Forward-Looking Echoic Flow for Guidance of an Unmanned Aerial System

Undergraduate Thesis

Presented in Partial Fulfillment of the Requirements for Graduation with Honors Research Distinction in the Department of Electrical and Computer Engineering of The Ohio State University

By James Steven

The Ohio State University

April 2020

Thesis Committee

Dr. Inder J. Gupta, Advisor

Dr. Joel Johnson

Abstract

Echoic flow is a formula derived from natural phenomena that has the potential to control vehicles with great efficiency using range information. Initially studied in bats, echoic flow allows animals to use sonar as a navigation tool. Downward-facing echoic flow used in the vertical landing of an Unmanned Aerial System (UAS) has been studied in past research. Forward-looking echoic flow on a UAS could allow for new approaches to braking and following techniques in the horizontal plane of motion towards both fixed and moving targets. The goal of this project was to demonstrate forward-looking echoic flow guidance towards a fixed target using a quadcopter and to gather data showing the accuracy and precision of the process. In initial forward-looking tests, a modified Parrot AR Drone with an added ultrasonic sensor and Raspberry Pi were used as the UAS. Preliminary findings showed erratic and often inaccurate range finding measurements. These measurements were attributed in part to the inability of the UAS to aim directly at the small target. A software filter was designed to minimize the impact of erroneous measurements. Further tests conducted using a flat wall as the approach target still yielded trials that did not follow the ideal echoic flow approach accurately. In an attempt to improve the performance of trials, the equation used to convert velocities to motor thrust values was recalibrated. Though trial results did improve due to this modification, imprecise quadcopter movement control prevented the achievement of a smooth echoic flow approach. Finally, simulations of forward-looking trials were performed to test the impact of measurement and velocity error on the performance of echoic flow approaches. The values of measurement error that resulted in acceptable echoic flow performance were found to be lower than the expected values for the UAS in this study. Further forward-looking echoic flow research is recommended using a more accurate and robust rangefinder. A UAS capable of more precise horizontal plane movement is also recommended.

Acknowledgement

I would like to thank Professor Inder J. Gupta for his guidance and support throughout this research, and for making it a valuable experience even through unexpected circumstances. I would also like to thank Professor Joel Johnson for serving on the committee for this thesis. Finally, I would like to thank Justin Kuric for his advice on the Parrot AR Drone.

Table of Contents

List of Illustrations

I. Introduction

Echoic flow is a phenomenon identified in nature, specifically in bats [1], which can be used to control the rate of approach to a target. The much more thoroughly researched “optical flow” is a cousin of echoic flow. It is what allows humans and other animals to use vision to track self-position in an environment without performing computations [2]. Both echoic flow and optical flow are methods for controlling rate of approach, but optical flow uses light while echoic flow uses sonar/radar. Echoic flow in its simplest form can be represented by Eq. (1) [3].

$$\tau = \frac{r}{\dot{r}} \tag{1}$$

In this equation, all variables are functions of time with r representing the distance from the vehicle to the target, and $\dot{r}$ representing the change in distance between the vehicle and target (over a measurement time period). The resulting τ is a measure of time to collision between the vehicle and its target. The time derivative of tau, $\dot{\tau}$ is a dimensionless measure of the nature of the collision between the vehicle and the target. $\dot{\tau}$ can be obtained by taking the derivative of Eq. (1). This results in the second order differential equation Eq. (2) [3]. We can rearrange this equation to put it in terms of $\ddot{r}$, or in other words, acceleration. This formulation of echoic flow in Eq. (3) is the basis of what was actually used for control of the UAS.

$$\dot{\tau} = 1 - \frac{r\ddot{r}}{\dot{r}^2} \tag{2}$$

$$\ddot{r} = \dot{r}\frac{1-\dot{\tau}}{\tau} \tag{3}$$

By constraining $\dot{\tau}$ to different values, the velocity and acceleration of the approach can be controlled [3]. When $0 < \dot{\tau} < 0.5$, the UAS starts the approach breaking heavily, and gradually reduces breaking as it nears the target. When $\dot{\tau} = 0.5$, a constant breaking is applied over the approach. At $0.5 < \dot{\tau} < 1$, the UAS increases breaking as it approaches the target. Values where $\dot{\tau} > 1$ cause the UAS to accelerate into the target resulting in a crash. In this study, $\dot{\tau}$ is set to 0.5 for an even breaking approach. 0.5 was chosen because it offered the best results in previous real-world trials with a UAS in [4]. A graph of ideal echoic flow approaches using varying $\dot{\tau}$ values is shown in Figure 1.

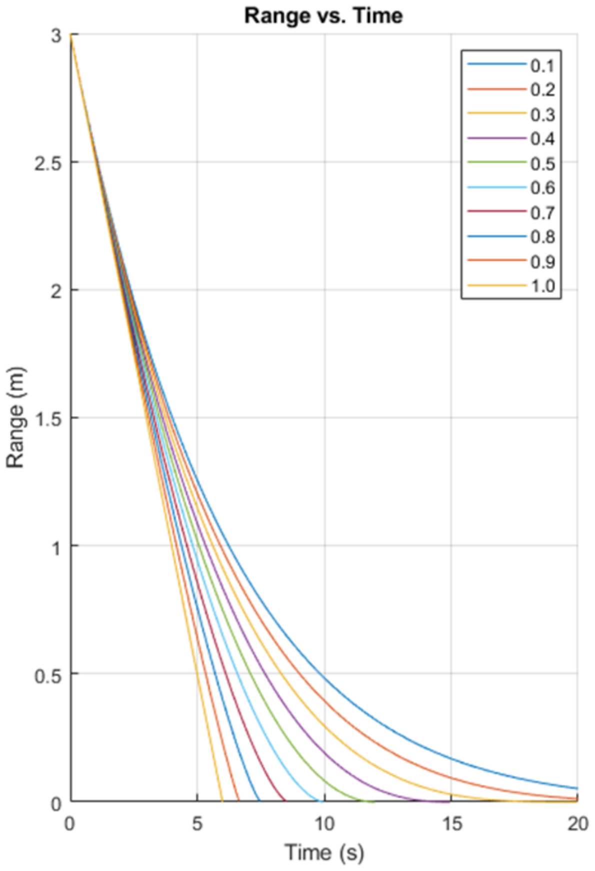


Figure 1: Echoic Flow Approach Using Constrained Tau Dot Values

The key takeaway from these equations is that for any UAS or vehicle to use echoic flow, it must be able to accurately measure two variables, time and instantaneous distance to target (r). The need to measure time accurately is clear as echoic flow is a function of time. The second variable, instantaneous distance to target, is used in conjunction with time between measurements to calculate instantaneous velocity ($\dot{r}$) and acceleration ($\ddot{r}$). The last requirement for a UAS to use echoic flow is the ability to control its velocity. Echoic flow approaches are guided by attempting to match the accelerations generated in Eq. (3). The only way to do this is by controlling velocity to achieve the target acceleration over a measurement period.

