Autonomous Racing Drone

ECE4011 Senior Design Project

Auto Quads
Project Faculty Advisor: Jennifer Hasler

Max Rudolph, EE, mrudolph8@gatech.edu
Dave Patel, CmpE, dpatel344@gatech.edu
Eddie Stevens, CmpE, estevens9@gatech.edu
Michael Bermudez, CmpE, mbermudez3@gatech.edu
Nyaire Najieb, CmpE, nyaire.najieb@gatech.edu
Rishov Sarkar, CmpE, rishov.sarkar@gatech.edu
Suhani Jain, CmpE, sjain340@gatech.edu

Submitted

November 18, 2019

Table of Contents

Executive Summary

In this modern age of autonomy, robotics are becoming more ubiquitous in society and our everyday lives. For instance, autonomous package delivery is beginning to become viable technologically and legally. These drone systems need to be reliable, safe, and, most importantly, fast. For this reason, we are designing an autonomous racing drone that can navigate a course without the interaction with a human pilot.

For this project, we expect to have the most difficulty when dealing with data pipelining and necessity for real-time control solutions. The drone needs to be fast enough to compete with a human without compromising the accuracy of the drone's trajectories. Thus, beyond the software limitations, we will face barriers when interfacing between the flight hardware (motors and ESCs) and flight software (path planners); specifically, we will need to decrease any lag in the whole software stack such that the drone accurately follows the planned trajectory.

We plan on using a commercial-off-the-shelf racing drone supplemented with a low-power embedded compute unit with a discrete GPU and a traditional visual sensor (either a 2D depth sensor or regular camera). As for software, there are several computer vision (CV) and path planning libraries that we will need to implement and fuse in order to successfully race a drone.

The bulk of our cost will consist of the hardware we plan to use. In total, we expect to need $1000 for drone parts and compute units. We are confident that we can achieve our goal because there have been several documented successes of autonomous drones navigating a course [1]. We hope to have a functioning drone that can autonomously fly at high speeds (30–40 mph) around a course made of hoops and obstacles by the end of the semester.

Nomenclature

PID = Proportional-Integral-Derivative

CV = Computer Vision

GPU = Graphics Processing Unit

ESC = Electronic Speed Controller

ROS = Robot Operating System

NVIDIA Jetson = micro computer kit

PixHawk = Flight controller for drone UAV systems

UAV = Unmanned Aerial Vehicle

RTPS = Real-Time Publish Subscribe

UART = Universal Asynchronous Receiver/Transmitter

TFR = Temporary Flight Restriction

PIC = Pilot in Command

QFD = Quality Function Deployment

Autonomous Racing Drone

1. Introduction

Our team, AutoQuads, is requesting $1,000 of funding to design and develop an autonomous racing drone: a drone that is able to autonomously pilot itself through professional drone racing courses. The design will require a drone kit for simplified customization, a Jetson Nano Development Kit for computations, and a Pixhawk flight controller to interface with the motors. Figure 1 helps visualize the structure and components of the autonomous flight system.

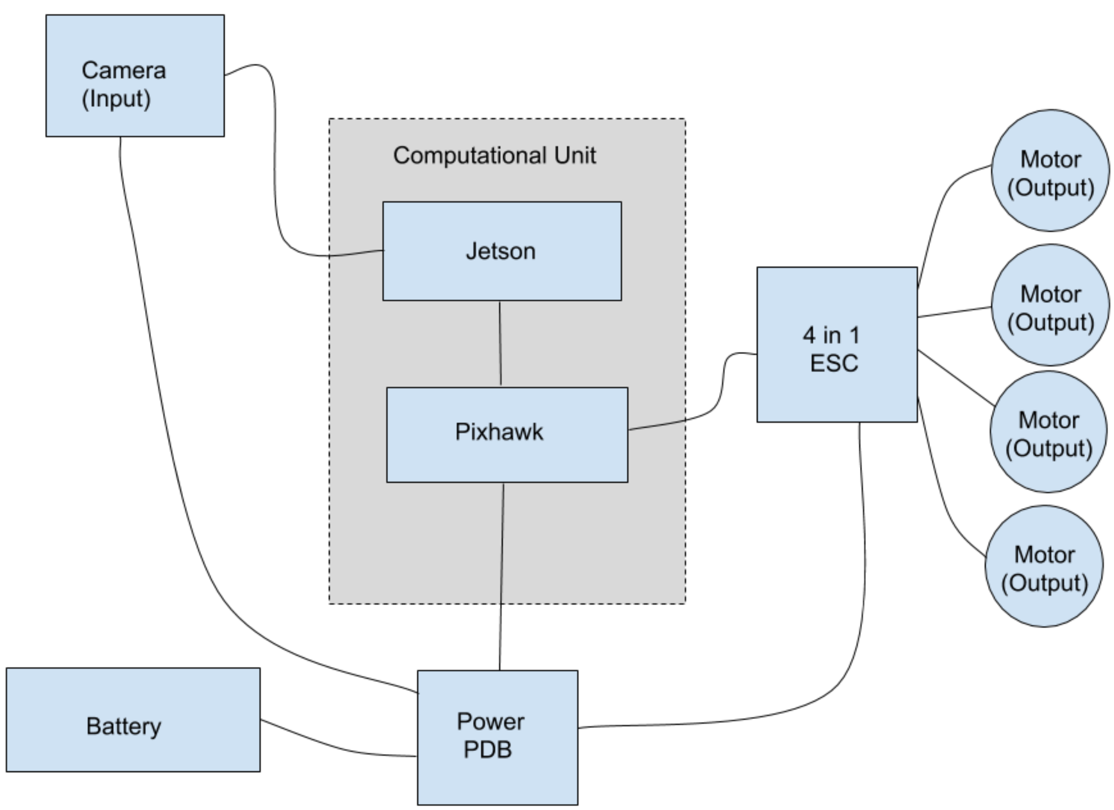


Figure 1. Block diagram of proposed autonomous flight system

1.1 Objective

The objective of the autonomous racing drone is to successfully compute a path to the goal and adjust the trajectory in real time as it senses its local environment and detects and prevents collisions. We will design an outdoor course with obstacles intentionally scattered throughout and test whether the drone is able to navigate itself to the goal position without crashing. We expect to focus on implementing the drone's ability to navigate and avoid collisions first before we chase higher speeds. However, being able to travel to the desired location quickly is a key part of our mission since we are designing a racing drone.

