sdmay18-17 MicroCART

MicroCART: Microprocessor-Controlled Aerial Robotics Team

Meet the Team

Team Members: Kyle Trost, Tyler Imboden, Peter Thedens, Jakub Hladik, Dane Larson, Matthew Kelly, Austin Rohlfing, Blake Pries

Clients and Advisors: Dr. Phillip Jones, Dr. Nicola Elia

•••

Website: http://sdmay18-17.sd.ece.iastate.edu/

Problem Statement

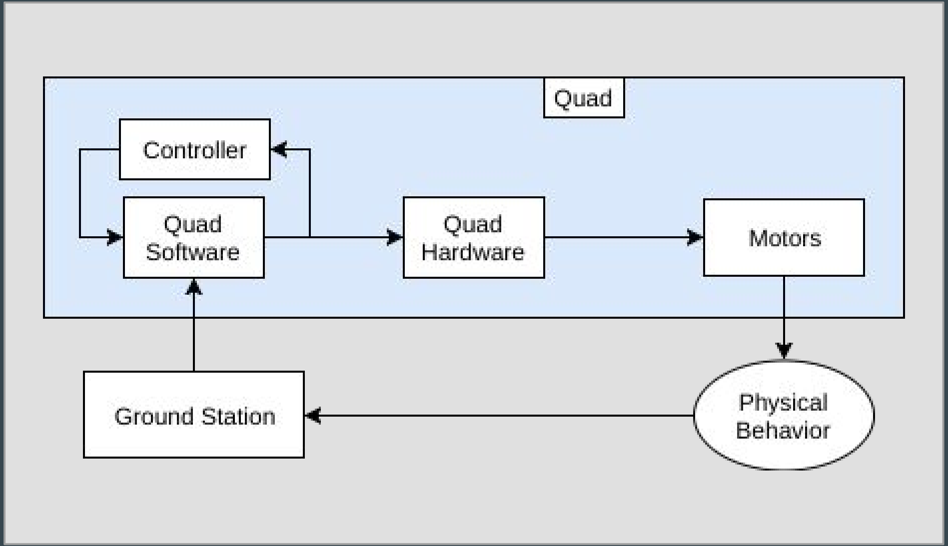
- Controls research platform
  - Modular
  - Robust
  - Low-Latency
- Focus on infrastructure
  - Controls is one small portion of project
- Common real-world application

Keywords and Definitions

- Quad/Quadcopter
  - Quadrotor helicopter
- Camera tracking system
  - Set of 12 infrared Opti-track cameras
- Ground station
  - Application running on a host computer that communicates with quad and camera tracking system via Wi-Fi
- Continuous integration
  - Automated process of running tests on every commit to the repository
- GUI
  - Generated user interface

Primary Areas of Development

- Quadcopter Software
- Ground Station
- Controls
- Testing and Virtual Quad



Functional Requirements

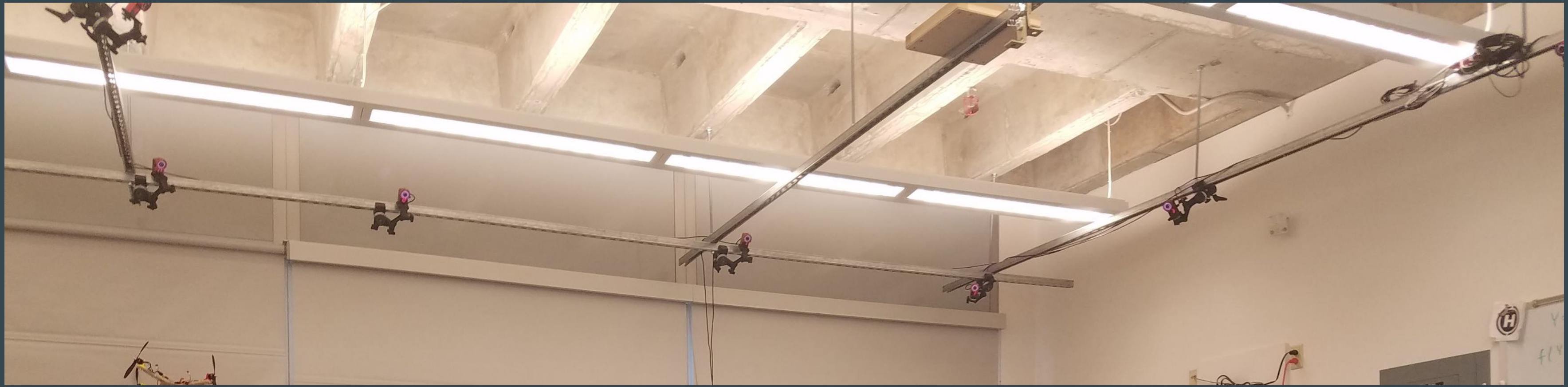
- Quad Software
  - Stability around a setpoint
  - Smooth autonomous movement
  - Real-time data logging
- Ground Station
  - Multiple real-time quad support
  - Real-time data logging and visualization
  - Support of new demos
  - Safety Features
- Controls
  - Enable autonomous flight with controls model
- Testing
  - Automated test suite

