Micro Autonomous Systems Research

Georgia Institute of Technology

Annual Technical Report

June 6th, 2014

Dr. Dimitri Mavris

Dr. Zohaib Mian

Sean Lawlor

Jason Connolly

Pete Mangum Jr.

Tejas Kulkarni

Arturo Santa-Ruiz

ASDL
...from Vision to Reality

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
16-06-2014	Annual Research Report	20121201 - 20131230

4. TITLE AND SUBTITLE

Micro Autonomous Systems Research
A Methodology for Quantitative Technology Assessment and Prototyping of Unmanned Vehicles

5a. CONTRACT NUMBER

LaRC/ARL Coop NNL09AA0A

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Zohaib Mian, Sean Lawlor, Jason Connolly, Pete Mangum Jr., Tejas Kulkarni, Arturo Santa-Ruiz, Dimitri Mavris

5d. PROJECT NUMBER

2A31

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Aerospace Systems Design Laboratory, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Army Research Laboratory
RDRL-VTV
6340 Rodman Rd
Aberdeen Proving Ground, MD 21005

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This material is based on work supported by the National Aeronautics and Space Administration, Langley Research Center under the Research Cooperative Agreement No. NNL09AA00A awarded to the National Institute of Aerospace. Any opinions, findings,

14. ABSTRACT

A vital requirement of the modern combat environment is to gain and maintain situational awareness to facilitate effective squad-level decision making. Over previous years, Georgia Institute of Technology (Georgia Tech) has supported the Army Research Laboratory (ARL) in developing design capabilities to assess the operational capability of micro autonomous vehicles to assist at the squad level. In conjunction with Phase I of the contract between ARL and the Aerospace Systems Design Laboratory (ASDL), development of an autonomous vehicle has evolved to concept selection and building, with testing and validation to occur post fabrication. The vehicle will be chosen from a large set of platform and sensor configurations, to be assessed on criteria pertinent to the combat scenario where it will be implemented. The primary means for defining design selection criteria will be compliance with the Department of Defense Architecture Framework (DoDAF), which serves as the Operational Architectural definition for the team. This framework essentially relates the high level capabilities of any vehicle to subsequent activities, and in turn to specific functions. Capabilities, activities, and functions of importance are subjectively ranked for any specific mission, such

15. SUBJECT TERMS

Micro autonomous systems, matrix of alternatives, unmanned system, conceptual design, sizing, architecture, agent-based modeling, experimentation, scenario simulation, technology tradeoffs

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Eric Spero
Unclassified	Unclassified	Unclassified	UU	28	19b. TELEPHONE NUMBER *(Include area code)* (410) 278-8743

Reset

Introduction

A vital requirement of the modern combat environment is to gain and maintain situational awareness to facilitate effective squad-level decision making. Over previous years, Georgia Institute of Technology (Georgia Tech) has supported the Army Research Laboratory (ARL) in developing design capabilities to assess the operational capability of micro autonomous vehicles to assist at the squad level. In conjunction with Phase I of the contract between ARL and the Aerospace Systems Design Laboratory (ASDL), development of an autonomous vehicle has evolved to concept selection and building, with testing and validation to occur post fabrication. The vehicle will be chosen from a large set of platform and sensor configurations, to be assessed on criteria pertinent to the combat scenario where it will be implemented. The primary means for defining design selection criteria will be compliance with the Department of Defense Architecture Framework (DoDAF), which serves as the Operational Architectural definition for the team. This framework essentially relates the high level capabilities of any vehicle to subsequent activities, and in turn to specific functions. Capabilities, activities, and functions of importance are subjectively ranked for any specific mission, such that the need from a vehicle will change depending upon the medium of operation. The vehicle to be designed will have a primary mission of interior building reconnaissance (IBR) in a benign, urban workspace. However, there are multiple potential missions requiring additive capabilities, in which the vehicle may operate. Thus, the design chosen must not only fulfill the needs of interior reconnaissance, but must also address, at least at a high level, the ability to operate in diverse environments. In addition to this primary mission, the vehicle may also be used as a test bed to gather data that will help assess the feasibility of future concepts. The process used to design the appropriate vehicle, and the proposed experiments to validate vehicle performance, are discussed in more detail throughout the rest of this paper.

Mission Breakdown

A large amount of ambiguity, concerning the project, exists in customer requirements and in defining the customer specifically. While the focus will be on the needs of the Warfighter, the needs of the scientist or technologist must also be considered since they are the gatekeepers to technological advancement. The Warfighter requires more immediate situational awareness, at a level of detail that provides adequate, real time support. Improved situational awareness should assist in providing the Warfighter with tactical superiority to reduce risk of injury, and in turn improve mission effectiveness. To achieve this situational awareness, however, the significant gap between technologists, such as scientists

and engineers, and the warfighter must be addressed. Scientists understand the capabilities of technologies, but those capabilities do not necessarily match the needs of the warfighter, nor do scientists understand how a technology could be deployed. Warfighters, however, have a very specific need for some product or technology, but are unaware of what is available or feasible to address the required capability needs. By doing so, the optimized system will be accomplished, fulfilling the requirements and specifications of both scientists and soldier adapted to managerial prerequisites.

System configurations and technologies will be assessed in an environment analogous to a benign combat scenario without hostile forces. While the focus for Phase I will be in this benign environment, the chosen mission is ultimately a down selection to the bare roots of reconnaissance. Additional complexities and hazards may be added in further testing in non-benign environments, where hostile enemies will actively seek to interfere with vehicle operation, with less geometric boundaries, such as a cave or collapsed building, or mobile convoys of a single or more squads. By filtering the mission scenario to the bare roots, the vehicle may be better assessed to prove system capability before confounding of the vehicle and mission. Thus, the designed vehicle will traverse an established space to provide a detailed map of the room, including doors, windows, and obstacles contained within the center of the room. When assessing the performance of each technology alternative, it will be important to consider each of the mission scenarios and the impact they will have on vehicle performance. However, fully operational support in some of the described missions, such as a moving perimeter, will require a more networked system of systems application; where multiple vehicles operate in unison to provide stationary or moving perimeter surveillance. Further, it is evident that the first step in validating the vehicle performance must begin with a modest task less intricate than a swarm of vehicles in a moving perimeter. Most of the design and requirement focus will be for the benign urban environment as a foundation for future teams to expand from.

Experimental Setup

Primary Experiment: Indoor Area Exploration

The primary vehicle objective is:

- To explore at least 90% of a single room, by area
- Under nominal ambient lighting
- Benign environment
- Secondary objective to determine:
    - Total number of doors
    - Total number of windows
    - Number of other obstacles including tables, chairs, and humans.
- Perform the listed tasks in 10 minutes
- Reserve battery capacity of 2 minutes

Time criteria are based on the acceptable maximum time to acquire data for adequate situational awareness. Any longer and the Warfighter will remain in a potentially exposed position at short distance from hostiles, and any shorter would require too much from the vehicle. In addition, it is imperative that the system is not detectable, both aurally and visually. The noise levels for deployment and operation are:

- 70 dB at the furthest boundary of the mission space during operation
- 50 dB at the furthest boundaries of the mission space in deployment

This noise limit is based on noise perception by humans 50 dB, considered quiet, similar to a large transformer at 100 ft (outdoors). 70 dB is moderately loud and represents the noise associated with a radio or other household appliance.

