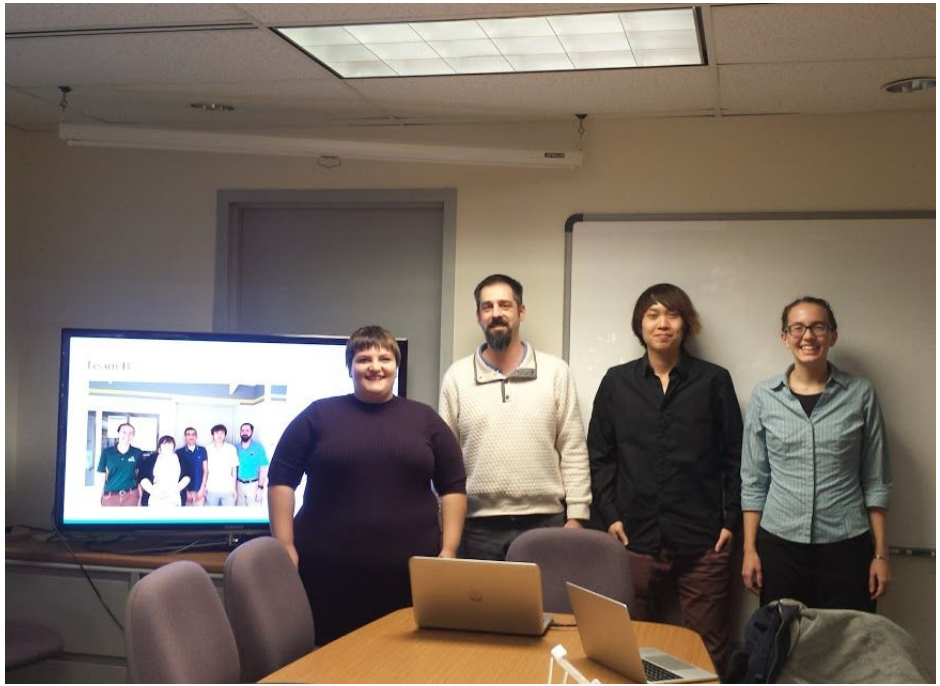
Don’t Fly Here

Senior Design II
Team B

Team B



Emma Sum - EE
Amanda Wolfe - CS
~~Feras Al Otaibi - EE~~
Huanxi Wu - CS
Ted Hood - EE

Table of Contents

Project Goals

Emma Sum

What We Designed

Autonomous unmanned aerial system (UAS) flight control system (FCS) capable of no-fly zone (NFZ) detection and avoidance

- Real-time location awareness
- Restricted airspace awareness
- Large obstacle awareness
- Automated flight path planning

What We Built

Two software solutions using one UAS development platform

- Hardware: DJI Matrice 100
- Software I: iOS application based on DJI Mobile SDK
- Software II: Android application based on DJI Mobile SDK

Design Approach: Hardware

Ted Hood

Hardware

- DJI Matrice 100
  - Developer platform
  - N1 Flight Controller
  - 3.6KG max takeoff weight
- DJI Guidance
  - 5 Sensor pods with Ultrasonic sensor and two 4mp cameras
  - On board processor capable of detecting objects within 0.2 - 20m
- Configuration
  - Battery bay on top
  - Guidance system on bottom
- Controls
  - DJI Controller
  - 10” ASUS Tablet

Images form DJI.com

Design Approach: Software I

Huanxi Wu

Path Planning

- Original Design: A* algorithm
  - Basics:

Manipulating sets of nodes(available path) OpenSet, CloseSet and PathSet. Finding the shortest path within a network infrastructure (e.g Motor way system, maze etc.)

  - Challenge & issues:

In our specific case, there is not a network, while we can build it specifically for it to cooperate with A* algorithm (by using the Point of Interest optimization - discussed during CDR). However, it’s redundant and suffer the performance.

- Alternative Approach: Mathematical approach

Mathematically analysis the problem.

Mathematical Analysis

Start Location = current drone location (x1, y1)

End Location = User Input (x2, y2)

