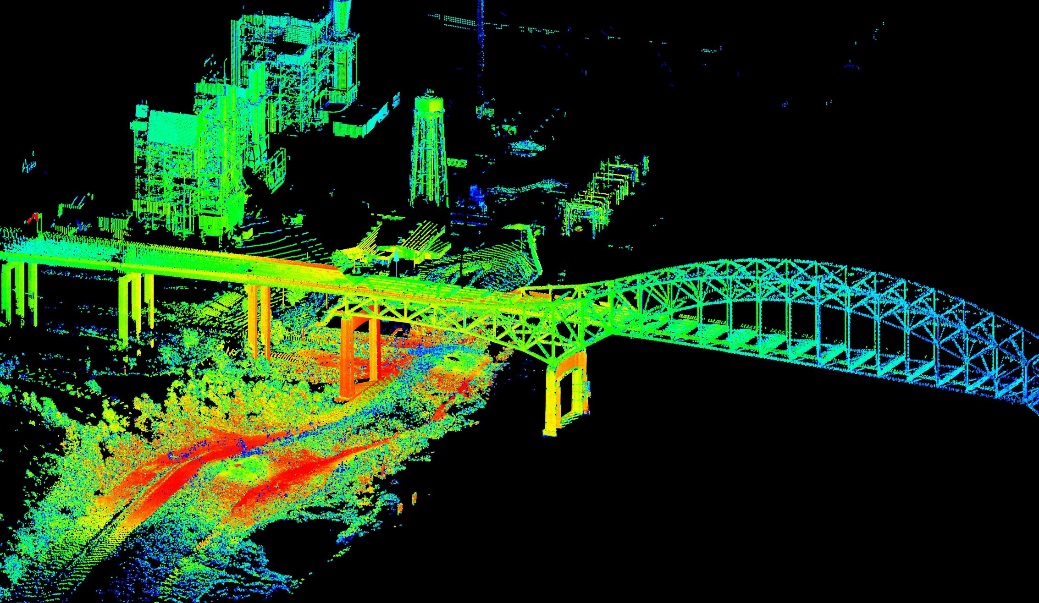
Autonomous Drone Racing



Replace with your graphic.

**Author:** Mid. 1/C Credle  
**Date:** Dec. 5th, 2019

**Adviser:** Prof. D. Evangelista

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_  
(signature) (date)

**Department Chair:** Prof. B. E. Bishop

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_  
(signature) (date)

Autonomous Drone Racing

A. Credle and D. Evangelista

**Abstract**

Autonomous racing drones are not only beneficial to the drone racing sport, but also in autonomous vehicles and in military drones flying through windows, chimneys, etc. Research in this field is often based around either race gate recognition or flight controls, but this project will incorporate both. We will create a process for a drone to modify its given flight path based on the visual recognition of the gate in order to fly through the gate. The processor will identify the gate, locate it in relation to the drone’s current path, and modify the path to fly through the gate. This project will demonstrate this process by using simulations and physical experiments. Simulations will consist of individual tests of recognition, location, and path modification processes, and then tests for all three together. Physical experiments will test all three process together in a real environment. Success will be measured by the difference in distance between the path and the center of the gate, as well as the difference in trajectory at the same point. The total cost of the project is $29,666- with $1586 of that being new equipment. This project is scheduled to have the coding and simulation done within the first semester and physical experiments done by the end of the second. The biggest risk involved is with the drones crashing, so the physical experiments have more time than simulations to ensure safety. If successful, this will be the first step in creating an autonomous racing drone at USNA.

I.BACKGROUND AND MOTIVATION

The concept of drone racing is straightforward: a group of people fly unmanned aerial vehicles (UAV) through gates, the first one to the finish line wins. The UAVs, or drones, come in a variety of sizes, ranging from an inch in diameter to over a foot, and are flown through communication with a remote controller. Races can take place in a variety of conditions including time of day and location. Gates can be configured in an unlimited number of patterns and often come in circular or rectangular form, though they are not limited to these shapes. Limitations are often placed on the power and size of drones that are raced, as owners are often expected to bring their own equipment [1]. While the concept of racing is not new, drone racing is one of the fastest growing sports in the world [2]. Its fame is quickly growing, and sports broadcasting companies like ESPN are signing contracts to broadcast competitions [3]. There are no limitations on age, gender, ethnicity, or physical prowess, making drone racing available to everyone.

