Reduced Gravity Flight: TEARDR0P

Andrew Brahim, Jourdain Francis

Dept. of Electrical and Computer Engineering
University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Since man has achieved the capability to enter the final frontier, experimentation in a microgravity environment has exponentially increased our understanding of the Universe. The Terrestrial Experimentation and Research Drone Regulating 0-G Parabola (TEARDR0P) aims at providing a low cost and time effective means for achieving reduced gravity experimentation. The drone implements autonomous flight missions to carry a small payload meant for experimentation on a trajectory that simulates a reduced gravity environment similar to that found in space. With the use of a camera feed and blackbox datalogging, information is parsed and exported in an easy to understand format.**

***Index Terms* — AC motor, aerospace electronics, aircraft navigation, global position system, motor drives, radio communication, telemetry**

I. INTRODUCTION

The need for low cost microgravity (reduced gravity) experimentation platforms exist for researchers who may not have access to drop tower facilities or the funds to carry out existing methods of collecting data in reduced gravity environments. Parabolic flight requires modified commercial airline equipment that researchers do not have, and renting such a flight or service is not a feasible cost for the average university faculty scholar or researcher. Ergo, we were given a proposal from Northrop Grumman to make a drone-based system to carry a payload for experimentation at predetermined altitudes for free fall drops or parabolic flight. This project set out to create reduced gravity conditions via drone flight for a length of time necessary for payload experiments to be recorded. The payload environment must also be recorded and instrumentation such as drag, acceleration, temperature, and velocity are necessary.

We were tasked to work with an assortment of other engineers from the Mechanical and Aerospace (MAE) and Computer Science (CS) department. Specifically, Electrical and Computer (ECE) engineers have designed the power distribution circuit, implemented an embedded system required for sensor data logging, and developed and configured the control system.

The interdisciplinary team's goal is to carry a payload minimum 10 pounds at roughly the size of a shoebox via drone and perform flight maneuver to simulate reduced gravity within 10^{-1} to 10^{-4} G's for 2.5 seconds or more. The data is then recorded and is stored on an onboard device for later interpretation.

Due to the altitude, payload, and air speed variables in this project, our design is limited by the upper limit of the Federal Aviation Administration (FAA) regulations for unmanned aircraft. An aircraft weighing less than 55 lbs falls under part 107 of the Federal Code of Regulation, which imposes regulations on model/hobby or small industrial/commercial grade unmanned aircraft. Part 107.51 of FAA regulation limits air speed of such craft to 100 mph and imposes a maximum flight ceiling of 400 feet above ground level.

II. FLIGHT TECHNICAL OBJECTIVES

A. Microgravity Flight Time

The drone is designed so that microgravity can be achieved for somewhere between three to seven seconds. In this time frame, experiments will be in the condition necessary to record any and all relevant data and be stored to be reviewed for a later time. The success of reaching the desired measurement requirement is highly correlated with the payload weight and dimensions. With more weight and higher dimensions, the drone may have a more difficult time achieving microgravity for such a time frame.

The stability of the freefall can also mean the difference between a controlled freefall microgravity environment or a high gravity situation due to uncontrolled circumstances such as rotation and noise from vibrations.

$$H_{initial} = (Max\ Allowable\ Height) - \frac{(Initial\ Velocity)^2}{2(32.2)} \quad (1)$$

$$T_{ff} = \frac{(Initial\ Velocity) + \sqrt{(Initial\ Velocity)^2 - 4(\frac{-32.2}{2})(H_{initial} - H_{critical})}}{32.2} \quad (2)$$

B. Gross Weight

The weight of the drone will be around twenty pounds, or 9.07 kilograms. This weight is the sum of the physically drone, electrical and mechanical components, and the payload itself. The design and materials used in the creation of the drone are essential in maintaining the desired parameters of the weight. A heavier craft will require higher power output in order to stay within the constraints of the other engineering requirements. The drone is built using mostly carbon fiber and 3D printed components.