1.2 Motivation

As a team of computer engineers and electrical engineers, we wanted to pursue a project that challenged our shared interest in robotics, computer vision, and embedded systems. Designing an autonomous racing drone involves all three disciplines and more. We will have to use our existing experience with embedded systems to correctly interface between the hardware and sensors, but we will also be exposed to new techniques such as parallel GPU programming and real-world PID control. Furthermore, the algorithms and technologies necessary to achieve our goal of high speed autonomous drone racing are applicable to many autonomous systems as evidenced by the interest of Lockheed Martin and their Alpha Pilot Competition.

1.3 Background

To sense its environment, the drone will perform image processing onboard using a camera and an NVIDIA Jetson Nano, which is a GPU-enabled embedded computing platform. Then, after processing the image and deducing whether there is an obstacle in the flight path, the Jetson Nano must relay the necessary control information to the Pixhawk flight controller. All of the computations will be done on-board for speed and simplicity as a product. The major challenge of this project will be in developing software that detects the obstacles and ensures the flight controller enacts the necessary evasive maneuvers. However, there are many successful controller models today, so we know it is feasible. For example, Georgia Tech's own AutoRally research platform uses a model predictive controller to plan paths for off-road autonomous rally cars. The controller, called Model Predictive Path Integral, samples thousands of stochastic and achievable trajectories and averages them to obtain one reliable and optimal path [2]. Model predictive controllers are highly parallelizable and thus well suited to real-time control using the onboard GPU of the Jetson Nano. Figure 2 shows how a standard flight controller implements PID control to modify a drone's flight behavior. Our approach will combine PID control with real-time path planning to compensate for objects in the flight path.

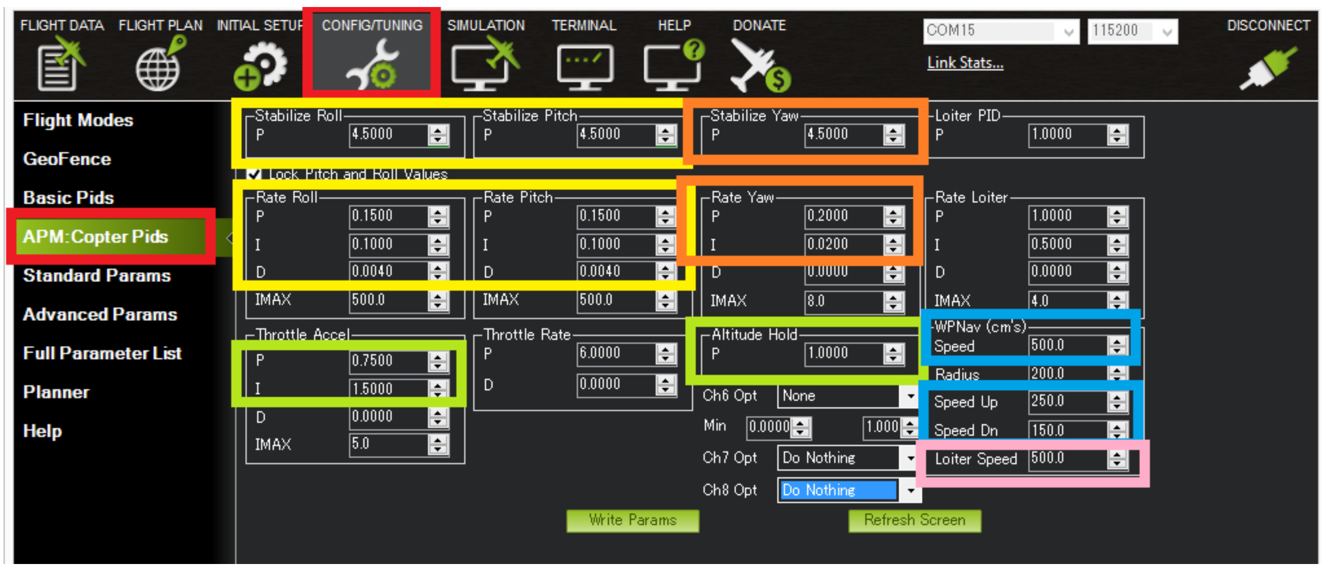


Figure 2. Quadcopter PID example

2. Project Description, Customer Requirements, and Goals

Our team will be assembling an autonomous racing drone using commercial off-the-shelf drone parts, including high-speed motors to enable the drone to fly at high speeds, and a GPU-enabled single-board computer such as an NVIDIA Jetson to interpret sensor data and control the drone. We will be writing custom software to take input from the drone's camera modules, perform computer vision in real time, compute a flight trajectory to pass through goal points and avoid obstacles, and send commands to the drone accordingly.

From a consumer standpoint, the final product should be able to autonomously navigate a racing course through goal points identified in real time while avoiding all obstacles. The drone should fly at speeds greater than 40 mph for at least five minutes, the typical length of a drone race. It should be as small and lightweight as possible. The product would be targeted towards drone enthusiasts, and the target price should be around $999, considering that the parts together cost around $479, and labor, marketing, and development costs for the product would be around $321, leaving $199 as profit.

From an engineering standpoint, the drone should be able to accomplish certain functions, which can be measured by certain objective metrics. It should be able to fly through goal points autonomously, measured by the tolerance between a goal point and the point on the actual flight path closest to it. It should be able to avoid all obstacles, measured by the number of collisions with an obstacle on a test race course. It should be able to fly greater than 40 mph when racing, measured by the average speed of the drone during a test race. It should be able to fly at least five minutes, determined by the average power consumption of the drone during a test race and the capacity of the battery.