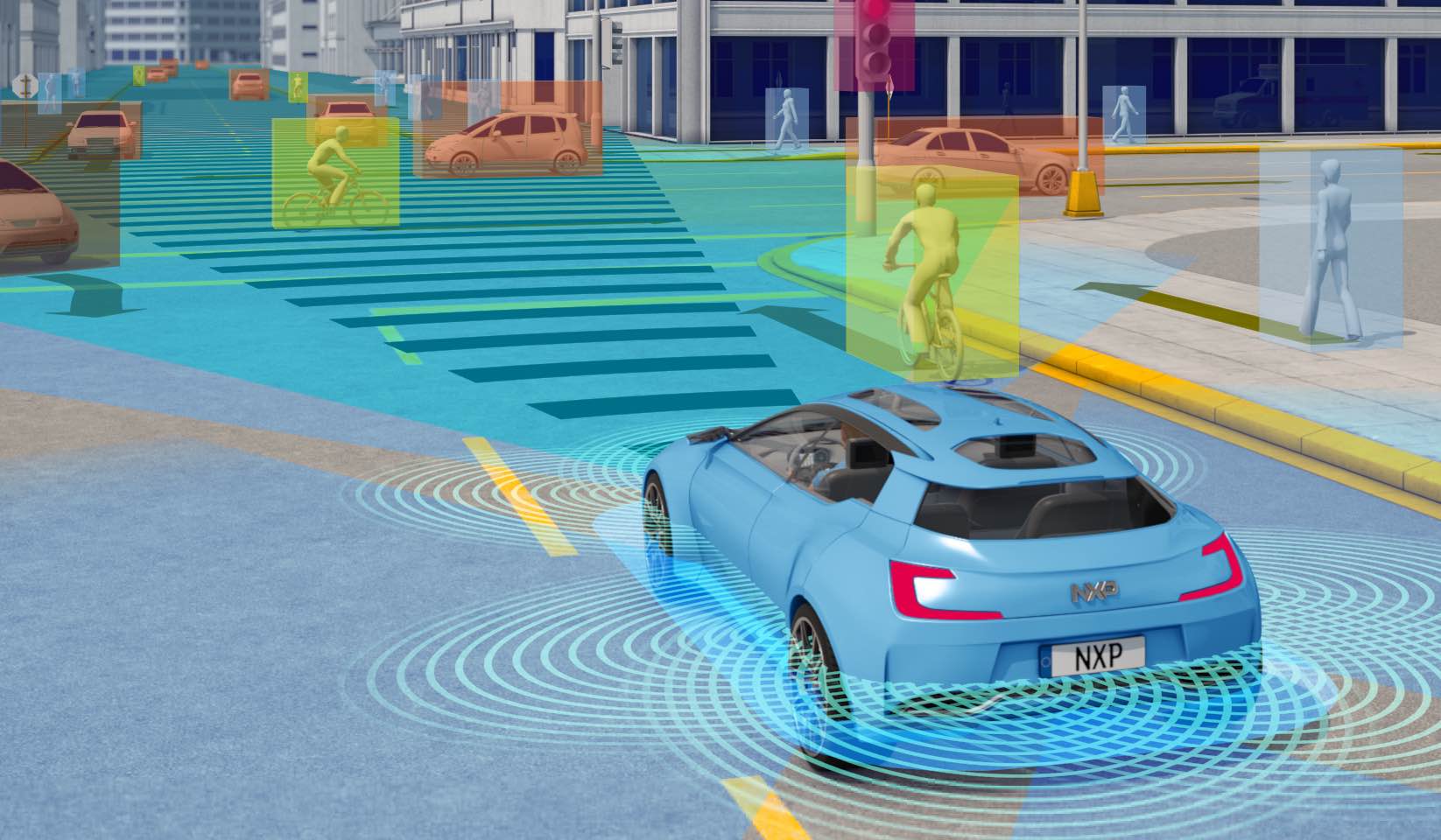


www.droneracingleague.com

Figure 1: Two drones racing

The concept of autonomous drone racing is even newer, though its potential goes beyond that of its sport. In concept, a human drone racer goes through a series of ques and maneuvers to race the drone. If these ques and maneuvers are broken down into systematic steps, it is possible to automate the process and have the drone fly itself through the course. In modern autonomous vehicles, the specific location of the vehicle is often known, through GPS, a tracking system, etc. While this method of localization works for vehicles with static environments, it does not take into account the location of its surrounds, or the vehicles location in relation to specific objects.