C. Telemetry Hardware

(1) Initial Height Equation
(2) Freefall Time Equation

The hardware selected is able to accurately communicate data between the in-flight drone and the senior design group on the ground. Though the data recorded is stored onboard the drone until landed at the ground station, the RC telemetry is required to have a manual failsafe at all time. This makes the telemetry component one of the greatest and most important factors when deciding and setting the height, and henceforth time, that the microgravity drone will be able to stay and record the processes and experimentation of being in a microgravity environment. Our telemetry allows us to control and protocols throughout the entire duration of our trajectory.

D. *Reduced Microgravity Environment*

The reduced microgravity environment is the primary purpose of our senior design project. Our engineering requirement is to create an environment that can maintain 10^{-1} G throughout the duration of freefall. During this time, there drone will hold a payload that will contain small experiments to be carried out. These experiments will be saved, recorded, and held for later analysis while maintaining this state. The gravitational force experienced on the payload will be constantly recorded by an accelerometer. While freefalling, the drone will mechanically adjust itself to sustain stability and keep the payload at 10^{-1} G.

E. *Drop Height*

An appropriate drop height is needed to achieve an optimal duration of free-fall and recovery time. The maximum drop height is restricted by the maximum reach of the telemetry. The drop height therefore matches the maximum range of the telemetry hardware at 3000 feet, however we are limited to 400 feet via FAA.

F. *Flight Pattern*

There are several ways to simulate microgravity on a payload. These different methods, whether it'd be a fixed wing parabolic flight or freefall, require different constraints, configurations, and overall design and implementation by all team branches to simulate the desired results. For the purpose of this project, it was established that our group would implement freefall by a quadcopter drone. The drone is programmed to fly up to its maximum height and drop. While free-falling, the drone is required to not only maintain stability, but also add enough, if any is needed, downward momentum to counteract the drag and maintain the desired measurement of gravitational force on the payload.

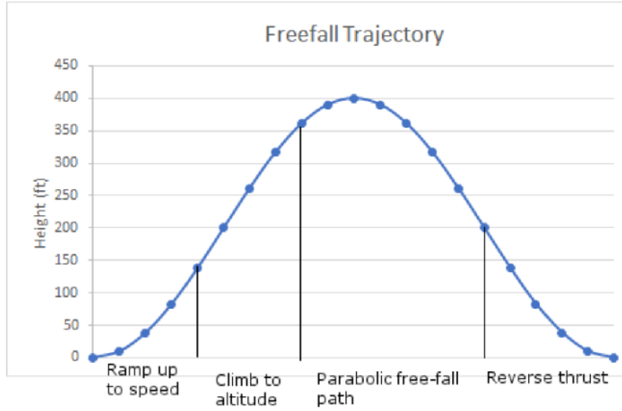


Fig. 1. Flight Pattern of the Quadcopter drone to experience reduced gravity

G. *Cost*

Our cost parameter is based on the budget given to us by Northrop Grumman. This budget is shared among the three branches of our team and will be used for the construction and purchasing of the necessary parts and components for every part of this project. The different components and the complexity of its design drastically change the cost. Finding a balance between quality and cost is essential in the completion of this project.

III. DEFINING THE PROBLEM

In the field of reduced gravity there are many ways to achieve reduced gravity, but each way comes with its own set of flaws and inconveniences. In order to produce and replicate reduced gravity experiments as desired by Northrop Grumman, we must be able to simulate a reduced gravity environment.

One way to obtain the amount of gravity, or lack thereof, necessary is by loading up the experiment onto a rocket and shipping it up to the International Space Station. The International Space Station serves as a central hub for man low earth gravity experiments with its expansive gallery of scientific laboratories. Another alternative involves the use of a plane that can fly at certain angles allowing for a parabolic flight. One such vessel, originally owned by Nasa and now currently operated by the Zero Gravity Corporation, is known as the Vomit Comet. The Vomit Comet serves its purpose in being far more accessible than sending research material to the International Space Station, but still not a good enough alternative to be considered for this project in regards to cost and ease of usability.

