6 Degree of Freedom Autonomous UAV

Long Vo, Liam Berti, Ritesh Misra, Levi Burner
Department of Electrical and Computer Engineering

PITT | SWANSON ENGINEERING
ELECTRICAL & COMPUTER

Background

- Quadcopters adjust orientation to translate; they can only control 4 Degrees of Freedom (DOF)
- This limits possible motions and decreases mobility
- A 6 DOF UAV is not restricted in this manner
- Designs found in literature are complex which makes construction and controller design difficult

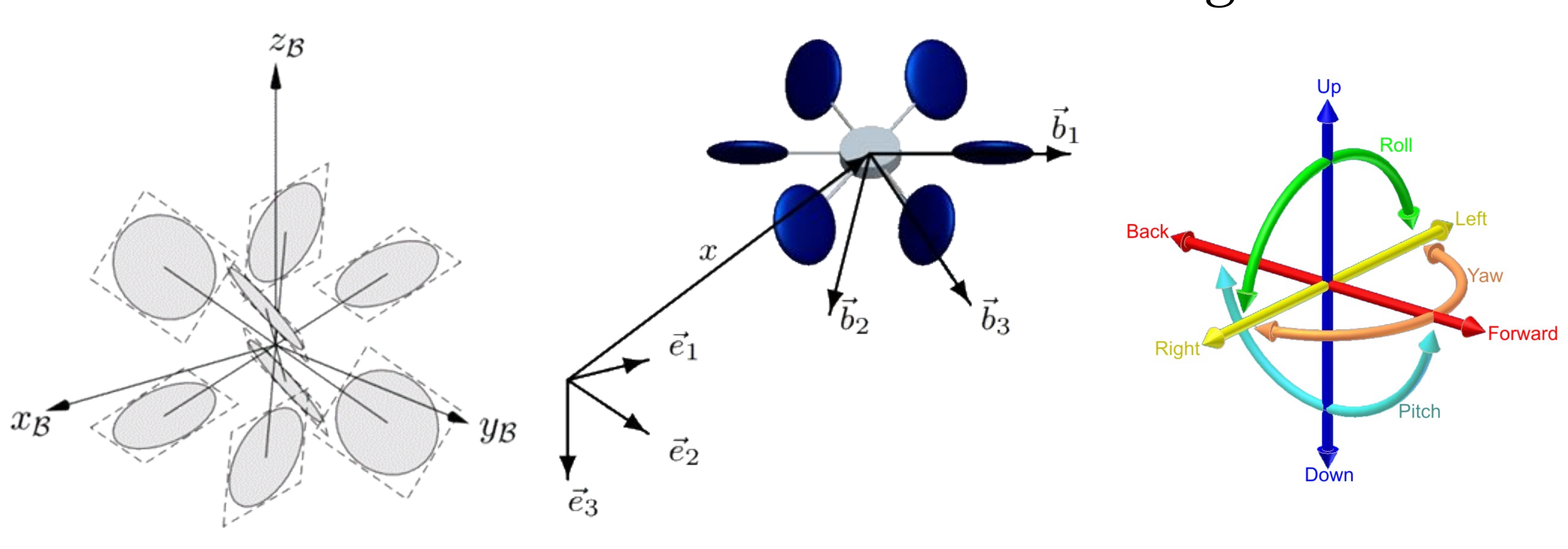


Figure 1. Examples of 6 DOF drones found in literature and an illustration of their controllable degrees of freedom

Design Considerations

- Goals:
  - Full 6 DOF control when UAV is close to level
  - Maximum agility when in the level position
  - Autonomous height hold and velocity control
- Solution:
  - 4 side rotors mounted on an octorotor frame
  - Suitable for close tracking of moving ground targets and high-speed navigation in constricted environments