Past experimentation in [4] focused on the application of echoic flow in vertical landings of a UAS, specifically, a Parrot AR Drone 2.0. The drone had an onboard downward facing ultrasonic rangefinder for distance measurements and was controlled over Wi-Fi using a laptop. Experiments were successful in showing that the UAS was capable of landings using echoic flow for guidance. Further research was recommended in the implementation of a Kalman filter, and the exploration of forward-looking echoic flow.

The goal of the research presented in this thesis was to build upon the work in [4] to achieve accurate forward-looking echoic flow approaches to a horizontal-plane target. We used the same drone used in [4] with the addition of a forward-looking ultrasonic sensor, and Raspberry Pi. The integration of these components allowed the drone to take new distance data and fly without control from a laptop.

Over the course of the research project, several large obstacles to our goal were encountered. The first issue was a large amount of variance and incorrect measurements in readings from the rangefinder on the UAS. Through a series of experiments focused on rangefinder accuracy, we concluded that inability of the UAS to maintain position directly in front of the small target contributed to incorrect readings. We attempted to mitigate this by performing further tests with a wall as a much larger target. When met with poor echoic flow accuracy even after reducing incorrect rangefinder measurements, we re-evaluated the velocity conversion equation that converted m/s values to commands that the UAS could follow. After creating a new, more accurate conversion equation, we were still unable to achieve acceptable horizontal approach results under echoic flow guidance. Finally, we ran a series of simulations to test several types of error in echoic flow approach trials. We concluded that measurement error had a very high impact on echoic flow approaches and that a sensor error standard deviation of under 0.01 m was necessary for accurate approaches even with Kalman filtering.

In the next section, Equipment, the details of the modifications and additions to our UAS are explained. In the Forward-Looking Trials and Testing section, the echoic flow guidance experiments we ran are discussed. We also discuss issues we encountered and solutions we found in this section. Finally, the Conclusion presents a summation of our findings along with recommendations for further research.

II. Equipment

2.1 – Sensors and Components

The unmanned aerial system (UAS) used in this research is a Parrot AR Drone 2.0. It is a quadrotor aircraft capable of communication over Wi-Fi 802.11n protocol. For the forward-looking sensor, we selected the HRLV-MaxSonar ultrasonic rangefinder. This sensor has a manufacturer listed operating range of between 30 cm and 5 meters with a precision of 1mm. It runs on an operating voltage of 2.5-5.5V with a nominal current draw of 3.1mA (at 5V). The physical component itself takes up less than 1 cubic cm of space. The low power draw and small size made it a viable option for forward-mounting on the UAS. The final component was a single board computer capable of sending commands over Wi-Fi to the UAS. A Raspberry Pi 3 Model B was selected primarily for its integrated wireless module and ability to execute Python programs. This latter feature was particularly important as almost all programs used on the UAS were written in Python.

The first step in wiring the components was powering the Raspberry Pi, which would in turn provide power to the rangefinder. The UAS used a 1500 mA h LiPo battery capable of delivering enough power for approximately 18 minutes of flight. At a cost of additional flight time, we elected to use the UAS's flight battery to power the Raspberry Pi as well. This decision was made to avoid the added weight an extra battery would add to the UAS. A voltage regulator was soldered to the battery connection in series with a manual switch to control 5V power to the Pi.

To connect the Pi to the ultrasonic rangefinder, the V+ and ground pins on the sensor were linked to the corresponding pins on the Pi. The start/stop pin on the sensor that controls when ranging measurements should take place (discussed further in the communication protocol section) was linked to a general purpose I/O (GPIO) pin on the Pi. Finally, the serial output pin on the sensor was connected to the Raspberry Pi's serial RX pin. The full wiring diagram of the system is shown in Figure 2.