Drop towers are also an option typically used in attempting reduced gravity experimentation. The issue with this method is that building such a large tower takes a high amount of capital. Once constructed, the actual state of reduced gravity is directly limited by the height of the tower.

Once in development, other issues began to arise. The major problem being the ability to autonomously have a controlled reverse thrust while keeping the drone in freefall. Though programs exist that allow bidirectionality in compatible motor, there are very few ways at the moment to setup the need for reverse thrust autonomously.

IV. HARDWARE DECISIONS

There are a plethora of hardware components in that market that could allow for our team to achieve the desired outcome of our project. When researching, the final decisions were made with the constraints given by the desired design crafted by the MAE team, as well as the budget given by Northrup Grumman. Though most of the physical design of the craft were designed and implemented by the MAE team, all electronic components were selected and finalized at the digression of the ECE team.

A. Battery and Power Distribution

In order for our drone to operate effectively, there is a need for an extensive amount of power and a battery that can provide the necessary capacity. We decided on having one big battery for the sake of this project. The heavy weight of the battery has also given the added benefit of serving as a counterweight that sits at the bottom of the drone and stabilizes the craft.

The battery is the Lumenier 16000mAh six cell LiPo Battery with a C Rating of 20c and a burst of 40c. The battery connects to the Lantian 7oz 200A. This power distribution board allows for the simultaneous power for all four speed controllers as well as the onboard flight controller. For the case of the flight controller, a switching regulator (UBEC) is connected to the power distribution board to supply the device with 5V.

B. Electronic Speed Controllers

The speed controllers selected for our drone are meant to sync and control the rate, direction, and power the motors will exhibit during flight as dictated by the flight controller. Due to the high-power requirements dictated by the MAE team, the speed controllers picked have the following specifications:

- Input: 2-6s LiPoly
- Constant: 51 A
- Burst: 80 A
- DSHOT 1200 protocol
- Bidirectional mode
- Latest Bl Heli 32 firmware
- GD32 ARM Cortex-M MCU, 72mhz clock speed, PWM freq 48khz

The motors require a significant amount of current in order to operate. A high current speed controller is necessary to be able to withstand the bursts during sudden or prolonged thrust of the motors. In order to achieve a reduced gravity environment, the motors must be able to quickly shift from reverse to normal thrust in an instance.

C. Printed Circuit Board

In addition to an off-the-shelf flight controller, GPS module, and telemetry transceivers, the ECE team has elected to implement a small, lightweight, robust black box flight recorder. The recorder is self-contained and data logs without wireless transmission in order to reduce printed circuit board complexity and wireless interference with other signals (e.g. telemetry, flight controller remote). The design is based around an ATMEGA 328P-AU chip, an 8-bit microcontroller primarily used in the popular Arduino R3 Uno prototyping board. For this application, we will not be using the prototype board, but the microchip the board is based around.

The eagle cad electrical schematic is given in figure below. The AVR microcontroller is a RISC (reduced instruction set count) low power CMOS. Our 9 degree of freedom sensor will record x, y and z accelerations in the microgravity environment. We have chosen the MPU9250 sensor due to cheap cost and noise and drift immunity properties.