By mixing both an assumed path and gates for the drone to fly through, the exact location of the drone is not required; its relative location to the gates is all that is needed to fly a successful race. This concept will be critical for future autonomous systems, such as self-driving cars, that are required to navigate based on an assumed path mixed with the vehicles relative to the environmental objects or barriers (Fig. 1) [4]. For cars, the environmental objects would include lane lines, other vehicles, curbs, objects in the road, etc [5]. By integrating objects of the environment into the core of the navigation process, autonomous vehicles will be able to keep both passengers, pedestrians, and wildlife safe. Autonomous vehicles of the future must be able to navigate based on both an assumed path and a relative location in order to efficiently and safely navigate dynamic environments.



www.electronicsweekly.com

Figure 2: An autonomous car sensing its environment

As the first autonomous drone-racing project at USNA, this will open the door to future drone research for midshipmen. EW281 and EW282 provide opportunities for midshipmen to experiment with drones on a hardware and flight-testing level, but this project will allow for future drone development on the software and autonomy level. Additionally, this research is only the first step in creating a fully autonomous racing drone that rivals a human’s performance. Future steps include path optimization, recognition with visual uncertainties, racer collision avoidance, etc. This research will be one small step in the larger picture of fully autonomous drone racing.

II. PROBLEM STATEMENT

Given a three-dimensional flight path, drone accelerometer readings, and the image of a drone-racing gate in a three dimensional environment, this research intends to begin the process of developing autonomous drone flight through a drone-racing course. Given a three-dimensional path that does not fly through the gate, we will find and test the feasibility of a guidance system that creates a similar path that does. This system will be tested by placing a small quadrotor in multiple scenarios in relation to a race gate(s), with varying pre-determined paths, and directing the guidance system to fly the drone through the race gate(s). Success of this system is based on the ability to autonomously correct the path and fly the drone through the center of the gate without contact. This will be measured by the difference in distance between the path at its intersection with the gate, and the center of the gate. The competing objective will be to maintain the same trajectory at the intersection with the gate as if the path had not been modified. The final output of this project will a guidance system that could be implemented on any racing drone with a camera and IMU and fly through race gates autonomously.

New Path

Trajectory at Gate Intersection

Original Path

Figure 3: Depiction of flight path correction (original path in green, new path in blue, trajectory at gate intersection as red arrow, gate intersection point as red dot)

III. LITERATURE REVIEW

This paper is the one of the first developments of a drone autonomous flight program, so the background research encompasses papers that focus on multiple aspects of autonomous flight. The main categories that these papers include navigation, flight controls, and visual recognition. This research focuses on the visual recognition of the gate and navigation, but these flight controls is necessary for the background research because it affect the performance of navigation and is necessary for basic demonstration.

In order to understand the movement patterns of the drone, [6] provides a set of equations that could be used to accurately model the induced drag and thrust forces. These equations were derived from proofs using properties of physics, and then the coefficients were identified through flight experimentation. This provides suitable estimations for the forces acting on the drone, which allows for control of the drone through the gate. This is connected to [2], which created a state space model that could accurately predict the movement of a small drone with aggressive flight. By determining these equations through modeling and live testing, these state equations can be coupled with the flight force equations from [6] to form an accurate estimate of the drone’s location. Coupling these with a flight controller will allow the drone to have a basic understanding of its location within the general space. This research intends to use visual servoing to adjust a drone’s flight paths, and in order to understand the drone’s relation to the predetermined flight path, [6] and [7] provide an accurate location.

While [6] and [7] provide what is believed to be an accurate location of the drone in-flight, its location in relation to the preprogrammed flight path is not always certain. [8] provides the groundwork for a concept known as “nano mapping”, which refers to modeling a 3D data structure depicting obstacles around the drone. This technique is different from a traditional mapping approach because traditional mapping is based off a common world frame. [8] is based solely on the frame of the drone, allowing for navigation based off the drones currently location, rather than its location relative to a known point. This concept is also crucial to the development of this research because the coupled flight path planning approach requires both the common world frame and the drone world frame.

