8 May 2019

From: Asst Prof Evangelista
To: MIDN 2/c Marcello
Subj: EW502 PROPOSAL - COMMENTS
Encl: (1) Markup of EW502 proposal

1. A markup with some comments on the EW502 proposal is attached.

2. **Overall comments:**

a. There appear to be some typos and missing citations which we will add in future versions of the text as it evolves.

b. A single Crazyflie 2.1 is not sufficient; we should budget for and purchase spares as well as the Crazyflie flow deck for direct measurement of altitude and horizontal velocities.

c. Additional discussion is needed on geometric similarity and which form of dynamic similarity is to be used in planning a scale version of the maneuvers.

d. A concern raised previously but not discussed here is the latency in a ROS message passing system and what it will do as more extreme maneuvers are attempted. In biological maneuvers in general, the maneuver is often much quicker than allows for full state high gain feedback; animals use feed-forward along with clever exploitation of passive dynamics and fast inner loops to get by well enough. If the project is pushed into these by control system limitations, the result would be exciting and notable.

e. The proposal focuses on position measurements "from an IMU"?, on the other hand, hummingbirds and racing quad pilots rarely use direct position information, and rarely try to directly control pose, instead controlling velocity and angular rates (e.g. ACRO mode versus LEVEL mode in quadrotors). It is unclear how this figures into the planned work. Along with (d), I expect full state feedback position control to only be possible at severely time-scaled versions of the trajectory, and anticipate more clever things will need to be done as time approaches real hummingbird time.

3. For future work in EW495/EW402, use of Latex, Git/Github shared repositories for deliverables and code, etc will be required.

D EVANGELISTA

Copy to:
EW502 instructor
File (Marcello capstone 9020)

Some comments noted

Extreme Autonomous Trajectory Flight to Copy Bird-Like Maneuvers

Student: Midn 2/C E. J. Marcello   Adviser: Prof. D. Evangelista

Abstract— **I propose to use a small indoor quadrotor (Crazyflie 2.1) and a multi-camera tracking system (OptiTrack) to attempt to recreate extreme maneuvers observed in flying animals. My primary goal is to recreate the courtship display dives of Anna's Hummingbirds (*Calypte anna*). In being one of the first research topics into flying Anna's Hummingbird dive trajectories with an autonomous platform, this paper aims to discover an advantage to this specific type of maneuver for autonomous aerial vehicles. To execute this task, I will be relying heavily on prior work completed in autonomous control of the Crazyflie quadrotor. Additionally, I will be using previously obtained data for *C. anna* flight trajectories. To measure the feasibility of the maneuver I plan to conduct trials both in simulation and in proof of concept demonstration that will compare the trajectory of the quadrotor to a geometrically-similar hummingbird flight trajectory using a root mean square error and maximum distance error between sampled data points along the trajectory paths. The total projected cost of the project is $28,695 including a cost of $570 in new materials. I propose a timeline that estimates project completion in 24 weeks, with the highest risk being my requirement of learning the ROS package library, and any hardware damage caused through repeated hard landings.**

I. Background and Motivation

Animals moving in a real environment face many challenges that are currently unsurmountable for typical engineered devices, including small size, difficult-to-achieve power-to-weight ratios, ability to operate in multiple environments, and the ability to control complex maneuvers even in the presence of environmental disturbances such as wind, water flow, or turbulence, variable lighting, or even attack from predators or conspecific rivals. These alone make them worthy of engineering study, but such interdisciplinary work is not a one-way street.

Biologists are also interested in using engineering techniques to learn how organisms overcome the challenges of their natural environments. As an example, consider the aerodynamic principles of "helicopter" samara seeds such as in maples (*Acer sp.*) and convergently evolved in many other groups. Samaras are able to slow their descent to the ground using an spanwise asymmetric weight distribution and wing structure to facilitate autorotation, allowing them longer hang time during which, on rare occasions, they are swept quite far by a lucky gust of wind. The resulting long dispersal distances are quite advantageous for the saplings, who no longer have to compete with the mother tree. The study of the first autorotating seeds in the fossil record made heavy use of engineering techniques uch as autorotation theory, consideration of stability from the vertical separation of the center of gravity and the center of pressure, and nondimensional coefficients governing flight [1]. As for applications[1], researchers in [2] developed a monocopter based off of this concept, and were able to develop control algorithms for actively steering such a device.

cite stevens

While organisms are a potential source of inspiration for improving robots, robotic systems are also an excellent way to study the form and function of organisms. In [3], the strange movement behavior seen by the sidewinder rattlesnake (*Crotales cerastes*) is studied by creating a robot to imitate it. Sidewinders get their name from their strange locomotion across hot desert sand. They use a combination of body twisting and bending to create moving points of contact with the ground and achieve motion in a direction lateral to their main body axis. The robot was developed like a snake, and was programmed to copy as closely as possible the movement of the sidewinder. The initial test had success on level ground, but no such luck on steeper slopes. On closer observation of the live snake, it was discovered that the sidewinder used two independent wavelike motions to move instead of just a single wave as the researchers had originally thought. Once this dual-wave control was implemented to the design, the robot was able to navigate just like the sidewinder, and a new form of ground travel was made attainable by robotic machines. This discovery has applications of movement over a variety of different terrain that a typical wheeled robot would have significant trouble with, and its design is again owed to the study of biological behaviors.

I will be attempting to enhance a quadrotor's maneuverability by studying the flight patterns and maneuvers of Anna's Hummingbirds (*Calypte anna*), a 5 g hummingbird native to western North America. Male Anna's Hummingbirds perform a display dive in order to win mates with choosy females. The dive reaches speeds above 20 m s^{-1} [4] and ends with a 9G pullout maneuver (full 3D field kinematics determined in [5]) culminating in a display of red iridescent feathers on his gorget (throat) and a high frequency tweeting noise made by flow-induced vibration of the distal two retrices (outer tail feathers) [6]. Such a maneuver would normally cause G-induced loss of consciousness in a human pilot without a g-suit. The difficulty of the maneuver makes it an honest signal [7] of mate quality to Anna's Hummingbird females as it requires skill and power to complete. Thus, it is expected to be difficult for a 27 g quadrotor[2] to imitate, and presents a worthy engineering challenge. I intend to discover how feasible the dive maneuvers actually are for quadrotors, and explore their operational limits in this respect.

Unmanned aerial systems are becoming increasingly more autonomous for tasks with position control at comparatively low speeds; a guiding research question is how extreme maneuverability can augment their capabilities by building on previous work in quadrotor

Authors are with the United States Naval Academy, Department of Weapons, Robotics, and Control Engineering

[1] We emphatically believe applications of basic biomechanics research may not be immediately obvious but that this is not a reason to ignore biomechanical systems of interest.

[2] The Crazyflie 2.1 mass is approximately the same as the largest living hummingbird species, the Giant Hummingbird (*Patagona gigas*), native to the Andes.