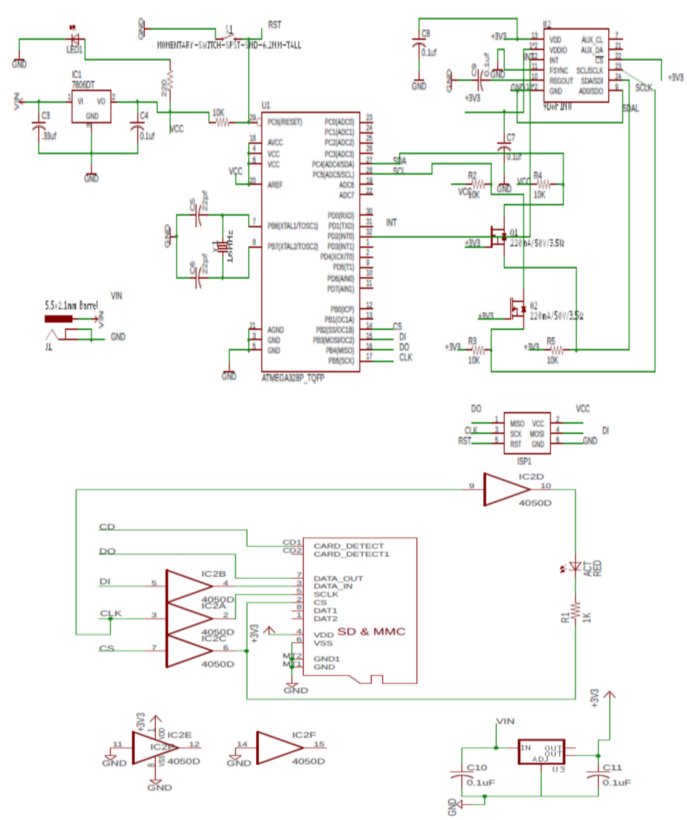


Fig. 2. The basic schematic of the PCB components before being implemented into a layout for the board

It is plenty robust for this small application involving only a 9 axis IMU and a data-logging unit (SD card storage). However, the microcontroller only contains 32 kilo-bytes of programmable flash memory and 1 kilobyte of EEPROM, leaving few options for long term data storage. Hence, we've added an SD card storage peripheral into the PCB. The SD card schematic is based on an Adafruit breakout board, which can directly connect to the ATmega328P via integrated level shifting and a 5v to 3v regulator. It has been modified for our application.

The board is set to record the accelerometer data throughout the duration of the flight. This feature was initially implemented to make up for a feature that was lost when choosing between flight controllers. The redundancy allows for a reaffirmation data received from the flight controller.

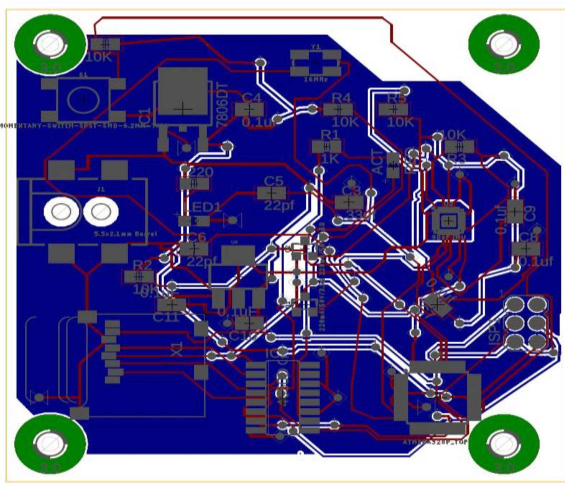


Fig. 3. Final PCB layout

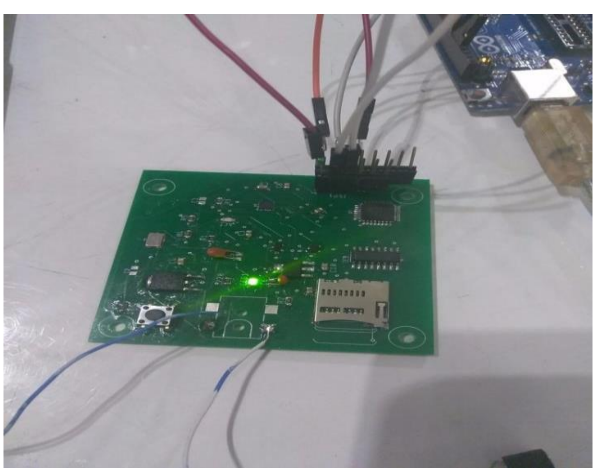


Fig. 4. The final revision of the PCB upon being boot loaded and programmed from an external Arduino Uno using In-System Programming (ISP)