The following QFD chart provides an overview of how the customer needs interact with the engineering requirements for this product.

	Speed	Processing power	Power consumption	Battery capacity	Manufacturing cost
High goal point accuracy	x	x			
Low obstacle collisions	x	x			
High speed	x	x	x		
Long flight time	x			x	
Small and lightweight	x			x	
Low cost	x	x		x	x

Figure 3. QFD diagram for the autonomous racing drone

The design of the product will need to take into account certain constraints. For example, for the purposes of safety and monitoring, the drone will need to include a 5.8 GHz wireless transceiver that will transmit real-time video of what the camera is seeing and allow for a remote override or “kill switch” to prevent damage to property or harm to people in the early stages of autonomous flight controller testing. Our choice of hardware and software will also impose constraints; most notably, the use of a Pixhawk flight controller will likely require our software to use ROS to interface with that controller.

3. Technical Specifications

As with any robotics project, there are two systems that generate requirements and specifications: hardware and software. Our drone must be able to fly quickly for a long period of time. These two specifications require that the drone be energy efficient and light. The outlined specifications in Section 3.1 describe a drone that meets the aforementioned requirements.

Because the drone is doing its trajectory planning and localization in real time, the software we build needs to process data and generate motion commands at a rate of 30 hertz (same as camera frame rate). Thus, the software specifications outlined in Section 3.2 describe the requirements needed to achieve real-time image processing and path planning.

3.1 Hardware Specifications

Table 1 contains the specifications for the hardware components of the drone

Item	**Requirement**
Total Weight	< 1 lb
Computation Power	> 4 cores + discrete GPU
Horizontal Speed	> 40 mph
Communication Range	200 meters
Communication Frequency	5.8 GHz
Computation Power consumption	< 5 Watts
Total Power Consumption	~ 700 Watts

3.2 Software Specifications

Table 2 contains the specifications for the software requirements necessary to navigate the course at high speed

Item	**Requirement**
Data transfer	50 Mbits/sec
Image lag	30 ms
Structure	Highly parallelizable and
Language	C++

4. Design Approach and Details

4.1 Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

Design Fulfillment Requirements

Complete three laps around a designated course without crashing. The racing drone needs to be able to complete the course in at least one battery charge life while being able to avoid crashing into other racers or obstacles, and being able to travel through checkpoints whenever necessary.

Support a minimum battery life to complete the course (three laps) in one fully charged battery. The battery life of the drone should be capable of running all the other necessary equipment such as sensors and the camera, while satisfying all other functions.

Fly through checkpoints such as hoops. If a hoop were to be missed, the drone would have to turn around and travel through it again. In the cases where, the drone misses the hoop and must travel back, the drone should be able to turn around and go through the hoop while avoiding other racers and obstacles, and then get back on track to finish the race. Adding checkpoints to the course would serve as extra functions to add to the project. The drone should be capable of distinguishing the difference between an obstacle on the course and the hoop that it must travel through.

As part of collision avoidance, the drone would be capable of avoiding crashing into other racers and obstacles. Part of designing an autonomous drone, would mean making a drone capable of having collision avoidance. Since in this case the drone will be a racing drone, it should be capable of avoiding racers and any obstacles in the course; however, it should be able to recognize hoops as checkpoints and not obstacles.

A target weight that will be aimed for the racing drone with all features added will be about 1kg. After adding all of the features such as the camera and sensor, our goal for this autonomous racing drone will be to keep it at about 1kg modeled after some racing drones from the Drone Racing League (DRL).

Concepts for Fulfillment

Completing the Course: The first step would be locating an area where could start building a "racing course". Adding simple obstacles to check for obstacle avoidance and then adding more difficulties to the course by adding turns and height changes. The drone would have to detect other racers and avoid crashing into them as part of its collision detection. The drone should also be able to complete the course in a flight time of around three minutes. Once all these other requirements are met, checkpoints can be added as extra steps that the drones must complete throughout a race.

Having the drone complete a race in one battery life is the minimum requirement. Assuming we have a battery capacity of 2200 mAh and we have on average a 40 A draw, this would give us a flight time of about 3:30 minutes. Depending on the final build and testing, these figures could change and perhaps a better battery might be needed or a reduction on sensors or cameras might be needed.

Part of flying through checkpoints such as hoops, would require some modification in the collision detection software. The drone must be able to distinguish the difference between a hoop and any other obstacle/drone and be able to travel through it without crashing. The other feature the drone must do is the in the case that the drone fails to travel through a checkpoint, it must turn around and pass through the checkpoint and then remain on course to meet the other checkpoints. All of this must be done through while avoiding obstacles and other races. Modifications in software will have to be done to the point where the drone is able to recognize the difference between a hoop that it failed to pass through and is now behind it and a hoop that is supposed to be next in line to be met.

Collision avoidance will first have to be dealt with by setting up the cameras and sensors. Once proper testing is done to ensure all of the equipment is working properly, the next steps would be to see if drone can fly itself in a straight line without colliding into any obstacles. Setting up the drone to not collide with obstacles will have to be set up in the coding section of the project. It would need to detect what are obstacles, checkpoints, and other racers. It would also need to determine when it considers itself outside of the track.

Meeting the target weight of 1kg would be met after finalizing the products that will be used on the drone. A of all hardware components and their respective weights would be made to ensure that the weight is kept at 1kg or lower. By having the weight lower, this gives us an opportunity to test the drone with its given part. Should it fall short of our expectations and provided we still have funds left over, we could invest in more hardware features to meet our expectations while maintaining our 1kg weight goal.

