems Laboratory testbed facility, robust enough to be used for the future plans of the lab.
4. Derive and implement the control algorithms for a series of coordinated tasks for both operational blimp vehicles to conduct. This will start as a simple Lead/Follow scheme, and evolve to steadily more difficult tasks should project time allow.

In order to fulfill all objectives, a plan of work was determined prior to the first milestone report. The initial deadlines were chosen based on the skill and experience level of the assigned members with respect to each task. One of the general assumptions taken into account at the start of the project was that control design process would not consume an inordinate amount of time. This turned out not to be the case, thus the deadlines for the Controls subsystem were altered accordingly. An updated task list, organized by project subsystem is provided in Table 1.

Table 1: Formation Flying Blimps Project MS3 Task list

Subsystem	Task	Members	Initial Deadline	Current Deadline	Progress
Hardware					
1	Identify and order necessary structural components, motors and propellers	All	4/9	5/7	100%
2	Finalize motor configuration	Boardman, Walker, Wintermute	4/9	4/9	100%
3	Thrust test motors for plant model	Boardman, Walker, Wintermute	4/16	4/28	100%
4	Develop blimp and gondola CAD model	Boardman, Walker	5/7	5/7	100%
5	Final blimp and gondola design	Boardman, Odland, Wintermute	5/14	5/14	100%
6	Construct blimps	All	5/21	5/21	75%
Software					
Subsystem	Task	Members	Initial Deadline	Current Deadline	Progress

1	Develop VICON camera system interface	Boardman, Odland, Wintermute	4/16	4/16	100%
2	Research and define software system architecture between VICON camera system, central computer and microcontroller	All	4/23	4/23	100%
3	Establish WiFi communication between Microcontroller and Central Computer, validate with simple task test	Boardman, Odland, Walker	4/30	4/30	100%
4	Develop Microcontroller to circuit board communication	Bui, Odland, Wintermute	5/7	5/7	100%
5	Develop control law code	Bui, Boardman, Odland	5/19	5/19	50%
6	Implement control law onto microcontroller	Bui, Odland, Walker	5/21	5/28	30%

Controls

1	Develop Plant Model	Bui, Walker, Wintermute	4/16	4/16	100%
2	Develop Waypoint Tracking Control Law	Bui, Walker, Wintermute	4/30	5/7	90%
3	Develop coordinated task and formation control law	Boardman, Odland, Walker, Wintermute	5/7	5/21	10%
4	Develop Simulink Model for control law validation	Odland, Walker, Wintermute	5/14	5/7	100%

Power

1	Identify and order necessary electrical components	Bui, Walker	4/9	4/28	100%
2	Research Microcontroller–actuator interface, battery configurations	Bui, Boardman, Odland	4/30	4/30	100%
3	Preliminary circuit board design, validate via breadboard	Bui, Walker	5/7	5/7	100%
4	Final circuit board design, construct on prototyping board	Bui, Walker	5/14	5/14	75%

Several prominent delays that affected the fundamental scope of the project were addressed in Milestone Report 3. Since that time there have been no other project altering delays.

Blimp Structure:

The main function of the gondola is to house the onboard components. The primary design requirement was to have enough surface area for mounting all the components. The secondary design requirement is mass limitation. It has been estimated that the structure mass should be about 50 grams for a neutrally buoyant blimp. The next design requirement is to comply with the performance requirement of being able to rotate 360 degrees in 30 seconds. Also for performance, the motors must be far enough away from the main structure to avoid air inference. The final design requirement is to keep the manufacturing processing simple.

Keeping these design requirements in mind, a preliminary design was formed and built. To keep the gondola compact, a double deck configuration was chosen. For simplicity, the main structure was designed as a cube. Version1 of the gondola is in Figure 1, which has a mass of 92 grams. 92 grams is nearly twice that of the estimated mass, which could be too heavy for the blimp envelope to lift. The motor arm length of 20 cm allows the blimp to rotate 360 degrees in 30 seconds which meets the design requirements. But, the motor arms are attached poorly as when the motor is fired the arm swings back about half a centimeter. A version2 was built to fix version1's problems.