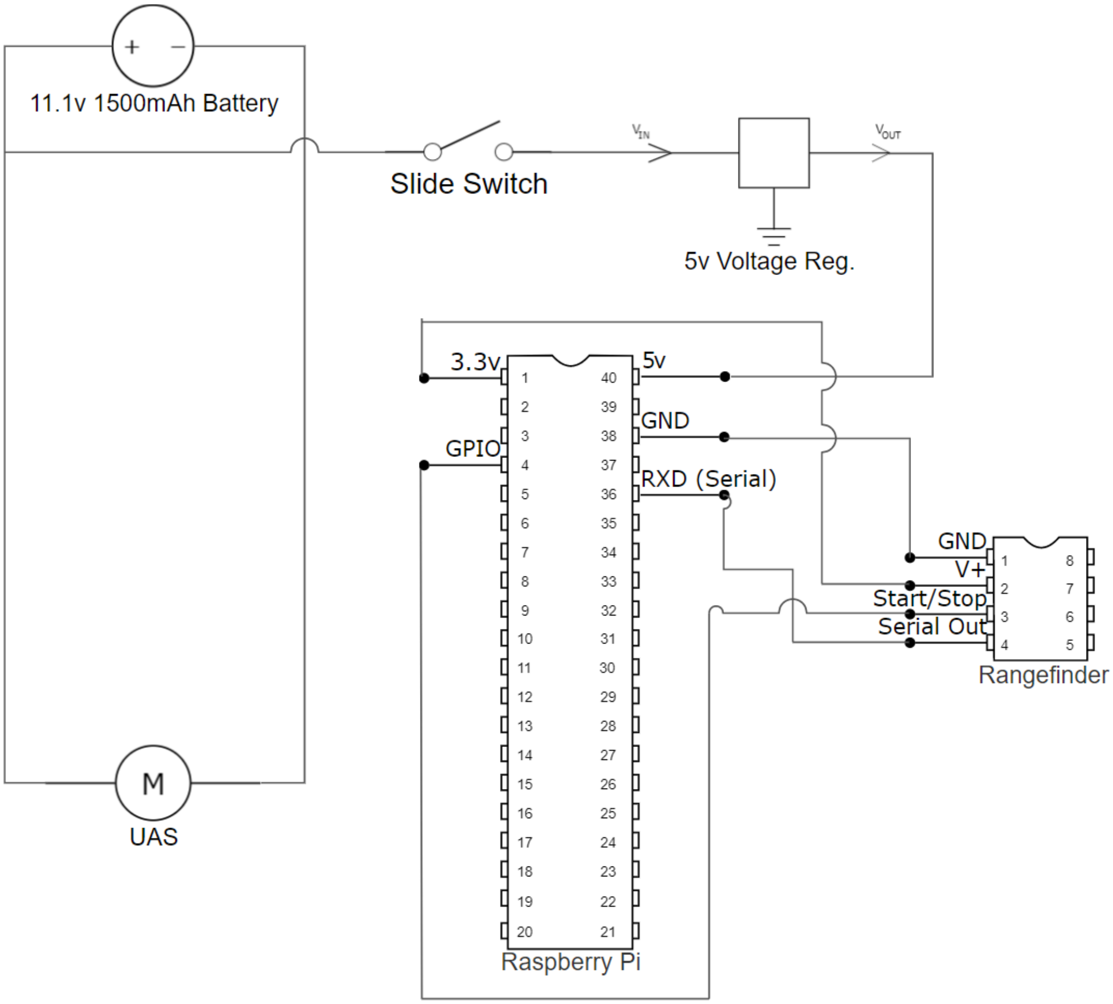


Figure 2: UAS and Rangefinder Integrated Wiring Diagram

2.2 – Communication Protocol

An RS232 serial connection with a baud rate of 9600 was used for communication between the Raspberry Pi and the rangefinder. The rangefinder had a measurement update frequency of 10 Hz capable of sending a new value on the serial link ten times per second. To utilize this 10 Hz update rate, we ran the sensor in its “free run” mode by holding the start/stop pin high.

On the Raspberry Pi, Python code was written to communicate to the rangefinder on the serial port. Within the Python program, the serial link was established once at the start of execution. From this point forward, the Raspberry Pi polled the rangefinder at a rate of 10Hz to match the update rate of the sensor. Results were returned to the Pi in 6-byte words. The start of the response was always the letter “R” followed by four digits corresponding to the range in millimeters. The response terminated with a carriage return. An example response for a range measurement of one meter would be “R1000\n”.

A Python library called PS-Drone was selected for communication between the Raspberry Pi and the UAS. This library provided several functions that allowed the Raspberry Pi to connect to and communicate with the Parrot AR 2.0 UAS. Among the most used functions were the takeoff and land commands, the calibrate function to reset the drone’s gyroscope, the retrieve flight data command that allowed sensing of the drone’s altitude from the onboard downward facing ultrasonic sensor, and the move command that allowed directional thrust setting.

Velocity control is critical to braking using echoic flow, so the move command within the Python library was used heavily. The issue with controlling the velocity of the UAS via move command was that the move command operates using arguments of thrust proportions, for example 20% thrust. Echoic flow equations, however, generate target thrust values in m/s. In order to convert from thrust proportions to m/s, we made use of a conversion equation carried over from previous echoic flow research using the Parrot AR 2.0 drone [4].

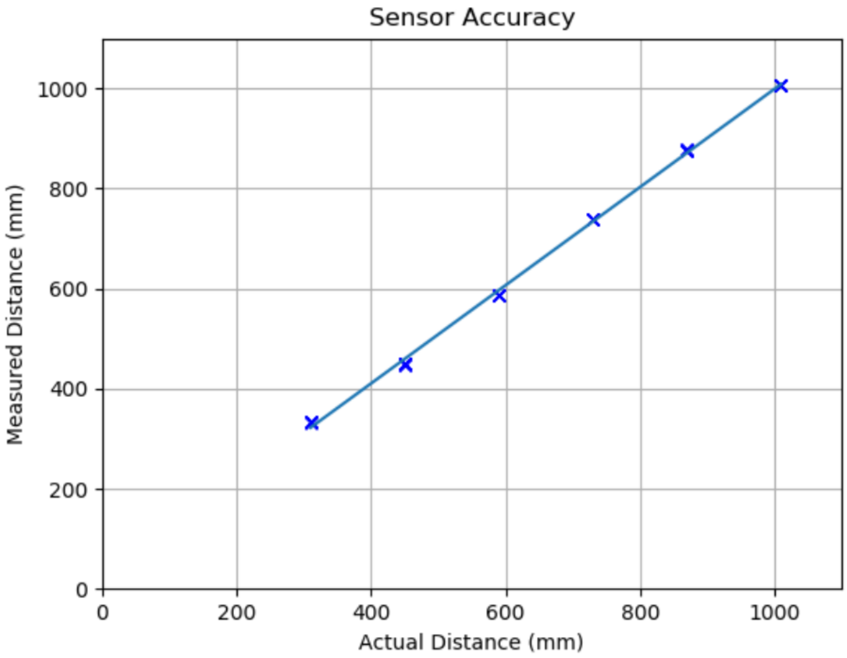
$$\text{Vertical Velocity} = -0.148 \, * \, Com.^2 \, - 0.65 \, * \, Com. \, - 0.004 \tag{4}$$

Equation (4) is quadratic and offers the necessary conversion. All calculations were performed on the Raspberry Pi and sent to the UAS over Wi-Fi.

2.3 – Integration Testing

To test how the addition of the forward-looking rangefinder and Raspberry Pi affected the UAS, several integration tests were performed. First to test the power impact of the new components, a takeoff command was sent to the modified UAS. After stabilizing at an altitude of approximately 4 feet, the drone hovered, and the Raspberry Pi pinged the rangefinder continuously at a rate of 10Hz until low battery forced a landing. The drone was able to function for approximately 8 mins 30 s, a roughly 50% reduction in flight time from the unmodified UAS. This was expected based on the Raspberry Pi’s nominal current draw of about 500 mAh. These results were deemed acceptable for further testing with the modified UAS.

Next, a simple experiment was run to test the ultrasonic sensor’s accuracy. The sensor was fixed horizontally at a height of 3 feet off the ground. Directly in front of the sensor, a 24”x18” rectangle of cardboard was hung off a thin metal rail. The experiment consisted of taking ten range measurements at six different distances from the cardboard target. The actual distances were measured with a tape measure. Aggregating absolute value measurement differences of the results indicate that the sensor distance differs from the actual distance by 7.6 mm on average (note that sensing at 31 cm had higher error because it is the lower limit of the sensor’s range). A graph of all the trials is shown in Figure 3 along with a linear regression. Finally, these data are displayed in Table 1 along with the standard deviation for each trial.