Concepts/Possible Solutions for Each (Sub)function and Technical Tradeoffs

The biggest trade off in completing the course would be time completing the course vs accuracy in not crashing into obstacles or other racers. Traveling as fast as possible through a course would in theory result in crossing the finish line first; however, this gives less time to perform calculations and detect fewer potential collisions. Accuracy is considered the most important factor in our case over speed, since finishing the race is more valuable than finishing first or not finishing at all.

The tradeoff when it comes to battery life is weight vs efficiency. Having a higher weight will result in less flight time as the higher weight drives up the power consumption from the battery and makes the flying harder throughout the course. The positive would be that the drones have the best sensors and cameras which would help in navigating the course and avoid obstacles. In terms of efficiency, having the lowest weight possible with sacrificing as little accuracy as possible would be the most desirable approach. Sacrificing by having less cameras and cheaper motors would reduce the weight but at the same sacrifice accuracy. Determining the best ratio between accuracy and weight will be a pivotal factor in completing the course.

The tradeoffs that deals with the ability of the drone being able to fly through the hoops comes down to weight. Having a higher weight would mean having the ability to have stronger collision detection and the ability to not miss the hoops in the first place; however, in the cases where the hoops are missed, having the extra weight on the drone would make it harder to travel back around to meet the checkpoint. Having a lower weight would allow the drone to be able to fly back faster and reach the hoops quicker in the cases that they are missed. Also, by having a lower weight, the drone would have a much easier time maintaining a stable path to travel through the hoops and not miss them.

The tradeoffs dealing with collision avoidance involves the weight vs efficiency. Having the higher weight would mean having better sensors and more cameras which improves the collision detection and would improve accuracy of the course. The issues come down to sacrificing battery power and maneuverability throughout the course to meet the necessary time limit.

Weight requirement: The tradeoffs with dealing with meeting the weight requirements will come down accuracy vs speed. Having a lower weight will mean having a lighter drone capable of having a solid battery life capable of traveling through the course faster and meeting checkpoints along the way. On the other hand, if the weight of the drone exceeds the target weight, then we can assume that it'll have an easier time detecting obstacles and other racers thus, improving accuracy at the cost of a shorter battery life and the case where the drone doesn't finish the race or takes longer than expected.

Integrated Concepts

We decided to go with the Jetson Nano by NVIDIA over the Raspberry Pi both of which are small computers that are responsible for running our cameras and sensors. The Jetson allows for quicker and more computations to be performed which the Raspberry Pi falls short. For example, the Raspberry Pi 3 only offers 6.2 gigaflops while the Jetson provides us with 472 gigaflops. Gigaflops are a measure of the performance of the computer's floating point unit.

We considered purchasing a store-bought drone to use for our project ;however, the main concern was that with the added weight of the Jetson, the store-bought drone would not have enough power to maintain optimal flying capabilities. The store-bought drone also lacks the power to maintain the cameras and sensors. In place of this, we have considered going with a custom-built drone that will have enough power to support the Jetson along with the cameras and sensors.

Design Factors

Global: Global factors do play a role in the design of the autonomous racing drone as it affects the current market. Since we are going to purchase a drone from the available market, global factors will play a role as it can limit or widen our choices.

Cultural: There is no cultural factor that affects the design factor. If anything, there'd only be a cultural factor in racing drones that could affect what's available in the current market.

Economic: Economic factors do play a role in the design factor for the autonomous racing drone. We only have a limited budget to buy the drone and its parts, so localizing the most effective part for the price will be the challenging part.

Environmental: Environmental doesn't play a role in the design aspect the autonomous racing drone. The drone would only need to be charged to keep flying. Solar rechargeable drones would be a better environmental solution ; however, it would be more expensive and out of the project budget.

Sustainability: Sustainability doesn't play a role in the design aspect of the autonomous racing drone. The drone would only need to be charged to keep flying. Solar rechargeable drones would be a better sustainable solution ; however, it would be more expensive.

Manufacturability: Making a drone would be difficult. Having to buy one off of the market would probably be the safest choice. On the market currently, there is a wide variety of different drones for sales, therefore, manufacturability wouldn't be an issue as a design factor.

Ethical: There are no ethical issues with designing an autonomous racing drone.

Health and Safety: We have to abide by a set of codes and regulations that have been placed on flying drones for the safety of the general public. For example, we can't fly the drone near an airport or military installations.

Social: With an autonomous racing drone, there isn't social issues when designing the drone. As long as the drone is being raced in a closed off area away from the public, there are no issues.

Political: Political factors do not affect our design factors when deciding on our drone and the other hardware components we are going to use.

Hardware/Software Interaction and Trade-offs

i. General flow of information
    i. Camera array (real sense sensor) captures images at 30 FPS or greater
    ii. Jetson will analyze frames using an iterative point algorithm to locate goal locations and calculate optimal trajectory
    iii. Jetson will send processed trajectory information to Pixhawk via robot operating system (ROS) nodes
    iv. Pixhawk will be responsible for motor outputs via ESCs
    v. Feed forward feedback can also be sent to Jetson for further processing
ii. Hardware and Software interaction will rely on two major interactions.
    i. Hardware Input - Cameras will deliver information for software processing
    ii. Hardware Output - Pixhawk will use processed data to output four independent voltage signals to motors.
iii. Some trade-offs include
    i. Including more cameras for improved input and spatial awareness will not only add weight but computational power and bandwidth required to process extra data