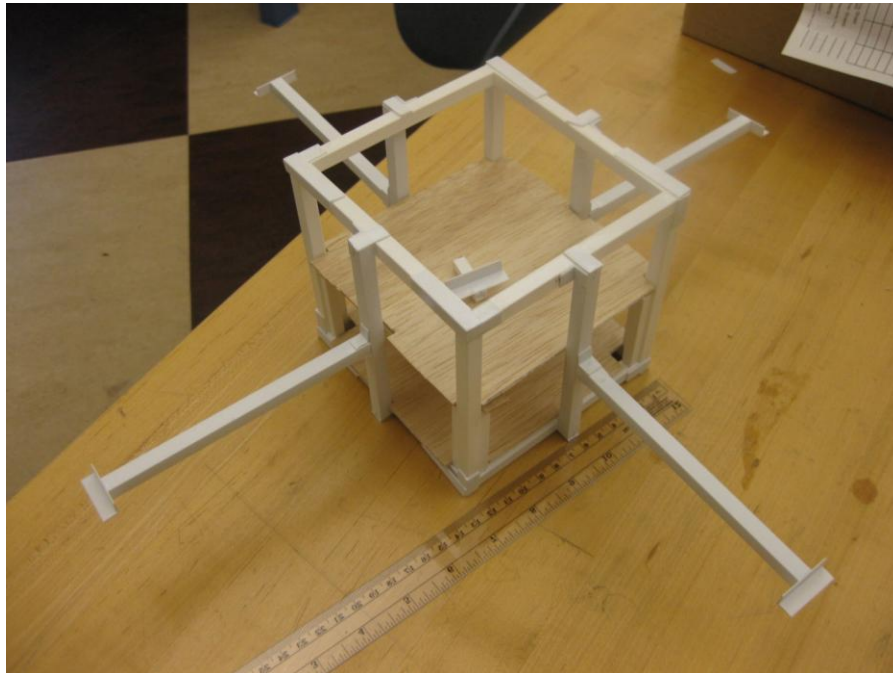


Figure 1: Blimp Gondola Version 1

Version2, in Figure 2, is a squat double deck cube. The deck spacing is just enough that the Arduino and xBee hardware will fit comfortably. The mass of the version2 gondola is 49 grams, extremely close to what was estimated. The motor arm length was kept at 20 cm, to meet the design requirement. But their attachment was redone. By attaching the arms to a horizontal strut rather than a vertical one the arms are much more stable. The motor to move the blimp in the z-direction is 10 cm from the bottom of the blimp. This is enough space to mitigate any unwanted disturbances caused by air flow against the gondola. The main structure is composed of 7.9 mm square styrene tubing. Plastruct Bondene is used to attach the styrene tubing together. An epoxy will be used to attach the motor to the motor arms.

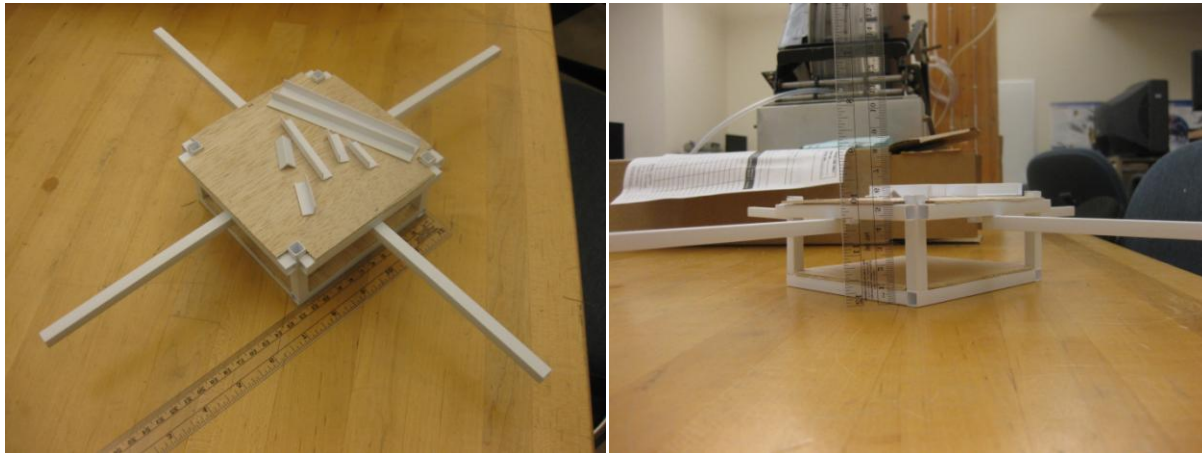


Figure 2: Blimp Gondola Version 2

The first motors we chose had ducted fans, due to their increased efficiency. After evaluating the motors and propeller it was decided upon to go with a bi-directional propeller blade. When the bi-directional propeller was attached to the motor used for the ducted fan, they over heated and burned out. New motors were found that are good for running bi-directional propellers.

