IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

MicroCART
May 17-16

Microprocessor Controlled Aerial Robotics Team

Introduction

Project Statement: *To create an aerial robot as a modular platform for research in controls and embedded systems.*

Intended Users

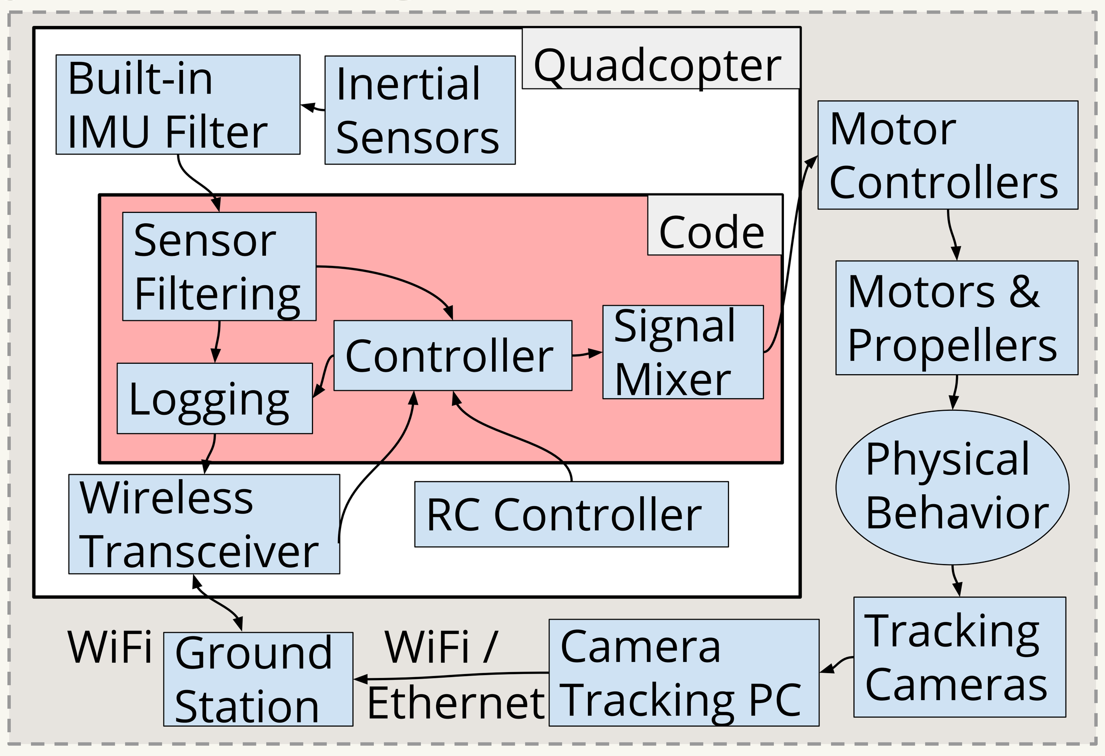
- Controls students who want to experiment with and test different control structures
- Controls research graduate and PhD students

Goals

- Autonomous flight with waypoint navigation
- Independence from camera system
- Develop a mathematical model of quadcopter