Regression Line:

*Sensor Value (mm) = 0.980 * Actual Dist (mm) + 18.3*

Figure 3: Ultrasonic Rangefinder Accuracy Test

Table 1: Aggregate Results of Ultrasonic Sensor Testing

Trial	1	2	3	4	5	6
Actual Distance (mm)	310	450	590	730	870	1010
Avg. Measured Distance (mm)	333.6	449.2	587.9	739.3	876.5	1006.5
Measured Distance S.D. (mm)	0.699	0.789	0.316	0.483	1.08	0.527

III. Forward-Looking Trials and Testing

3.1 – Rangefinder Variance

After completing the additions to the UAS necessary for forward-looking echoic flow, we began to experiment with stationary target trials. To perform these tests, a startup procedure for the drone was developed. This startup procedure was aimed at ensuring each trial was as uniform as possible. The procedure began with sending startup and calibration commands to reset the onboard sensors. Next the drone took off and hovered while trim calibrations were performed. Finally, the drone ascended to a height of one meter and moved forward or backward to position itself within 5cm of the selected horizontal starting distance from the forward target. In each measurement cycle (occurring at a rate of 10 Hz), the UAS collected a new distance reading, ran it through a smoothing filter, and compared it to the previous reading along with the time delta to find an instantaneous velocity. The instantaneous velocity and range were then used in Eq. (1) to find the current τ value. Finally, all of the collected values were used to calculate a target acceleration for the UAS from Eq. (3). In an attempt to match this target acceleration, a new command was sent to the UAS to raise or lower its velocity for the next measurement period. This cycle continued as the UAS approached its target until it reached a set distance threshold. At this point the UAS was sent a land command and the trial was complete.

Our first tests of forward-looking echoic flow used a 24”x18” cardboard target held at a height of approximately 1 meter. From the very first trial, it became immediately apparent that a large number of jumps in rangefinder measurements were posing an issue for the echoic flow algorithm. The flight graph from one trial is shown in Figure 4.

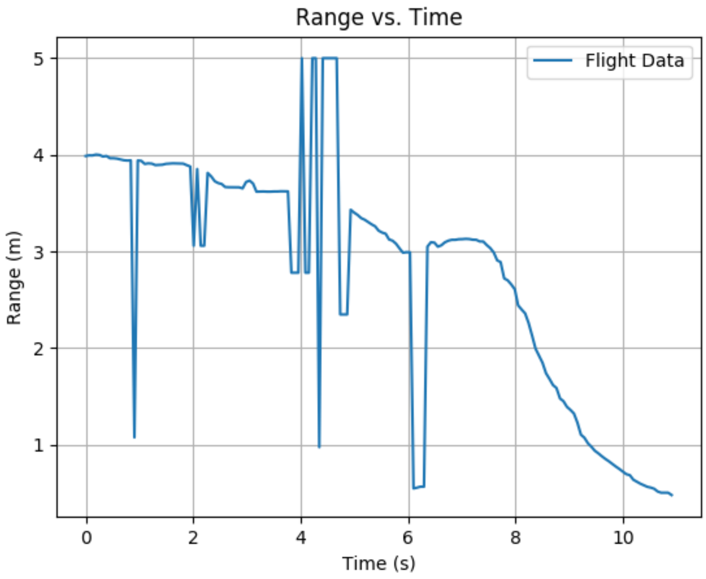


Figure 4: Forward-Looking Echoic Flow with Persistent Rangefinder Variance

In this trial, the UAS takes off at a distance of 4 m from the target, moves in a linear fashion to its starting horizontal distance of 3 m, then executes echoic flow until it reaches 0.35 m. Clearly within the first 6 s of the test, there were many jumps in rangefinder measurements. These large jumps in rangefinder measurements were present in almost every forward-looking trial performed thereafter. The reason the UAS was still able to approach its target stopping distance of 0.35 m despite the jumps in rangefinder readings was the least squares quadratic fit used in the echoic flow algorithm. This least squares regression takes a buffer of the last set amount of range readings and uses a quadratic fit to predict the next range. Through use of this method, the UAS was able to approach its horizontal target, though not at all with the speed and timing set by the echoic flow formula. For this reason, it was necessary to explore methods of mitigating the large jumps in rangefinder measurements.

We suspected that some of the incorrect readings given by the rangefinder were due to the UAS angling up, down, and to the sides during flight. The slight angling of the rangefinder up and down is inherent due to how quadcopters move forward. Angling of the rangefinder side to side is due to imprecise control limited by the capabilities of our Python library and UAS. The first step taken to confirm this theory was to perform continuous range finding tests while the UAS was not in flight. In these tests, the UAS was placed on a platform 1 m off the ground. While the UAS was immobile on the platform, rangefinder measurements were continuously taken as the 24"x18" target was moved towards the sensor. For the first two seconds of the trials, the board was held still. At this point the board was moved linearly towards the sensor until it reached a distance of 0.3 m. The rangefinder data from one such trial of this experiment is shown in Figure 5.

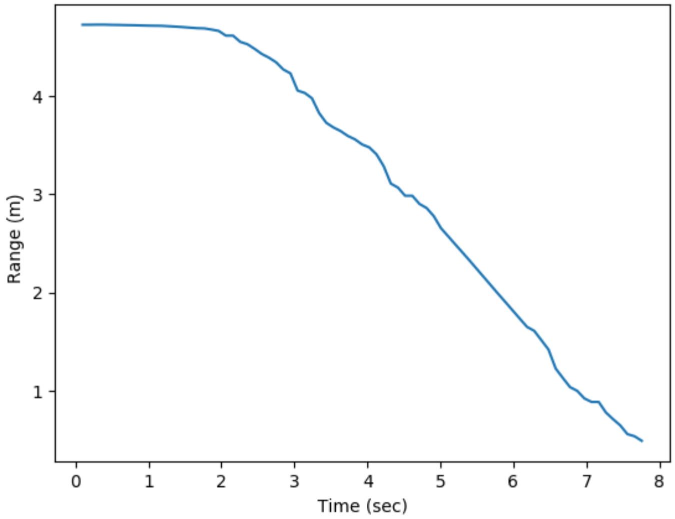


Figure 5: Static Forward-Looking Rangefinder Test from 5 m

Clearly the erratic measurements seen in the forward-looking approach tests were not present in this static trial. Other repetitions of this experiment offered similar results with few to zero jumps in range. With these results, we determined that the ultrasonic sensor itself was capable of stable continuous range finding measurements in an ideal situation with the sensor immobile and the target directly in line with the sensor.

3.2 – Target Size Testing

The next step we took to explore the erratic readings was to test the manufacturer stated beam characteristics of the ultrasonic sensor. Ultimately, we looked to determine whether the size of the target in conjunction with the rangefinder's inability to stay directly in line with the target had a significant impact on range measurements. Shown below in Figure 6 is the beam detection pattern for an 11-inch-wide board taken from the ultrasonic sensor's data sheet [5].