D. Flight Controller

The flight controller plays an extremely important part in the ECE portion of this senior design project. As a team, there was an ample amount of research implemented into choosing a flight controller that can fit the constraints given to us by the MAE department and our own engineering requirements. For the purpose of this project, we have selected to use the Racing F4 EVO Flight Controller. The Racing F4 EVO Flight Controller provides an assortment of useful features that can be coded and modified to work with the components provided by the other branches of our senior design team.

The F4 EVO gives us access to built-in sensors as well as the ability to add our own. With the use of INAV, a branch of Betaflight the open source flight controller, autopilot and flight control patterns can be implemented and manipulated to incorporate parabolic flight patterns necessary for reduced gravity. The customizable firmware, based in the C Programming language, provides diversity in the electrical components, sensors, and overall capability.

A main feature provided by the Racing F4 EVO Flight Controller is compatibility for reverse speed control. Reverse speed control compatibility gives the drone accessibility to bidirectional motor control. This feature provides an extra degree of motion and maneuverability. In accordance with other components such as the gyroscope, barometer, and the accelerometer, bidirectional control of the quad motors will be implemented for a variety of purposes. This downward thrust provides stability as well as a canceling force while traversing along the reduced gravity path of trajectory.

TABLE I
Comparison between SP Racing F4 Evo and 3DR Pixhawk Flight Controller

Parameter	SP Racing F4 Evo	3DR Pixhawk
GPS	Not included (Hardware can be added)	Yes
Autopilot	Available through external means	Yes
Onboard IMU	ICM 20602 IMU	ICM 2060
Barometer	BMP280	MS5611
Processor	STM32405, ARM	STM32F427, ARM
Compatible Software	Cleanflight/Betaflight BLHeli32	Arducopter
Bidirectional Support	Yes	No
SD Storage	Yes	No
Cost	$37	$250

Originally, the choice flight controller was the 3DR Pixhawk Mini. This alternate flight controller offered an extremely user-friendly setup as well as certain components that are considered as external components for the SP Racing F4 Evo board. Shown in table I, there are certain comparable differences between the two boards.

The biggest factor of using the SP Racing F4 Evo over the 3DR Pixhawk is the bidirectional support. As mentioned earlier, bidirectional support is an integral part in the proper execution of a reduced gravity flight. Above, table 2 outlines the specifications for the onboard ICM-20602 accelerometer, which will be used to determine the success of all microgravity experimentation.

V. SOFTWARE

The SP Racing F4 Evo flight controller requires the cooperation and synchronization between three different flight navigation programs. During the development of the SP Racing F4 Evo, the initial firmware used to control the craft allowed for certain features over others. With the development of new software, certain features brought upon by the use of the firmware has been altered throughout its course. To achieve maximum usability during the implementation of the board with our product, the SP Racing F4 Evo board will be running a custom version of INAV. In order to fully understand INAV, one must understand the other firmware it is based off of.