ii. We could choose to have two input processing loops one fast and one slow

1. fast loop would be used for object avoidance

2. slow loop would be used to develop trajectory to next hoop

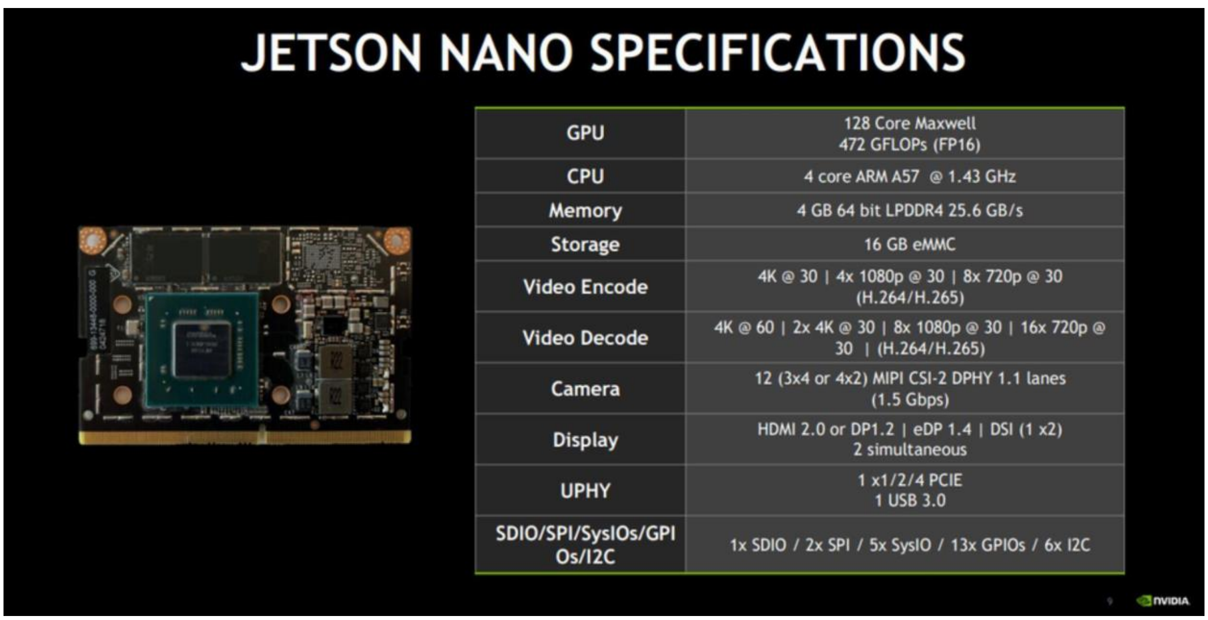
iii. If one loop is preferred all input information will flow fast but only differences will be processed to reduce computational power required

4.2 Preliminary Concept Selection and Justification

One of the main choices we had to make was whether we should use a store-bought drone or a disassembled drone kit. There are advantages and disadvantages to both, but since we will be attaching modules and an onboard computer, it makes sense to have a minimal frame to which we can add hardware. While a store-bought drone offers a readily assembled drone with well-established hardware, it comes at high cost with very low repair ability. We anticipate failed flights at the start of our testing, and we must be able to quickly swap damaged parts. Kits often come with multiple parts, and even replacements are always cheaper than the store-bought drone parts. Figure 4 illustrates a typical drone kit with various parts such as batteries, propellers, motors, and a camera module.

Figure 4. Typical drone kit with individual parts

Another important decision we made involved our selection for the computing engine. The widely known Raspberry Pi is often perfect for embedded systems projects due to its small size, low cost, and incredible functionality. However, in order to detect objects and calculate the desired path at high speed, we need a dedicated GPU to support the massive amount of parallel computations. While the latest Raspberry Pi 4 is fitted with an SoC containing a 500 MHz Videocore VI 3D Unit, it simply lacks the processing capability we require [5]. We found the NVIDIA Jetson Nano with a dedicated GPU, and it was even smaller than the Raspberry Pi. As Figure 5 shows, the Jetson Nano is properly equipped with 128 GPU cores to handle the parallel computing workload of our autonomous system. It costs more than the Raspberry Pi, but the size and performance of the Jetson Nano are more suitable for our project.



GPU	128 Core Maxwell 472 GFLOPs (FP16)
CPU	4 core ARM A57 @ 1.43 GHz
Memory	4 GB 64 bit LPDDR4 25.6 GB/s
Storage	16 GB eMMC
Video Encode	4K @ 30 | 4x 1080p @ 30 | 8x 720p @ 30 (H.264/H.265)
Video Decode	4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 16x 720p @ 30 | (H.264/H.265)
Camera	12 (3x4 or 4x2) MIPI CSI-2 DPHY 1.1 lanes (1.5 Gbps)
Display	HDMI 2.0 or DP1.2 | eDP 1.4 | DSI (1 x2) 2 simultaneous
UPHY	1 x1/2/4 PCIE 1 USB 3.0
SDIO/SPI/SysIOs/GPIOs/I2C	1x SDIO / 2x SPI / 5x SysIO / 13x GPIOs / 6x I2C

Figure 5. Jetson Nano Specifications

Given that autonomous drone racing has been explored with many cases of success, we don't expect many major roadblocks; however, our reliance on visual telemetry may be an issue. If we find that visual telemetry cannot accurately and appropriately detect obstacles on its own, we may have to consider investing in LIDAR technology. We are refraining from using LIDAR unless absolutely necessary simply due to cost. LIDAR is undoubtedly more accurate than just using a camera for detecting objects, and we have seen the company, Terabee, demonstrate drone collision avoidance using their own time-of-flight LIDAR sensors [7]. However, a camera module is much more affordable, and we should be able to get results comparable to LIDAR using image processing techniques.

4.3 Engineering Analyses and Experiment

Prototype testing will be done in conditions identical to our intended demonstration. With hoops and flags scattered throughout a track, the drone will be tested to see if it can successfully reach the end goal while avoiding collisions. Early prototype testing will involve analyzing power use, flight time, and the effectiveness of the collision avoidance systems. At this stage, the drone will be considered a success if it can fly for the entire duration of the course, avoid obstacles, and make it to the target location. A flight time of around five minutes is to be expected with a standard drone race, so we will have to keep this in mind as we attach various power-consuming modules. We must consider power drain from the motors and hardware as well as energy storage from the batteries. Weight can have an enormous impact on the power draw and efficiency of a drone, so we will have to experiment with the tradeoffs of increasing the battery size. A larger battery holds more charge, but it also weighs more, causing the motors to work harder and consume more power. Initially, we would like to focus on avoiding objects and reaching the final location, but getting there fast is just as important.