- How to scale maneuvers.
- ROS latency?
- Pos control vs Aero.

control [8], [9]. The applications of this research are twofold: it will both help in developing a greater understanding of the physical limitations of extreme maneuverability on quadrotors (engineering problem), and it will also provide a greater idea of the effects of such maneuvers on flying animals. The ability to automate the hummingbird dive maneuvers means that other forms of extreme maneuvers might also be autonomously executed, perhaps providing a toolbox to use when a UAS is presented with a maneuvering challenge during a mission. If combined with enhanced sensing abilities, autonomous MAVs (Micro Air Vehicles) would be able to navigate through obstacle-heavy environments with greater speed and ease, have increased evasion and penetration/infiltration capabilities, or simplify recovery.

In addition to this, a successful device could be used in behavioral playback studies in which live hummingbirds are presented with a controllable stimulus mimicking the male display dive. If a quadrotor were to execute this dive for a female Anna's Hummibgird, could the quadrotor generate a favorable response from her? If so, it may be possible to alter the pattern to present a super-normal stimulus or probe which aspects of the display are most appealing to her. Such a study in hummingbirds would be novel as biologists have not yet been able to produce the maneuvers themselves.

II. Problem Statement

Malmö

I aim to replicate the display dives of male Anna's Hummingbirds (*Calypte anna*) with an autonomous quadrotor platform. My project assumes the following are provided: a quadrotor and flight controller (e.g. Crazyflie 2.1, Bitcraze, Malm o, Sweden), and a method of obtaining three-dimensional (3D) position data of the quadrotor in test flights (e.g. OptiTrack, NaturalPoint Inc., Corvalis, OR). Example trajectories from live hummingbirds will be obtained from [5]. Given a desired hummingbird flight trajectory, depicted in Figure 1a, the system will autonomously generate and execute the control inputs required to successfully complete the maneuver—or a geometrically similar maneuver.

Δt

A successful maneuver is defined as a root mean square error calculation of less than 5 cm between the geometrically similar hummingbird trajectory and the quadrotor trajectory, where each trajectory is defined as a matrix array of positions in the *x, y, z* right handed coordinate frame with a given sample period Δt. The hummingbird trajectory will be scaled down by a constant factor to account for the physical limitations of the quadrotor and the lab space—e.g. the quadrotor will still fly the same trajectory shape, but it will be smaller than the full scale dive, and potentially at slower speeds. This is a necessary adaptation due to the impressive speed and acceleration capabilities of the Anna's Hummingbirds relative to their size [5]. The ideal result is to fly the hummingbird trajectory at the same speed as the hummingbird, but this may prove to be impossible due to the physical constraints of the quadrotor and lab space in Maury 201, where I will be completing my proof of concept experimentation.

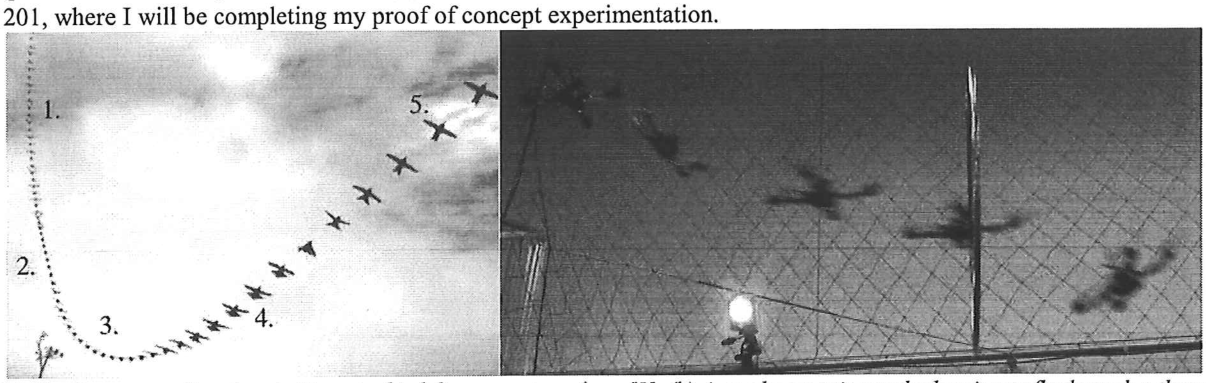


Figure 1: (a) The five stages of an Anna's Hummingbird dive maneuver, from [5]. (b) A quadrotor using path planning to fly through a thrown hoop, from [8]

As a stretch goal, I will also analyze and replicate a variety of different types of bird dives and maneuvers, to include evasive aerial maneuvers [10]. Success will be determined in the same fashion as described above.

Sholths too

III. Literature Review

The inspiration of this project proposal has its roots in the high maneuverability seen by hummingbirds. As such, it was essential to be able to gather data on hummingbird trajectories in order to study their feasibility of being replicated by a quadrotor. There have been many bio researchers that have delved into the study of hummingbirds, however I will mainly discuss the works [6] and [10]. The work in [10] determined the trajectory and body kinematics of four different hummingbird species in an evasive maneuver. This was done by startling the birds while they were hovering, and observing their movements with 3 high definition cameras to provide a 3D position [10], [11]. Water-soluble white paint was used to make dot markings on the hummingbird's body to help model the wing and head positions for each trial. The data they acquired in their experiments includes many more details than I will need to use, however they provide data on the hummingbird's velocity and trajectories for several trials which I can use to help develop my quadrotor trajectories. I will also be using a similar optical tracking method as in [10], [12], [13], [11] with the added simplifications of being able to install known, infrared (IR) relfective markers for automatic tracking and using a RANSAC algorithm to estimate position. Evasive maneuvers like those displayed by these hummingbirds can be useful for quadrotors if they are in threat from a counter-UAS drone denial system (e.g. USNA Project Midknight or similar). If I am able to copy the hummingbird's

cite 2018 2019

trajectory, further analysis and testing may prove that this type of trajectory provides a maneuvering or sensing advantage to the quadrotor in evasive flight.

The work in [5] obtained full 3D field kinematics of courtship dives in *C. anna* in order to study extreme locomotor performance in animals. [5] used multiple calibrated high speed cameras and manual digitization using Peak Motus 8 to reconstruct the birds' 3D position during dives. Splines were used to estimate acceleration and velocity from positions without undue amplification of noise. [5] also examined wing and tail movements and sounds produced during dives. As a measure of the biomechanical difficulty of the maneuver, [5] estimated the maximum stresses in the humerus during dive pullout; such a quantity would be extraordinarily difficult to measure *in vivo* but in my work there is a possibility of directly instrumenting the quadrotor to examine forces, torques, stresses, and engineering limits.

After obtaining these hummingbird trajectories, an effective method of modeling a quadrotor and conducting trajectory planning is required. In [14], the aim of the paper is to develop a general standard to measure against in terms of quadrotor maneuvering performance and constraints. This is achieved through the solving of an optimal control problem offline, and then using a machine learning technique to learn these trajectory solutions with the given constraints. This will then translate into an online general solution for near-optimal trajectories for a quadrotor. This was done in the x-z plane for point to point and perching maneuvers, as well as joint trajectories. To validate their solution, they flew these optimal trajectories using both simulink simulations, and proof of concept demonstrations. Since I will be using a quadrotor platform, this paper directly applies to my problem statement as a good reference base that I can use to springboard my exploration into more complex extreme maneuvering. The basis of this work will give me a much more quantitative measure of success in terms of how close my developed trajectories are to an optimal path. [14] is very thorough and provides a clear distinction and improvement on previous work in quadcopter trajectories, especially with regard to the joint trajectory problem. I can build on this by expanding into 3D trajectories instead of just working in a 2D plane, and I can also try to utilize their proxy-based joining method to create a desired path curvature.