Different methods for seeing the world around the drone exist, including range finding technology (used for altitude measurements), single camera, and multi-camera approaches. [7] is based on a single camera approach, while [4] found that a three camera setup was optimal for uncertain terrains. [9] is based on autonomous navigation of drones in a wooded area where the location of the trees is unknown before flight. The drone does not have a preprogramed path, but rather has a program to recognize key features along a route and to estimate the likelihood of needing to turn. While the racecourse for this research is assumed to be known, a single camera approach will be utilized for hardware simplicity; estimating the likelihood that the computer is seeing a gate, however, may be utilized in this research. If a neural network could be constructed to follow a path through a forest, a similar neural network could be formed to fly through an aerial path.

One of the main challenge that arises with flying through the gate is recognizing multiple gates. In small drone racing courses, it is likely that the drone will be able to see multiple gates within its field of view. [10] worked through this problem and found reasonable neural network parameters to mitigate this issue. By assuming the next gate in the racecourse would be the largest gate in the drone camera view (using a single camera approach), [10] removed the neural network layer of all image analysis and left only the recognition of the largest circle gate. By removing this process in the image analysis, [10] found a significant reduction in computation time while only reducing the accuracy of gate recognition by less than 10% (464ms to 34ms reaction time, 82.4% to 75.5%).

The other main challenge that arises after recognizing the proper gate to fly through is implementing a visual servoing program to direct the drone through the gate. [11] created a process of gate recognition that located the center of the gate in relation to the location of the center of the cameras view and redirecting the drone towards the center of the gate. This research is the closest process to the research in this paper, as it specifies a visual recognition process that is simplified into the computational level of an inflight processor.

While many of these papers are useful in the development of an autonomous drone, it should be noted that there lacks a consistency among drone researchers in the assumptions and underlying assumptions. As described earlier, some researchers utilize a single camera operating system, while others use multi-camera systems, and even some include range-finding technology. Many of the subjects of these papers delved into different subsections of drone autonomy, so it is reasonable that their methods varied. For research such as [6], [7], and [8], their focus was more for drone flight in general, so this critique does not apply as much as it does for projects such as [9], [10], and [11], which had varying sensor and visual capabilities. As the latter 3 projects had more to do with direct visual recognition, it would have been more advantageous for the autonomous drone community had the projects been embarked on in a similar manner.

These projects center around the research in this proposal in two ways: by giving basic flight control and navigational equations to use in baseline location estimations and flight controllers, and by providing simplistic approaches to visual recognition to model my approach. While my approach uses both a whole world frame and a drone view frame to form a novel approach to gate recognition, many of the approaches depicted in [9], [10], and [11] can be used to model a realistic approach to melding the two frames. This project will take these approaches and form a simple and novel method to visual gate recognition and path modification.

IV. Materials and Methods

*A. Materials*

The original budgeted proposal called for four Tello drones, 2 Vortex 180 drones, and 2 sets of LED race gates. Upon receiving the Tello drones, it was found that their preprogrammed flight controls had a minimum movement requirement per command of 20mm. To best utilize the Tello drones and calculate the future path correction accuracy, a larger gate became more advantageous. The USNA School of Drones has a set of 5’x5’ golden race gates, so the smaller LED race gates have been substituted. The Vortex 180 drones have not been purchased as they are apart of the stretch goal for this research, and will not be utilized until the spring.

*B. Methods*

The methods for proof-of-concept included three main phases: Tello drone flight by computer, Tello video streaming to computer, and computer gate recognition. These three sections were developed separately and will be integrated together in future research. Each is operated by a separate python program on the same Linux system.

For Tello drone flight, the first phase was to connect to the Tello drones flight controls in any manner possible to understand its capabilities. The easiest method was a connection over a smartphone through an app developed by DJI through the Tello’s Wi-Fi. Upon completion of this flight, focus was shifted to flight under computer control. A python program was written utilizing the base code provided by DJI to open a socket between the drone and Linux terminal. This connection was also established over the Tello drones Wi-Fi. The Tello is preprogramed to receive text commands over the socket and perform a take-off maneuver, landing maneuver, or translational/ rotational movements. Each of the preprogrammed commands were tested and verified to operate according to the DJI operational instructions REFERNCE.

For Tello video streaming to a computer, a python program was developed using the base code provided by DJI to connect the Tello’s on-board camera to the same Linux system. This worked in a similar manner to the flight controls by connecting over the Tello’s Wi-Fi using a socket connection, but differed in the required port connection. The connection allowed the computer to receive real-time flight video data, which was converted using an h256 decoder. The video stream was then integrated into a GUI for ease of viewing, and tested to ensure minimal connection lag and accurate color imaging.