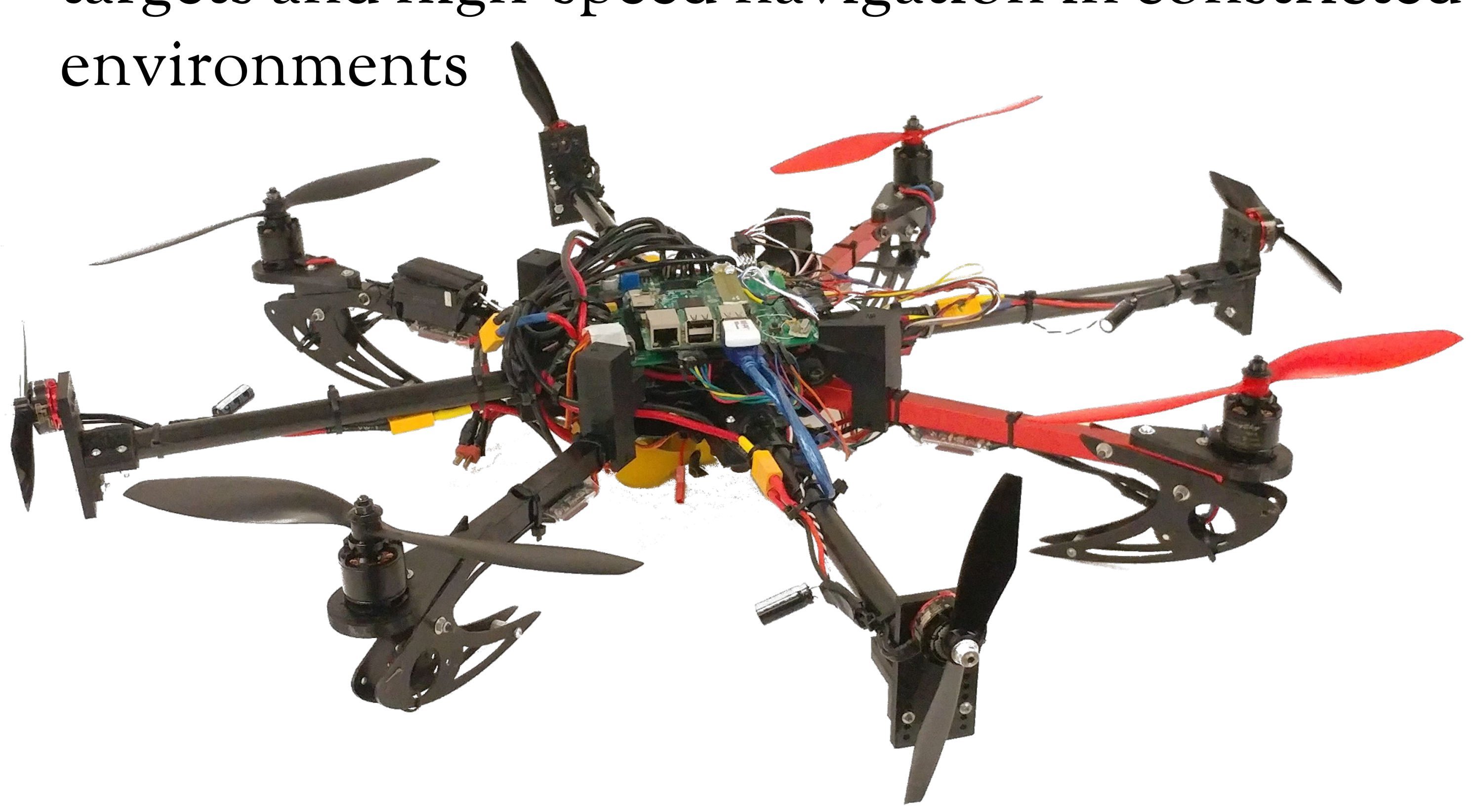
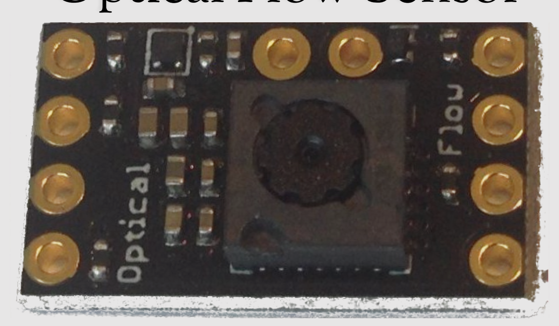


Figure 2. The 6DOF UAV built for this project.
The rotors are powered by a 3 Cell 5000 mAh LiPo battery.
Sensors and electronics are powered by a 3 Cell 1000 mAh battery.
It weighs ~2.3kg and can hover for 7-8 minutes.