Defining

$$d_x = x_2 - x_1$$
$$d_y = y_2 - y_1$$
$$d_r = \sqrt{d_x^2 + d_y^2}$$
$$D = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} = x_1\, y_2 - x_2\, y_1$$

gives the points of intersection as

$$x = \frac{D\, d_y \pm \operatorname{sgn}^*(d_y)\, d_x \sqrt{r^2 d_r^2 - D^2}}{d_r^2}$$
$$y = \frac{-D\, d_x \pm |d_y| \sqrt{r^2 d_r^2 - D^2}}{d_r^2},$$

where the function $\operatorname{sgn}^*(x)$ is defined as

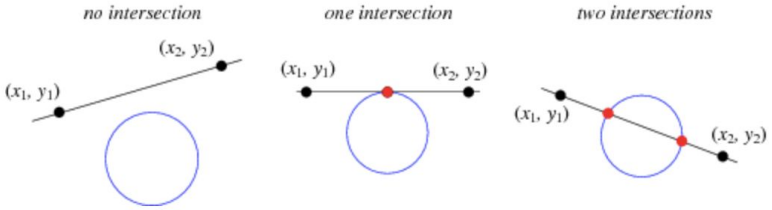
$$\operatorname{sgn}^*(x) \equiv \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{otherwise.} \end{cases}$$

The discriminant

$$\Delta \equiv r^2 d_r^2 - D^2$$

therefore determines the incidence of the line and circle, as summarized in the following table.

Δ	incidence
$\Delta < 0$	no intersection
$\Delta = 0$	tangent
$\Delta > 0$	intersection



Find intersections using root calculation for quadratic formula, which grant the entire algorithm a complexity of O(n), where n is number of the NFZ we need to analysis.

Efficiency comparison:
MA = O(n), n = number of NFZs

A* = O(log h (n)), h = heuristic

Design Cont.

With the Intersection generated for each NFZ encountered, place the waypoint along the NFZ (currently implementation add a midpoint between two intersection as the detour point) to move around the NFZ.

Platform: iOS Environment: build iOS 11.2 Any Device

Language: Objective-C, SWIFT

Software-hardware(Flight Control) integration: DJI Mobile SDK (written in both Objective-C & SWIFT)