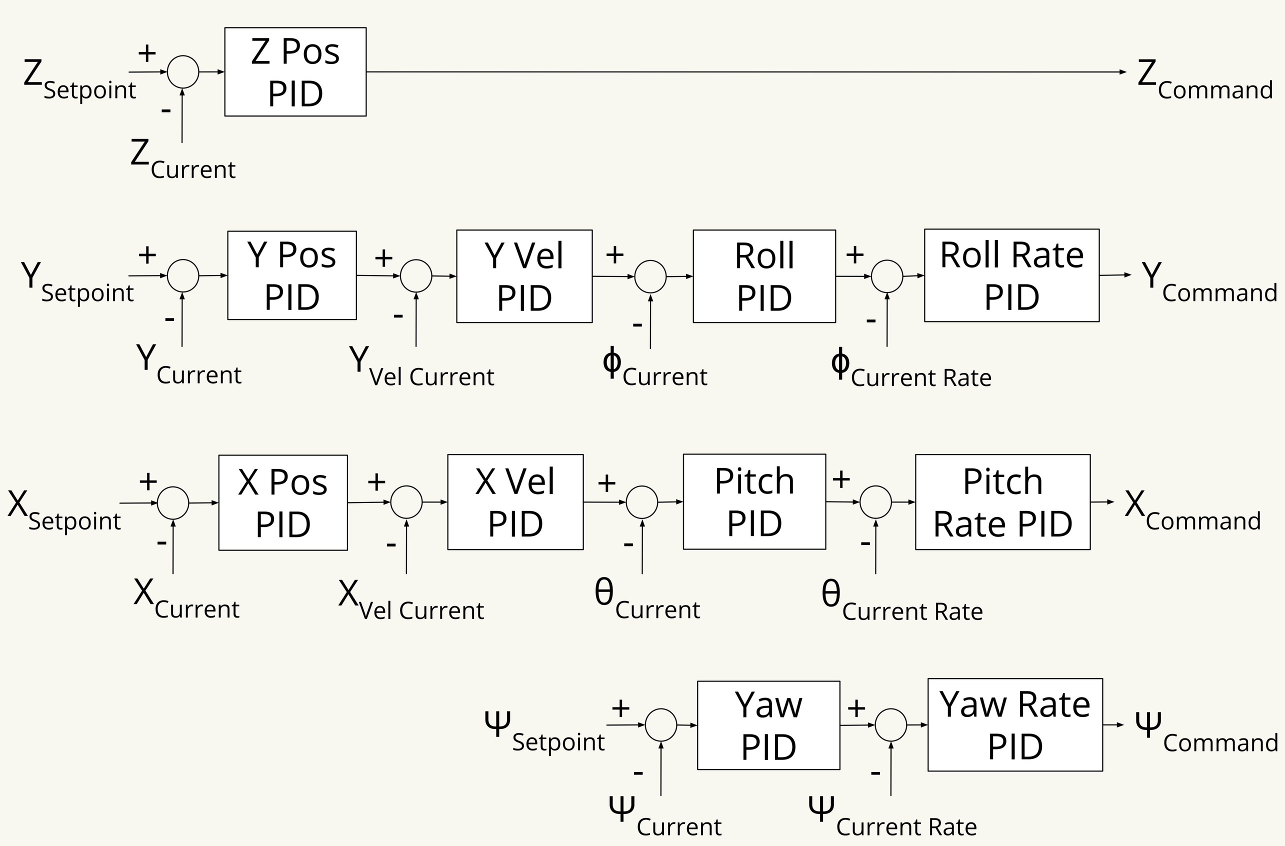
High level Diagram

System Diagram



Mathematical Model and Control Structure

- Created Simulink model of physical quadcopter and sensors
- Designed control structure around this open-loop model of the system
- Added ability to insert signals and log signals at any point in model

