Single Operator Multi-Drone Control Platform Development

ME-94 Project by Jeremy Kanovsky
Research Adviser Jason Rife, Ph.D.

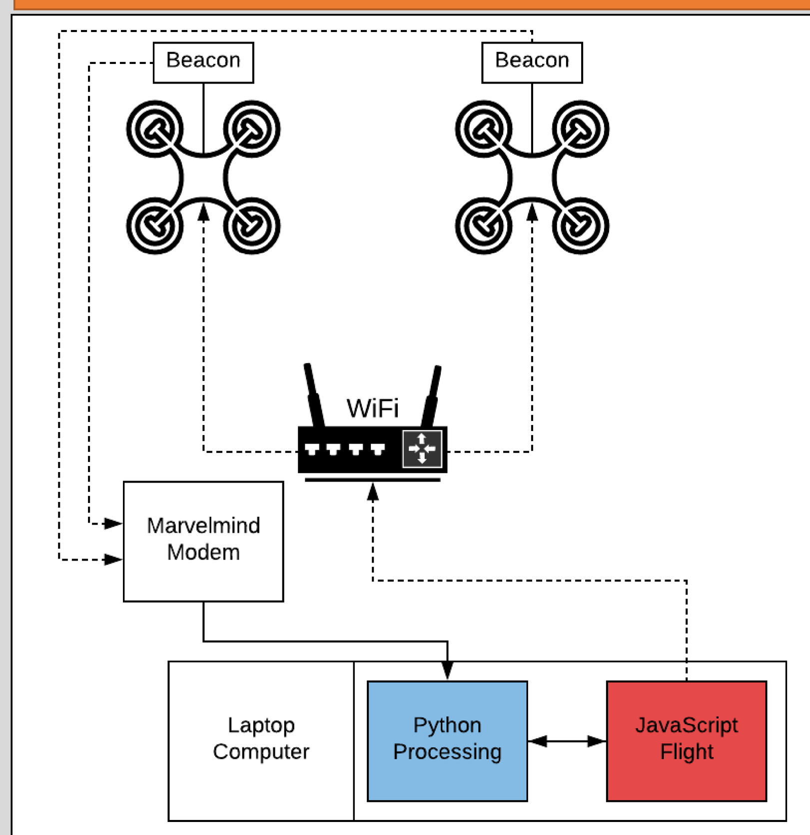
ASAR Lab

Tufts University School of Engineering

Introduction

The advent of commercial drones and unmanned aerial vehicles becoming cheaper and easier to operate has caused a boom in the number of feasible applications. Uses like search and rescue, disaster relief and aid, geographic mapping and surveyance are well within the realm of possibly, yet currently untapped in their potential. The problem, however, is that relatively few people are trained to properly operate and fly a drone safely, especially over inhabited areas. Current technology is limited to a single operator with direct control over a single aerial vehicle with regards to flight path. At the present there is not a platform by which a single operator may control several drones at a time and maintain a looser control on all of them while at the same time still being able to selectively direct each a particular aircraft in real time. The goal of this research project was to develop a software platform by which a single operator from a single source controller (i.e. from a single laptop or joystick) would be able to direct the actions of multiple drones simultaneously. This project also included development of a semi-autonomous flight controller whereby the operator may set a destination marker to which the drone would fly before prompting the user for more guidance.

Platform Architecture



Results

The result of this project is an open-source modular platform for control of a Parrot AR 2.0 drone. This platform can integrate directly with the MarvelMind 3D positioning system to provide real-time feedback as to the position of the drone within ±2 cm. The code platform itself can be run on any operating system and has low computational overhead. The system developed is theoretically capable of managing a near infinite number of drones by virtue of subnet masking. The source code developed in this project is openly available at https://github.com/0xJeremy/ME94-Drone.

Challenges

Challenge: Communication with multiple drones simultaneously on different Wi-Fi networks.
Solutions Considered:

- Use a Raspberry Pi as a "slave" for its network card
- Use a second network adapter to force multiple connections
- Find a third-party library for creating drone swarms
- Use a single common Wi-Fi network and issue commands via subnet

Chosen Solution: After investigating all of the approaches above, the best solution was to use a Telnet connection to shutdown the individual drones' networks and cause them to log onto a common open network. Each drone was assigned a unique subnet mask which could be entered into the program to create a "drone swarm" object. Commands could then be issued to each drone individually or as a whole. This solution allows for a near infinite number of drones to be controlled simultaneously. This system also allowed each drone to hold its own positioning data privately without potential interference between drones in the swarm.