Non-Functional Requirements

- Documentation
  - Working with a continued repository
- Performance
- Testing

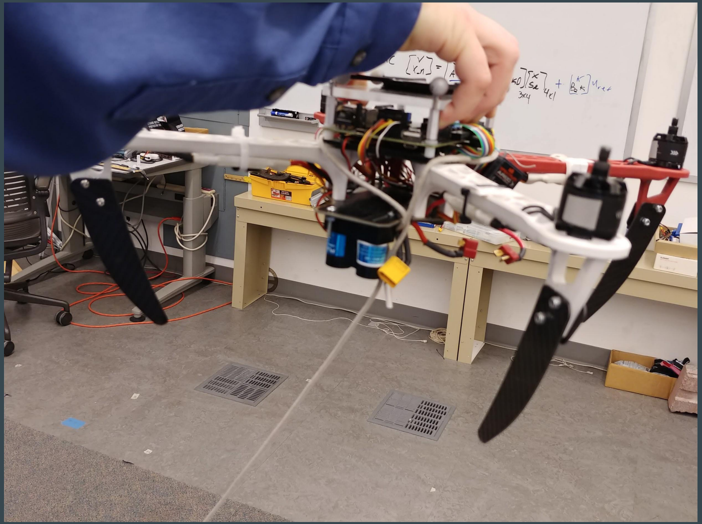
Technical Constraints and Considerations

- Camera tracking system
  - Screened area boundaries
  - Decreasing accuracy when approaching boundaries
- Battery endurance
  - 10-15 minute flight time



Potential Risks & Mitigation

- Safety hazards
  - Spinning rotors
  - Potentially unstable quad
  - LiPo battery charging
- Risk to Quadcopter
  - Testing controls can cause unknown behavior
  - Loss of position information
- Precautions
  - Quadcopter tether
  - Simulation testing
  - Camera safe zones
  - Battery bags
  - No unattended batteries



Quad with tether attached

Market Survey

- Georgia Tech: DURIP, others
  - Remote-access testing
- University of Maryland: TERP
  - LQR controller
  - MATLAB implementation
- Stanford: STARMAC
  - Basic controls
  - Research focused on collision avoidance
- M.I.T.: RAVEN
  - Multiple vehicle types
- Iowa State: MicroCART
  - Modular controls implementation