For computer gate recognition, the computers onboard camera was utilized so that integration of the previous step was not required. A python program was developed to view the real-time camera data and segment it based on HSV color segmenting. Appropriate values were found to segment out all colors other than the yellow/gold of the actual race gate, and visual race gate created using a sticky-note on a black board. The segmented sections of the video feed were then connected and analyzed for size. Small clumps were removed in an attempt to keep only the race gate. The remaining objects in the image were then analyzed for their number of edges. Knowing that the race gate has 8 edges, any shape without 8 edges was removed. This left only the race gate in the segmented imaging.

Integration of these pieces will allow for path correction to be tested. Future research will connect the video feed from the Tello drone into the gate recognition process. This allows for the drones relative location to the gate to be calculated and integrated with the flight controls to pass through the gate.

*C. Results*

V. CONCLUSION

In conclusion, this research will create a process for a drone to modify its flight path based on the visual recognition of racing gates. This is the first steps in the process to create a fully autonomous racing drone. Future steps will include path optimization and processing optimization. The process for this research will identify the gate, locate it in relations to the drone, and modify the flight path to fly through the gate. The processor will have a visual feed, the three dimensional acceleration data of the drone’s flight, and the pre-determined three-dimensional path. The process will output a new three-dimensional path for the drone to follow in order to fly through the gate. This project will demonstrate this process by using the Liftoff Flight Simulator and physical experiments to refine and optimize the program and measure results. Success will be measured by the accuracy of the new three-dimensional path as it passes through the gate. Accuracy is based on the difference in distance of the path and the center of the gate. The biggest risk involved with this research is damaging the drones during the experimental demonstrations, but this risk will be mitigated by extensive use of the simulator first.

ACKNOWLEDGEMENTS

We thank Prof Esposito, MIDN 2/C Ethan Marcello, and the EW502 class for helpful comments which were incorporated into this proposal. We thank MIDN 1/C Canlas and MIDN 1/C Cunniff for their advice and suggestions.

REFERENCES

[1] Red Bull UAE, “Drone racing: The Sport of the Future,” Red Bull, 18-Oct-2018. [Online].

[2] J. Condliffe, “Is Drone Racing A Sport Yet?,” *MIT Technology Review*, 19-Sep-2016. [Online].

[3] A. Marshall, “ESPN Has Decided Drone Racing Is a Sport Because Internet,” Wired, 03-Jun-2017. [Online].

[4] “How do Self-Driving Cars Work?,” IoT For All, 05-Oct-2018. [Online].

[5] S. Rayej, “How Do Self-driving Cars Work?,” Robohub, 03-Jun-2014. [Online].

[6] J. Svacha, K. Mohta, and V. Kumar, “Improving quadrotor trajectory tracking by compensating for aerodynamic effects*,” 2017 International Conference on Unmanned Aircraft Systems*, *ICUAS 2017*, pp. 860–866, 2017.

[7] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, “Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2017.

[8] P. R. Florence, J. Carter, J. Ware, and R. Tedrake, “NanoMap: Fast, Uncertainty-Aware Proximity Queries with Lazy Search over Local 3D Data,” *Preprint!*, 2018.

[9] A. A. Zhilenkov and I. R. Epifantsev, “The use of convolution artificial neural networks for drones autonomous trajectory planning,” *Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2018*, pp. 1044–1047, 2018.

[10] S. Jung, S. Hwang, H. Shin, and D. H. Shim, “Perception, guidance, and navigation for indoor autonomous drone racing using deep learning,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2539–2544, 2018.

[11] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, “A direct visual servoing-based framework for the 2016 IROS Autonomous Drone Racing Challenge,” *Journal of Field Robotics*, vol. 35, no. 1, pp. 146–166, 2018.