A thrust and torque test was done on the new motors. The estimated thrust was used in the blimp's model and simulation. The resulting curves from the motor thrust test were included in Milestone Report 3. The motors will be running in two different directions, which, at 6 V, produce two different thrusts. To produce the same amount of thrust in each direction, the dominant direction will have to be run at a lower voltage. The maximum thrust the motors can produce in the weaker direction is 0.25 N, running at -6 V. To achieve this same thrust in the dominant direction, the motor will need to be run at 4 V. The torque test proved that the motor torque is negligible and therefore can be removed from the simulation for simplification purposes. The maximum torque produced by the motors is 0.023 N-m. Each motor will be attached to the gondola at the end of the motor arm by sitting in a type of two sided cradle, see Figure 3.

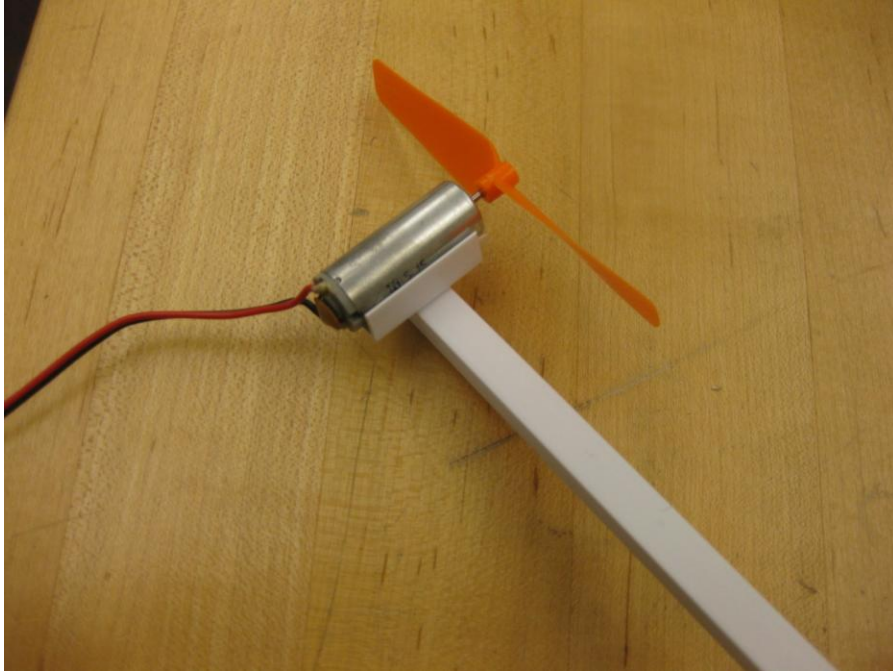


Figure 3: Motor Configuration

It was estimated that the total mass needed to be lifted by the balloon is approximately 0.4 kg, as detailed in Milestone Report 2. With a spherical balloon, this led to an envelope size of 1.0 m in diameter, using helium.

Originally, the envelope was going to be purchased from Mobile Airships, but upon talking to the company they do not make a blimp envelope small enough for our needs. A weather balloon was chosen instead. The weather is spherical and has a burst diameter of 1.03 m.

The gondola construction is complete. The next step is to work on efficiently filling the envelope to its required size and easily attaching the gondola to the envelope. A loop of string that is slightly less than that of the envelope diameter will be slid over the top of the envelope, four wires will extend from the loop of string to the corners of the gondola. The wire should be stiff enough to prevent compression and twisting, will still being light weight. The envelope will be hard to fill precisely, so we will overfill the envelope slightly and add a small amount of mass close at the center of gravity. The method for attaching said mass is still being investigated. One method could be to create a basket that can be filled with pennies.

Power Systems

After finalizing the blimp motor configuration and the microcontroller, the baseline requirements for the power system were set. In order to translate about the DSSL in any direction regardless of orientation, bidirectional control of the motors was deemed necessary. These factors dictated the remaining criteria:

- Operate from supply voltage of 5V (from the Arduino)
- Motor can turn in both directions when commanded
- Minimal overall weight
- Minimal overall power consumption
- Minimal total part count
- Minimal total price

Design Procedures:

List of initial components:

- One 3 cell 1350 mAh 11.1 V Lithium Polymer battery
- Five single 5 A H-Bridges
- One generic array logic GAL22V10D
- Two quad optocouplers
- One dual optocouplers
- Supporting diodes, resistors and capacitors