TABLE II
ICM-20602 ACCELEROMETER DATASHEET

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY							
Full-Scale Range	AFS_SEL=0			±2		*g*	2
	AFS_SEL=1			±4		*g*	2
	AFS_SEL=2			±8		*g*	2
	AFS_SEL=3			±16		*g*	2
ADC Word Length	Output in two's complement format			16		bits	2
Sensitivity Scale Factor	AFS_SEL=0			16,384		LSB/*g*	2
	AFS_SEL=1			8,192		LSB/*g*	2
	AFS_SEL=2			4,096		LSB/*g*	2
	AFS_SEL=3			2,048		LSB/*g*	2
Sensitivity Scale Factor Initial Tolerance	Component-level			±1		%	1
Sensitivity Change vs. Temperature	-40°C to +85°C			±1.5		%	1
Nonlinearity	Best Fit Straight Line			±0.3		%	1
Cross-Axis Sensitivity				±1		%	1
ZERO-G OUTPUT							
Initial Tolerance	Component-level, all axes			±25		m*g*	1
	Board-level, all axes			±40		m*g*	1
Zero-G Level Change vs. Temperature	-40°C to +85°C	X and Y axes		±0.5		m*g*/ºC	1
		Z axis		±1		m*g*/ºC	1
OTHER PARAMETERS							
Power Spectral Density	@ 10 Hz			100		µ*g*/√Hz	1, 3
RMS Noise	Bandwidth = 100 Hz			1.0		m*g*-rms	1, 3
Low-Pass Filter Response	Programmable Range		5		218	Hz	2
Accelerometer Startup Time	From sleep mode to valid data			10	20	ms	2
Output Data Rate	Low-Noise mode		3.91		4000	Hz	2
	Low Power Mode		3.91		500	Hz	

A component present in the 3DR Pixhawk that is missing from the SP Racing F4 is the GPS unit. The GPS unit is a required component for our software to be able to incorporate autonomous flight. We are using the Ublox M8N. This device allows for accuracy up to 2.5 meters with a 0.5° heading accuracy and 0.1 m/s velocity accuracy. This accuracy is crucial in determining the speed, and henceforth acceleration, needed in order to achieve the results desired.

The first of these firmware is the open source BetaFlight firmware. This piece of software will be the base firmware in processes regarding flight performance, sensor monitoring, telemetry logging and protocol, and third-party support. BetaFlight allows for an arrangement of tools to be integrated a modified to accommodate the motors, and sensors involved in the design of the quadcopter.

The basic firmware code is structured in the C programming language. Changes for the firmware is necessary to allow for the implementation of other key features in later firmware.

BetaFlight allows for the configuration of the flight mechanics that will need to be set and adjusted before and during the flight. The program will allow for the control of the motors and the rudders. While working in conjunction with other firmware, the quadcopter drone will autonomously control the thrust and angle that has been planned prior to launch.

The second of these firmware is INAV. INAV is a program that adds certain navigational features that the previously mentioned firmware lack. The presence of a strong navigational component provides the possibility for autopilot. INAV includes a waypoint system. The waypoint system lets us, in accordance with the flight controller, physically map and trace out the course of action for the quadcopter drone. With mission planning in place, the drone will be able to take off, complete the experiment, and land safely without any and all external interactions.

The INAV program waypoint system is accessed and programmed by a Linux application called MWP tools (Figure 5). INAV includes several other features, such as access to in flight adjustment. .Running with this firmware on the flight controller enables the craft to follow the planned trajectory to achieve a reduced gravity environment suitable for testing.

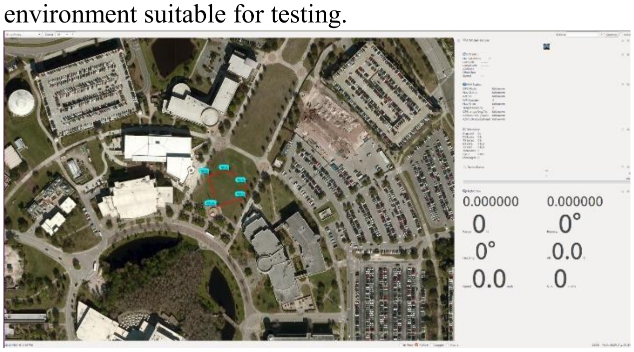


Fig. 5. MWP tools being used to draft a waypoint mission

A key adjustment to INAV was created by a user KenImhof on GitHub [1]. His branch of the program, named INAV3D, revolutionized the way in which the default INAV accounts for motor bidirectionality. This is down through certain changes in the navigation system of the code to account for the need of reverse thrust if dictated in a waypoint.