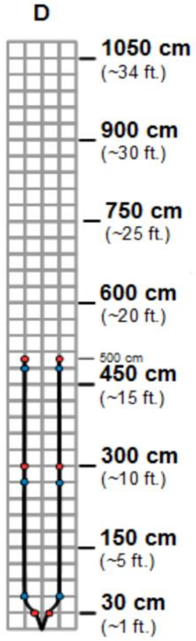


Figure 6: MB1013 HRLV-MaxSonar-EZ1 Beam Characteristics

This graph shows the position of the rangefinder at the bottom center of the graph. Each grid box denotes a 30x30 cm square of space spanning directly in front of the sensor. An 11-inch target is visible by the sensor when it is within the black lines. According to the graph, the rangefinder should be able to accurately range an 11-inch-wide target up to the maximum range of 5 m and at a horizontal distance of 30 cm to each side of the rangefinder. Considering that the target we used was 24 in wide, the rangefinder should have been more than capable of stable and accurate readings. To test the impact of using a larger target, we ran an experiment using a flat wall as the target. In this experiment, the UAS took off at a distance of approximately 1.8 m from a flat wall. The drone then hovered and took range measurements for several seconds before landing. The range results of one trial of this experiment are shown in Figure 7.

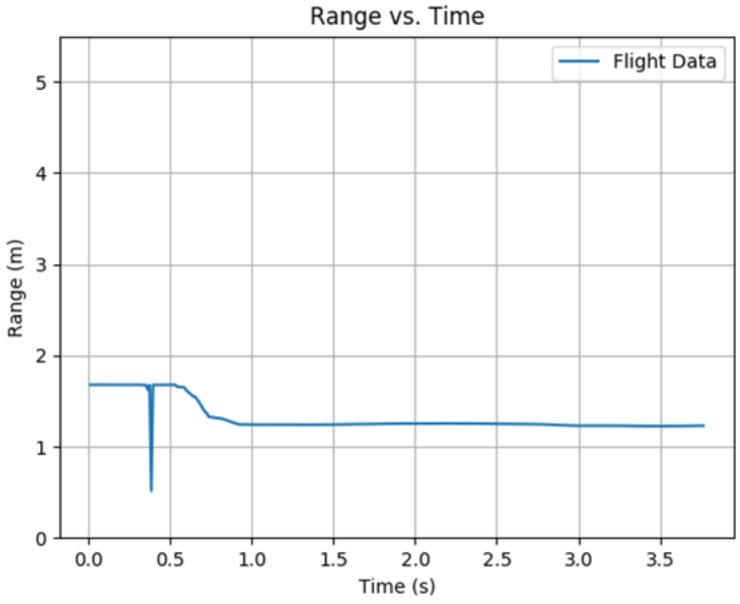


Figure 7: Hover Flight Graph Facing Flat Wall

Clearly the range finding measurements in this trial are much more stable than the tests performed with a smaller target. This indicates that the size of the target plays a large role in the ability of the rangefinder to take accurate measurements. For this reason, all ensuing tests were performed using a flat wall as the target.

3.3 – Rangefinder Measurement Buffer Algorithm

To mitigate the erratic rangefinder measurements still present after switching to wall targets, we started by adding a simple filter to the core distance measurement collection loop that ignored readings with a greater than 50% difference to the previous reading. After running several echoic flow approach tests, a flaw in this method became immediately apparent. Often in these tests, two back to back erratic measurements could be taken and dropped by the filter but in the time that passed since the last accurate measurement, the drone could have moved much closer to the target resulting in every new measurement to be dropped by the filter.

After the failure of the simple range filter, we decided to implement a slightly more complex filter using the range data buffer. This filter evaluates the range measurements contained in the buffer used in the least squares regression fit. It computes the standard deviation of the ranges in the buffer and drops values that are outside of two standard deviations. By dropping range values out of the buffer rather than in real time, the UAS is much less likely to get caught in a situation where all measurements are dropped as they were with the simple filter. Figure 8 shows a forward-looking echoic flow approach to a wall using the new filter with a large target.

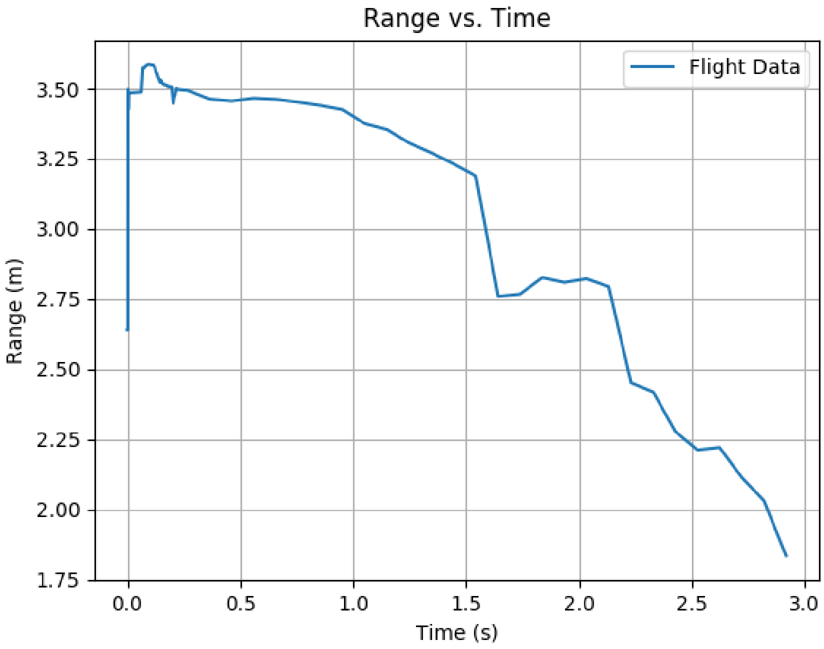


Figure 8: Forward-Looking Echoic Flow Approach to Wall

Clearly the flight graph from this trial has less variation than in previous trials. However, issues still remained with the UAS's ability to execute an accurate echoic flow approach. With a $\dot{\tau}$ of 0.5 used in these tests, we would expect to see a concave slope that approaches the target much slower than the convex sloped graph in Figure 8. An ideal echoic flow approach from 3.5 m is shown in Figure 9.