As discussed, the main focus for this stage of design will be on the benign urban environment, so the major requirements put forth by the stakeholders, as well as those derived from the environment, which will ultimately influence the design include: mapping 90% of space in 10 minutes or less, with a 2 minute reserve. The vehicle must be reliable, rapidly deployable, and portable, the setup time should be no more than 5 minutes. The ability to hover, and a configuration which supports hover, such that the vehicle can occupy a hide site, or maneuver more covertly, is additionally essential. The vehicle must also be quiet and undetectable given the urban operational environment. Noise emissions represent the biggest gap in technology deficiency at current capabilities. The vehicle must navigate independently

throughout the space, which includes avoiding obstacles. Obstacle avoidance is twofold: first to decrease the noise emitted by the vehicle, and second to decrease damage taken by the vehicle.

Secondary Experiments

With the functional quadcopter, there are several tests that can be done to aid in the design of future quadcopters. A preliminary list of tests is shown below:

1. Power consumption vs RPM – This test will provide a more accurate understanding of the amount of power consumed by the system for a given propeller. This is useful in determining the endurance of the Quadcopter, and the optimal operating point of the motor-rotor combinations.
2. Lift vs RPM – Currently, the data provided is for singular points; nevertheless, it would be beneficial to understand the entire lift vs RPM curve. The development would further bring benefits by allowing the designer to attain optimal throttle settings for given propellers.
3. RPM/Propeller-pitch v Noise – There is currently very little data on the noise a motor or propeller produces while in flight. The first step of this experiment would be to determine which component influences noise creation the most; either the motor or the propeller. The second step would contain an understanding of ways to mitigate noise through design variables exchanges.

DoDAF Breakdown

As the vehicle will be operating in an urban war scenario, it is beneficial to utilize the Department of Defense Architecture Framework (DoDAF) which will allow for translation of high level capabilities to more specific system activities, which in turn define functions of the system. These major requirements roughly define a list of capabilities which the vehicle will need component control, positioning, surveillance, reconnaissance, deployment and recovery, and affect the environment. Component control focuses primarily on how an individual vehicle, even in a system of systems, is controlled autonomously. Autonomous control, the set of activities for this vehicle, includes routing plans from constructed maps of the surrounding terrain along with obstacle avoidance. Ultimately, this single system will additionally be able to manage tactical information from command, other squads, or vehicles within the swarm. Positioning of the vehicle defines how the vehicle moves about the mapped space. Activities include occupying a hide site to avoid detection, evasion when in open space, and whether the vehicle moves in open space or more discretely. Surveillance of the mission location primarily involves establishing both ground and air entrances and exits, and establishes tracking on these locations based on the current vehicle position. Autonomous awareness of the vehicle location will be important for efficiently navigating the three dimensional space. Reconnaissance then follows surveillance, where the vehicle will enter a mission space to complete a map of the immediate entry area and the rest of the environment, time permitting. It is important to recall that the reason for the vehicle is to provide information to the warfighter prior to entry into a space. Thus, as an additional activity within the reconnaissance capability, the vehicle must identify key obstacles and overall traversability such that the warfighter is given realistic path options through the space. Deployment and Recovery simply define the manual portions of the mission, where the vehicle must be deployed by the warfighter, where it will then surveil, infiltrate, and reconnoiter the mission space. The vehicle must then return, or recover, to the starting location. Recovery is ideal because of the reduced cost in procurement of additional vehicles. Affecting the environment is a secondary requirement at this level, where the vehicle is not expected to be lethal, but instead passive. The next level above passive, before becoming lethal, would involve jamming enemy signals, creating distractions, or weakening gates and other entry points for the warfighter.

Concept Selection using M-IRMA

The next phase of the system selection involved mapping capabilities, and associated activities, to specific functions of the vehicle. This step was performed using the MAST Interactive Reconfigurable Matrix of Alternatives (M-IRMA). M-IRMA synthesizes mission-specific weightings of the capabilities and activities, and function specific weightings from a list of configuration alternatives, or technologies. Some of these alternatives involve the method of locomotion including track based, flapping wing, rotor, or any hybrid combination. Others include power source, sensor type, and communication mediums, each with a set of alternatives to evaluate and choose from. Mission-specific weightings essentially define which capabilities and functions are most important given the operating conditions of any mission. For example, in the urban environment, noise emissions are extremely important because there is typically little ambient noise, and thus a vehicle which emits a lot of noise will likely be discovered. In a jungle environment, however, there will likely be a significant amount of ambient noise to mask some of the vehicle noise emission. Thus, the importance of operating quietly is not so important for the jungle mission, and thus the functions that go along with quiet operation are not as vital, when compared against the urban environment. Specific to the Urban IBR mission, the most important functions include analyzing for ground and/or air entrance to ultimately generate an entrance plan, classification of objects or targets, determining covert movement plan, generate reconnaissance plan, generate interior map, generate trafficability pattern, and plot path to desired position and back to return area. Collectively, this function set defines most of the basic mission requirements to infiltrate, map, and exit an unknown space in a relatively short time period. Additional functions, such as creating interference or distractions, are not necessary at this basic level. Again, the goal of Phase I is to validate the single vehicle performance. The additional functions which were not ranked as high at this level may become more important as validation proceeds to more complex environments with more complicated tasks. The result of M-IRMA analysis was a series of ranked vehicle configurations for each mission scenario. While this analysis synthesized a complete vehicle, with all of the applicable power systems, sensors, and control unity, the main goal was to determine a common platform from the locomotion alternatives. Of the ground, air, and hybrid vehicles, the flapping wing and rotor configurations appeared most frequently. Further, the flapping wing platform comprised roughly 70% of the systems chosen. However, while the flapping wing was theoretically chosen as the “best” platform, the limitations of current technology do not allow the flapping wing to produce sufficient lift to support flight, even without any additional mapping or navigation sensors. Thus, despite the prominence of the flapping wing in the top ranked systems, a quadrotor platform was ultimately chosen due to extensive, proven performance, and reliability in operation. Either platform, flapping wing or rotor, would

allow for hover or forward flight, so the decision to utilize a quadrotor platform was ultimately a matter of reliability, and the proven effectiveness of the quadrotor system. Table 1 summarizes high level characteristics of the 10,000 configurations analyzed in the M-IRMA. As shown, only 5% of the 10,000 cases were not air based. Further down selection to the top 100 vehicles eliminated all but the Quadrotor and flapping wing vehicles.