Upon further inspection of the code, there were further certain adjustments that had to be made to the already custom INAV3D firmware in order to achieve microgravity. Inserted into INAV are certain failsafe in the navigation system. This set up would constrain our drone to only being able to accelerate at one-fifth Gs and limiting velocity to that of 9 miles per hour.

For the drone to meet the specifications of each point of its trajectory, the drone would have to be able to break these certain conditions. Certain waypoints, in order to maintain microgravity, will need to achieve velocities over 90 miles per hour. By modifying the navigation system implemented into the code, we are able to break the limits set before.

A third program was also used extensively during the process through a BetaFlight pass through. The BLHeli program, though separate from the BetaFlight configurator, can be used to connect and configure the onboard speed controllers. BLHeli is a program that allows for flashing and configuring onto a speed controller. These configurations allow for certain settings being set, such as the capability for reversed motor function.

VI. Simulation and prototyping

A series of tests are necessary in order to adjust and adapt the firmware changes of the flight controller and the implantation of autopilot directives present in the Racing F4 EVO Flight Controller. In an effort to test the programming of the flight controller before the final development of the final drone project as created by the Mechanical and Aerospace teams, as well as preserve the integrity of the final product, the Electrical and Computer engineering branch of this senior design project has elected the use of simulation software.

For the sake of simulation, we have selected to use Matlab Simulink, or more specifically an addon called QuadSim. As the drone itself is a costly endeavor, this allows for testing without risking any damage to the physical drone that is attached to the electrical components for the end product.

Our reduced gravity unmanned aerial system, as defined by the engineering design constraints, is scheduled to reach nearly 400 feet above the ground and perform controlled and precise movements. A key part of these maneuvers is the presence of freefall. The lack of control or the occurrence of an error in the flight controller would leave the craft spiraling downwards towards a certain destruction.

Testing different control system approaches on a physically craft could equate to the accumulation of cost for necessary components that would consistently break upon impact. The motivation behind simulating the aircraft and reduced gravity environment is two-fold: the ability to test various control system approaches to best obtain clean reduced gravity, and ironing out kinks in our flight dynamics, system identification, and reduced gravity trajectories that could cause our quadrotor to become unstable. That latter case is an effort by our interdisciplinary engineering team to prevent collisions to the craft. This practice greatly increases the chance of success on the fun on the final craft. The quadcopter drone needs to be able to remain stable through the freefall while also being able to retain enough control to have a safe landing.

Reusability and cost are two major factors that this reduced gravity drone is tasked to solve. Meticulous testing through simulation ensures the future reusability of the craft by lowering the number of critical failures to the craft through testing. Money is also saved on the craft by not needing to produce another physical drone every time an experiment needs to be run. A system that crashes frequently also runs the unacceptable risk of destroying not only the materials used in the experiment, but also any potential data that could have been recorded throughout the duration of the flight.

Matlab is a widely-used and well supported by Mathworks in addition to 3rd party applications and add-ons. The interface is simple and allows easy access to live scripts, Simulink examples, and help/tutorial guides for many relevant topics pertaining to a quadrotor simulation.
QuadSim was used in order to get a better understanding in how to tune the PID as well as get a better grasp on the power to weight ratio of our drone. Figure 6 and Figure 7 below shows the results of this testing.

Fig. 6. Simulation of Drone reaching and hovering at a position

In order to get an accurate reading, intensive thrust testing was done in order to determine the RPM for major throttle points and was compiled into a table. QuadSim took these values and created the necessary constants used later in the simulation.