Challenge: Real-time data from the MarvelMind Robotics positioning system (more specifically, communication between JavaScript and Python in real-time).
Solutions Considered:

- Create a Python child process and return the positioning data
- Write data from Python to a local file, read by JavaScript
- Run JavaScript as a child process of Python
- Use a local data socket to read and write position data

Chosen Solution: While most of the solutions above work theoretically, there is a large issue with latency of communication between the two programming languages. Creating new processes has too much overhead to continually do every few milliseconds. The most elegant solution was found to be using a local data socket (creating by the JavaScript process) to write positioning data to the Python script. This also allowed most advanced forms of communication between the programs and a near zero communication delay.

Challenge: Transforming raw MarvelMind data into a meaningful coordinate system
Solutions Considered:

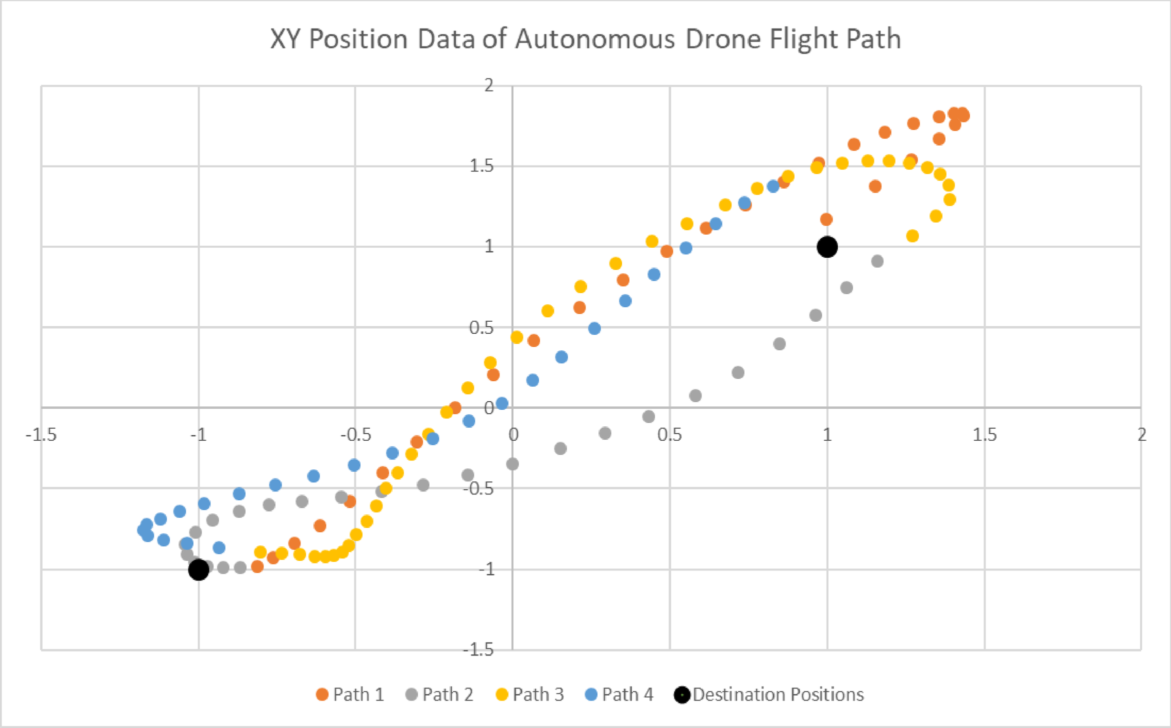
- Recalibrate the system after every test to zero the starting position
- Hardcode the positions of each beacon and use an arbitrary x-y axis
- Use a rotation matrix re-zeroed around the drone starting position