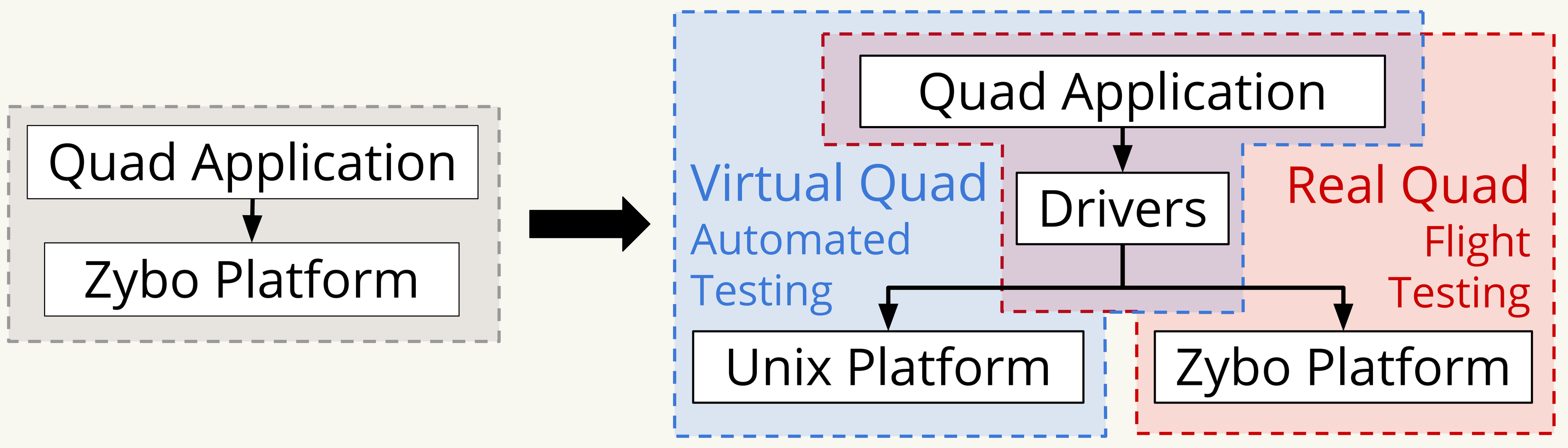


- Four parallel position movement options:
  - X-axis (Latitude)
  - Y-axis (Longitude)
  - Z-axis (Altitude)
  - Yaw
- Nested PID structure
- Added X and Y Velocity PIDs to original design
- Filtered derivative terms

Software Designed for Testing

Problem: Quad application was tightly coupled to the Zybo platform, forcing us to exclusively use end-to-end testing, slowing down the development process.

Solution: Re-design software using interface-like drivers, enabling us to run the application on any platform, including a Unix environment we use for unit and functional testing.



Original Testing Strategy

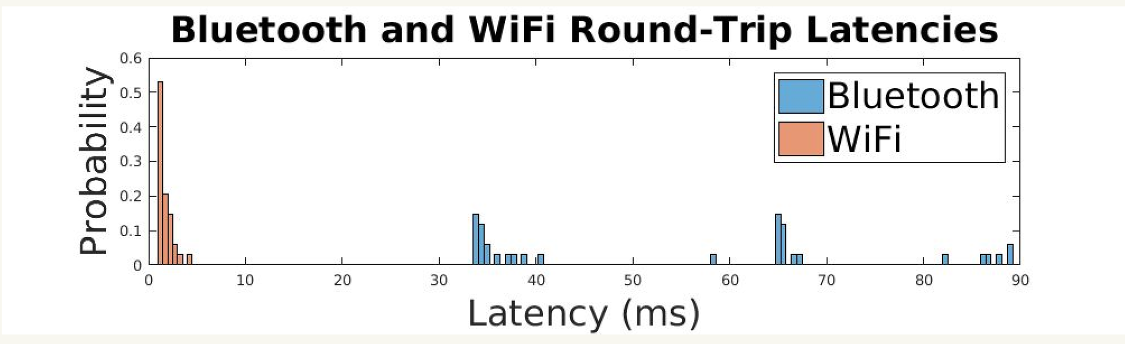
- End-to-End testing

New Testing Strategy

- Unit/Functional Testing (Continuous Integration)
- Dedicated Hardware Tests (Manual Scripts)
- End-to-End Testing

Communication

- Quadcopter
  - ESP8266 converts between WiFi and UART
  - Async UART connection to ESP8266
- Ground Station
  - TCP connection by backend
  - Reliable stream to backend
  - Backend handles client multiplexing with Unix domain socket

