MicroCART (sdmay18-17)

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

● **FPGA**
  ○ Field-programmable gate array

● **Ground station**
  ○ Application running on a host computer
  ○ Communicates over Wi-Fi to quad and camera tracking system

● **PID and LQR controllers**
  ○ Proportional–integral–derivative
  ○ Linear-quadratic regulator

● **Camera System**
  ○ 12 camera IR tracking system
Primary Areas of Development

- Quadcopter Platform
- Ground Station
- Controls
- Testing and Simulation
Functional Requirements

● Quad Platform
  ○ Ground station communication
  ○ Extract sensor data from sensors
  ○ Arbitrate which control graph to use

● Ground Station
  ○ Multiple quad support
  ○ Support of demos
  ○ Option for AP and Client on ground station PC

● Controls
  ○ Enable autonomous flight with controls model

● Testing and Simulation
  ○ Automated test suite
  ○ Sanity check of the quadcopter software through simulation
Non-Functional Requirements

● Documentation
  ○ Working with a continued repository
  ○ Document new and existing progress

● Performance

● Testing
Technical Constraints and Considerations

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

● Improving existing system
  ○ Documentation
  ○ Upgrade testing framework
  ○ Second Quad
  ○ Vivado

● New features added
  ○ Multi-vehicle navigation
  ○ Manual “assisted” mode
  ○ LQR controller
  ○ Additional demos
  ○ Flight simulation
Hardware and Software Platform

- **Quad hardware**
  - FPGA

- **Quad software and controls**
  - Wi-Fi
  - PID or LQR controller

- **Ground station**
  - Linux workstation
  - Camera system

- **Testing**
  - Continuous integration server
  - Quadcopter simulator
Functional Decomposition
Design and Implementation
2nd Quad

- **Challenges:**
  - Accelerometer biasing/placement
  - Radio controller settings
  - ESCs/DJI Documentation

- **New Versions of Components**
  - Lidar Lite 2 vs Lidar Lite 3
  - Controller / Radio Receiver

- **Improvements over 1st Quad**
  - Wiring
  - Component Mounting
## Resource/Cost Estimate

<table>
<thead>
<tr>
<th>Part</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame &amp; Speed Controllers</td>
<td>$270</td>
</tr>
<tr>
<td>Zybo Board</td>
<td>$189</td>
</tr>
<tr>
<td>Lidar Sensor</td>
<td>$134</td>
</tr>
<tr>
<td>Optical Flow Sensor</td>
<td>$120</td>
</tr>
<tr>
<td>Radio Receiver</td>
<td>$40</td>
</tr>
<tr>
<td>Remote Controller</td>
<td>$110</td>
</tr>
<tr>
<td>Batteries</td>
<td>$35</td>
</tr>
</tbody>
</table>
Vivado Upgrade

● Major Strides Upgrade from XPS to Vivado
  ○ More stable
  ○ Newer software
  ○ Faster FPGA hardware development

● Ran into many compatibility issues

● Results
  ○ Was flying, but unstable
  ○ Now building but failing in runtime
  ○ Findings well documented
  ○ Good base for future members
Ground Station

- Added multiple vehicle support
  - Quads, Crazyflies, and Roombas
- Ground Station access point
- Additional features to GUI
- Manual “assisted” mode
  - User input with a controller to modify PID setpoints
User Interface (Ground Station GUI)
Detailed Design of Controls

- **LQR Controller Design**
  - Generic system linearization
  - Testing with existing Simulink model
  - Porting to quad-side C

- **Weight Selection**
  - Multiple weighting patterns
  - Constrained minimization with MATLAB

- Design based on past research [1]

- Goal: implement fundamentals for future
  - State-feedback control for quad

![Graph showing setpoints and actual position with diagonal weighting and LQR setpoint tracking.](image-url)
Quad Simulator

- Uses JSBSim
- Visualization through FlightGear
- Data logging and data input over sockets
  - Plug-and-play control algorithm over sockets
- Joystick integration with real-time simulation
- Automated testing through scenario scripting and event detection
Test Plan

● Software
  ○ Unit and integration tests
  ○ Upgrade testing framework [5]

● Controls algorithm
  ○ Simulink

● System
  ○ Flight simulation
  ○ Flight tests

```shell
make -C src/computation_graph test
make[1]: Entering directory `/builds/danc/MicroCART/quad/src/computation_graph'
gcc -g -o run_tests test/computation_graph.c obj/computation_graph.o -I..../inc -L..../lib -lunity -lm -lgraph_blocks -DUNITY_INCLUDE_CONF
IG_H
./run_tests
# test/computation_graph.c:244:test_adding_2_numbers:PASS
```
Value Added

● Improving existing system
  ○ Documentation
  ○ Upgrade testing framework
  ○ Second Quad
  ○ Vivado

● Adding new features
  ○ Multi-vehicle navigation
  ○ Manual “assisted” mode
  ○ Flight simulation
  ○ LQR controller
  ○ Additional demos

Future Work

● Communication between Quads
● Linux Running on second core (OpenCV?)
● Strong real-time logging framework that is configurable for use in research
● More complete usage of testing framework
● Integrate flight simulation into continuous integration
Questions
References


Backup Slides
Ground Station
Latency Tests

- Sends 10,000 Packets
- TX and RX tied on the ESP Chip
- When running with multiple Clients tests were run on multiple processes
- AP is the old configuration
Controls Media

PID
Modification of Control Graph Implementation