Prototype testing in the later stages will build on our early analyses and will center on fine tuning control parameters to allow high speed flight. At this second stage of prototype testing, the drone will be considered a success if it can meet the same criteria at higher speeds of 30 to 40 miles per hour. Power considerations will again have to be made since the power draw from the motors will increase at higher speeds. We expect to experience challenges in avoiding obstacles as we push the drone's speed, but we should be able to tweak the drone's path planning and control parameters to handle abrupt, immediate turns as we run more trials. Unfortunately, since we have not gotten ahold of the hardware yet, we have not had the opportunity to perform any experiments. However, we have been able to conduct a cost analysis since we already identified the appropriate materials for the autonomous drone.

4.4 Codes and Standards

1. FAA regulations will need to be followed during testing. Since the drone will weigh less than 55lbs we can operate as recreational flyers as long as we abide by the FAA guidelines. These guidelines should be easy to follow with our design process [12].
    a. Register drone and affix registration number to it
    b. Fly below 400 feet and outside of controlled airspace
    c. Never fly over crowds or vehicles
    d. Be aware of TFR
2. The Fast RTPS protocol allows for communication between ROS and the Pixhawk natively [11]. Since the Pixhawk recognizes this protocol it will allow us to:
    a. Communicate natively without the need for translation layer such as MAVROS
    b. Enable integration of our Jetson module for Obstacle Avoidance and Autonomous piloting
    c. The use of two API layers one for usability and one for the finer inner workings of RTPS
3. The widely used Universal Serial Bus standard will be used for most of the data flow on the drone; the use of USB should be achievable between all major components.

a. The connection between Camera and Jetson will be USB 3 with a data rate of 5Gb/s which will be more than sufficient.

b. Connection between Jetson and Pixhawk will most likely have to be USB to UART

    i. UART is not a communication protocol but more of a physical circuit so a USB connection throughout is achievable

c. The USB standard can also be used for ground station communication via the SiK Telemetry Radio [13]

    i. This will be good to use for testing purposes during development to track telemetry data, but should not be needed for the final product.

5. Project Demonstration

We will record a video which will demonstrate our racing drone. There will be multiple obstacles, such as hoops and flags, set at multiple heights which the drone must attempt to avoid or pass through depending on the obstacle. Figure 6 below visually represents the 2019 qualifying course

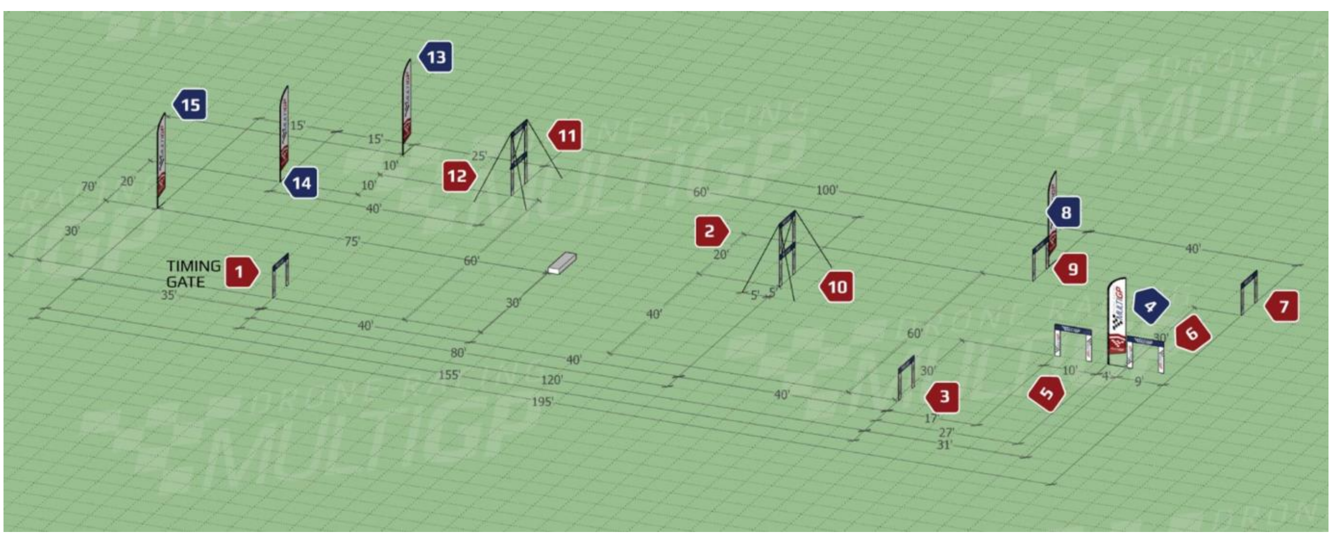


Figure 6. 2019 MultiGP Qualifying Course Layout

the project test course could be modeled after this. Additionally Figure 7 supplements the course layout with a proper trajectory for clarity. Ideally the video will show the drone attempting to make it through