Stanford STARMAC platform
from: http://ai.stanford.edu/~gabeh/research.html

Resource/Cost Estimate

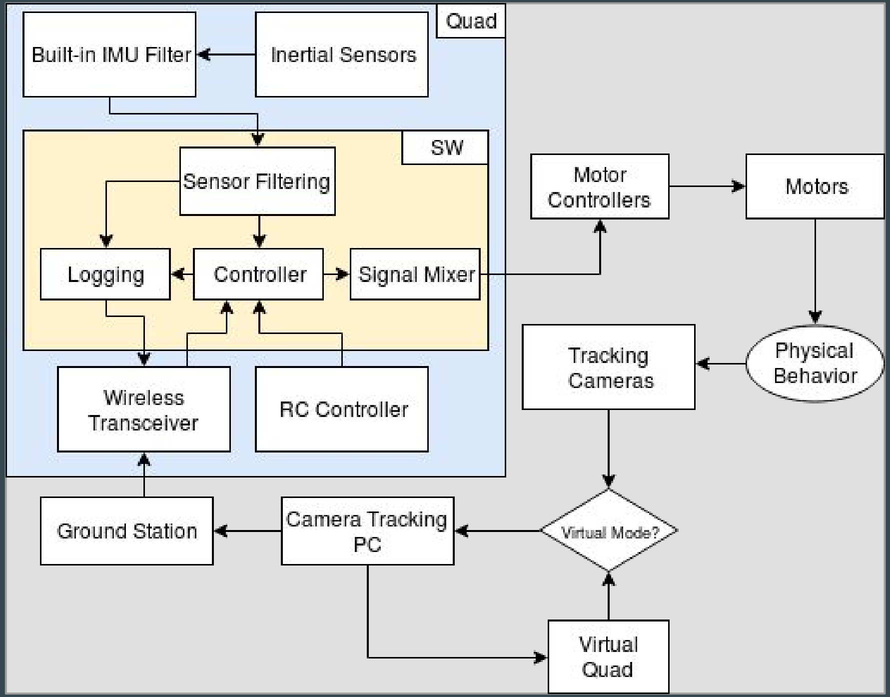
Part	Cost
Frame & Speed Controllers	$270
Zybo Board	$189
Lidar Sensor	$134
Optical Flow Sensor	$120
Radio Receiver	$40
Remote Controller	$110
Batteries	$35

Project Deliverables

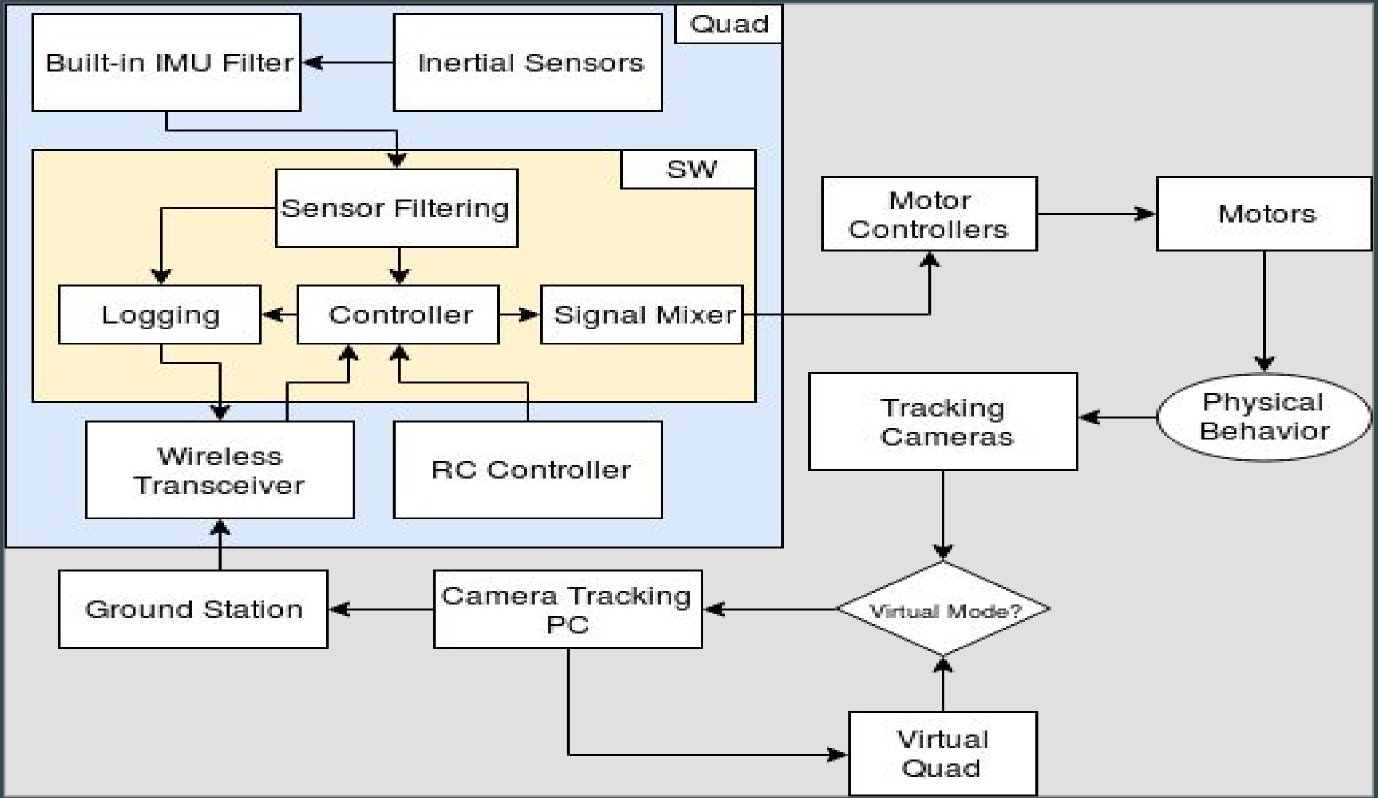
- Improving existing system
  - Documentation
- Adding new features
  - Multi-vehicle navigation
  - Manual “assisted” mode
  - Flight simulation
  - Linearized control model
  - Flashy demo