APPENDIX

Table I

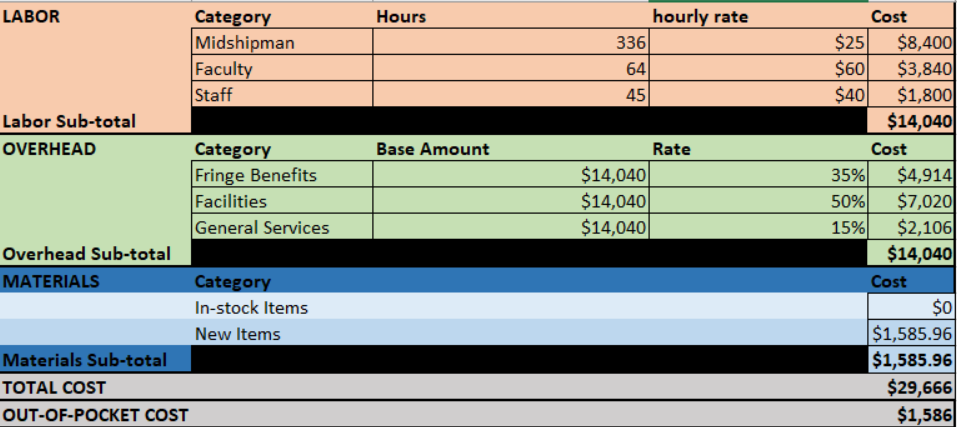
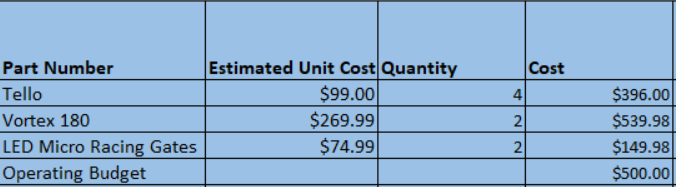


Table II



Enclosure 1: Timeline

**Timeline-Fall AY2020**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week (M)** | | **Note** | **Planned Activities** | **Actual Hrs** |
| 1 | Aug 19 | Class begins on M | Relearn Python (learn and practice) | 4 |
| 2 | A 26 |  | Relearn Python (learn and practice) | 4 |
| 3 | Sep 2 | M= Labor Day  T ← M | Relearn Python (learn and practice) | 4 |
| 4 | S 9 |  | Create Pseudo-code for flight controller | 6 |
| 5 | S 16 |  | Integrate visual input into code | 3 |
| 6 | S 23 | Ac Reserve | My 6 week deliverables is: Pseudo-code for flight controller  Create code to detect gates | 4 |
| 7 | S 30 | 6 wk grades due Tues | Integrate pre-determined path into code | 5 |
| 8 | Oct 7 |  | Integrate acceleration data reading into code | 5 |
| 9 | O 14 | M = Columbus | Create code to calculate location of gate | 4 |
| 10 | O 21 |  | Finish creating code to calculate location of gate | 4 |
| 11 | O 28 | Ac Reserve | My 12 week deliverable is: input integrated code  Begin control algorithm creation | 4 |
| 12 | Nov 4 | 12 wk Grades due Tues | Continue control algorithm creation | 4 |
| 13 | N 11 | M = Vets Day | Finish control algorithm creation | 4 |
| 14 | N 18 |  | Test gate recognition in simulation | 5 |
| 15 | N 25 | Tgiving Thurs-Fri | Test gate location in simulation | 5 |
| 16 | Dec 2 | R= last day class | Test path modification in simulation | 5 |

**Timeline-Spring AY2020**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week (M)** | | **Note** | **Planned Activities** | **Actual Hrs** |
| 1 | Jan  6 | Begins Tues (M sched) | Modify code based on simulation testing | 4 |
| 2 | J 13 |  | Modify code based on simulation testing | 4 |
| 3 | J 20 | M= MLK | Set up experimental testing | 3 |
| 4 | J 27 |  | Begin experimental testing (single gate) | 4 |
| 5 | Feb 3 |  | Continue experimental testing (single gate) | 4 |
| 6 | F 10 | Ac Reserve | My 6 week deliverable is: experimental testing data  Analyze experimental data | 3 |
| 7 | F 17 | M = Pres Day 6 wk grds W | Modify code based on experimental testing | 4 |
| 8 | F 24 |  | Modify code based on experimental testing | 4 |
| 9 | M 2 | M = Columbus | Set up experimental testing for multiple gates | 3 |
| SB | M 9 |  |  |  |
| 10 | M 16 |  | Begin experimental testing (multiple gates) | 4 |
| 11 | M 23 |  | Continue experimental testing (multiple gates) | 4 |
| 12 | M 30 | Ac Reserve | My 12 week deliverable is: multiple gate experimental data  Analyze experimental data | 3 |
| 13 | Apr 6 | 12 wk Grades T | Modify code based on multiple gate experimental data | 5 |
| 14 | A 13 |  | Submit poster to MSC for printing, work on Capstone presentation | 5 |
| 15 | A 20 |  | Capstone day | 5 |
| 16 | A 27 | T= last day of class | Schedule technology transfer with adviser | 3 |