Title:	Computer Controlled Auto-Follow Drone
Student:	Andrew Festa
Faculty sponsor:	Jeremy Brown
Academic term:	2195 - Spring 2020

---

1 Introduction

Most small, DIY drones are configured to either be controlled from a remote control or completely from an on-drone system which fully acts as the flight controller and auto-pilot. The former are generally limited by the number of control channels on the transmitter, and the pilot must have some method for observing the environment in which the drone is operating. The latter configuration is limited by the power and resources provided by the on-board computer. The most important resources, computational capability and power, are typically heavy and expensive, making resource management and system addons a critical design decision. Can a camera be added to the board without overly taxing the already limited processing power? What about a tracking algorithm? Many approaches for these problems are becoming more and more resource demanding, such as training of neural networks or running reinforcement learning agents, and so a different configuration might be better suited to building an extensible and scalable flight control system.

The ultimate goal of this independent study is two-fold: to explore machine learning and control systems on a resource-constrained system and to offer practical experience into the design decision required for building such a system. An additional benefit is the end-creation of a computer controlled drone which can be readily extended for control by other systems which may interface with other types of systems, such as BCIs or VR headsets.

The closest course offering to this project is CSCI-632, Mobile Robot Programming, traditionally taught by Zack Butler. However, this project differs in that flight offers a particular set of challenges compared to ground-based navigation. Where ground-based navigation is effectively operating in a 2D space, flight systems operate in a 3D space which is also a transition that is a challenge for humans learning to fly. Beyond that, resource management is vastly more important when controlling a system in the air. When on the ground, if power runs out. In the case of a flying system, this option is tantamount to crashing. The final difference is that this project intends to take a more hands-on approach to the design and building of the drone itself, resulting in a deeper understanding of the necessary components going into a flight controller.

2 Planned Work

During the course of this project, I intend to explore creating an arduino-based drone which uses the board merely for sensory input and for sending and receiving information to and from an off-drone computer, referred to as the control computer. The control computer will be responsible for the processing of this information and deciding on the control logic for flying the drone. This offloads a lot of the computational tasks from the resource-constrained arduino and reduces the power requirements for operating the drone. The operation of the drone will come in two modes. In `Mode 1`, the user will have full, manual control of the drone whereas in `Mode 2`, the tracking system will have control and the drone will fly fully autonomously.

This project arguable reaches into many clusters of computer science. However, the primary two are the domains of networking and intelligent systems. The networking aspect comes in the form of establishing a secure and reliable communication channel between the drone and the control computer. This channel is responsible not only for allowing the drone to send inputs and receive the control signals, but it will also have to send video and sensor data to the control computer in order for it to act on those inputs to generate the control systems. The drone must also be able to handle the case of a dropped connection in a reasonable manner. The intelligent systems aspect comes from implementation of `Mode`

2. This requires the control computer to analyze the video from the drone in order to track an object or person as well as adding the capability for the control computer to use input from an AI agent.

2.1 Project Phases

At a high-level, the project can be conceptualized into three phases. The first phase will primarily consist of building the hardware and setting up the low-level systems such that they are capable of using inputs, such as up, down, left, and right, and flying the drone accordingly. The second phase is where the control computer is designed and implemented and is effectively implementing what was previously referred to as `Mode 1` operation. This will allow for a human to use a computer to fly the drone manually. The third and final phase is effectively where `Mode 2` operation is added to the operational capabilities of the drone. This will allow for a computer-based agent to fly the drone autonomously.

2.2 Schedule

The table below captures the expected tasks to be completed during the course of this project and how long each portion is expected to take. It is expected that this independent study will require at least 200 hours of work and will likely not exceed 180 hours of work. As this semester as migrated to online delivery, no in-person meetings will occur. Rather, at least one hour a week will be dedicated to meeting with the faculty advisor to review recent progress.

Task and Anticipated Length	**Technical Goals**
Literature review 20 - 25 hours	Review approaches for flight control systems Analyze different computer vision based tracking algorithms Layout concrete steps for the remainder of the project
Architect the system 10 - 15	Layout a software plan for sending and receiving information from the control computer Design the structure for manually taking input from the user Design the structure for allowing an automated system to interact with the drone
Building the drone 35 - 40	Decide on the necessary hardware components Design the desired functionality of the flight controller Implement the flight controller
Implement the control interface 45 - 50 hours	Implement Mode 1 operation Create simple web-interface for visualizing drone and user inputs
Implement automated control 25 - 30 hours	Implement Mode 2 operation Hook into interface to display decisions made by control computer
Build database of recorded flights 10 - 15 hours	Design database for tracking flights for future learning Add ability to replay flights through web-interface Extensively play with drone
Finalize report and presentation 20 - 25 hours	Edit and revise final report Create presentation and demonstration content Finalize all remaining minutiae

Table 1: Independent Study Schedule

3 Deliverables

The independent study is expected to produce a physical product, in the form of a drone, as well as a software component, in the form of the control computer. Additionally, a final report will be provided which details the steps necessary for recreating the work done during the course of the project.

1. Quadcopter drone capable of being controlled from a computer interface
2. Guides specifically detailing the hardware construction and software configuration of the drone
3. Github repository containing all scripts and code created during the course of the project
4. Video demonstration of the drone in operation
5. Final report
6. Optionally, a web-interface for visualizing the inputs from the drone and tracking the flight decisions made by the control computer

4 Evaluation

The main consideration for evaluation of success will be simply: does it work and how does it operate in various environments? It is desired that it is able to operate within a single network without crashing and provide a non-painful interface for interacting with the drone. Beyond these considerations, the project will also be evaluated the basis of logical design and code architecture. As one of the benefits of this project is meant to be a tangible and extensible product, the code base should follow the principle of good code design in the sense that it is readable, maintainable, and scalable. The grade will take these considerations into account and will be assigned as a standard letter grade.