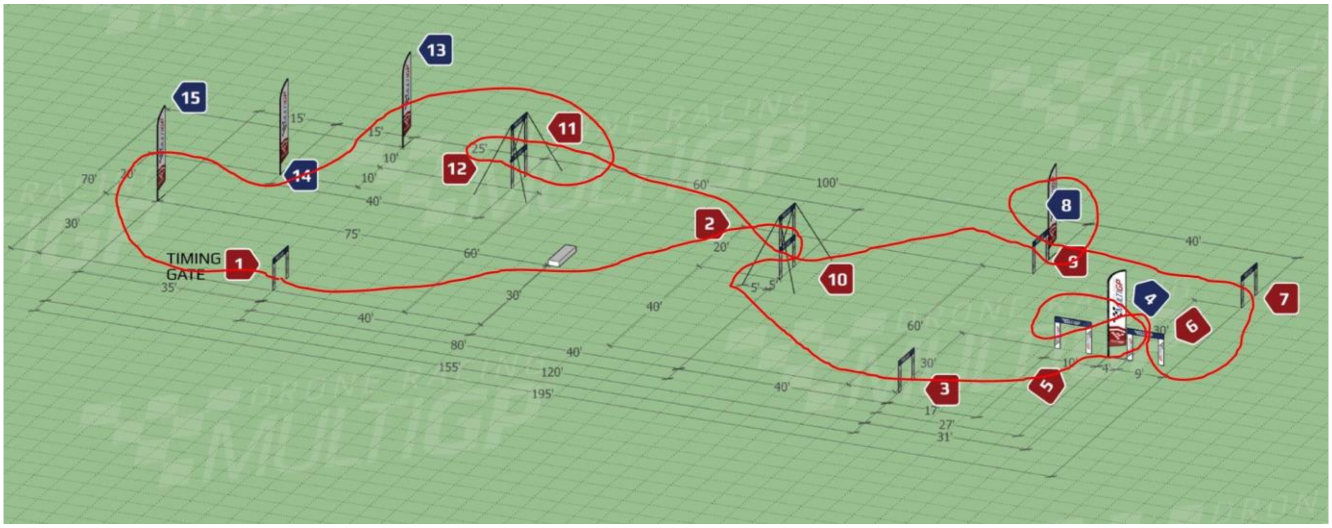


Figure 7. MultiGP Course with Proper Trajectory

the course at a high speed autonomously. Additionally, there will be a real time mapped path of the drones course, to provide real-time visualization of the simulation.

6. Schedule, Tasks, and Milestones

The AutoQuads team has decided to break the project into 5 building blocks; Hardware, Computer Vision, Autonomy, Budget/Procurement, and Final Testing. Over the course of the next 15 weeks, the team will be designing and developing an autonomous race quad. In Appendix A the provided GANTT chart illustrates the proposed tasks, owners, and deadlines. Phase 1 and 2 of our project will allow us to work on Hardware, Computer Vision, and Autonomy simultaneously to ensure all parts work independently before final assembly. After finalizing data networking between all components in week 6, we hope to start phase 3 in March. This will allow us to focus the following 3-4 weeks on Path Planning and failsafes. Phase 4 will begin in April and provide us with 3 weeks of final testing. If all major milestones are reached as stated in the GANTT chart we should complete the project a week before the Design Expo.

7. Marketing and Cost Analysis

7.1 Marketing Analysis

The target market would mostly be made up of people who are interested in the use of racing drones, which in the future could lead to a wide array of uses but for now would mostly be utilized for entertainment purposes. There are other autonomous drones currently on the market, such as the Skydio 2, which is currently being sold for $999, but is not a racing drone like the drone for this senior design project. [1]

7.2 Cost Analysis

The total development cost in order to make a prototype autonomous racing drone would be about $1157.00. The majority of this cost comes from the price of the actual drone. Since a lot of the invention is dependent on the software and algorithms, there are not a lot of materials that are necessary in order to create this product. The price breakdown is represented by the following table.

Table 3. Equipment Costs

Product Description	**Quantity**	**Unit Price ($)**	**Total Price ($)**
Drone	2	280.00	560.00
NVIDIA Jetson	3	119.00	357.00
Camera	2	80.00	160.00
Batteries	4	20.00	80.00
Obstacles	15	5.00	75.00
Total Cost			1232.00

Assuming that we are paying laborers $45.00 per hour, development costs were able to be determined and are shown in the following table. The highest number of labor hours are designated to coding for the drone. The determined development costs to design the drone summed out to be $21,531.00.

Table 4. Development Costs

Project Component	**Labor Hours**	**Labor Cost ($)**	**Part Cost ($)**	**Total Component Costs ($)**
Autonomous Drone				
Building	80	3600	1157	4199
Simulation/Testing	40	1800	75	1800
Racing & Collision Detection				
Coding Algorithms	100	4500	-	4500
Debugging	50	2250	-	2250
Testing	50	2250	-	2250
Demo Preparation	20	900	-	900
Meetings	150	6750	-	6750
Total Labor Costs	-	19800	-	-
Total Part Costs	-	-	1232	-
Total	-	-	-	21531

The total development costs for the autonomous racing drone is $50,590 which is shown in the table below. We utilized a 35% fringe benefit of the total labor and 130% overhead of materials and labor.

Table 5. Total Development Costs

Parts	$1,232
Labor	$19,800
Fringe Benefits	$6,930
Subtotal	$26,830
Overhead	$23,760
Total	$51,323

In order to determine the selling price and profit per unit, it will be based on a production of 5,000 units over a period of 5 years at $999.00 per unit forming about a 52% margin between cost per unit and selling price and matching the price of the other autonomous drone that is on the market. In bulk, the Jetson will be 16.67% off. Advertising is budgeting at 5 percent of the cost per unit. Therefore, the expected revenue from this product is estimated to be $995,000, which is a profit of $199 per unit. This information is summarized in the following table.

Table 6. Selling Price and Profit per Unit (5000 units over 5 years)

Parts Cost	$479
Assembly Labor	$20
Testing Labor	$20
Total Labor	$40
Fringe Benefits	$14
Overhead	$129
Advertising	$48
Average Development Costs	$50
Selling Price	$999
Profit	$199

8. Current Status

Currently, we are working on the development of the project scope, specifically working on the project proposal. In this proposal, we have outlined the components and budget needed, and once approved, we will be able to start designing and development of the drone.