Table 1: Concept Selection Summary

Category	Subsystem Technology	In top 10,000 Performers	In top 10,000 performers (%)
Locomotion	Hover Capable	9644	96%
Structure	Flex Joints	6828	68%
Power	Primary: Lithium-Ion	7530	75%
	Secondary: Fuel Cells (miniature)	8843	88%
Sensors	IMU/LIDAR	7713	77%

Additional sensors will be chosen based on the mission to be performed, and may ultimately be interchangeable. The interior building reconnaissance type missions one method of determining distance for mapping and navigation is LIDAR. Additionally, to control altitude, sonar was utilized. However in the future, stereo-optical sensors may simultaneously provide mapping and live feed of the mission space to the Warfighter, providing an alternate capability in tactical advantage. That is not to say stereo-optical sensors are superior, only that they offer additional capabilities that LIDAR does not.

Mission Based Sizing

Background

Previous MAST teams worked to create a modeling and simulation environment for a swarm of micro-air vehicles. The environment included regressions for weight estimations of the individual components of the vehicles as well as estimations for the flight characteristics: lift, drag, and power of the vehicle (either a flapping wing or a quadcopter).

The flight characteristics are calculated using a combination of the vortex panel method and blade element momentum theory. The 2013-2014 MAST team modified this environment to be suited for the analysis of a single vehicle. The code was corrected against known experimental data to ensure a high level of accuracy and reliability. Finally the tool was used to perform a design of experiments and the results are filtered to determine the solution that best fits stakeholder requirements. The overall sizing process is shown below.

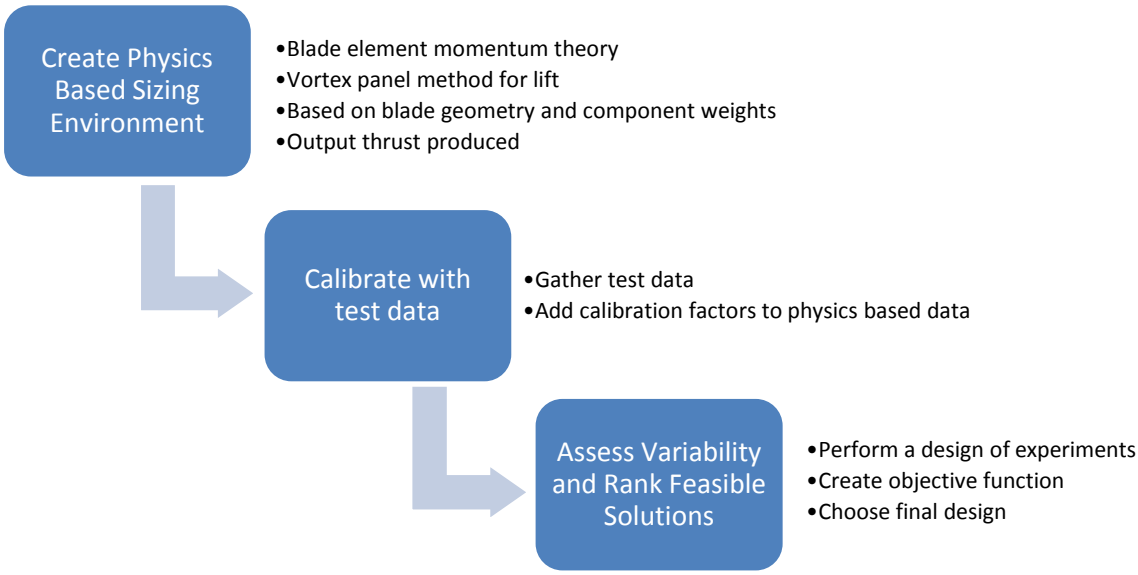


Figure 1: Mission Based Sizing Outline

Aerodynamic Background

The sizing code incorporates vortex panel method with blade element momentum theory in order to use any airfoil on the blade without knowledge of its lift and drag properties. This allows for rapid

development of designs as a full aerodynamic analysis does not have to be completed for each individual blade. In addition it is important to note that the sizing code does not accurately model flow separation, boundary layers etc. This leads to the need for calibration factors as discussed in the next section.

The vortex panel method is a common way to analyze the lift across a finite wing. It involves dividing the wing up into small 2-dimensional airfoil sections. A vortex sheet of strength γ is placed on the airfoil. The circulation is integrated over the airfoil section to estimate the lift and drag coefficients. The coefficients are then used with blade element momentum theory to estimate lift and drag of the rotor.

Blade element momentum theory (BEMT) divides the rotor into small airfoil elements to estimate the aerodynamic loading. With the lift and drag coefficients known, as well as the local angle of attack, BEMT can estimate the lift and drag of the rotor. However, rotors have an issue with inflow of the air due to the large vortices generated by the rotor tips. In order to correct for this, the effective angle of attack needs to be found; this is the pitch angle minus the angle generated by the inflow. With this angle, a better estimate can be found for the lift and drag of the rotor blade.

Experimental Correction Factors

The flight characteristics code, originally developed by Aaron Harrington at the University of Maryland, was calibrated against a known set of propeller and motor data[1]. A 13,094 set of propeller and motor characteristics were run through the code in order to obtain an estimate of the thrust produced by the particular propeller at a specific RPM setting. In running these cases, the following assumptions were made.

- No tip losses are present
- The propeller has a linear twist from 45º in the root to the pitch at the ¾ span
- Linearly tapered propeller blade

Running the known data through the code, there was an average of 6.09% error across all the cases; however, the distribution was very large, as shown in Figure 2: Distribution of Percent Error of Thrust Calculation, Original Code below.

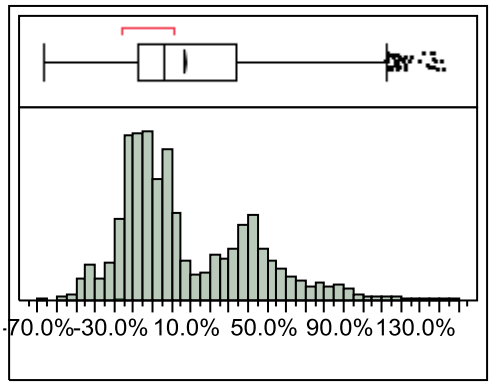


Figure 2: Distribution of Percent Error of Thrust Calculation, Original Code

To experimentally correct the code, the output of the sizing tool associated with the figure above was compared with known data1. The inputs to the code that had the greatest impact on the lift calculation were identified as blade radius, blade pitch, twist, and blade RPM. For each input a neural net based experimental correction factor was applied in order to match the output calculated thrust to the experimental thrust. After rerunning the same cases through the experimentally corrected code for lift calculation, an average of 0.90% error across all cases was achieved, with a much more normal distribution of percent error.