As the motor configuration matched that of the 2008 Blimp Group, the first step of the design process was to investigate and improve upon their original circuit. The first step in minimizing the weight for the current vehicle was to decrease the battery mass. The 2008 vehicle employed 4 7.2V Li-Polymer batteries to power their controller and motors. Even though it presented a decrease in operation time, it was decided to reduce this to a single 11.1 V Li-Polymer battery. This battery would directly supply the Arduino, which in turn would regulate the 5V supply required for the integrated circuits in addition to that required of the motors. There were several reasons why this particular battery was chosen. The normal operating voltage of the Arduino was between 7 and 12 V. Its minimum operating voltage was 6 V; however, it was unclear how the Arduino would behave between 6 V and 7 V. Moreover, Arduino board had an output voltage pin that would give 5 V that was equal to the logic voltages required to run the programmable logic chip and the H-bridges. This battery was cheap and light-weighted (94 g) while

having reasonable capacity and rate to charge and discharge compared to some other batteries. In addition, it can be charged using the same charger that another group operated had ordered, which saved the lab and us around \$100.

Secondly, the CN12-RXC motors chosen by the team based on the thrust requirement consumed 1.5 A at its peak tolerant supply voltage of 7 V; therefore, the original 3A 55V H-Bridge LMD18200 may get burned. So, to be safe, 5A H-Bridge TLE52052 made the list of components. These H-Bridge required two inputs and two outputs. Its different combinations of inputs would produce different combinations of outputs to make the motor turn clockwise, counter clockwise or stop as shown below in Table 2.

Table 2: 5A H- Bridge TLE52052's Original Truth Table

Input 1 (PWM signal)	Input 2 (Direction)	Output 1	Output 2	Action
L	L	H	L	Motor turns clockwise
L	H	L	H	Motor turns counter clockwise
H	L	L	L	Brake
H	H	Z	Z	Open Circuit

This kind of H-Bridge, unfortunately, accepted CMOS/TTL compatible inputs and had a rather unusual truth table combination for the inputs as seen above, which a basic programmable or generic array logic chip (PAL/GAL) could help fix as shown below in

Table 4 (The truth table that the generic array logic chip had inside its code is shown in Table 3). There are several options to choose from the EE store. It is anticipated that there would be five motors, so chip GAL22V10D with fourteen possible inputs and ten possible outputs seemed to fit perfectly.

Table 3: GAL22V10D's Truth Table for TLE52052

Input 1 (Old PWM signal)	Input 2 (Old Direction)	Output 1 (New PWM signal)	Output 2 (New Direction)	Supposed Action
L	L	H	H	Open Circuit
L	H	H	H	Open Circuit
H	L	L	L	Motor turns clockwise
H	H	L	H	Motor turns counter clockwise

Table 4: Circuit's Truth Table after Inserting GAL22V10D

Input 1 (PWM signal)	Input 2 (Direction)	Output 1	Output 2	Action
L	L	Z	Z	Open Circuit
L	H	Z	Z	Open Circuit

H	L	H	L	Motor turns clockwise
H	H	L	H	Motor turns counter clockwise

Even though the TLE52052 H-Bridges were made to deliver up to 5 A, after the component orders were delivered and tested, it was clear that they got hot very quickly. Their overtemperature protection would have shut them down after 7 – 10 mins of running the motors continuously at average to maximum voltages; hence, the circuit needed another kind of H-Bridges, the 4A dual H-Bridge L298N that seemed not to get as hot as rapidly as the TLE52052.

The 4A dual H-Bridges, L298N, have proved feasible. They still reach high temperature after the motors continuously ran for a while, but they did not get above the thermal limits. These dual H-Bridges could receive signals directly from the Arduino board. The truth table for each combination of input signals for the L298N is included in Table 4. However, they required three inputs instead of two like other H-Bridges that were looked at, while the Arduino board did not have enough input pins for all five motors. So, the GAL22V10D was put a new code as shown below in Table 6 so that there were only two inputs for each motor needed to give out from the Arduino as seen in Table 7.

Table 5: 4 A Dual H-Bridge L298N's Original Truth Table

Enable (PWM)	Input 1 (Positive Input)	Input 2 (Negative Input)	Action
L	Z	Z	Power Off
H	L	L	Brake
H	H	L	Turns clockwise fast
H	L	H	Turns counter clockwise fast
H	H	H	Brake
P (Pulse)	H	L	Turns clockwise slow
P (Pulse)	L	H	Turns counter clockwise slow

Table 6: GAL22V10D's Truth Table for L298N

Input 1 (Direction)	Output 1 (Positive Input)	Output 2 (Negative Input)	Action
L	H	L	Turns clockwise slow
H	L	H	Turns counter clockwise slow