Drone and Hardware	**Percentage Completed**
List of potential required parts	100%
Research required parts	75%
Build/Assemble	0%
Flight Test	0%
Determine flight parameters (flight time, max lift)	10%
Data networking	0%
Final Testing	0%
Computer Vision	
Test depth camera vs..regular	0%
Object Detection	0%
Object Classification	0%
Object Tracking	0%
Autonomy	
control system (controller PID / State)	0%
Parameter Tuning	0%
Failsafes	0%
Path Planning (MPC or MPPI)	0%
Budget/Procurement	
Distribute Funds (quad/replacement parts/sensors)	10%
Determine Procurement Deadlines	15%
Track Inventory	0%
Test / System	
Test overall build and system	0%

9. Leadership Roles

Rishov, Webmaster. In charge of creating a website to showcase the project, as well as handling the videography and mapping of the drone's path.

Max, Communications Lead. In charge of all contact with the supervisor and the professor to ensure the completion of tasks.

Eddie, Hardware Lead. In charge of the physical design of the drone, as well as deciding all additional components to be added for functionality.

Nye, Software Lead. In charge of the primary coding and algorithms for the drone to ensure high speed and accurate movements.

Suhani, Team Coordinator. In charge of managing meetings, deadlines, and the scope of the project.

Dave, Design Lead. In charge of all schematics of the drone and planning the integration of all components.

Michael, Research Expert. In charge of researching similar projects and determining what components are needed.

10. References

[1] Skydio, “Pricing Page-Skydio” *Skydio, 2019.* [Online]. Available: https://www.skydio.com. [Accessed: Nov. 17, 2019].

[2] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots and E. A. Theodorou, "Information Theoretic MPC for Model-Based Reinforcement Learning," [Online]. Available: https://www.cc.gatech.edu/~bboots3/files/InformationTheoreticMPC.pdf. [Accessed 18 November 2019].

[3] ArduPilot, "Tuning Common Things To Change," [Online]. Available: http://ardupilot.org/copter/_images/Tuning_CommonThingsToChange.png. [Accessed 18 November 2019].

[4] Drone Kit, “Amazon Drone Listing” *Amazon, 2019.* [Online]. Available: https://www.amazon.com/FEICHAO-4-Axis-Flight-Controller-Integrate/dp/B07DQJTCXK/ref=sr_1_8?keywords=250mm+drone+kit+pixhawk&qid=1574088906&sr=8-**8**. [Accessed 18 November 2019].

[5] Raspberry Pi Foundation, "BCM2711," [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/README.md. [Accessed 18 November 2019].

[6] Jetson Nano Specifications, “Nvidia Developer site” *Nvidia, 2019.* [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit[Accessed: Nov. 18, 2019].

[7] Terabee, "DRONE COLLISION AVOIDANCE FOR BEGINNERS," Terabee, [Online]. Available: https://www.terabee.com/drone-collision-avoidance-beginners/. [Accessed 29 September 2019].

[8] MultiGP Qualifying Course, “MultiGP 2019 site” *MultiGP, 2019.* [Online]. Available: https://www.multigp.com/2019-multigp-series-qualifying-track-released/ [Accessed 18 November 2019].

[9] “Drones Reporting for Work at 45 MPH,” *Draper*, 03-Aug-2017. [Online]. Available: https://www.draper.com/news-releases/drones-reporting-work-45-mph. [Accessed: Sep 17, 2019].

[10] A. Kamenev, “Skypad TBS Discovery Setup,” *GitHub.com*, 22-Mar-2018. [Online]. Available: https://github.com/NVIDIA-AI-IOT/redtail/wiki/Skypad-TBS-Discovery-Setup

[11] T. Nguyen, “Integration of ArduPilot and VIO tracking camera for GPS-less localization and navigation,” *ArduPilot*, 15-May-2019. [Online]. Available: https://discuss.ardupilot.org/t/gsoc-2019-integration-of-ardupilot-and-vio-tracking-camera-for-gps-less-localization-and-navigation/42394.

[12] L. Dorr, “Small Unmanned Aircraft Regulations,” *Small Unmanned Aircraft Regulations*. FAA, 2018. [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615

[13] “SiK Telemetry Radio,” *SiK Telemetry Radio - Copter documentation*. [Online]. Available: http://ardupilot.org/copter/docs/common-sik-telemetry-radio.html. [Accessed: Nov. 17, 2019].

Appendix A GANTT

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Days)	PCT OF TASK COMPLETE
1	**Quadcopter and Hardware**					
1.1	List of potential required parts	Max R	10/18/19	10/21/19	3	100%
1.2	Research required parts	Eddie S	1/6/20	1/9/20	3	75%
1.3	Build/Assemble	Eddie S	1/14/20	1/20/20	6	0%
1.4	Flight Test	Max R	1/20/20	1/29/20	9	0%
1.5	Determine flight parameters (flight time, max lift)	Max R	1/27/20	1/31/20	4	0%
1.6	Data networking	Michael B	1/13/20	1/24/20	11	0%
1.7	Final Testing	Michael B	3/8/20	3/27/20	19	
2	**Computer Vision**					
2.1	Test depth camera vs.regular	Rishov S	1/20/20	1/30/20	10	0%
2.2	Object Detection	Dave P	1/27/20	2/14/20	17	0%
2.3	Object Classification	Suhani Jain	1/28/20	2/15/20	17	0%
2.4	Object Tracking	Suhani Jain	1/29/20	2/16/20	17	0%
3	**Autonomy**					
3.1	control system (controller PID / State)	Eddie S	1/20/20	1/29/20	9	0%
3.2	Parameter Tuning	Eddie S	1/13/20	1/18/20	5	0%
3.3	Failsafes	Eddie S	2/13/20	3/9/20	26	0%
3.4	Path Planning (MPC or MPPI)	Max R	2/11/20	3/9/20	28	0%
					0	0%
					0	0%
4	**Budget/Procurement**					
4.1	Distribute Funds (quad/replacement parts/sensors)	Rishov S	1/6/20	3/27/20	81	0%
4.2	Determine Procurement Deadlines	Rishov S	1/6/20	1/11/20	5	0%
4.3	Track Inventory	Dave P	1/6/20	2/15/20	39	0%
5	**Test / System**					
5.1	Test overall build and system	Nyaire N	2/20/20	4/15/20	55	0%