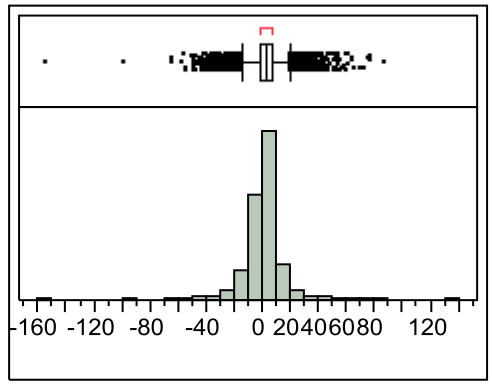


Figure 3. Distribution of Percent Error of Thrust Calculation, Corrected Code

Design Space Exploration

Following the experimental correction of the code, a Latin hypercube design of experiments of 10,000 cases was run through the code. It is used to find the propeller best suited for our mission. After filtering the results based on the selected motor and the required thrust, as well as minimizing the blade

radius and minimizing the absolute value of the pitch which reduces the noise output by the system. The resulting point is shown below, as compared with the entire design space.

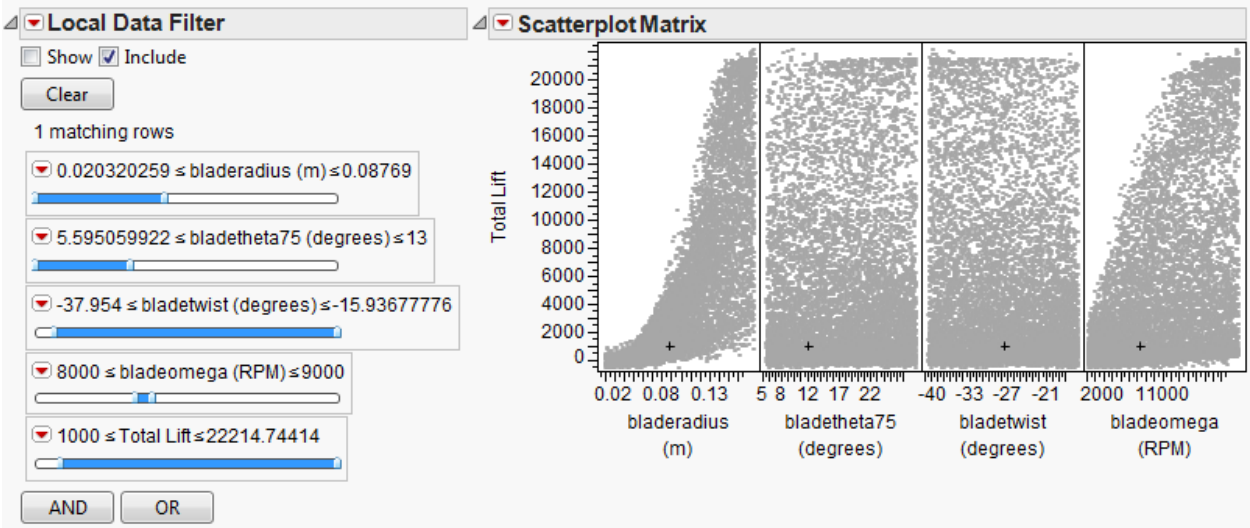


Figure 4: Code based Solution

The design parameters associated with the highlighted point are shown in the table below. Based on the analysis done using the available experimental data mentioned earlier, the team decided to purchase a 7x4.5 propeller, 7" (17.78cm) in diameter and 4.5" (11.43cm) pitch. The sizing code accurately predicts the minimum size propeller required to fly the entire quadrotor system.

Table 2: Filtered Propulsion Values

Design Parameter	Value
Blade Diameter (inches)	6.85 in (0.174m)
Pitch	12.01734º (4.58 in)(.166m)
Twist (degrees)	-27.46798379º
RPM	8251

Vehicle Design

Once the platform was determined to be a multi-rotor aerial vehicle; in this case a quadcopter, the specifics of its design were considered. With environment considerations and mission requirements taken into account; the payload included a LIDAR, sonar, and an IMU. Moreover, the focus moved to the integration of the entire setup in a device that would not only carry the aforementioned payload, but could properly achieve all other aspects of the mission as well. A list of possible choices of components was created and populated with commercially off the shelf components. Subsequently through an iterative sizing process involving thrust requirements, which will be explained below, the final design is composed of the following components found in Appendix A.

Platform Evaluation and Sizing

The choice of a specific frame to use in the design came about through an iterative sizing process. The reason being, that frame size affects gross weight proportionally, as it is scaled up and down. Moreover, W_0 creates a limitation on the rotor blades' size the autonomous vehicle can hold. The process is described graphically below

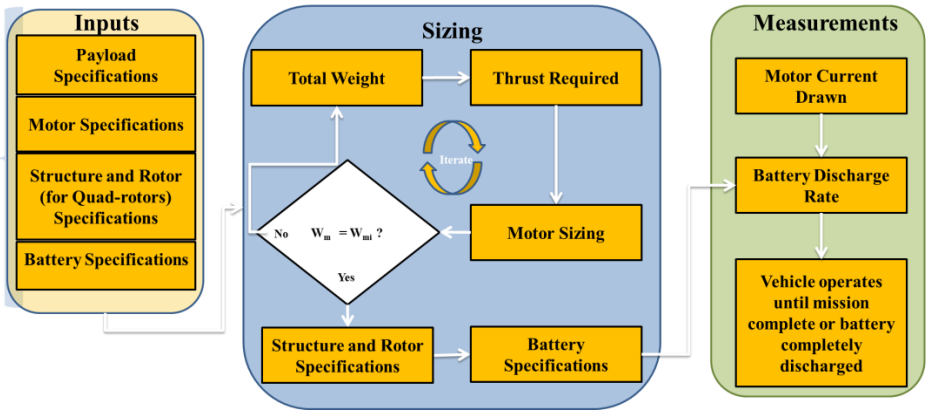


Figure 5: Sizing Process

Creating a Catalog

In the process of selecting the frame, motor, rotor, battery characteristics and more; a list of possible choices is in order. This catalog is populated primarily from products available at rctimer.com and hobbyking.com, because of reliable information and previous successful interactions. In order for the quadcopter to fit through a doorway the frame has to be less than 19.7 in (.5 meters). A compiled list of frames is displayed in Table 3. The motors were chosen to give a range of possible maximum RPM values and weights. These ranged from 35 grams to 200 grams with RPMs/V from 350KV to 2300KV; a sample of the list is

provided in Table 5. The rotors were researched thoroughly and chosen from 4in (0.1016m) to 10in (0.254m) diameters with varying pitch. The pitch available for each of the lengths is displayed in

Table 4.