Table 7: Circuit's Truth Table after Putting New Code in GAL22V10D

Input 1 (PWM)	Input 2 (Direction)	First Output from 2 (Positive Input)	Second Output from 2 (Negative Input)	Action
L	Z	Z	Z	Power Off

H	L	H	L	Turns clockwise fast
H	H	L	H	Turns counter clockwise fast
P (Pulse)	H	L	H	Turns counter clockwise slow
P (Pulse)	L	H	L	Turns clockwise slow

At initial calculations, to go with the TLE52052 H-Bridges, there would be ten optocouplers to accompany two inputs of each of five motors, which two quad optocoupler PS2501-4 chips and one dual optocoupler PS2501-2 chip would have been enough. These optocouplers' job was to provide an electrical isolation between the Arduino board with the GAL chip and the H-Bridges with the motors, and thus protect both sides from getting affected just in case something went wrong on one of these two sides. However, since the current dual H-Bridges needed three inputs for each motor, the optocoupler chips were switched to four quad optocoupler PS2501-4.

It turned out that ordering a customized PC board was expensive and many companies did not even do small boards with simple designs like ours. So, at first, a regular PC board was bought from Radio Shacks and the circuit was going to be soldered on this board. Yet, with the initial schematic, it was difficult to place all necessary resistors that were connected to ground without crossing some other wire paths. Although there was an engineer who could help make a customized PC board in the AA department, it was found out not to be possible to draw a single layer PC board schematic for this design, so it was determined to have the optocouplers removed completely.

In the end, our electrical design only contains three main components along with some supporting resistors, capacitors and diodes:

- One 3 cell 1350 mAh 11.1 V Lithium Polymer battery
- Three 4 A dual H-Bridges
- One generic array logic GAL22V10D