In [15], the authors developed a linear model of a quadrotor in planar motion using Newton's and Euler's laws. The careful process by which the quadrotor dynamics are identified and modeled will be helpful in my own research as I develop my own model for the quadrotor that I will be using. In [15], their modeling method is done for 3 different linearization methods and each of these is compared to each other by running a Simulink simulation with each controller. Quantities compared include several attributes of the step response, and the actual trajectory of the quadrotor compared to the desired trajectory. This comparison method between the different trajectories is similar to the validation work that I will need to do on my own simulation. As such, this work will help me to better understand ways of determining the accuracy of my trajectory testing in simulation, and in proof of concept demonstration. This paper, while a good starting point for my work, does not attempt to go into more complex maneuvers. These are discussed in greater detail in the following works.

In [16], the development of trajectories and path planning for UAVs was accomplished. They did this by determining the maximum overload, minimum turn radius, and maximum flight endurance of the experimental quadrotors in order to come up with feasible aggressive trajectories. Trajectories had the constraint that they had to follow a sixth order (or lower) polynomial trajectory. Much like my proposed concept, this work develops an attitude and trajectory controller with appropriate initial and final conditions, as well as a boundary "tube" which the quadrotor must stay within for every trajectory. The work in this project is heavily relevant to my proposed work, as they achieve a working simulation of aggressive trajectories with their path-planning algorithm and onboard controllers. I would like to expand on this work by flying a shorter trial with hummingbird-like flight patterns.

Finally, [8] offers some of the closest work to exactly what I am proposing for my own work. The main focus of this paper is to create trajectories for quadrotors in real time in an indoor or constrained environment. They also pay particular attention to the velocity and acceleration vectors of the quadrotor throughout its maneuver. I will also need to be able to achieve these types of measurements from my system, and be able to change my controller to affect them in an appropriate manner in order to fully achieve a trajectory flight path that replicates a hummingbird maneuver. [8] also uses temporal scaling to fly their trajectories at different speeds, which is exactly what I will need to do when and if I find that flying the hummingbird trajectory at full speed is either not possible or extremely dangerous.

In summary, my research will build from multiple sources concerning both hummingbird flight kinematics, and the control and modeling of quadrotor extreme trajectory-flying. I will be synthesizing 3D kinematic hummingbird flight data from [5] with the quadrotor dynamic control schemes of [8], and [9] with a foundational understanding of motion-capture data collection from [10], [12], [13], and [11] for the overall objective of achieving autonomous extreme maneuvering and trajectory flying with a quadrotor.

IV. Demonstration Plan

To demonstrate the objective of replicating *C. anna* display dives, I will simulate the quadrotor system in MATLAB and Simulink. This will be followed up with a proof-of-concept demonstration, provided I am able to achieve successful trials in simulation first. The simulation will be more general and allow the easy changing of parameters to more completely explore the space; simulations will also allow examination of quadrotor behavior at actual hummingbird flight speeds and full scale trajectories. Simulations make simplifying assumptions by necessity, so actual testing and demonstration using real quadrotors is also planned.

For my demonstrations, I will be modeling the rigid body dynamics of the quadrotor using the decoupled system as shown in Figure 2. below.

explain?

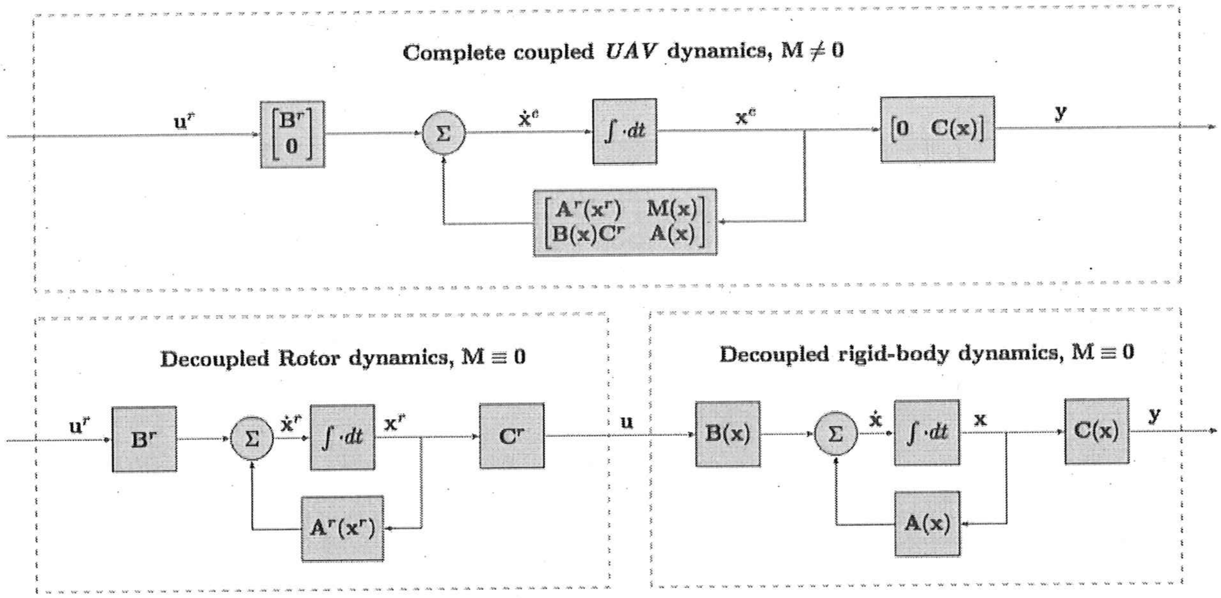


Figure 2: Coupled and Decoupled dynamics of the Crazyflie quadrotor [9].

The decoupled rotor and rigid-body dynamics of the quadrotor are achieved through the assumption that translational rigid-body movement has no effect on the rotor behavior. This is an important assumption, as this simplifies the quadrotor control to a combination of four single input single output (SISO) motors, whose outputs directly serve as the input to the rigid body dynamics of the quadrotor [9].