Table 3: Frame Options

Frame options	Resizable (Y/N)	Height (m)	Length (m)	Width (m)	Area (m^2)	Weigh t (g)	Suggested Rotor Size (diameter)
Hobbyking X650F	N	0.265	0.55	0.55	0.3025	598	9" to 11"
HobbyKing Mini Quadcopter Frame V	N	0.0505	0.539	0.539	0.2905 21	238	9"
Turnigy Integrated PCB Micro-Quad	N	0.085	0.25	0.25	0.0625	145	5"
DIY Frame from foam	N	N/A	custom	custo m	custo m	180	custom
Blackout Mini H Quad Frame Kit	N	N/A	0.13	0.175	0.0227	90	5"
R450 Glass Fiber Quadcopter Frame 450mm	N	N/A	0.45	0.45	0.2025	500	8"
RM450 V1	N	N/A	0.45	0.45	0.2025	500	N/A

Turnigy Talon Carbon Fiber Quadcopter	Y	N/A	0.498	0.498	0.2480 04	240	N/A

Table 4: Possible Rotor Choices

Rotors

Length(in)	**Pitch**	**Source**
4.1	4.1	rctimer.com
4.5	4.5	rctimer.com
4.75	4.75	rctimer.com
5	5	rctimer.com
6	3	rctimer.com
	4	rctimer.com
	5	rctimer.com
7	3	rctimer.com
	4	rctimer.com
	4.5	rctimer.com
	5	rctimer.com
	6	rctimer.com
8	3.8	rctimer.com
	4	rctimer.com
	5	rctimer.com
	6	rctimer.com
9	3.8	rctimer.com
	4.7	rctimer.com
	6	rctimer.com
10	5	rctimer.com
	6	rctimer.com
	7	rctimer.com

Table 5: Possible Motors Choices

Motors

Name	**RPM/V**	**Weight (g)**
2725 Brushless Outrunner	1600KV	35
Motor AC2830-358, 850Kv	850 KV	62
HP2808(3530)	1400KV	90
HP2812	750KV	90
HP2812	880KV	90
HP2812	1000KV	90
HP2812	1300KV	90
HP2808(3530)	1760KV	90
HP2217	1500KV	90
HP2212	2300KV	90
HP2212	1450KV	90
A3530-8	1700KV	100

A3530-10	1400KV	100
A3530-14	1100KV	100
HP2814(3536)	1280KV	120

5010	530KV	130
HP2826(3548)	1060KV	200
24N22P HP4114	350KV	200

Iteration of alternative

The frame chosen for the first iteration was a somewhat large to ensure a large rotor could be utilized. Initial frame weighted 500grams and spanned 17.72'' (.45m); therefore, accompanying motor and rotors were large to generate the required thrust. However, this alternative provided excess thrust, and a new iteration was performed with a smaller frame. This process was continued until the final alternative was decided upon. The final design featured a carbon fiber frame weighing 240g. The frame design allows for resizing of the arms to be able to decrease its size while maintaining its structural integrity. Due to the lower frame weight, smaller motors and rotors can be utilized. A total reduction in weight decreases the required thrust compared to the original heavier frame. It is important and intuitive to note that battery size and ESCs were adjusted as required in order to provide the motors with enough power to drive them and generate the required RPMs.

Thrust Requirements

In determining the amount of thrust needed from the motor and rotor combination, the weights of the payload: frame, batteries, motors, rotors, and ESCs were defined. The breakdown of these weights can be analyzed in **Error! Reference source not found.**5. After determination of the vehicle's total weight, the desired thrust was calculated to be about two times the weight W_0. For this particular case the required thrust is equivalent to a weight of 2026 grams, allowing the vehicle to have enough energy to execute maneuvers.

The calculation of thrust from a given motor and rotor combination was approximated by empirical-historical data1 gathered. Based on the motor chosen and the battery voltage supplied, the maximum RPMs were determined. Next, the rotor size was given an initial guess. These values were compared to similar results obtainable from the experimental data. The result of the comparison provides an analogous thrust that should be generated by the combination. Afterwards, the resulting thrust was then compared to the weight of the vehicle. The final motor and rotor selections can be seen on Table 2. From the historical data comparison of a similar combination, each motor and rotor should be able to

produce an equivalent of around 680 grams of thrust, for a corresponding total of 2720 grams. Desired thrust will be achieved. A visualization of the proposed system is shown in Figure 6: Quad-Copter Isometric View below.

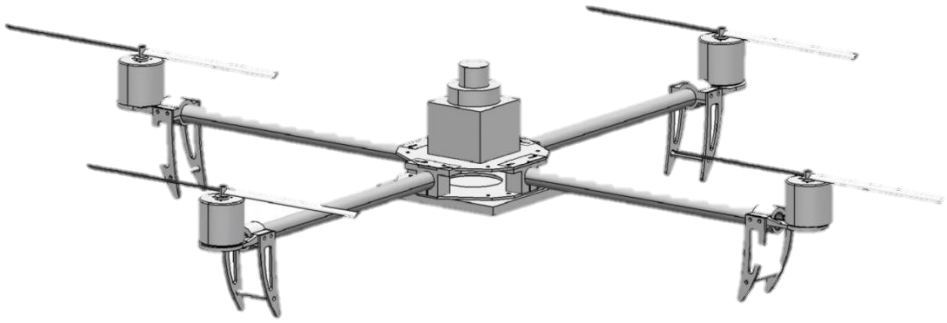


Figure 6: Quad-Copter Isometric View

Navigation Implementation

Family of algorithms

In considering the type of navigation to utilize, there were a few aspects to review. First, the general types of systems were explored: reactive and deliberate. The reactive systems sense the environment, and perform maneuvers based on the current information. The deliberate systems not only react to the environment, but also utilize knowledge about previous sensor data. While a deliberate system would be more ideal, to help eliminate the confounding effects on navigation logic and physical system performance, the simpler reactive system was chosen. The deliberate system will be implemented in future work.

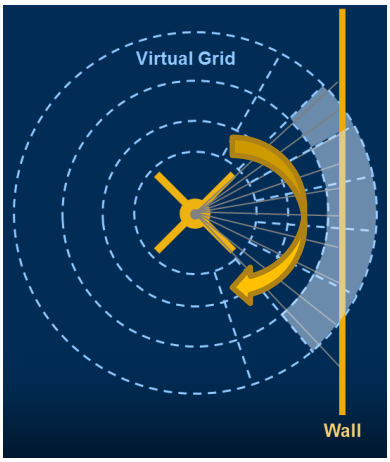


Figure 7: Reactive System where the quadrotor yaws right when too close to the wall.