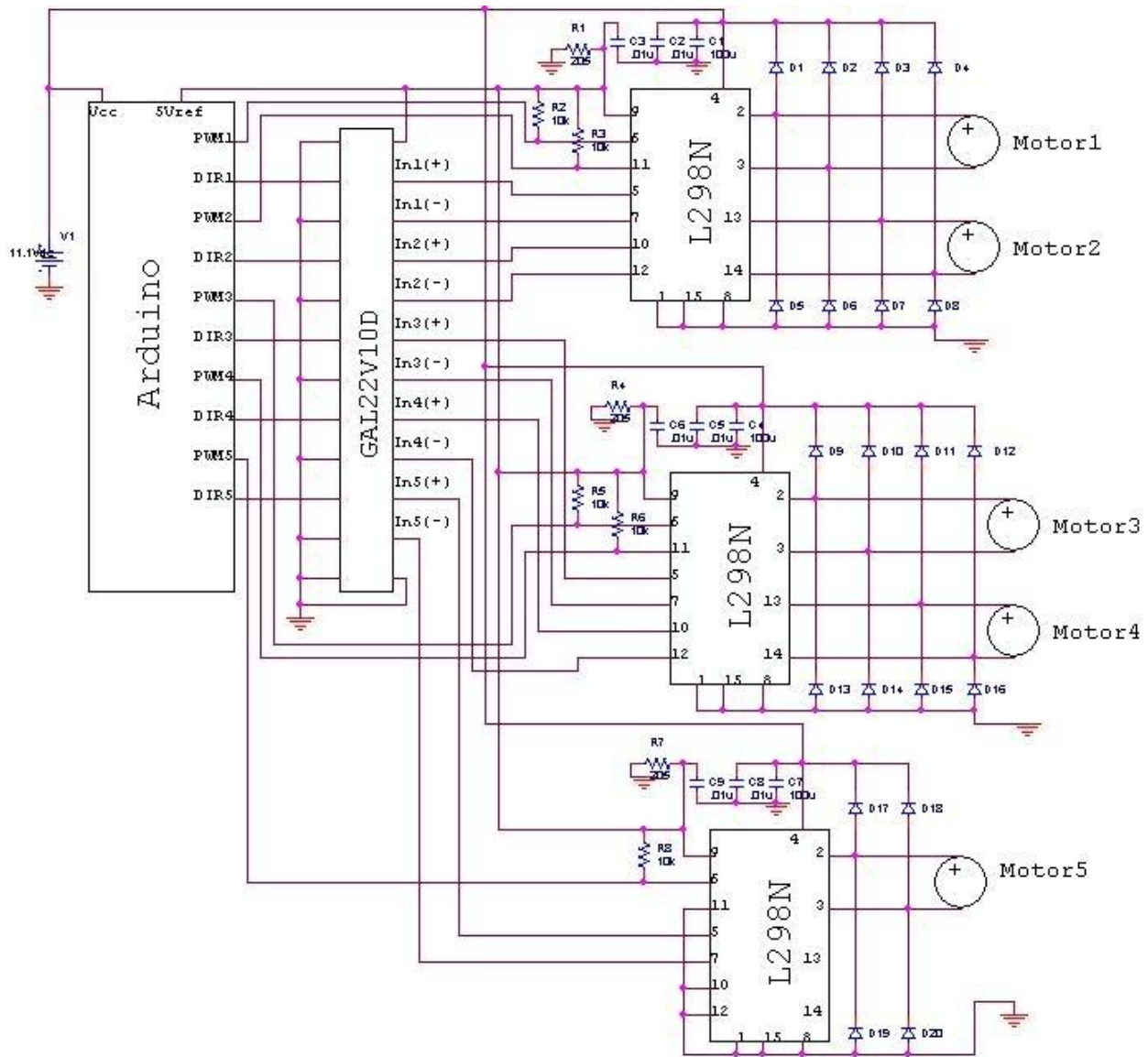


Figure 4: Circuit diagram for Blimp

System Architecture

The current system architecture for the blimp project is shown in Figure 5. The initial plan for the software implementation was to run the collection of sensor data and the control loop in Simulink, and then send the output data to the blimp via serial connection. Data is gathered from the VICON using a client programmed in Matlab; the client is capable of polling data and storing it, or sending it via serial, but it was not able to interface in real time with Simulink. For this reason, we next attempted to utilize the C# VICON interface which was created by the DSSL. Though the C# interface is fully functional, our team members are more comfortable working with Matlab. The customer also stated that a working

system using Matlab would be desirable for future use by the lab. For these reasons, it was decided that the final system architecture would implement the Matlab VICON client to poll data and send it to the Arduino without using Simulink. The controller is implemented entirely on the Arduino itself. Once the current and reference positions are received, the internal states of the system will be calculated using the estimator designed using the linearized system. After the full state of the system is known, the controller will be implemented in a code using simple matrix operations.