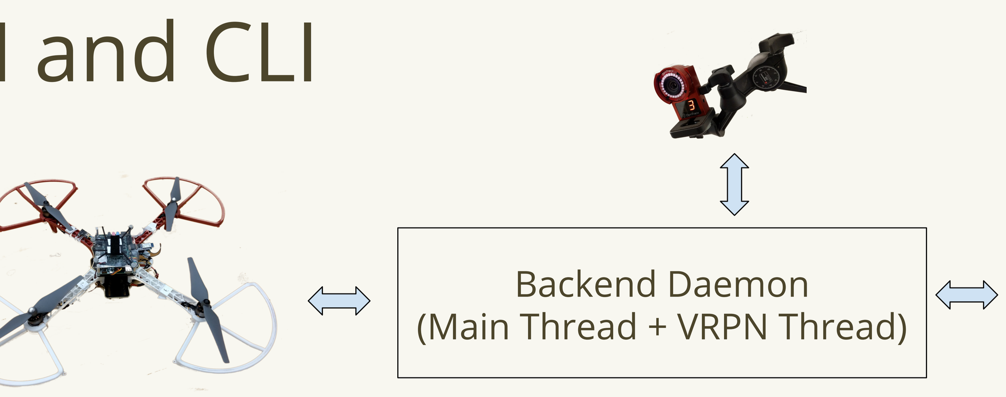


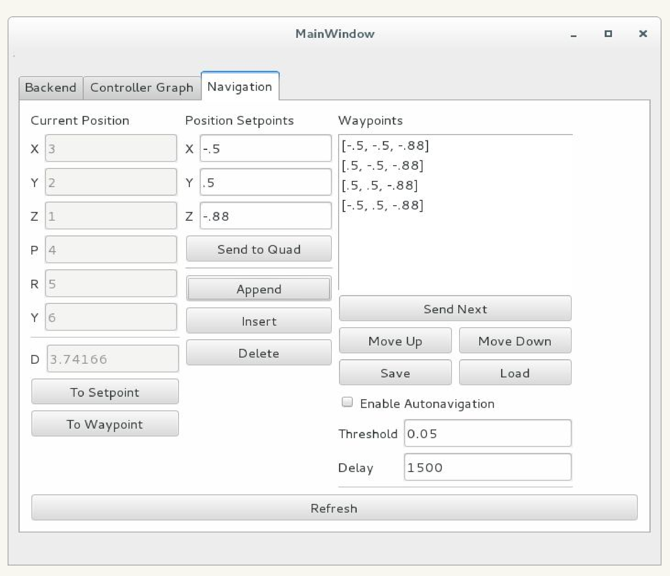
Added Capabilities

- Agnostic Height Sensing
  - Accomplished through LIDAR, a laser-based proximity sensor
- Stable Autonomous Flight
- Waypoint Navigation
- Automatic takeoff and touchdown

Lidar Sensor

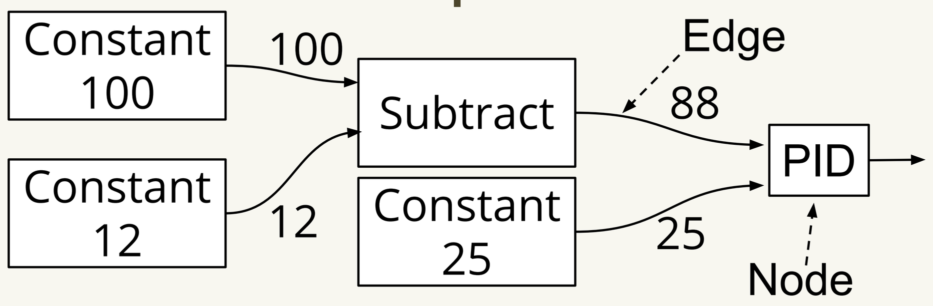
GUI and CLI





- Backend Daemon
  - Manages quad connection and tracking system
- Decoupled Frontend
  - GUI or CLI
  - Monitor quad position and orientation
  - Modify control structure
  - Send quad waypoints and navigation commands
  - CLI provides access to full functionality and scriptability

Directed-Graph Based Calculations



- Each computation resides within a single node.
- Output can be calculated via a depth-first search.

Applied to the Control Structure

- Advantages
  - This allows the controller to be easily reconfigured without modifying software
    - Can be modified from the ground station in real-time
  - Each node has a well-defined function, increasing testability and extensibility
  - Controller can be visualized in the GUI

Team Members

Brendan Bartels, Kris Burney, Joe Bush, Jake Drahos
Eric Middleton, Tara Mina, Andy Snawerdt, David Wehr

Clients and Advisors

Dr. Philip Jones
Dr. Nicola Elia

Acknowledgements

Matt Rich
Ian McInerney