Navigation script

In the implementation of the autonomous quadcopter, a type of reactive system in the form of a wall following script was utilized. In this, the quadrotor is set in an initial space, knowing nothing of its environment, except for distance readings taken from a LIDAR mounted on top of the frame. LIDAR will be discussed in detail in the Simultaneous Localization and Mapping Section below. At a high level, however, LIDAR is a laser range finder that takes 682 distance readings along a 240 degree arc, at 0.3519 degree increments, centered at the forward body line of the quadrotor. The current navigation implementation utilizes two LIDAR points, one directly forward of the LIDAR along positive X, and another along negative Y, in a North-East-Down (NED) body fixed frame. Upon initialization, the quadcopter translates forward by commanding a nose down pitch (negative pitch) until it reaches a "wall" in front of the body, or nose right yaw (positive yaw) when the "wall" is too close to the left of the body. The right LIDAR reading along positive Y is neglected so that the wall following algorithm anchors to the left side of the quadrotor. Including the right reading would over constrain the logic. Within the algorithm, there are distance thresholds which must be maintained, unique for forward and lateral, or left side. The higher priority threshold is forward, where on each navigation loop, the forward threshold is checked first, followed by lateral distance to the left. Once the quadrotor locates the target wall, and anchors to it, the quadrotor will pitch forward with the wall on the left side, as depicted in Figure 7, with forward toward the bottom of the figure. If the anchored wall falls outside of the distance tolerance, as when the quad reaches an open corner, the platform will yaw left around the wall so that the wall may again be anchored at the distance threshold, with the body X parallel to the wall. Additionally, if the quadrotor comes across an object directly in front, the quadrotor will halt any forward progress (i.e. pitch) and yaw until the object

is no longer blocking. This wall following algorithm was developed based on the idea of solving a maze. The quadcopter can make it through many different layouts for the building area as long as it can keep the wall to its side and avoid obstacle collision as it navigates the room.

Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) is a method by which two- or three-dimensional maps are generated with the fusion of LIght Distance And Ranging (LIDAR) and vehicle Odometry. LIDAR data represents how far away obstacles are from the vehicle, and is the radial distance from the center of sensing to that obstacle. LIDARs are essentially laser range finders on a mechanical wheel which spins at 10 Hz. LIDAR data at any single instant in time is comprised of the full sweep of laser ranges, ordered in a 1xn dimensional array, where n represents the number of scans generated. Each index of the data array corresponds to the angle at which that specific range was determined. Thus, with a single LIDAR data array, both angle and radius are available, and are internally generated by the LIDAR sensor. Odometry represents the vehicle position in space, and is important for vehicles that translate so that the origin of any LIDAR scan may be oriented appropriately. Odometry has three components: X position, Y position, and heading. X and Y position are distance relative to any orthogonal axes chosen by the user for convenience. The magnitude, or scale, of the odometry positions is important, and must be coordinated with the LIDAR scale otherwise the distances reported by the LIDAR will not sync with the approximated location in space given by odometry. Heading deals with the orientation of the vehicle in space at the prescribed X and Y position; essentially the direction of vehicle forward relative to the origin. Together, the position and heading form the vehicle pose.

The specific SLAM implemented is DP-SLAM[4], which SLAM functions by synthesizing “low” and “high” maps as the LIDAR scans progress through time. A single “high” map is maintained throughout operation with global data, while “low” maps contain local data taken at a variable timestep. As an example, the high map is always global, and contains LIDAR readings taken from algorithm implementation to the current timestep. Low maps, however, contain LIDAR data from the current timestep to an offset timestep, such that either one or more LIDAR readings may be combined into one low map. At the end of each low duration, the recently generated local map is added to the existing high map. Low duration influences processing time, as the time required to process images is significantly more than the time required to collect LIDAR data.

Changes to what would be considered the “native” DP-SLAM package were minimal, but necessary for the specific vehicle and implementation performed. To start, the assumed spacing of each LIDAR reading was adjusted from 1.0 degrees to 1.056 degrees, the reason for which is discussed in the LIDAR section below – it is due to the specific LIDAR implemented. Further, the quadrotor was required to operate SLAM in a “live” environment, where a map of the traversed space would need to be generated essentially as the vehicle was traversing that space. Most SLAM implementations are post-processors, where LIDAR and odometry data may be recorded during operation, but ingested by SLAM at a later time or location. Thus key locations within the SLAM were updated to include algorithms for handling the continuous acquisition of LIDAR and odometry data. With the limited time to complete the entire system design, synthesis, and testing, a temporary method by which the navigation algorithm read and recorded LIDAR and odometry data to separate files at each LIDAR update was implemented. As a result, the two systems were allowed to operate independently, which was the driving requirement for any algorithm development. Due to the RAM requirements of SLAM, and the processing capabilities of the single board computer, SLAM required nearly 400 milliseconds to complete one iteration. At that rate, the navigation algorithm would not have been able to control the quadrotor or prevent collision. Further, the purpose of the navigation algorithm at a basic level is to run much faster than every other component to continuously ensure accurate position estimation and obstacle avoidance. So again, the handling of LIDAR and odometry by SLAM could not hinder the update rate of the navigation component. LIDAR and odometry data were recorded to their separate files at a rate near 100 ms, the minimum time to compete a single LIDAR loop. Typically, 3-4 iterations of navigation, and thus LIDAR and odometry recording, were completed between SLAM updates. The final change made to the SLAM algorithm was in the low duration, which was previously discussed. Instead of coalescing a set number of LIDAR readings in to a single low map, the low map was allowed to iterate essentially without limitation, until a stop command was entered into the command console that signified the end of map generation. As a result, the lost processing time between low iterations in outputting a local map was removed, and the SLAM algorithm was able to remain up to date with the vehicle position. Inclusion of a set number of LIDAR readings into a low map resulted in significant delay between LIDAR readings, and ultimately resulted in a disjointed map that was not resolvable.

The data provided to SLAM from the LIDAR was composed of 181 evenly spaced points across the forward facing plane shown below. The 181 points are currently a requirement of the SLAM algorithm implemented, but future work may allow for variable sensors in a plug-and-play configuration with either

more than or less than 181 readings. If the vehicle were stationary, LIDAR measurements would be sufficient to describe a 2-dimensional space in proximity to the vehicle. However, since the system is translating, it was also necessary to understand where the LIDAR measurement was taken with respect to previous readings.

LIDAR

The LIDAR implemented was a Hokyuo URG-04LX-UG01, with a total detection sweep of 240 degrees. The native measurements start at an angle of -30 degrees from positive Y axis, which is denoted as 0 degrees in Figure 8, again in a NED body fixed frame, with a maximum sensing distance of 4 meters, and data points taken every 0.3519 degrees.

As mentioned previously, SLAM required the data to be conditioned into a pseudo 180 degree sweep comprised of exactly 181 points. In order to accomplish this, LIDAR readings were taken from around 30 degrees to around 210 degrees, relative to the figure above. The initial and final angles of reading were chosen to ensure that 90 LIDAR points were included to either side of the “forward” or +X reading. In addition, every third data point was taken to trim the data set down to 181 points. With a degree increment of 0.3519 between readings, every third reading results in a delta of 1.056 degrees, which is why the sweep was referred to as pseudo 180 degrees. 181 points at 1.056 degrees between, results in a total sweep angle of 191.08 degrees. What is ultimately important at this stage is the 181 points, including 11.08 additional degrees is of no consequence to the SLAM algorithm.