Design Approach: Software II

Amanda Wolfe

Design Approach: Software II

- Will be discussed with Test & Analysis

Test & Analysis: Hardware

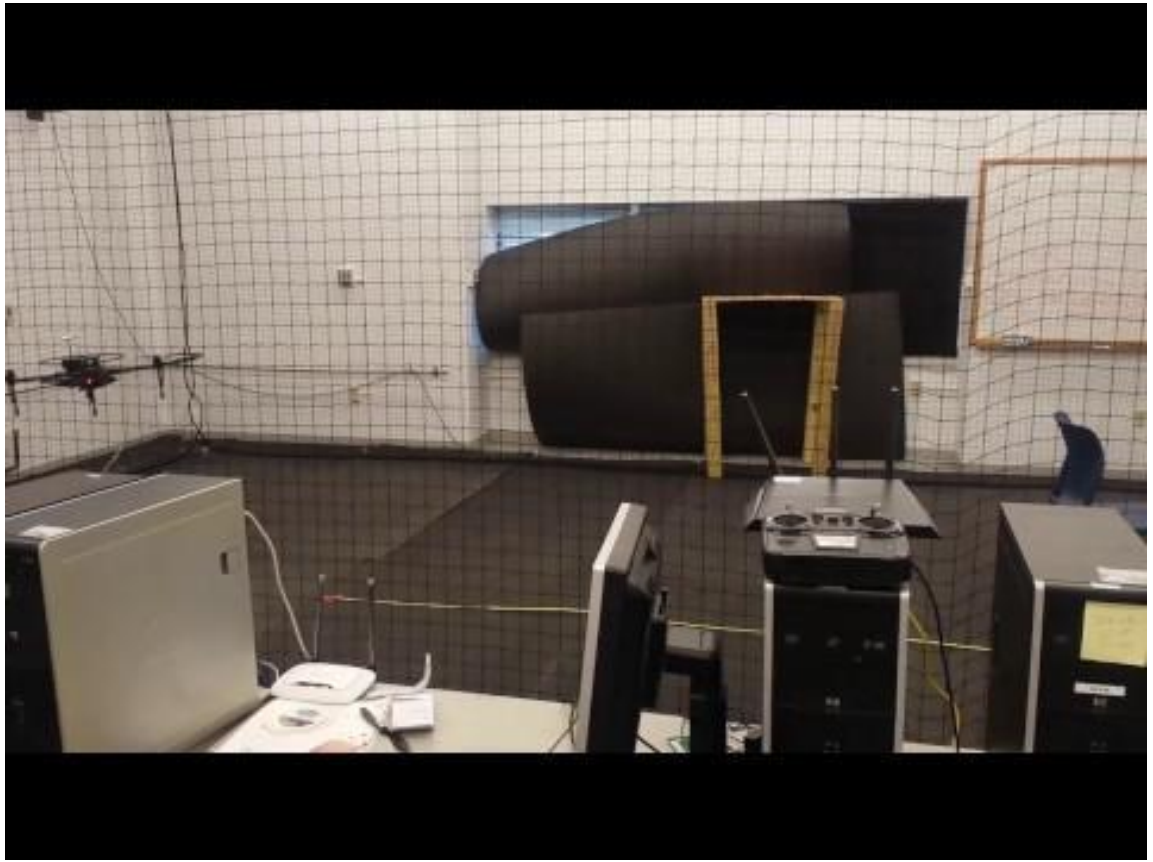
Ted Hood

Initial Flight Tests

- Flight tests
  - Inside
    - Issues with space and wall detection
    - Limited room
  - Outside
  - Soccer Field near Russ
    - Able to use space to test all our flights
- Guidance system
  - Object Detection
  - Test with Chair
    - Does not do object detection when low flight
- Hardware tests
  - All tests required for the flight of the drone passed

Initial Flight tests

During this flight, the drone was set to see objects at 10 meters. It detected all areas around the drone and only allowed pilot to control altitude

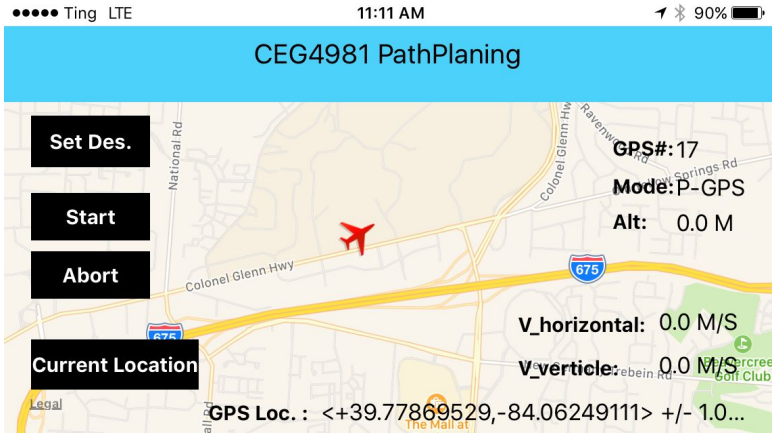


Test & Analysis: Software I

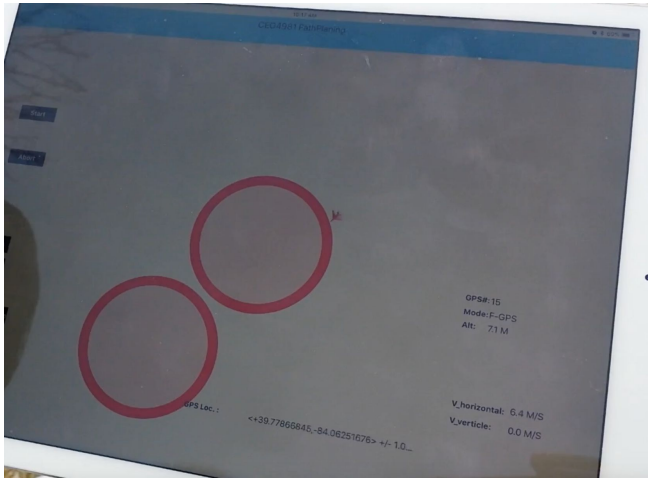
Huanxi Wu

Testing

- Pathfinding application test
  - Test on iPhone 5s - iOS 11.2 with actual flight
  - Test on iPad - iOS 11.2 with actual flight
- Application startup, load map view and current drone location(GPS coordinate)



Screenshot from iPhone 5s with info shown



Screen from iPad with NFZ shown during flight

Test & Analysis: Software II

Amanda Wolfe

Software II

- DJI SDK Demo App uploaded into Android Studio and modified to operate on Asus Zenpad
  - Demo App is compatible for Matrice 100, which meets our requirement
  - Demo App is not as “out of the box”, depending on how often DJI updates the GitHub
  - Android Studio utilizes Gradle framework to build apps
    - Updating Gradle during development will require a complete restart
    - SDK number must match across device, Gradle, Android Studio, and internal libraries
  - Demo App can only connect to the drone if the remote controller is synced to the Matrice 100
    - No obvious way to determine if the remote isn’t connected to the Matrice 100 or the app is not connected to the remote

Software II cont.