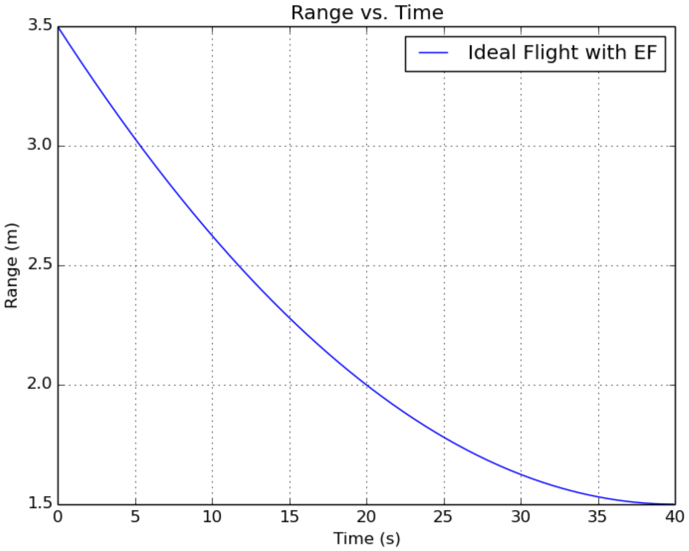


Figure 9: Ideal Forward-Looking Echoic Flow Approach from 3.5 m

3.4 – Target Velocity to Motor Command Conversion

In an attempt to improve the echoic flow approach performance of the UAS, we recalibrated the equation that converts velocities in m/s to commands that the UAS can execute. This equation is central to how the UAS follows echoic flow. Initially we were using a conversion equation that was based off of vertical tests with the onboard ultrasonic sensor. We suspected that the conversion between horizontal velocities and motor commands might be different than that for vertical.

To recalibrate the equation, we focused on motor command values (essentially thrust values for the UAS) between 0.025 and 0.125 or 2.5% and 12.5%. We found that below 0.025 the UAS essentially did not move at all and above 0.125 it had difficulty stopping before its target. We performed three trials of each of the following motor commands: 0.025, 0.05, 0.075, 0.1, 0.125 for a total of 15 trials. For each of these trials, we selected a section of the graph to perform a linear fit on. we attempted to eliminate the beginning of the graph while the UAS was accelerating to get an accurate speed reading. Any clear outliers were also removed from the fit as well. An example trial and regression are shown in Figure 10.

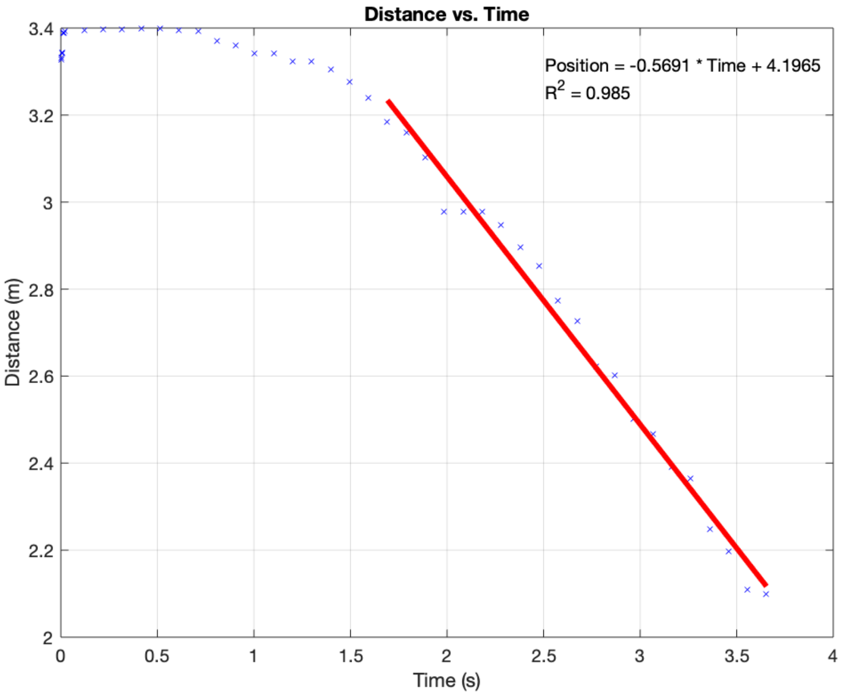


Figure 10: Forward Velocity Regression with Motor Command 0.075

Following this process for each trial, we recorded all of the linear coefficients relating motor commands to true velocity. The three recorded coefficients for each motor command were then averaged and plotted on the chart shown in Figure 11. Finally, a quadratic fit was performed on the five datapoints which yielded an equation relating motor commands to true velocity.

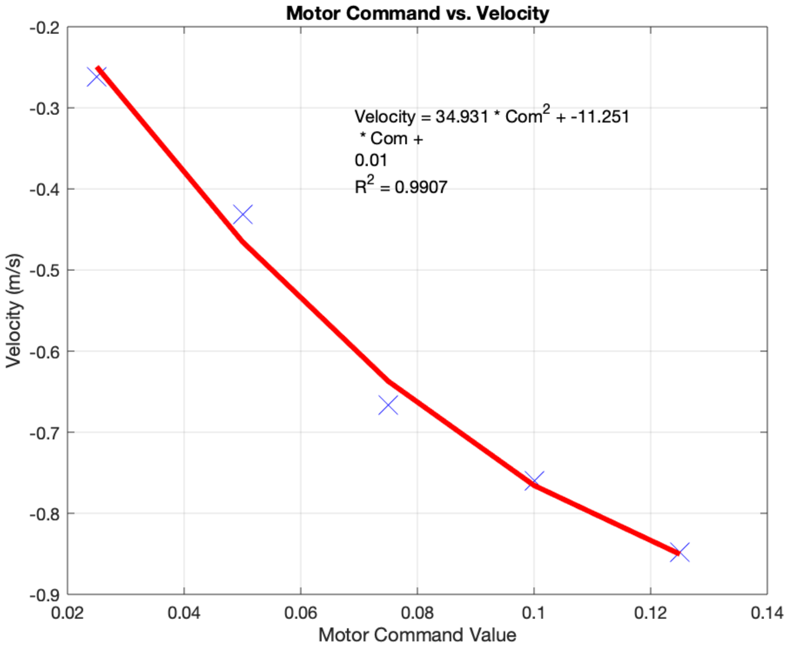


Figure 11: Compiled Forward Velocity Regression and Resulting Equation

$$\text{Forward Velocity} = 34.931 * Com.^2 - 11.251 * Com. + 0.01 \tag{5}$$

$$\text{Vertical Velocity} = -0.148 * Com.^2 - 0.65 * Com. - 0.004 \tag{6}$$

Equation (5) above was the product of our forward velocity testing. It is a clear departure from Eq. (6) which is what we had been using previously. After entering the newly calibrated equation into the Python program, we repeated our wall approach test with a starting velocity of 0.5 m/s. The resulting flight graph along with the corresponding ideal echoic flow approach graph is shown in Figure 12.