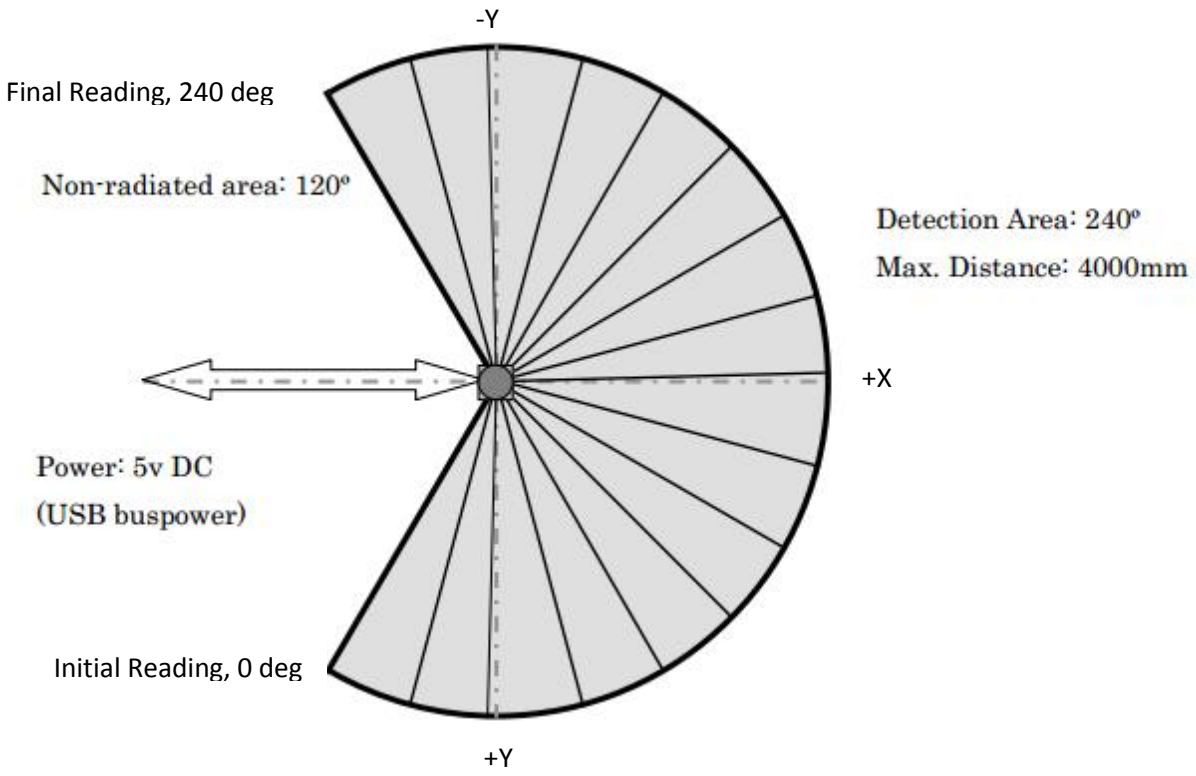


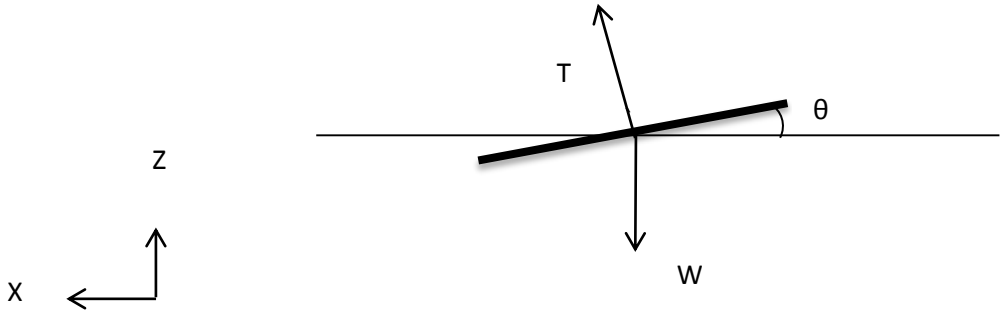
Figure 8: LIDAR information

In code, data was extracted from the LIDAR utilizing an available software framework from the LIDAR manufacturer, the URG framework, that allows a port to be opened and data streamed continuously utilizing the built in classes. Given that the LIDAR is a mechanical device, continuous reading occurred at 100 ms, the time required for the LIDAR to make one complete revolution.

Odometry

Collecting odometery information is a far more challenging task, due to the assumption that the vehicle is operating in a GPS denied environment. This means that GPS data was not available to determine position of the vehicle. Instead, onboard measurements were used to determine translation distance. The measurements available were translational acceleration and Euler attitude. It is important to note that the Euler angles are calculated from the accelerometer data internal to the autopilot along with a correction from the onboard gyros. Over the course of testing the autopilot system, the accelerometer sensors experienced significant noise associated with operation, and ultimately resulted in

very inaccurate position estimates. To avoid such significant noise, position was instead estimated with Euler attitude. The derivation is based on the assumption of steady flight, where lift generated is equal to weight, and the more important assumption that deflections will be relatively small, on the order of 1 degree or less. Note that the small deflection assumption was not the same as the typical small angle approximation. Small angle would assume that the forward component of thrust would be negligible for any pitch or roll angle less than 15 degrees. The small deflection assumption utilized instead seeks to minimize momentum generation. To adequately track odometry, this method assumes that any angle, positive or negative, directly influences translation. For example, if the vehicle is pitched forward at 0.5 degrees, the vehicle was assumed to be in forward translation. At the next instant, a negative, or backward pitch of 0.5 degrees would instantaneously result in rearward translation, without any further forward translating. This assumption would ultimately drive the autonomous navigation behavior, where commanded attitude would be strictly regulated and minimized to maintain the zero momentum condition. The diagram below provides a diagram of the force relations utilized to determine odometry.



$$T_z = W$$

$$T_z = T\cos\theta = W$$

$$T = \frac{W}{\sin\theta}$$

$$T_x = T\sin\theta = \frac{W\sin\theta}{\cos\theta}$$

$$T_x = \frac{W\sin\theta}{\cos\theta} = W\tan\theta$$

$$T_x = mg\tan\theta$$

Odometry is calculated under the assumption that a small change in pitch angle will result in a minute displacement, forward if the pitch is negative or nose down, or aft displacement if he pitch is positive or nose up. Total thrust will be increased such that the vertical component is equivalent to the quadrotor weight, which in turn will result in a prescribed forward thrust. No matter the deflection, the automatic throttling will maintain vertical lift, and the resulting forward thrust is a function of the vehicle mass and the angle of deflection only. From the final relation provided in the equations above, force, in this case X component of thrust, is related to acceleration by dividing out mass, utilizing Newton's second Law of motion. The resulting acceleration along the X axis is equal to gravity times the tangent of theta, shown below.