Hardware and Software Platform

- Quad hardware
  - FPGA
- Quad software and controls
  - Wi-Fi
  - PID controller
- Ground station
  - Linux workstation
  - Camera tracking system
- Testing
  - Continuous integration server
  - Virtual quadcopter simulator



Functional Decomposition



Detailed Design of Ground Station

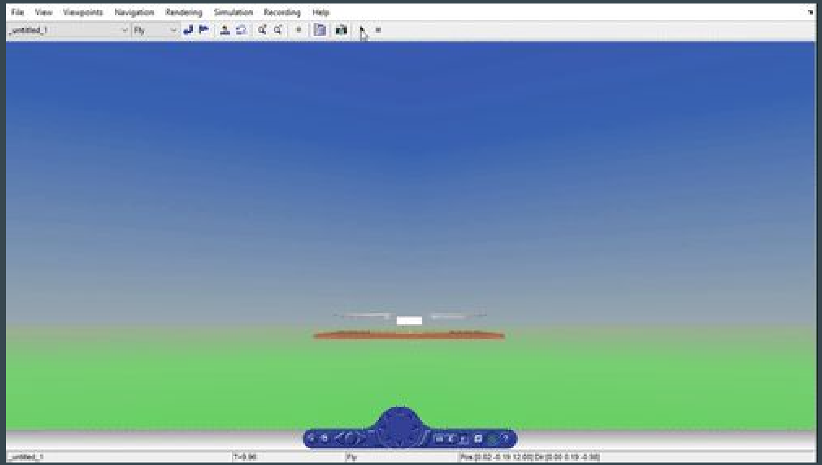
- Add multiple vehicle support
- Add multiple vehicle type support
- Additional features to GUI
- Data analysis tool
- Documentation update

Detailed Design of Quad Software

- Manual "assisted" mode
  - Take away direct input to output connection
  - Use input to modify PID setpoints
  - Makes quad easier to fly because the PID will stabilize
- Real-time data logging
  - Move communication to second core
- Update tool flow

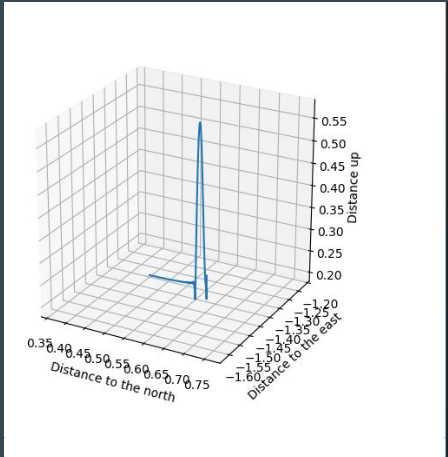
Detailed Design of Controls

- Integrated simulation
  - Verify future changes
  - Interact with ground station
- Linear Controls Model
  - Collaborate with PhD student Matt Cauwels
  - Verify model with simulation
  - Refine model with physical flights