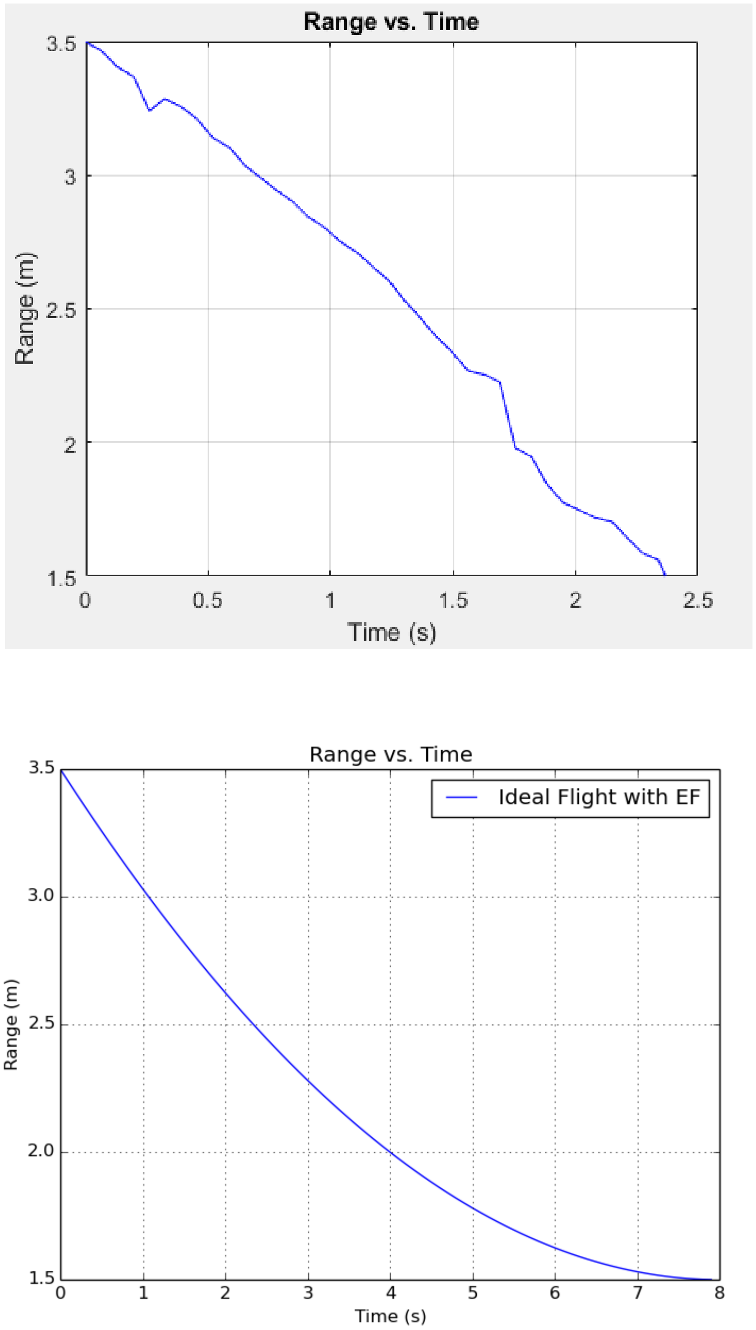
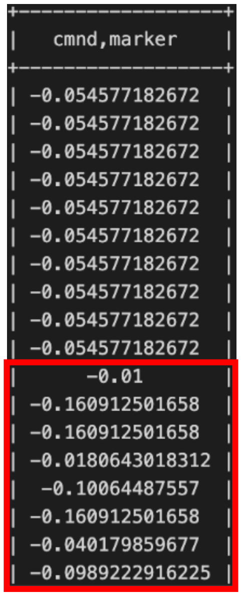


Figure 12: Forward Looking Echoic Flow Approach with Ideal Calculated Approach

Clearly the UAS's approach still differs drastically from the ideal approach described by the echoic flow formula even with the modified velocity conversion equation. Similar results were found in repeated trials of this experiment. Looking more closely at the trial in Figure 12, it appears that the UAS held a constant speed during the approach, however, the raw data indicates otherwise. Figure 13 shows a subsection of the motor commands given to the UAS during the flight (the first ten commands are the same because they are part of the pre-echoic flow buffer). The key takeaway from these data is that motor commands vary widely between consecutive commands. In the highlighted box, commands range from the lowest possible thrust all the way to the highest thrust within 0.8 s.



```
+-------------------+
|    cmnd,marker    |
+-------------------+
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
| -0.054577182672   |
|       -0.01       |
| -0.160912501658   |
| -0.160912501658   |
| -0.0180643018312  |
|  -0.10064487557   |
| -0.160912501658   |
| -0.040179859677   |
| -0.0989222916225  |
```

Figure 13: Raw Flight Data from Echoic Flow Approach

The reason for the varying commands may be traced back to the calculated velocities. Even small errors in the range measurements resulted in large changes in calculated velocity, further resulting in rapidly changing motor commands. To explore the impact of error in range measurements, we turned to simulating echoic flow approaches.

3.5 – Simulations of Forward-Looking Echoic Flow

We evaluated two main types of error within the simulation of forward-looking echoic flow. The first was velocity error. This is the error present in deriving UAS velocity from discrete range measurements. Though we measured the UAS to move between two points in space over a certain time, it did not actually move at that corresponding velocity for the entire distance. For the purposes of these simulations, we estimated the velocity error to have a standard deviation of 0.05 m/s based on prior research [4].

The second type of error is more central to the difficulties we experienced exploring forward-looking echoic flow. This error is the discrepancy between the measured range of the quadcopter to its target and the true distance; it is termed measurement error. With our specific UAS it was difficult to exactly quantify the measurement error due to the Parrot AR Drone's tendency to drift in the horizontal plane when given hover commands. If we had more time to explore our UAS's measurement error, we would have performed a series of experiments commanding the UAS to maintain a specific distance from the target while recording range measurements.