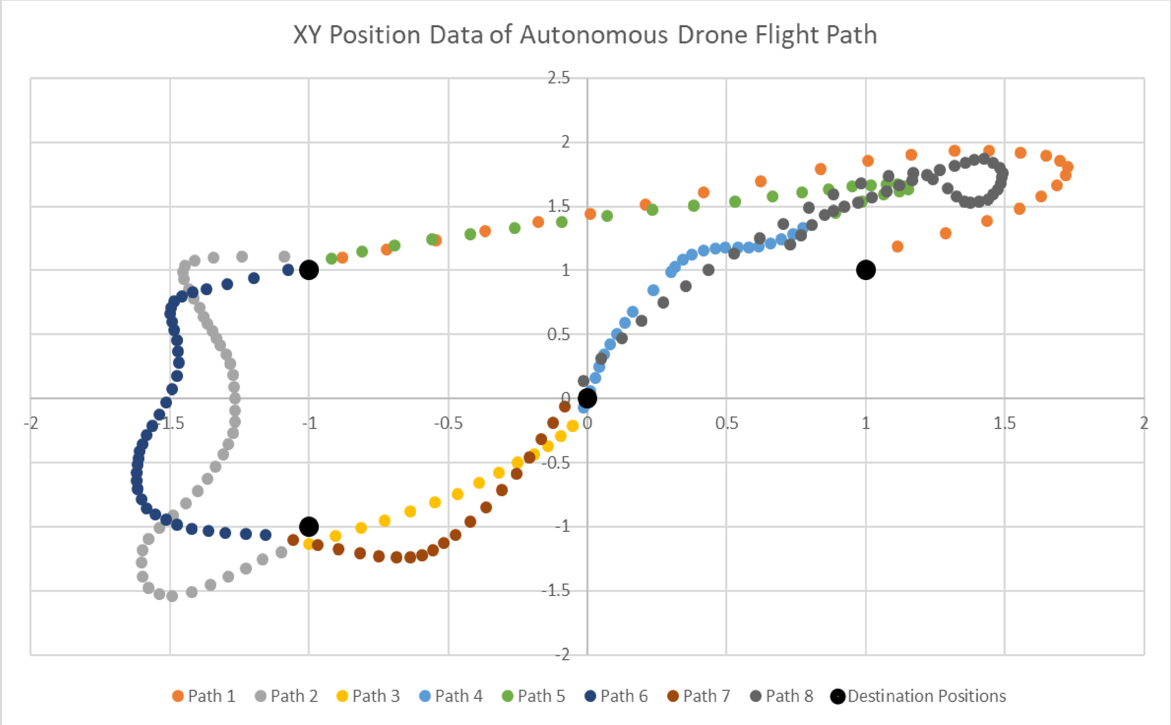
Chosen Solution: Each of the solutions above was used at some point in the design process. The first two solutions were inevitably problematic due to the inflexible nature, and the time taken to manually recalibrate the system after each test. The final solution is the most computationally intensive, but also the best. A rotation matrix allows the x-y data from the MarvelMind system to be reoriented around an arbitrary axis. This axis is the initial position of the drone at takeoff, which is automatically found during each test. This allowed the MarvelMind data to be transformed to meaningful data for the drone operator.

Conclusion

The platform developed as a result of this course is one that allows for easy control over multiple drones simultaneously and solves many of the problems associated with controlling a drone swarm from a single operation station. The modules of the system are designed such that they can be changed in and out to allow for further functionality such as a larger number of drones, satellite interfacing, and control of different types of drones. The platform also allows relatively untrained operators to take control of any drone in the swarm and give it high level tasks without needing to give individual instructions. Future work would involve expanding the system to support a further range than a Wi-Fi network will allow, removing reliance on the MarvelMind system for positioning, and integrate multiple types of drones into the same program.

Logged Positioning Data





Positioning data (as seen from above) of the path as a result of autonomous drone flight.

References

- Felixge, Node AR Drone, 2013, GitHub Repository, https://github.com/felixge/node-ar-drone
- MarvelMind Robotics, Python Library for Beacon, 2018, GitHub Repository, https://github.com/MarvelmindRobotics/marvelmind.py
- Parrot AR 2.0 Drone Power Edition, Parrot Drones SAS, https://www.parrot.com/us/drones/parrot-ardrone-20-power-edition
- MarvelMind Starter Set, MarvelMind Robotics, https://marvelmind.com/