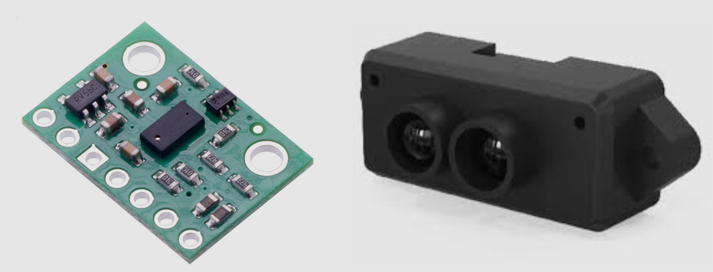
How it Works

Optical Flow Sensor



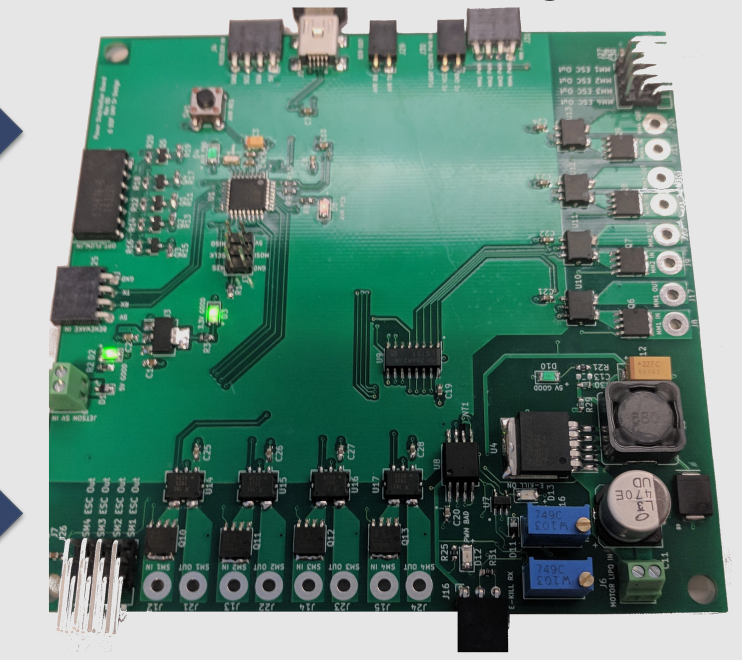
- *Camera and ASIC compute translational velocity estimate at 50 Hz*

Range Finders



- *Range finders use reflected lasers to detect height of drone*
- *First is used for 0-1.2 meters*
  - *Updates at 30 Hz*
- *Second is used for 1-10 meters*
  - *Updates at 100 Hz*

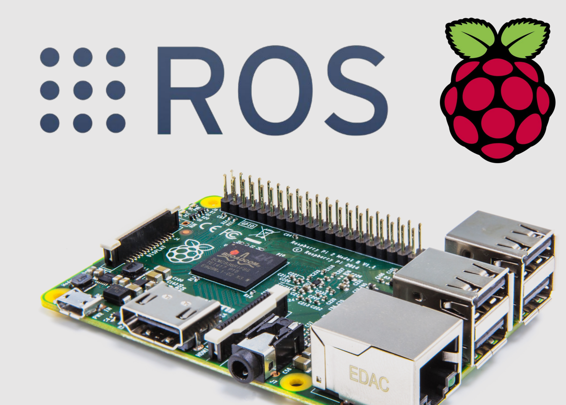
Custom Power Distribution and Sensor Interfacing Board



- *ATMEGA328P microcontroller reads sensors and commands side rotors*
- *Isolates noisy motor circuitry from computing and sensing circuitry*
- *Returns real time current and voltage usage statistics*
- *8 channel wireless circuit breaker safely shuts down motors*
- *30 A per channel - 240 A total*