Detailed Design of Testing Tools

- Flight simulation
  - Design a virtual quadcopter system and flight simulator
- Use robust third-party test framework




```
Event 0 (throttle up 80%):
  Executes once if all of the following are true:
    sim-time-sec >= 2.0000
  Actions taken:
    {
      set throttle-cmd-norm to 0.7500 (constant via step)
      set throttle-cmd-norm to 0.7500 (constant via step)
      set throttle-cmd-norm to 0.7500 (constant via step)
      set throttle-cmd-norm to 0.7500 (constant via step)
    }

Event 1 (throttle back down to 0%):
  Executes once if all of the following are true:
    sim-time-sec >= 5.0000
  Actions taken:
    {
      set throttle-cmd-norm to 0.0000 (constant via ramp with
      set throttle-cmd-norm to 0.0000 (constant via ramp with
      set throttle-cmd-norm to 0.0000 (constant via ramp with
      set throttle-cmd-norm to 0.0000 (constant via ramp with
    }


  Event 0 (throttle up 80%) executed at time: 2.0001


  Event 1 (throttle back down to 0%) executed at time: 5.0000

SIMULATION FINISHED
```

Test Plan

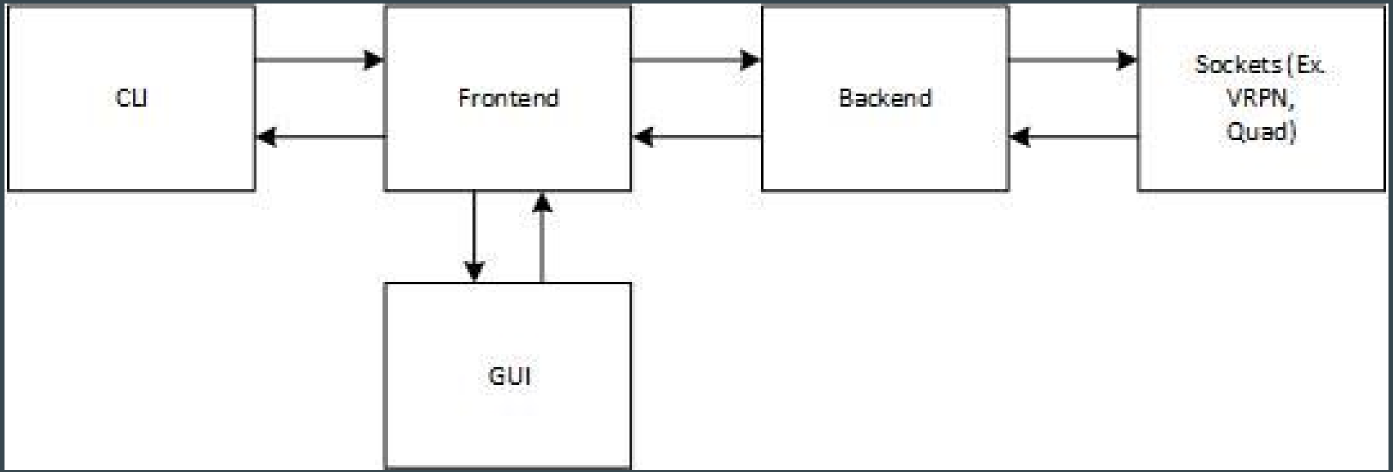
- Software
- Controls algorithm
- System