$$a_x = g \tan\theta$$

From this relation, position along the X is double integrated in time through velocity to position, which results in an estimate of change in position along X with respect to pitch change. The same relation was derived for bank angle phi, where the force relations provided above modified to include bank angle instead of pitch angle. The derivation is identical, and resulted in a similar estimate of lateral acceleration, shown below.

$$a_y = g \tan\theta$$

While each of these relations may suffice at a high level in perfect operation, each assumes that the other angle, i.e. the bank angle in the pitching equation or the pitch angle in the banking equation, is equal to zero. This assumption is erroneous, because the bank angle cannot be assumed to be the source of lateral displacement but also zero in the longitudinal or forward displacement. Thus, the equations must be coupled to include the effect of bank on the perceived thrust in the longitudinal or pitching axis, and the effect of pitch on the thrust in the lateral or banking axis. The result, summarized below, proved to be less sensitive to the noise of operation than the double integration with respect to time of pure translational acceleration.

$$\Delta X = -\frac{g}{2}\tan\theta\ \Delta t^2 \cos\psi - \frac{g}{2}\tan\phi\ \Delta t^2 \sin\psi$$

$$\Delta Y = \frac{g}{2}\tan\theta\ \Delta t^2 \sin\psi + \frac{g}{2}\tan\phi\ \Delta t^2 \cos\psi$$

Upon implementation of this displacement estimation system on the vehicle during flight, the assumption that deflection angles would be small, and that translational momentum would remain small and negligible did not stand. In both manual and autonomous flight, commanded deflections of 10 degrees resulted in a build up of forward velocity. While the magnitude of commanded deflections calculated by the navigation algorithm were expected to be on the order of single degrees, a counterproductive behavior was noted at commanded deflections near 5 degrees or below. The quadrotor continued to gather translational momentum, though less than in the 10 degree deflection, but was ultimately unable to course correct when confronted with a forward obstacle or when the anchored wall loomed too close. Either further tuning of the commanded deflections is required, or a more accurate position estimation method will need to be implemented. Preliminary literature review conducted prior to system design pointed to Kalman Filtering techniques applied to the previously discarded translational acceleration values. As a next step, these advanced Kalman methods may provide a more accurate and realistic estimate of odometry.

Pilot Hardware and Software

Open Pilot Interface

The OpenPilot board is an open-source system that can be easily modified to fit the needs of any autonomous system. The OpenPilot Revolution (Revo) board offers many useful tools for autonomous navigation. The board contains a 3-axis accelerometer, a 3-axis gyroscope, a 3-axis magnetometer, and a barometric pressure sensor. It also has built-in serial ports and a micro USB port for easy connection to a computer.

The OpenPilot has a very useful software architecture. To send messages from the Revo to the ground control station or the navigation script, it uses a protocol known as UAVTalk. This protocol was developed by the OpenPilot team and offers a very simple way to communicate information. UAVTalk uses UAVObjects to send data back and forth. These objects can range from current attitude to the battery

state or the next GPS waypoint. There are a predefined set of objects within the current OpenPilot software but a user can create his own objects that would fit a specific task or mission.

The UAVTalk protocol is built in to the OpenPilot ground control software for easy use but in order to access the information outside of the ground station, a software interface is necessary. To accomplish this, software that was developed at the University of Porto by Gustavo Olivera and his team was utilized. This software opened up a USB connection to the OpenPilot and used the UAVTalk protocol to transfer the UAVObjects from the Revo board to the program that was controlling the navigation of the system.

By using this USB API connection, commands were able to be calculated and sent to the Revo board using the UAVTalk protocol. These commands would then be translated into PWM signals and sent to the motors to control the quadcopter.

ArduCopter

The ArduCopter is an Arduino-based pilot board. It has a IDE for the development of custom firmware. This firmware was edited to enable an externally guided mode. This mode was able to control pitch and roll commands based upon the commands calculated by the navigation algorithm.

ArduCopter uses the MAVLink protocol to communicate with both the GCS and the navigation script. MAVLink has a set of predefined objects, similar to that of UAVTalk. Each object has its own identifier and a specific message. Objects and be user-defined to achieve better utility of the pilot board. By using MAVLink and the custom firmware. The quadcopter was able to be controlled by the on-board computer while still maintaining user control via the receiver.

Design Changes

During preliminary testing, there were large amplitude vibrations that were severely affecting the accuracy of the accelerometer and gyroscope data on the pilot board. This was because the pilot board was originally hard mounted to the quadcopter frame. In order to counteract these vibrations, the pilot board was soft mounted, using a block of foam to hold it in place on the frame. This allowed the board to be securely held in place yet still be isolated from the vibrations caused by the motors. The rest of the structure needed to be raised slightly to account for the extra space taken up by the foam. This raised the center of gravity for the quadcopter up slightly but still within the bounds for stability.

Conclusion

Over the course of the first six months of this project, previous work developed in the last years, has been leveraged into two products; the first a feasible design that is capable of completing the mission set forth by the customer, as well as an environment that can be used to size a Quadcopter based on experimental data. Over the next six months we plan on building the proposed concept and performing the mission as well as gathering more experimental data that is currently unavailable.

---

[1] D. P. Millener, "RC Propeller Thrust Curve and Efficiency Test Data," 07 May 2012. [Online]. Available: http://www.rctoys.com/pr/2012/05/07/rc-propeller-thrust-curve-and-efficiency-test-data-251-unique-propellers-tested-including-quad-copter-counter-rotating-propellers/. [Accessed Nov 2013].

[2] *Department of Defense Architectural Framework Version 1.5* s.l. : Dept. of Defense, 23 April, 2013

[3] Mian, Zohaib. *A multidisciplinary framework for mission effectiveness quantification and assessment of micro autonomous systems and technologies.* Atlanta, GA . Georgia Institute of Technology, 2013.

[4] E. Austin, R. Parr, “DP-SLAM.” Available: http://www.cs.duke.edu/~parr/dpslam/

Appendix A: Component Detail Sheet

Component Breakdown Sheet	**Total Weight (g)**			983
	Wire weight estimate (g)			30
	Total Weight (g)			**1013**
Frame				
Name	**Weight (g)**	**Length (m)**	**Width (m)**	**Notes**
Turnigy Talon Carbon Fiber Quadcopter Frame	240	0.5	0.5	Resizable arms
Motors				
Name	**Weight (g)**	**RPM/V**		
2725 Brushless Outrunner Motor 1600kv	35	1600		
Rotor				
Name	**Weight (g)**	**Pitch**	**Diameter**	
7x4.5E Nylon Multi-Rotor Propellers L/H and R/H Rotation (2 pairs)	24	4.5	7	
Battery				
Name	**Weight (g)**	**mAh**	**Voltage**	
Lipo Battery Black-B	190	2200	11.1	
Sensors				
Name	**Weight (g)**			
Lidar	140			
Sonar	8			
PandaBoard	77			
OpenPilot	20			
TOTAL	**245**			
Additional				
Name	**Weight (g)**	**Amp**		
ESC	18	18		