- Path Planning algorithm
  - 2m intervals, test area is not large enough for 5m and 10m intervals
  - Base graph is a square or a rectangle of nodes
  - Three paths, determine the shortest path and the path with the least obstacles
    - Determine obstacle occurrence and distance with Guidance system
    - Account for rectangle graphs, difference between two GPS points is odd
  - Difference between GPS points and interval count is determined by Haversine formula
    - Provides data validation of base graph and location of points
- Analysis
  - Output of Path Planning module was verified through output in NetBeans
    - Developed module separately to conserve unit integrity
    - Easier to evaluate results without external module interference
    - Java is compatible with Android Studio
      - Downside: not all Java syntax is appropriate with Android Studio

Summary of Results

Emma Sum

Test Matrix

- 26 tests total
  - 5 on ground
  - 4 airborne
  - 17 on ground or airborne
- 18 tests passed
  - Number of tests varied per requirement
  - 1 requirement only partially passed
  - 69% success rate
- 8 tests failed
  - Change in path planning implementation
  - Lack of time for debugging, testing

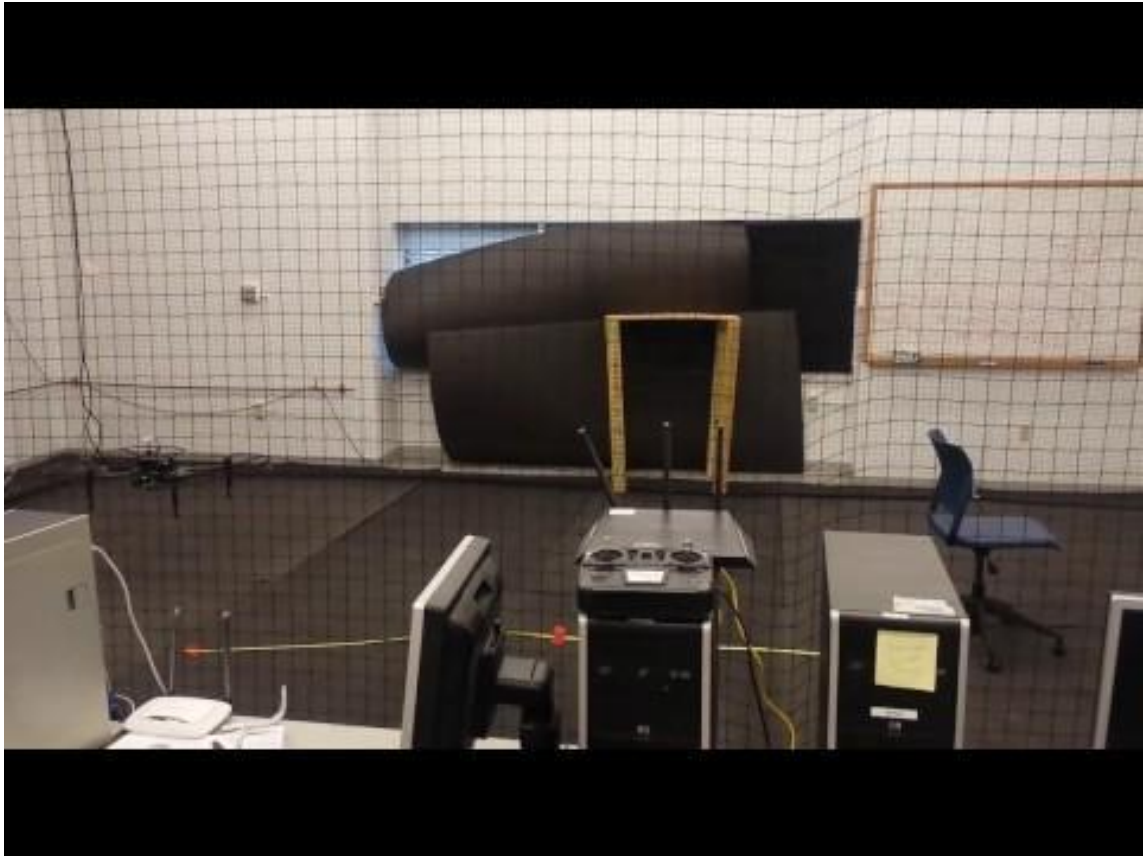
REQ	Summary	Passed Test?
REQ-010	Inclusion of UAS	1 of 1
REQ-020	Payload weight	1 of 1
REQ-030	Power consumption	1 of 1
REQ-110	Latitude trilateration	1 of 1
REQ-120	Longitude trilateration	1 of 1
REQ-210	LNFZ detection distance	0 of 2
REQ-215	LNFZ avoidance	0 of 4
REQ-220	LNFZ storage	1 of 1
REQ-310	ONFZ detection size	2 of 2
REQ-320	ONFZ detection distance	4 of 4
REQ-330	ONFZ avoidance	4 of 4
REQ-410	Path planning	0 of 1
REQ-420	Failure modes	1 of 2
REQ-510	Wireless	1 of 1