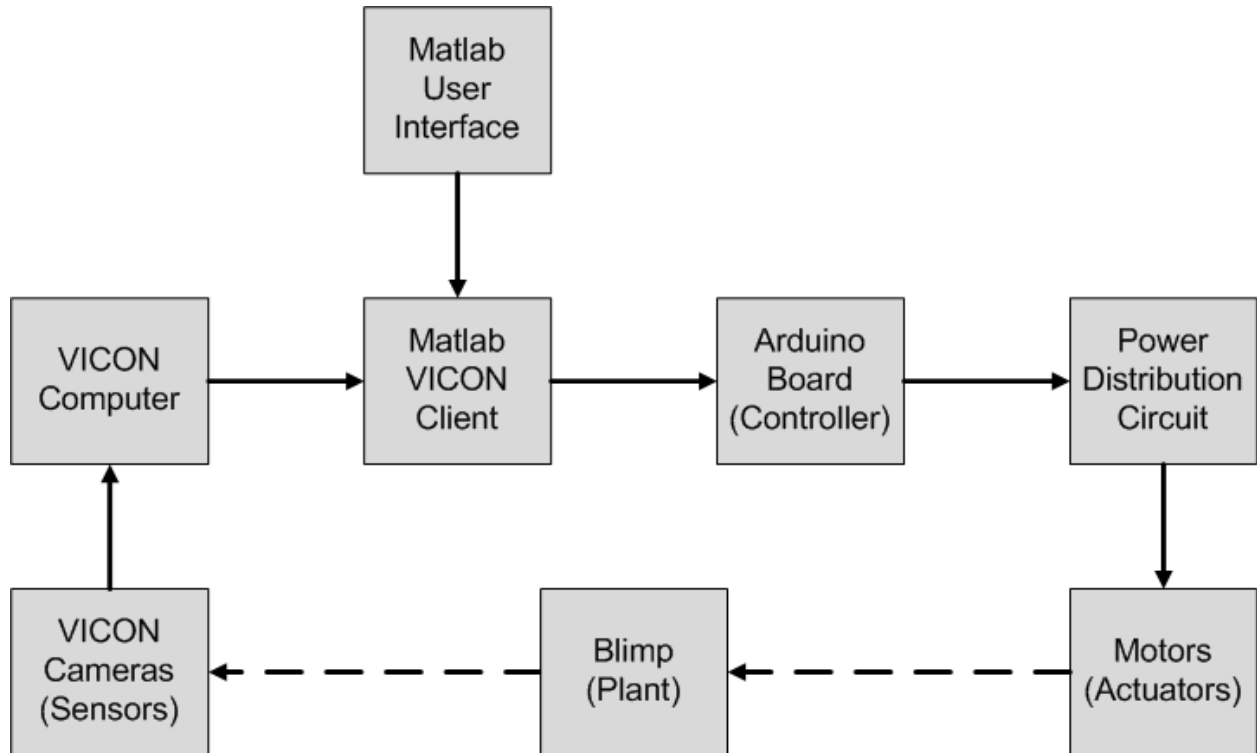


Figure 5. High Level Blimp System Architecture

VICON Interface

The interface with the VICON camera system is provided through an existing Matlab client. The client runs through a file which continuously loops, polling data from the VICON computer as it iterates. The data sent to the Arduino includes both the current position of the blimp (as measured by VICON) and the desired reference position towards which the blimp travels. In order to allow changes to the reference position while data is streaming from VICON, a graphical user interface was created. This GUI allows the user to input a desired position in x, y, z, and yaw, which is then read by the looping file and sent to the blimp and controller. A picture of the interface is shown in Figure 6.

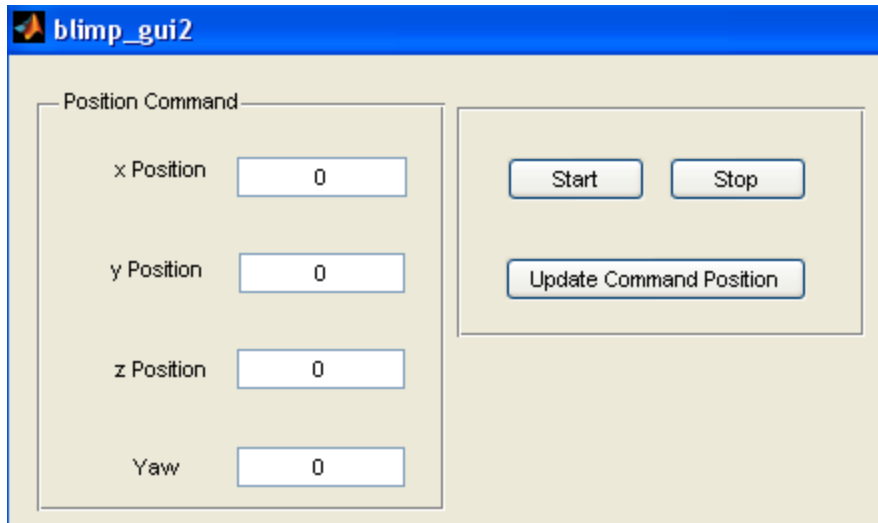


Figure 6. Graphical user interface for blimp reference position updates during flight

Once the data from the VICON and the GUI have been gathered, it must be sent via serial to the controller. This requires that the data to be sent be broken down into bytes that may be easily read by the Arduino.

Onboard Controller Implementation using Arduino

The first task of the Arduino is to receive new data over the serial connection, and reassemble the bytes into meaningful numbers. After this is accomplished, the Arduino will run through a series of tasks to calculate the control voltages for the motors. The basic flow chart of these operations is shown in Figure 7. Control calculations are accomplished using the linearized system model, which allows all the computations to be done using matrix multiplications.