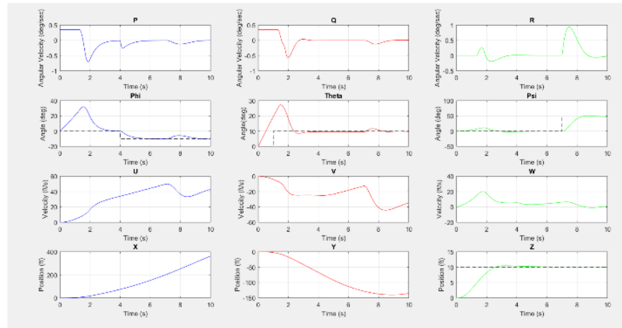


Fig. 7. Simulation of Drone reaching and hovering at a position

Other than simulation, testing was also done on a small test drone in order to receive accurate readings from the black-box sensor and test the different firmware modifications necessary to successfully execute the expectations for the main drone.

The initial concept of the PCB was also tested using a breadboard, Arduino, and breakout components. These tests aided in developing the Arduino code that would later become the final program implemented onto the PCB for data recording.
The raw data, shown below, is accessed post flight by the CS team and analyzed using their web application.

0:7:27	-0.124505	1.764018	-10.376999
0:7:42	-0.061055	1.752046	-10.402738
0:7:57	-0.070633	1.577859	-10.161510
0:7:72	-0.070633	1.577859	-10.161510
0:7:86	-0.108942	1.685604	-10.423688
0:7:2	-0.123906	1.772997	-10.263868
0:7:16	-0.083203	1.728103	-10.193235
0:7:31	-0.083203	1.728103	-10.193235
0:7:46	-0.126301	1.706554	-10.290205
0:7:60	-0.125702	1.731096	-10.506892
0:7:75	-0.116723	1.708948	-10.322528
0:7:90	-0.116723	1.708948	-10.322528

Fig. 8. Snippet of Raw Data from PCB

These easy to read graphs give us a visual way to view the trend of microgravity and its duration during the flight.

Fig. 9. Final Parsed Data

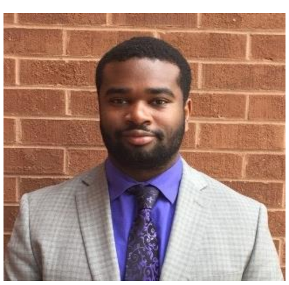
VII. Conclusion

The design constraints and decisions, as well as heavy integration with teams in both the Mechanical and Aerospace and Computer Science departments has allowed us to develop a mostly autonomous drone that is capable of running flight missions that would induce a reduced gravity environment on its payload. Large amounts of testing and prototyping has led us to successfully create the final large-scale drone. Additionally, inexpensive MEMS sensors and cheap, rapid, PCB prototyping allowed us to develop a simple, but accurate black box recorder, utilizing various techniques to reduce sensor noise and interference from outside sources or internal copper.

VIII. Acknowledgement

We would like to extend thanks and acknowledgement to Nathan Bodnar of the UCF radio club, and Kris Werner of the MPU9250 GitHub for consultation and guidance.

IX. Biography



Jourdain Francis will be graduating with his undergraduate degree in Computer Engineering in Spring of 2018. He currently works as a software engineering intern at L3 Mobile Vision. Upon graduation, Jourdain intends to move to Baltimore, Maryland in order to start his Master's degree at the University of Maryland, Baltimore County in Computer Engineering. Jourdain's primary interest are Software Development and Embedded Systems.

Andrew Brahim is presently a senior at the University of Central Florida. He intends to graduate with his Bachelor's of Science in Electrical Engineering in May of 2018. He is currently working at Mitsubishi Hitachi Power systems as an Engineering Associate, Instrumentation and Controls. He has no plans for graduate studies, however he has accepted an offer to work full time in the aerospace industry. Andrew's primary interests are computer simulation, sensors and instrumentation, and vehicle dynamics and control systems.

X. References

[1] Kenlmhof. *Inav3d: Navigation-enabled flight control software* (2018). https://github.com/KenImhof/iNav3d