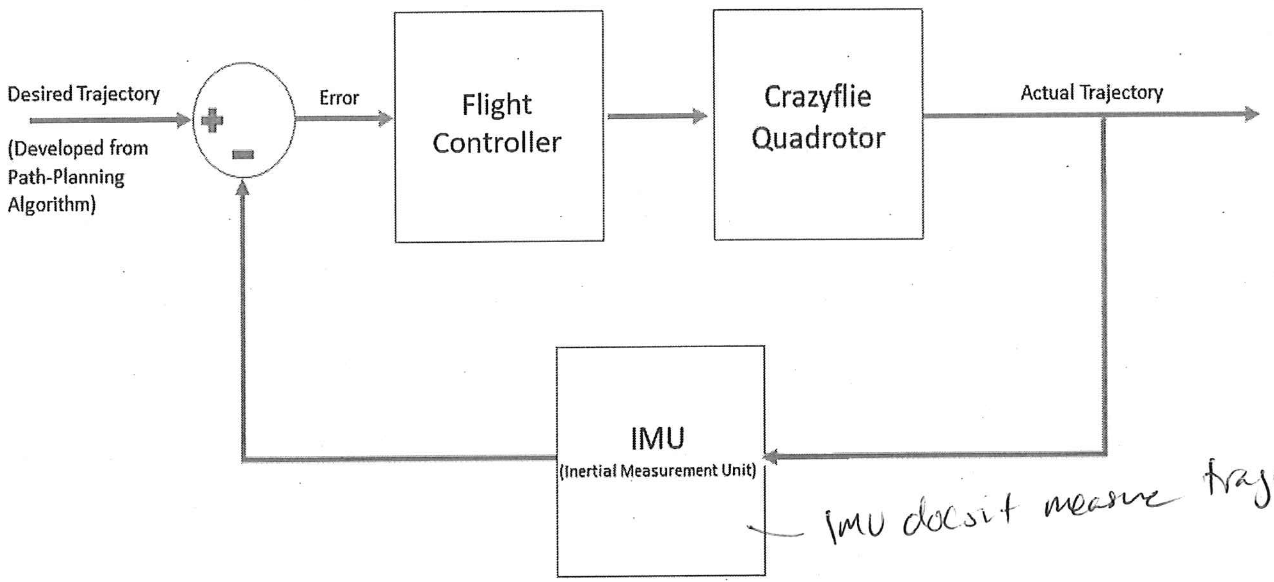
Additionally, due to the extreme nature of the hummingbird trajectories it will be necessary to represent rotation angles of the quadrotor using quaternions, as outlined in [9]. This is due to the limitations of the Euler model at high rotation angles, as a singularity can occur which causes the model to break down. Representing these rotations with quaternions will be robust at these extreme angles, and is therefore better suited for my experiment.

A. *Simulation*

To create a feasible simulation for this research, I will model the Crazyflie quadrotor platform to be used in simulation. This involves determining the various thrust vectors of the motors, and the calculation of many performance metrics. Thankfully, much of this work has already been done, and I will be relying heavily on the work in [9], and [17] to simulate the quadrotor in MATLAB and Simulink. Due to the extreme maneuvering of the quadrotor, I have chosen to use a quaternion coordinate system to describe its position, for flight control purposes only. The trajectory comparison will be conducted in the x, y, z Cartesian coordinate frame. A quaternion is defined as a 4D vector of one real and three imaginary vector directions, i, j, and k. Once this has been accomplished, I will obtain and upload Anna's Hummingbird flight trajectories into MATLAB. These will serve as truth, or the desired trajectories for my simulation to run. I will then develop, or obtain from another source, a path-planning flight controller algorithm that is able to take in a desired flight path and create an appropriate response to fly this desired flight path with minimal error. The feedback diagram of this control concept is shown below in Figure 3. In principle, a desired trajectory will be developed from a path-planning algorithm that takes in the desired trajectory and calculates the idealized pitch and motor torque and speed at each time step in order to fit this trajectory with as little error as possible. This signal will be combined with an inertial measurement true position feedback to produce the error signal to the onboard flight controller. The flight controller will send a signal to the motors on the quadrotor based on this error signal to come as close as possible to zero error for the next time step, and the cycle will repeat until the quadrotor has completed its maneuver. My simulation will replicate this decision-making process using numerical integration methods (e.g. ode45 or similar) to obtain the position data for every iteration.

[illegible]

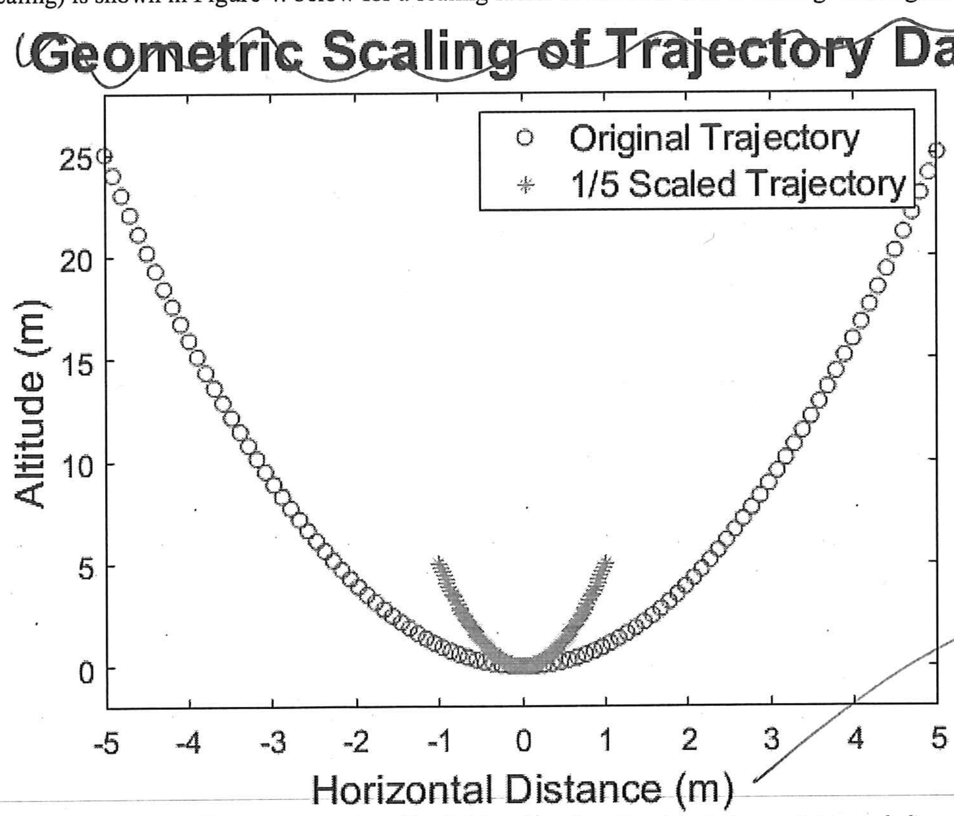
Sim limitations?



IMU doesn't measure trajectory?

Figure 3: Functional Block Diagram of the simulated controller feedback loop. This model assumes that the IMU is capable of direct measurements of position, velocity, and acceleration of the quadrotor.

Since I am completing simulations of the quadrotor behavior with the intent of replicating the simulated results in lab experiments, I must also run my simulated trials taking into account the physical constraints of the lab environment. One of those constraints is the physical space in the lab. The actual *C. anna* dive trajectories occupy a space much larger than I have available in Maury 201, the OptiTrack lab. I will therefore have to run trials at a scale factor smaller than the actual dive trajectories. This geometric similarity (titled ~~Geometric Scaling~~) is shown in Figure 4. below for a scaling factor of one fifth that of some given original trajectory.



too big.

Figure 4: Example of a geometrically scaled trajectory, achieved by dividing the value of each trajectory point in each dimension by a value of 5.

