Development of a Remote Interface and API for Networked Control of Multi-Vehicle Systems

Steve Granda, University of Illinois at Urbana-Champaign

ADVISORS: Prof. Geir Dullerud, Seungho Lee

---

**Goals**

- Develop an API for software development/control of the HoTDeC and Quadricopter testbed.
- Reverse engineer and integrate Quadricopters to work with existing network and lab vision system.
- Develop a client to provide remote users with the ability to control testbed safely.

**Fundamental Questions/Challenges**

- How much control should the testbed allow for remote users?
- In developing the API, should we build from the ground up or use an existing implementation?
- Is it possible to use the existing vision system to track the devices in flight, or will a new one need to be built?

**Related Work and References**


---

**Background**

- We currently have several autonomous hovercraft (HoTDeC) in a testbed that receive visual feedback from a system of networked overhead cameras.
- We would like to expand our testbed to include Quadricopters as well as create a development environment that would allow remote usage of the testbed by other institutions though a simple client.

---

**Research Plan**

- Find safest method of obtaining full access to Quadricopter hardware and software.
- Find a suitable server model for secure remote control of testbed.
- Find best language to implement remote client and API.
- Find effective way to expand the testbed without a large impact to existing system.

---

**Research Results**

- Quadricopters are fully open with possibility of adding custom onboard hardware or software for future projects.
- The Remote Client written in Python/Qt/ZMQ allows the user to send commands via the API to the control server as well as receive coordinates, video, and sensor data from the testbed.
- Control Server runs software written in Python/ZMQ tying the lab vision system and hovercraft to the Quadricopters and allowing secure control of the testbed with API.