Current Project Status

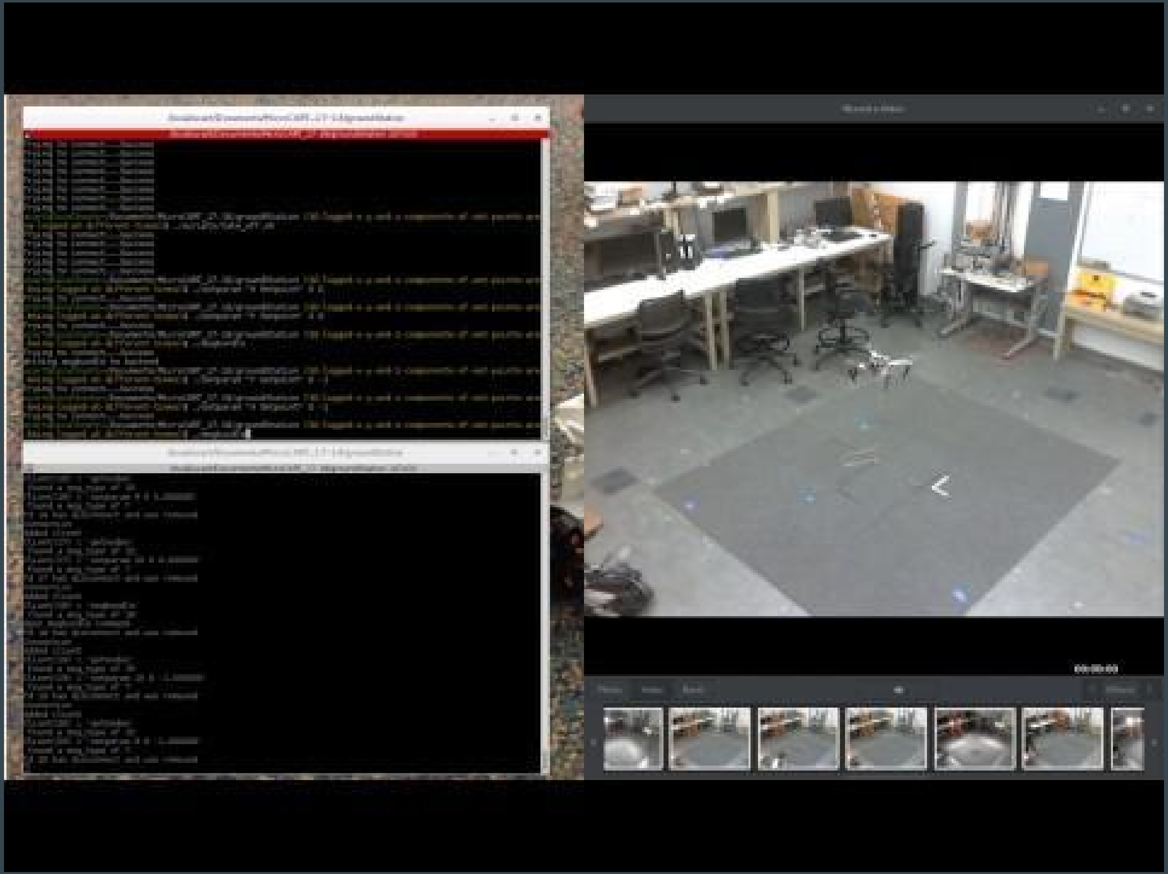
- Completed
  - Migrating unit tests to a more reliable testing framework
  - Overhaul of documentation
- In progress
  - Multiple vehicle ground station
  - Updating hardware specification tool to Vivado
  - Integrated flight simulator
- Future
  - Demonstration of new features
  - Real time logging / graphing
  - Manual assist mode

Questions

Ground Station Components



Message Bundle Testing Video



Work division details

Task	Assignee	Man-hours
Improve documentation	All	80
Understand existing code	All	40
Code improvements	All	30
Brainstorm Ideas for demo	All	14
Decide on final demo	All	8
Stabilize using lidar	Austin & Blake	60
Finish POC of virtual quad	Peter & Jakub	80
Fix GUI performance	Dane & Matt	120
Switch ISE to Vivado	Kyle & Tyler	20
Expand controls capabilities	Austin & Blake	180
Deploy additional testing	Peter & Jakub	160
Improve optical flow performance	Kyle & Tyler	140
Improve Virtual quad performance on Linux	Dane & Matt	100
Test demos	All	40
Get “Hard” numbers on performance	All	20
Presentation practice	All	16

Quad Software Block Diagram