Trials will be run at several different scaling factors in simulation to determine the effects that this type of scaling has on the velocities and accelerations experienced by the quadrotor. Trajectories will be geometrically similar before they are input into the system, and as such the similar trajectories will become the new desired trajectories—i.e. truth data.

Finally, to determine accuracy of the trial I will compare the actual flight path of the quadrotor with the desired flight path using two different error metrics. The first is a distance error based on the magnitude of the distance between each pair of actual and desired trajectory data points in the two datasets. The second is a root mean square error (RMSE) between the entire two position vs time datasets—i.e. the actual quadrotor trajectory data, and the desired trajectory data. The distance magnitude error will be calculated using a simple magnitude equation shown in equation 1 below:

$$e(t) = \sqrt{(x_a(t) - x_d(t))^2 + (y_a(t) - y_d(t))^2 + (z_a(t) - z_d(t))^2} \qquad (1)$$

where $e(t)$ is the error at a specific time step in the trajectory (time t), and the subscripts a and d represent the actual traveled trajectory by the quadrotor and the desired trajectory respectively. This error will be calculated for each time step, and the maximum error value will be determined. A successful flight will be defined as having a maximum error value of no greater than 10 centimeters.

The second error metric, the RMSE, is defined by equation 2 below:

$$RMSE_{3\times1} = \left[\sqrt{\frac{\sum_{i=1}^{N}(x_{a_i} - x_{d_i})^2}{N}} \quad \sqrt{\frac{\sum_{i=1}^{N}(y_{a_i} - y_{d_i})^2}{N}} \quad \sqrt{\frac{\sum_{i=1}^{N}(z_{a_i} - z_{d_i})^2}{N}} \right]^T \qquad (2)$$

where N is the total number of sample points in the trajectory path. The squared difference in position of the actual and desired trajectories are averaged over the entire dataset for each Cartesian direction x,y,z independently, and the square root of each result is taken. The result can be viewed in the form of a 3x1 matrix, where each component is the average error in each direction. The magnitude of this matrix will be the 3D RMSE. Without any other indications, a successful trial will be considered a 3D RMSE of less than 5 centimeters over the entire trajectory.

If I am able to accomplish a working simulation with the Anna's Hummingbird courtship dive maneuver, my stretch goal is to make the simulation slightly more general so it can apply to a wide variety of hummingbird dives and maneuvers. This can include evasive maneuvers, and potentially other types of aerial stunts.

B. Experimental work

Provided that the simulation is able to achieve several successful runs, a proof of concept demonstration will be deemed necessary for further analysis into the possibility of actually autonomously flying a drone through hummingbird flight trajectories. The hardware components necessary for this experiment include a fully functional Crazyflie quadrotor, and an OptiTrack system as used in [18] and [19] whose work I will be emulating in experiment. The quadrotor will be the object of the experiment, and the OptiTrack system is a highly accurate method of obtaining position data for the Crazyflie in flight using visual information from nearly 20 cameras staged around the perimeter of the testing area. Experimentation will be conducted indoors to minimize any aerodynamic noise; however, if early trials prove successful, I can use a fan to test controller robustness to external disturbances. I will need several batteries for the Crazyflie to ensure sufficient trial and testing periods, and I will also need to attach approximately five OptiTrack visual markers on the Crazyflie drone to ensure that it is able to be detected by the OptiTrack system. These marker additions will be taken into account in simulation first in order to ensure readiness to counteract any effect they may have on the dynamics of the quadrotor in the feedback loop.

As mentioned, this demonstration plan is an extension of previous work done by [18] and [19] using the same equipment for trajectory flying. These works were completed by other Midshipman at the Naval Academy, and I fortunately have access to all the same resources that they had in their experimentation, in addition to Midshipman Canlas' firsthand testimony of his experience. He will prove to be a valuable resource early on to help troubleshoot the system and jumpstart my experimental testing. I also plan to make use of their MATLAB scripts to obtain position and orientation data from the OptiTrack, in addition to their control algorithm to fly the Crazyflie through the desired trajectories. This necessitates the use of the Robot Operating System (ROS) libraries, and an Odroid for control of the quadrotor, since that was the control approach taken in [18] and [19].

To fly the trajectory, the quadrotor system will follow a feedback loop similar to the simulation loop portrayed in Figure 3 above, only the IMU block in that depiction represents both the IMU onboard the quadrotor, and state feedback from the OptiTrack system. When the quadrotor is ready to fly the trajectory, the OptiTrack system will obtain the position vs. time information of the Crazyflie using visual sensor data, accurate to one millimeter. The experiment component communication scheme is outlined in Figure 5 below.

slow?

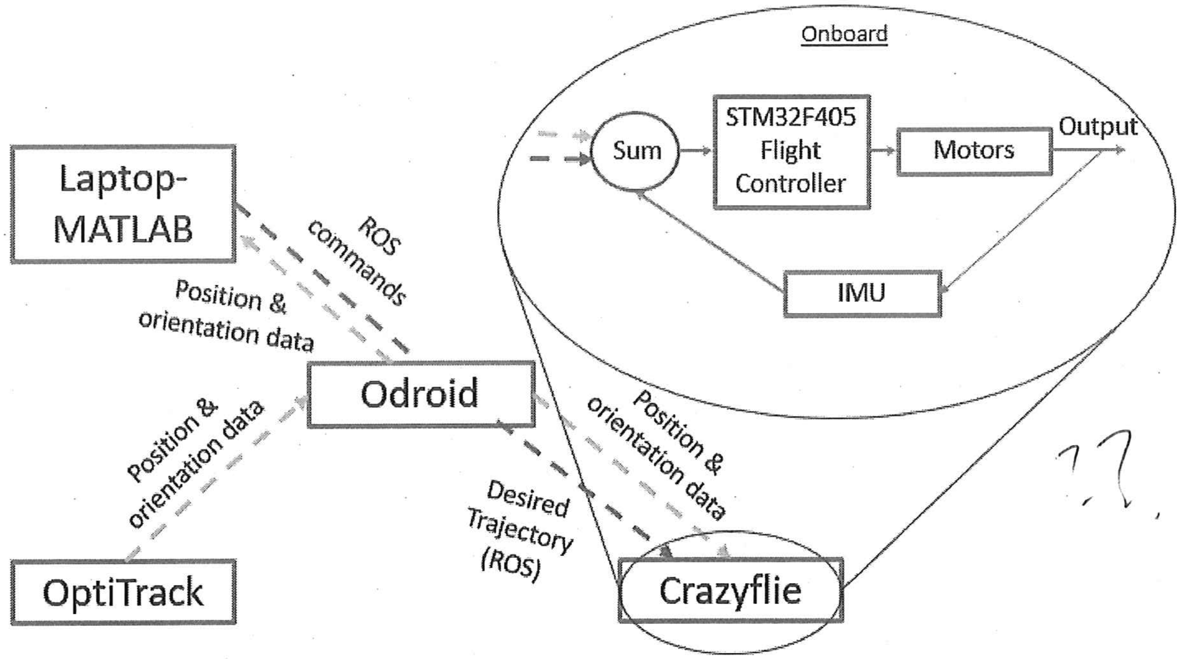


Figure 5: Diagram sketch of communication channels in lab experiment. Motor commands are executed as a function of the desired trajectory, state feedback from the OptiTrack motion capture system, and state feedback measured using the onboard IMU.