Demonstration

All Team Members

Object Detection

During this flight the drone detects the chair and stops forward movement

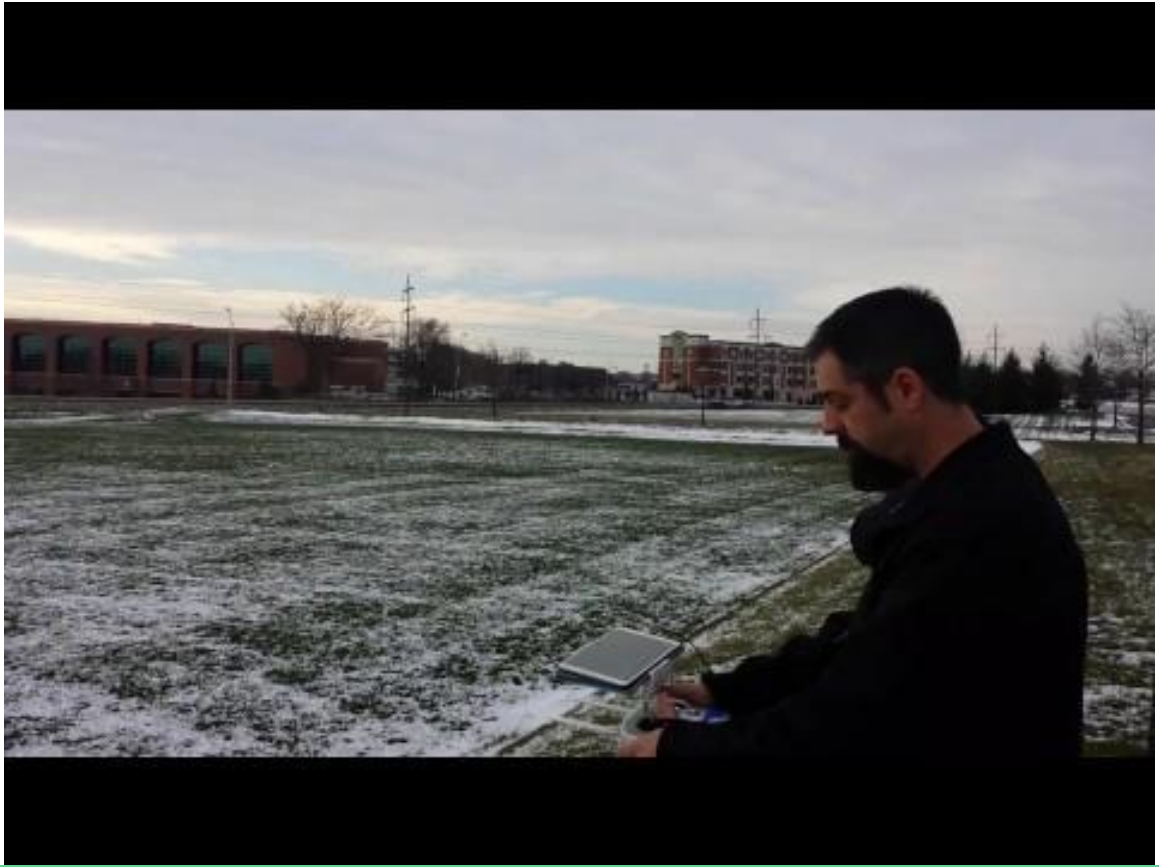


Object avoidance

During this video, the Drone detects object moving towards it and backs away without pilot control.

No Fly Zone Detection

This test demonstrates No Fly Zone detection. The drone stops forward movement even though pilot commands a forward flight path



Path around NFZ

During this flight we created a No Fly zone using our software (shown in red circles). The video shows a our software during flight of the drone avoiding the no fly zone

Conclusion

Ted Hood

Take aways

- DJI Matrice Drone/Guidance System
  - Stable flight even in high winds
  - To large for UAV lab
  - Easy assembly and configurability
- Software
  - Path planning could have been easier without A*
  - Issues with iPhone vs. Android with software
- Senior Design Process
  - Order process took too long (changed our timeline)
  - Needed about 2 or 3 weeks more testing time with Drone
  - Would have ordered drone at conclusion of Senior Design 1
- Drone Control
  - Multiple drones with multiple groups caused control issues
  - Limited control and issues with drone/controller pairing

References

1. DJI Official. (2017). Matrice 100: The quadcopter for developers - DJI. [online] Available at: https://www.dji.com/matrice100 [Accessed 13 Dec. 2017].
2. DJI Official. (2017). Guidance : The quadcopter for developers - DJI. [online] Available at: https://www.dji.com/guidance [Accessed 13 Dec. 2017].