To explore the impact of changing the measurement error on the ability of a UAS to accurately execute echoic flow, we ran a series of simulations. These simulations generated Gaussian random noise based on values for the two types of error. Velocity error was kept at a standard deviation of 0.05 m/s while measurement error was varied. Once the noise was generated, the simulated flight data was passed through a filter. Calculations that involve ratios like those used in echoic flow tend to amplify noise. Filters are necessary in both simulated and experimental scenarios to obtain aggregate values for distance measurements and thus avoid amplifying noise. In these simulations we tested a quadratic least regression filter as was used in all experimental forward-looking tests. We also evaluated a Kalman filter which has been posited as a better filter for UAS echoic flow approaches. The R-value for the Kalman filter was set to 100 to represent the low confidence in the rangefinder measurements. For each filtering system, 100 simulations were performed for each measurement error value. Since a range of 20 measurement error values were tested, a total of 4,000 simulations were run. The aggregated results are shown in Figure 14.

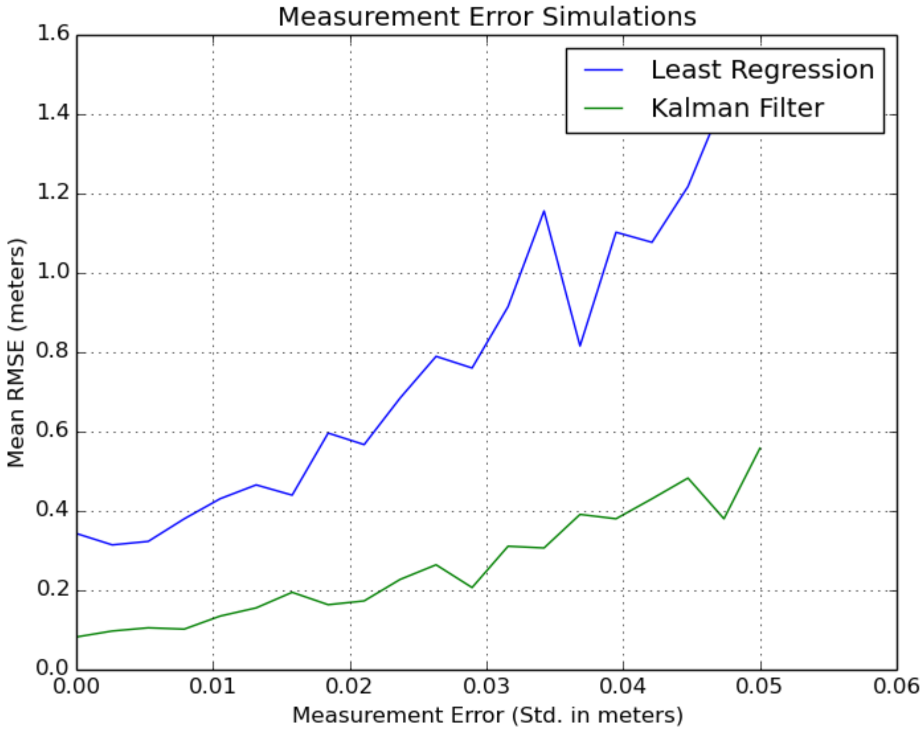


Figure 14: Least Regression and Kalman Filter Simulation Results

In this graph, mean RMSE is the average root-mean-squared error when comparing a simulation to the mathematically ideal echoic flow approach. Mean RMSE is used as a proxy for the accuracy of an echoic flow approach with past research deeming approaches with a value of under 0.1 m as very successful [4]. From the graph above, clearly the least regression filter is never able to achieve an average RMSE of 0.1 or lower. The Kalman filter is able to achieve successful approaches on average when the measurement error was lower than 0.01 m. Though this is only one set of simulations, the importance of accurate range measurements is clear in forward-looking echoic flow.

IV. Conclusion

The initial goal of this research was to explore forward-looking echoic flow in the guidance of a UAS. Components were selected and a UAS was constructed for use in the testing of forward-looking echoic flow. Once the system was complete, preliminary horizontal approach tests were conducted which revealed extensive incorrect range finding measurements. A series of tests were conducted to discover the cause of these incorrect readings and to attempt mitigation. Further approach testing was carried out and the UAS's velocity conversion equation was recalibrated for forward looking tests. After collecting data that showed the inability of the UAS to execute accurate echoic flow approaches, simulations were written and executed to explore the impact of measurement error on echoic flow. Several important conclusions were drawn from this study.

First, when using an inexpensive ultrasonic sensor, range readings can be impacted heavily by motion of the sensor and size of the target. A small target like the 24x18 inch cardboard rectangle led to many jumps in rangefinder measurements during trials. Even with filters in place to mitigate the impact of incorrect measurements, they still had a strong effect on the ability of the UAS to execute an echoic flow approach.

Second, velocity and measurement error are large factors for echoic flow that are negatively impacted by poor motor precision and poor rangefinder accuracy. When the UAS tends to drift in unexpected directions and does not move at exactly the speed that it is commanded, velocity error increases. When incorrect rangefinder measurements must be dropped by the UAS, measurement error increases. Even when rangefinder measurements pass through the filter on erratic measurements, if they differ from the true distance, measurement error will still increase. With the level of velocity and measurement error present in our UAS, accurate forward-looking echoic flow approaches were not possible using the methods described in this thesis.

Because much of the difficulty in pursuing forward looking echoic flow approaches resulted from inaccurate range measurements and imprecise UAS control, it is recommended that future experimentation use a more accurate range sensor and a more precise quadcopter. A faster update rate on the range sensor and more precise motor controls would likely contribute to more successful echoic flow approaches. Further research should focus on integrating higher quality components into the UAS and using the improved platform to perform forward-looking echoic flow approaches to large and small targets.

References

[1] D. N. Lee, J. A. Simmons, P. A. Saillant, and F. Bouffard, ‘‘Steering by echolocation: A paradigm of ecological acoustics,’’ J. Comput.Physiol. A, vol. 176, pp. 347–354, 1995.

[2] Smith, Graeme & Baker, C.J. “Echoic flow for radar and sonar”. Electronics Letters. 48. 1160-1161, 2012

[3] Baker, C.J. & Smith, Graeme & Balleri, Alessio & Holderied, Marc & Griffiths, H.D. Biomimetic Echolocation With Application to Radar and Sonar Sensing. Proceedings of the IEEE. 102. 447-458, Apr. 2014.

[4] Kuric, Justin. “Using Echoic Flow for the Guidance and Control of an Unmanned Aerial System”. (dissertation), 2016, Retrieved from https://kb.osu.edu/bitstream/handle/1811/76508/1/Undergraduate_Thesis.pdf

[5] Maxbotix, (2014). “HRLV-MaxSonar-EZ Datasheet”. Retrieved from https://www.maxbotix.com/documents/HRLV-MaxSonar-EZ_Datasheet.pdf