At the conclusion of each trial, the position data will be compared to the desired hummingbird trajectory in post-processing to determine the flight accuracy, just as in the simulation. Error metrics will be calculated in the same manner, with a similar error tolerance.

C. *Property Measurement*

The major property concerned with this research is the trajectory of the quadrotor. I must be able to determine the position of the Crazyflie, and sample this information at a similar data rate to the desired trajectory data. This is to ensure that there exists matching trajectory points in time in both data sets at regular intervals to be used for the error calculations discussed previously in subsection *A*. Position data of the quadrotor is obtained by the OptiTrack motion capture system, and these data are queried from the OptiTrack in a MATLAB script through the use of a ROS vehicle object. The full script is available in Appendix II, and its analysis has been excluded from this section for brevity.

To reiterate, successful trials will be considered to have a maximum distance error of no more than 10 cm between any pair of time-matched trajectory points of the actual and desired trajectory data, and the magnitude of the RMSE must be less than 5 cm. These numbers don't have any particular significance as of yet, and are subject to change as unexpected changes to the experiment occur.

D. *Technical risks and mitigation*

It is possible that the Crazyflie quadrotor will crash during experimental testing. [19] attempted to fly 2D trajectory patterns using Crazyflie 2.0 hardware, and had been repetitively set back through damage accumulation from repeated hard landings. The risks to damage on the quadrotor will be mitigated by 3D printing propeller guards to prevent hardware damage on unintended impacts. Since damage is inevitable, I have also taken initiative to order several replacement parts in the event that the system becomes too damaged to fly during testing. There is also the potential to switch to an alternative platform (DJI Tello) and follow other Midshipman work using that platform.

E. *Time risks and mitigation*

Learning ROS will be the most time-intensive part of this project, as well as finding or potentially developing a suitable control algorithm that will allow my quadrotor to fly the necessary trajectories, since the integration of controllers in [18] and [19] to my experiment may prove to fail. To mitigate these risks, I will be learning ROS concurrently with the rest of my research work to minimize any delays to progress that a lack of knowledge in this area might cause. Additionally, even if the controller used in [18] and [19] is not useable for my experiment, I still be able to use it as a basic control algorithm and build from it as needed for my application. A full timeline is included in Appendix I.

F. *Justification of special high risk activities*

In order for this research to be conducted successfully, I will likely need to learn how to use the ROS package libraries, and re-familiarize myself with linux shell and Python. This could potentially impose delays to my work while I learn these newer capabilities; however it is a necessary step in making full use of the Crazyflie developmental platform, and being able to communicate between it and the OptiTrack system. This is especially the case, since my work will be based on prior work done by [18] and [19], who used ROS and linux shell in his work. Python will only become necessary if the current control algorithm onboard the Crazyflie isn't suitable for my experimental trails. Additionally, I will be able to use his work as a springboard toward my research goals, which will outweigh any delays incurred from learning to use these aforementioned tools.

G. Budget

TABLE 1 BUDGET

LABOR	Category	Hours	hourly rate	Cost
	Midshipman	336	$25	$8,400
	Faculty	64	$60	$3,840
	Staff	45	$40	$1,800
Labor Sub-total				$14,040
OVERHEAD	Category	Base Amount	Rate	Cost
	Fringe Benefits	$14,040	35%	$4,914
	Facilities	$14,040	50%	$7,020
	General Services	$14,040	15%	$2,106
Overhead Sub-total				$14,040
MATERIALS	Category			Cost
	In-stock Items			$25
	New Items			$570.00
Materials Sub-total				$595.00
TOTAL COST				$28,675
OUT-OF-POCKET COST				$570

*Disclaimer: Labor and overhead costs are estimated only for EW502 training purposes and do not actually reflect real costs that would be supported by project sponsors.

Flow deck

I will be purchasing a new Crazyflie 2.1—possibly two—since many of the Crazyflies that we own have been crashed multiple times and have potentially lost functionality I will need for my highly precise and aggressive maneuvering. Also included in the new items category are extra propellers, batteries, motors, and OptiTrack markers. The extra propellers and motors are being purchased in the event that they need to be used to replace the original ones if they become damaged during experimental trials. Extra batteries will help make efficient use of lab time, as each battery only lasts for approximately eight minutes of flight. I will be able to cycle through batteries, using one while the others charge. I have also purchased extra OptiTrack markers to try different configurations of markings on my Crazyflie in order to get the most consistent tracking and flight data. An additional $25 has been included as an estimated cost in producing the 3D printed rotor guards used on the Crazyflie, which will be used to prevent damage to the Crazyflie in the event of a crash.

Not included in my budget table, but also of importance, is funding to attend a robotics conference such as SICB, AUVSI, or ICRA. My attendance at one of these may significantly increase the out-of-pocket cost of my project as I would need funding to cover registration fees and travel to and from the venue.

why not?

V. Conclusion

The proposed research discussed in this paper is concerned with the replication of the highly aggressive aerial maneuvers of the Anna's Hummingbird on an autonomous quadrotor platform. Specifically, this research aims to determine the extent to which these maneuvers are possible through analysis of a root mean square error and maximum distance error between the geometrically similar bird and quadrotor trajectories. Analysis will be conducted through both simulation and proof of concept demonstrations on the Crazyflie 2.1 platform. The proof of concept demonstration will utilize an OptiTrack motion capture system to ensure a high degree of accuracy is recorded for the quadrotor trajectory, and the experiment will be conducted inside to minimize any possible disturbances in the air.

This research is the first attempt to autonomously mimic a hummingbird dive trajectory, and it has promise in being successful at developing a greater understanding of the limits of extreme maneuverability in our currently utilized UAV quadrotor platforms. The biggest risk to the project's completion is my understanding of the ROS library. Lapses in my understanding could cause my research timeline to drag out longer than intended, and as a result could possibly inhibit my ability to draw any significant conclusions about the quadrotor's capability in executing these extreme maneuvers.

REFERENCES

[1]R. NORBERG, "AUTOROTATION, SELF-STABILITY, AND STRUCTURE OF SINGLE-WINGED FRUITS AND SEEDS (SAMARAS) WITH COMPARATIVE REMARKS ON ANIMAL FLIGHT", *Research Gate*, 1973. [Online]. Available: https://www.researchgate.net/publication/229965983_Autorotation_self_stability_and_structure_of_single_winged_fruits_and_seeds_Samaras_with_comparative_remarks_on_animal_flight. [Accessed: 04- May- 2019]

[2]"Maple Seeds Inspire Robotic Flight", *Aero.umd.edu*, 2019. [Online]. Available: https://aero.umd.edu/news/story/maple-seeds-inspire-robotic-flight. [Accessed: 19- Mar- 2019]

[3] Feltman, Rachel. . *Creepy Robots Help Researchers Understand the Mysterious Sidewinder Snake*. Washington: WP Company LLC d/b/a The Washington Post, 2014, https://search.proquest.com/docview/1609471314?accountid=14748 (accessed March 19, 2019).