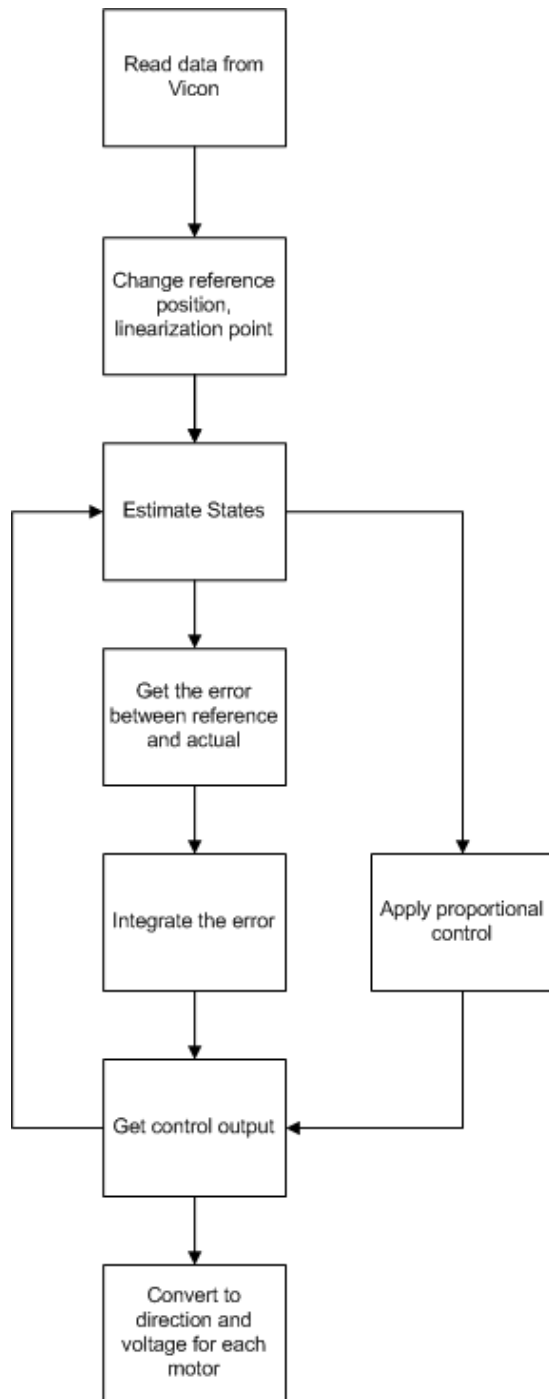


Figure 7. Schematic showing flow of data through Arduino control program

The controller on the Arduino will also include some specific provisions for the implementation of the controller in hardware. In order to implement bidirectional motor actuation, hard limits must be applied to

the output voltages calculated by the controller. These limits are in addition to the saturation limits that will be imposed to protect the motors. The controller currently outputs the necessary motor voltages as though the two motor directions perform the same. These voltages are then converted into duty cycles, and a down scaling factor is applied to signals in the forward direction (those with positive sign) to force the motors to behave in the desired manner. The end result will allow outputs from the controller in either direction to create the same net force with no dependence on directionality.

Bibliography

1. Bestaoui, Yasmina and Hamel, Tarek. *Dynamic Modeling of Small Autonomous Blimps. Methods and Models in Automation and Robotics*, Miedzyzdroje, Pl, Aug. 2000, vol. 2, pp 579 – 584.
2. Dorf, Richard C. and Bishop, Robert H. *Modern Control Systems*. Pearson Prentice-Hall, Inc., Upper Saddle River, NJ, 2005.
3. Heemstra, Brian. *Linear Quadratic Methods Applied to Quadrotor Control*. Department of Aeronautics & Astronautics, University of Washington, 2010.
4. Hughes, Kyle, et al. *Distributed Space Systems Laboratory Blimp*. Department of Aeronautics & Astronautics, University of Washington, 2008.
5. Lewis, Frank L., and Stevens, Brian L. *Aircraft Control and Simulation*. Wiley, Hoboken, NJ, 2003.
6. Loo, van de Jasper. *Formation Flight of Two Autonomous Blimps: The Atalanta Wingman Project*. Master thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2007.
7. Nise, Norman S. *Control Systems Engineering*. Wiley, New York, 2007.
8. Ogata, Katsuhiko. *State Space Analysis of Control Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1967.
9. Schutter B. De, *Minimal State-Space Realization in Linear System Theory: an Overview*, Control Laboratory, Faculty of Information Technology and Systems, Delft University of Technology, Netherlands 30 January 2000

Use of Resources:

1. Paper detailing the model determination portion of a project attempting a similar objective: autonomous waypoint tracking of a blimp vehicle. This was also used to validate our plant model with other research projects.
2. This textbook provided a comprehensive discussion for designing the state estimator.
3. Mr. Heemstra, a former Masters student in the Department of Aeronautics and Astronautics wrote his thesis on controlling a quadrotor vehicle for the DSSL. As the sensors for that project are identical to those used in for the blimp system, his thesis has been referenced over the course of the control design.
4. Mr. Hughes, a current Masters student in the Department of Aeronautics and Astronautics developed a similar blimp vehicle for this course in 2008. The final milestone report for that project has been a referenced over the course of the project.
5. This text book derived the transform matrices between body and inertial reference frames for vehicles in flight.
6. Thesis paper on similar project for formation flying blimp vehicles. This was used in validate our plant model with other research projects.
7. This was the textbook for our Introduction to Control Systems course, and provided the linearization procedure for state space representations, and the explicit equations for the controllability and observability matrices.
8. Used this reference to better understand the formal definitions of observability and controllability and the derivations behind the corresponding matrices. This source has been used throughout the course due the depth in its discussion of state space control methods.
9. This paper provided an overall discussion of the methods behind minimal realization of linear systems, and will be used when analyzing other operating points.