AUTONOMOUS QUADCOPTERS

Introduction

The current state of disaster relief lacks easy, safe, and reliable ways to assess damage and to prevent disasters, such as avalanches. The introduction of quadcopters provided a platform on which to survey affected areas but not without its flaws.

There is a specific need to survey areas in winter conditions including snowpacks to be able to predict and survey avalanche conditions. This project first and foremost supports the work of Professor Jacobs in the Civil and Environmental Engineering department relating to snow and snow melt to be able to better provide data relating to these topics by collecting data in the field.

Quad-X aims to develop autonomous quads for researchers, capable of surveying a predefined area as well as or better than a professional quad pilot by ensuring at least the same accuracy of measurements and with collection of data done in at least the same amount of time. This will be done to make field research more cost effective before the Undergraduate Research Conference.

With this impact area in mind, the team is focusing on establishing new lines of communication with the quads, improving existing data collection, and improving existing surveillance abilities of the quads.

Project Goals

Computer Science

- Assist graduate level students in their research by improving:
  - Quadcopter bring up time
  - Ease of use to launch a mission
- Implement a new line of communication between quadcopter and ground station

Mechanical Engineering

- Investigate the following variables involving a quadcopter while in flight
  - Thrust
  - Commanded Motor Output
  - Battery Voltage
- Design a test stand to measure these variables

Electrical/Computer Engineering

- Design and create a wireless charging station
- Integrate this wireless charging station onto another autonomous ground vehicle

Contacts

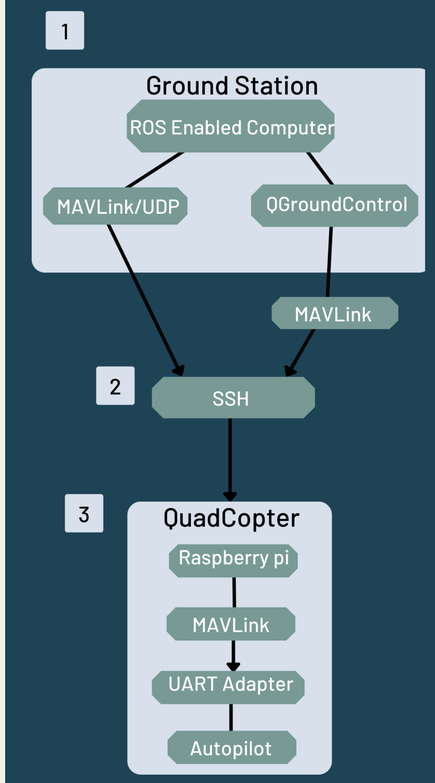
Project Contributors:

Spencer Janelle, Jonathan Snow, Nicholas Vincent, Harrison Mannering-Londres, Brendan Murphy, Stephanie Dondyk, Clay Greig, Jeb Collins, Matthew Kimball

Project Advisor:

Professor May-Win Thein

Design



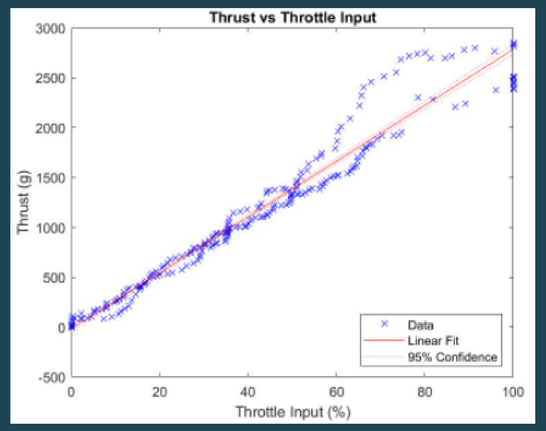
Physical Mission Communication

- Two computers are needed to run a mission.
  - one as a "Ground Station".
  - another physically on the quadcopter.
- 1. The mission Ground Station acts as hub for multiple quadcopters to connect to.
  - It runs QGroundControl and a GUI, to get the current state of the quadcopter.
- 2. The Ground Station communicates with each QuadCopter simultaneously, to send directions on how to complete a mission
- 3. The Quadcopter's computer is a raspberry pi 4.
  - It runs Robot Operating System (ROS), which is used to run Offboard scripts on the quads.

Results

Multi-quad Flight & SSH Comunication

- Opened new line of communication to quadcopters using SSH
- Used small travel router to set up LAN which the ground control computer uses to communicate with all the quads.
- Connecting and SSHing into the quads can be done more seamlessly in comparison to the previous system
- Decreased time to initialize and arm 3 quads for a mission by 95%
- Reduced the setup time of a Raspberry Pi 4 by 300%



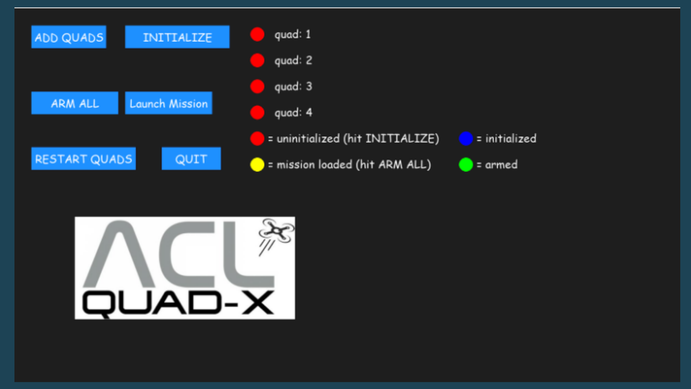
These measurements are used to determine:

- Best Components (Batteries, Propellers, ESCs)
- Flight Time at Different Takeoff Weights
- Maximum Payload Capacity

Multi-quad Flight Videos

Implementation

Quad-X Groundstation GUI



- Ground station GUI for communication with quadcopters, created using Python and the pygame library.
- The groundstation allows users to run commands on the quadcopters and check the status of each of the quads.

Groundstation GUI Buttons

- ADD QUADS: Prompts the user to enter the IP addresses of the quadcopters to be added, separating the values with a comma.
- INITIALIZE: Runs the px4 initialization command on each quadcopter.
- ARM ALL: Runs the rosservice command to arm on each quadcopter.
- Launch Mission: Prompts the user to specify a ROS package or python script to run on each quadcopter.
- RESTART QUADS: Runs a reboot command on each quadcopter, this will return each quadcopter to the “uninitialized” state.
- QUIT: Closes the pygame window.