[4] J. Larimer and R. Dudley, "Accelerational Implications of Hummingbird Display Dives", *The Auk*, vol. 112, no. 4, pp. 1064-1066, 1995 [Online]. Available: https://www.jstor.org/stable/4089044?Search=yes&resultItemClick=true&searchText=%28%28hummingbird%29&searchText=AND&searchText=%28dive%29%29&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3D%2528%2528hummingbird%2529%2BAND%2B%2528dive%2529%2529&ab_segments=0%2Fdefault-1%2Frelevance_config_with_defaults&refreqid=search%3A045626d07ba91ba3c65c3041c3083b18&seq=2#metadata_info_tab_contents. [Accessed: 08- Feb- 2019]

[5]C. Clark, "Courtship dives of Anna's hummingbird offer insights into flight performance limits", Proceedings of the Royal Society B: Biological Sciences, vol. 276, no. 1670, pp. 3047-3052, 2009 [Online]. Available: https://royalsocietypublishing.org/doi/pdf/10.1098/rspb.2009.0508. [Accessed: 19- Feb- 2019]

[6]B. G. Hogan and M. C. Stoddard, "Synchronization of speed, sound and iridescent color in a hummingbird aerial courtship dive," *Nat. Commun.*, vol. 9, no. 1, p. 5260, Dec. 2018.

[7]C. J. Clark and T. J. Feo, "The {A}nna's {H}ummingbird chirps with its tail: a new mechanism of sonation in birds," *Proc. R. Soc. B*, vol. 275, no. 1637, pp. 955–962, Jul. 2008.

[8]D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 2520-2525 [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5980809&isnumber=5979525. [Accessed: 19- Feb- 2019]

[9] M. Greiff, "Modelling and Control of the Crazyflie Quadrotor for Aggressive and Autonomous Flight by Optical Flow Driven State Estimation," Lund University Department of Automatic Control, 2017.

[10] B. Cheng, B. Tobalske, D. Powers, T. Hedrick, S. Wethington, G. Chiu and X. Deng, "Flight mechanics and control of escape manoeuvres in hummingbirds. I. Flight kinematics", *The Journal of Experimental Biology*, vol. 219, no. 22, pp. 3518-3531, 2016 [Online]. Available: http://jeb.biologists.org/content/219/22/3518. [Accessed: 08- Feb- 2019]

[11] D. H. Theriault *et al.*, "A protocol and calibration method for accurate multi-camera field videography," *J. Exp. Biol.*, vol. 217, no. February, p. jeb--100529, Feb. 2014.

[12] K. M. Sholtis, R. M. Shelton, and T. L. Hedrick, "Field Flight Dynamics of Hummingbirds during Territory Encroachment and Defense," *PLoS One*, vol. 10, no. 6, p. e0125659, Jun. 2015.

[13] B. E. Jackson, D. J. Evangelista, D. D. Ray, and T. L. Hedrick, "3D for the people: multi-camera motion capture in the field with consumer-grade cameras and open source software," *Biol. Open*, vol. 5, no. 9, p. bio.018713, 2016.

[14] T. Tomić, M. Maier, and S. Haddadin, "Learning quadrotor maneuvers from optimal control and generalizing in real-time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, no. Section III, pp. 1747–1754.

[15] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," KTH Royal Institute of Technology in Stockholm, 2015.

[16] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning Dynamically Feasible Trajectories for Quadrotors Using Safe Flight Corridors in 3-D Complex Environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1688–1695, 2017.

[17] David(dch33), "Quadrotor Dynamic Modeling and Simulation", Web. 13 Nisan 2015. https://github.com/dch33/Quad-Sim. (Date of access: 03.05.2019).

[18] J. Gainer, L. Devries, M. Kutzer, "Persistent Target Detection and Tracking by an Autonomous Swarm", ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2018.

[19] B. Canlas, L. Devries, "Shape Formation Control for an Autonomous Swarm", United States Naval Academy 2015.

APPENDIX I: TIMELINE

EW495 - Fall AY2020

Week (M)		Note	Planned Activities	Actual Hrs
1	Aug 19	Class begins on M	**Information turnover with necessary parties** • Conduct trial run of hardware with MIDN Canlas' supervision to become acclimated with experimental procedures and troubleshooting. • Affirm that I can collect data using OptiTrack **Familiarization with simulation capabilities:** • Test all demos	5
2	A 26		• Note how simulation responds to changes in certain parameters (weight, max thrust, etc.) • Buffer time to potentially find a new simulation if current one isn't suitable • Use Crazyflie flight characteristics for simulation	5
3	Sep 2	M= Labor Day T ← M	**Manual Flight & OptiTrack** • Establish manual control and fly Crazyflie to ensure working flight capabilities. • Use OptiTrack motion capture system to take position vs. time data of several manual flights. Plot Data obtained. **Fly basic trajectories in simulation** • Fly circle, straight line, and 2nd degree parabolic trajectories (2D-constant altitude) in simulation. Collect position vs. time data and plot simulated data. • Fly these trajectories at a wide range of different speeds and log data.	7
4	S 9		**Simulated data error calculations** • Calculate error between simulated flown trajectory and actual desired trajectories. • Plots for position, velocity, and acceleration. • Determine a relationship between error and speed. **Trajectory data collection** • Obtain dive trajectories of Anna's hummingbirds for MATLAB import.	5
5	S 16		**Buffer time.** Tie up any loose ends. **Additionally:** begin studying ROS and Python programming languages to be used in autonomous control of Crazyflie.	5
6	S 23	Ac Reserve	My 6 week deliverables are: • Working simulation with basic trajectories (or significant progress in this direction if unanticipated complications arise). • Obtain Anna's hummingbird flight data to test. • At least one manual flight with Crazyflie	
7	S 30	6 wk grades due Tues	**Programming** • Continue work in ROS and Python. **Simulation--3D trajectories** • Test 2nd degree polynomial trajectories with varying altitude in simulation at several different speeds. • Test humminbird dive trajectories in simulation at different speeds. • Calculate error in trajectories for each speed.	5
8	Oct 7		**Programming** • Continue work in ROS and Python. **Autonomous Flight** • Fly Crazyflie autonomously using MIDN Canlas' previous work. • Log flight data with OptiTrack.	5

9	O. 14	M = Columbus	**Programming** • Continue work in ROS and Python. **Autonomous Flight** • Fly Crazyflie autonomously with varying altitude. Complete point to point trajectory flight. • Log flight data with OptiTrack.	5
10	O 21		**Programming** • Continue work in ROS and Python. **Buffer time.** Clean up any work in progress.	5
11	O 28	Ac Reserve	My 12 week deliverables are: • Fly Anna's hummingbird trajectories in simulation at varying speeds. • Fly autonomous trajectories with Crazyflie and log flight data. • Obtain understanding of how the Crazyflie executes desired trajectories in Python/ROS code.	
12	Nov 4	12 wk Grades due Tues	**Controller algorithm** • Determine current trajectory controller (PD, PID, PI-Lead?) **Buffer time.** Clean up any work in progress.	5
13	N 11	M = Vets Day	• Find suitable controller algorithm alternative (if current controller is deemed insufficient for testing. **Buffer time.** Clean up any work in progress.	5
14	N 18		• Implement new controller on Crazyflie in both simulation and experiment. Compare errors obtained from logged data.	5
15	N 25	Tgiving Thurs-Fri	• Share draft report and poster with adviser for comments. Use template.	
16	Dec 2	R= last day class	• Submit poster for printing. • Revise plan for Spring semester as needed	

EW402 - 4 Credits - Spring AY2020

Week (M)		Note	Planned Activities	Actual Hrs
1	Jan 6	Begins Tues (M sched)	Refine simulation with new controller. Re-test trajectories for error relationship with speed.	5
2	J 13		Test Humminbird trajectories in lab at slow speeds	5
3	J 20	M= MLK	Determine max speed in simulation for RMS (root mean square) error of less than 5cm. Achieved through running iterations of the simulation with increasing speed.	5
4	J 27		Test Hummingbird trajectories in lab at speeds close to simulated speed. Obtain error from data collection.	5
5	Feb 3		Adapt controller gains in lab experiment to maximize speed through trial and error.	5
6	F 10	Ac Reserve	My 6 week deliverables are: • Final controller implementation. Fully functional simulation. Max quadrotor speed determined with suffering less than 5cm root mean square error in the trajectory	

7	F 17	M = Pres Day 6 wk grds W	**Buffer**	5
8	F 24		**Buffer**	5
9	M 2	M = Columbus	**Buffer**	5
SB	M 9			
10	M 16		**Buffer**	5
11	M 23		**Buffer**	5
12	M 30	Ac Reserve	My 12 week deliverables are: • Final project demonstration. Fly Hummingbird dive trajectories in lab and determine max speed quadrotor is capable with suffering less than 5cm RMS error.	
13	Apr 6	12 wk Grades T	• Share draft poster with adviser for comments. Use template.	
14	A 13		• Submit poster to MSC for printing.	
15	A 20		• Share draft report with adviser for comments. Use template. • Capstone day	
16	A 27	T= last day of class	• Schedule technology transfer with adviser	

APPENDIX II: MATLAB SCRIPT—ROS VEHICLE

WRITTEN BY MIDN GAINER

why?

```
classdef ros_vehicle < matlab.mixin.SetGet
  % This class creates a matlab vehicle object which can be used to send
  % commands and receive data from ROS Enabled vehicles
  %   Detailed explanation goes here

  properties
      name;
      type;
      pose_topic
      vel_topic;
      desvel_topic;
      path_topic;
      traj_topic;
      wp_topic;
      path_pub;
      wp_pub;
      des_vel_pub;
      pose_sub;
      vel_sub;
      pos;vel;yaw,pose;rpy,des_vel;

  end

  methods
      function obj = ros_vehicle(varargin)
          obj.name = varargin{1};
          if(nargin>1); obj.type = varargin{2};end;


          obj.path_topic = sprintf('/%s/path',obj.name);
          obj.traj_topic = sprintf('/%s/trajectory',obj.name);
          obj.desvel_topic = sprintf('/%s/des_vel',obj.name);
          obj.wp_topic = sprintf('/%s/waypoint',obj.name);
          obj.path_pub = rospublisher(obj.path_topic,rostype.nav_msgs_Path);
          obj.wp_pub =
rospublisher(obj.wp_topic,rostype.geometry_msgs_PoseStamped);
          obj.des_vel_pub =
rospublisher(obj.desvel_topic,rostype.geometry_msgs_Twist);

          obj.vel_topic = sprintf('/vr/%s/vel',obj.name);
          obj.pose_topic = sprintf('/vrpn_client_node/%s/pose',obj.name);
          obj.pose_sub =
rossubscriber(obj.pose_topic,rostype.geometry_msgs_PoseStamped,@obj.poseCallback);
          obj.vel_sub =
rossubscriber(obj.vel_topic,rostype.geometry_msgs_Twist,@obj.velCallback);
%             obj.desvel_sub =
rossubscriber(obj.desvel_topic,rostype.geometry_msgs_Twist,@obj.desVelCallback);
          obj.rpy = [0;0;0];
          obj.pos = [0;0;0];
          obj.vel = [0;0;0];

      end

      function poseCallback(obj,pose_sub,msg)
          obj.pos = [msg.Pose.Position.X;msg.Pose.Position.Y;msg.Pose.Position.Z];
```

never cited or mentioned in text.

```
            obj.rpy = [msg.Pose.Orientation.X; msg.Pose.Orientation.Y;
msg.Pose.Orientation.Z];
%               [r,p,y]=quat2angle([msg.Pose.Orientation.W msg.Pose.Orientation.X
msg.Pose.Orientation.Y msg.Pose.Orientation.Z],'XYZ');
%               obj.rpy = [r;p;y];
        end
        function velCallback(obj,vel_sub,msg)
              obj.vel = [msg.Linear.X;msg.Linear.Y;msg.Linear.Z];
        end
        function desVelCallback(obj,desvel_sub,msg)
              obj.des_vel = [msg.Linear.X;msg.Linear.Y;msg.Linear.Z];
        end
        function [pos,rpy] = get_pose(obj)

            pos = obj.pos;
            rpy = obj.rpy;
        end

        function vel = get_vel(obj)

              vel = obj.vel;
%             yaw = [msg.Pose.Orientation.Z]
        end
        function dvel = get_des_vel(obj)

              dvel = obj.des_vel;
%             yaw = [msg.Pose.Orientation.Z]
        end
        function send_vel(obj,vel)
            msg = rosmessage('geometry_msgs/Twist');
            msg.Linear.X = vel(1);
            msg.Linear.Y = vel(2);
            msg.Linear.Z = vel(3);
            send(obj.des_vel_pub,msg);

        end
        function send_wp(obj,pos)
            msg = rosmessage('geometry_msgs/PoseStamped');
            msg.Pose.Position.X = pos(1);
            msg.Pose.Position.Y = pos(2);
            msg.Pose.Position.Z = pos(3);
%             obj.wp_pub.send(msg);
            send(obj.wp_pub,msg);

        end

        function send_path(obj,path)
            msg = rosmessage('nav_msgs/Path');
            tmp_pose = rosmessage('geometry_msgs/PoseStamped');
            [m,n] = size(path);
            for i=1:m

                msg.Poses(i).Pose.Position.X = path(i,1);
                msg.Poses(i).Pose.Position.Y = path(i,2);
                msg.Poses(i).Pose.Position.Z = path(i,3);
            end


        end
```

```
        %           function obj = poseCallBack(obj,pose_sub,msg)
        %               obj.pos =
[msg.Pose.Position.X;msg.Pose.Position.Y;msg.Pose.Position.Z];
        %               obj.pos = pos;
        %               [roll,pitch,yaw] =
quat2angle([msg.Pose.Orientation.W,msg.Pose.Orientation.X,msg.Pose.Orientation.Y,msg.
Pose.Orientation.Z],'XYZ');
        %               obj.rpy = [roll;pitch;yaw];
        %           end

    end
    methods (Static)

    end

end
```