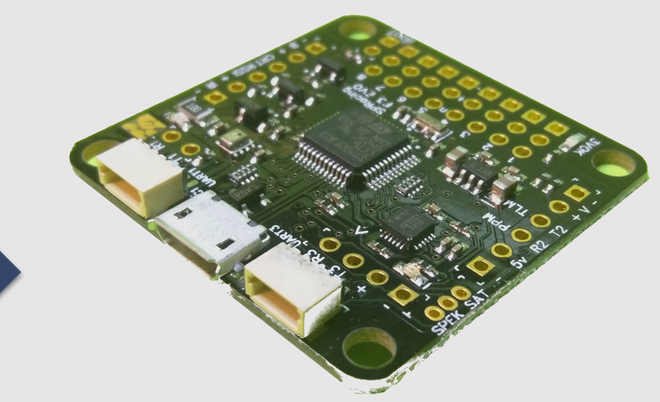
Companion Computer

ROS



- *Raspberry Pi 3 runs ROS nodes*
- *Kalman Filter fuses sensor data*
- *Motion profile generated at 10 Hz*
- *Custom PID controller and plant model update at 50 Hz*
- *Commands throttle, orientation, and side rotor thrust*

Flight Controller



- *SPF4EVO (Common racing drone flight controller)*
- *Runs orientation controller*
- *Accepts throttle and orientation targets from companion computer*
- *Relays IMU data at 100 Hz*

Side Thrust Motors

- *1900 kV*
- *6" dia. 4.5" p. prop*
- *0.8 kg peak thrust*

Main Thrust Motors

- *900 kV*
- *10" dia. 4.5" p. prop*
- *1.2 kg peak thrust*

Testing Results

Autonomous setpoint tracking was tested using a custom motion capture system and onboard sensors; jerk and acceleration were greater than possible with a traditional quadcopter.

Figure 3. (right) 3-D position is tracked within 4 cm using a styrofoam marker attached to the UAV

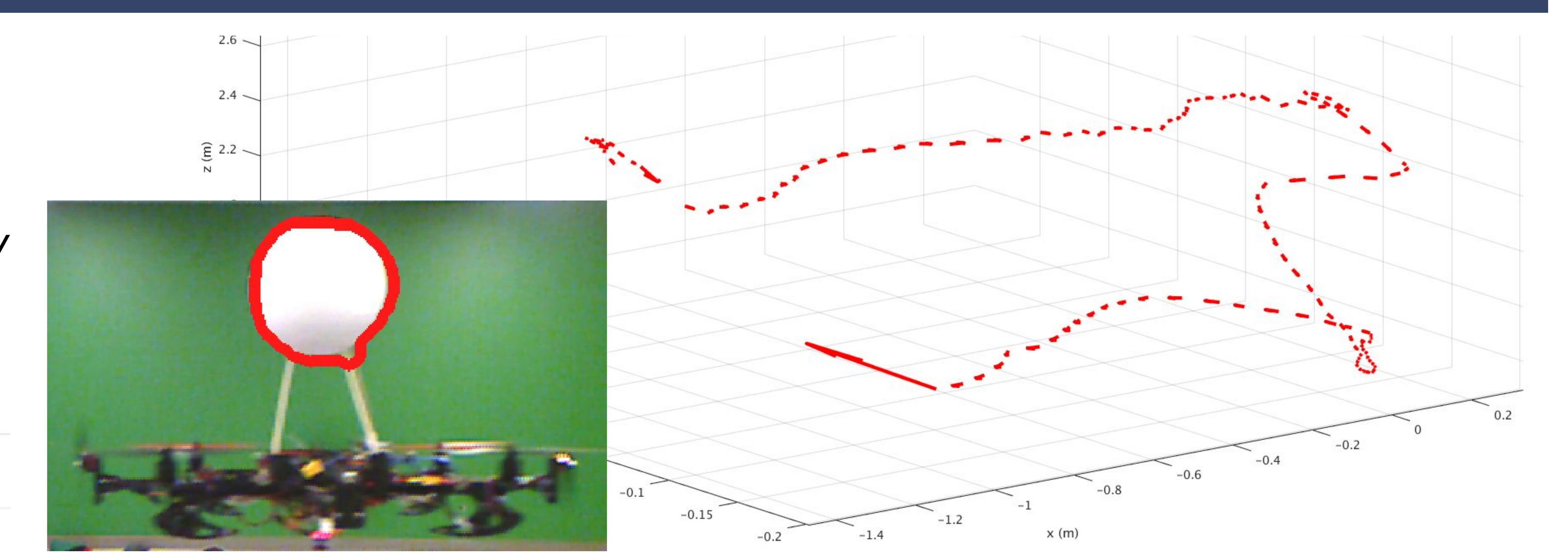


Figure 4. Measured height (red) versus target height (blue) when taking off and landing

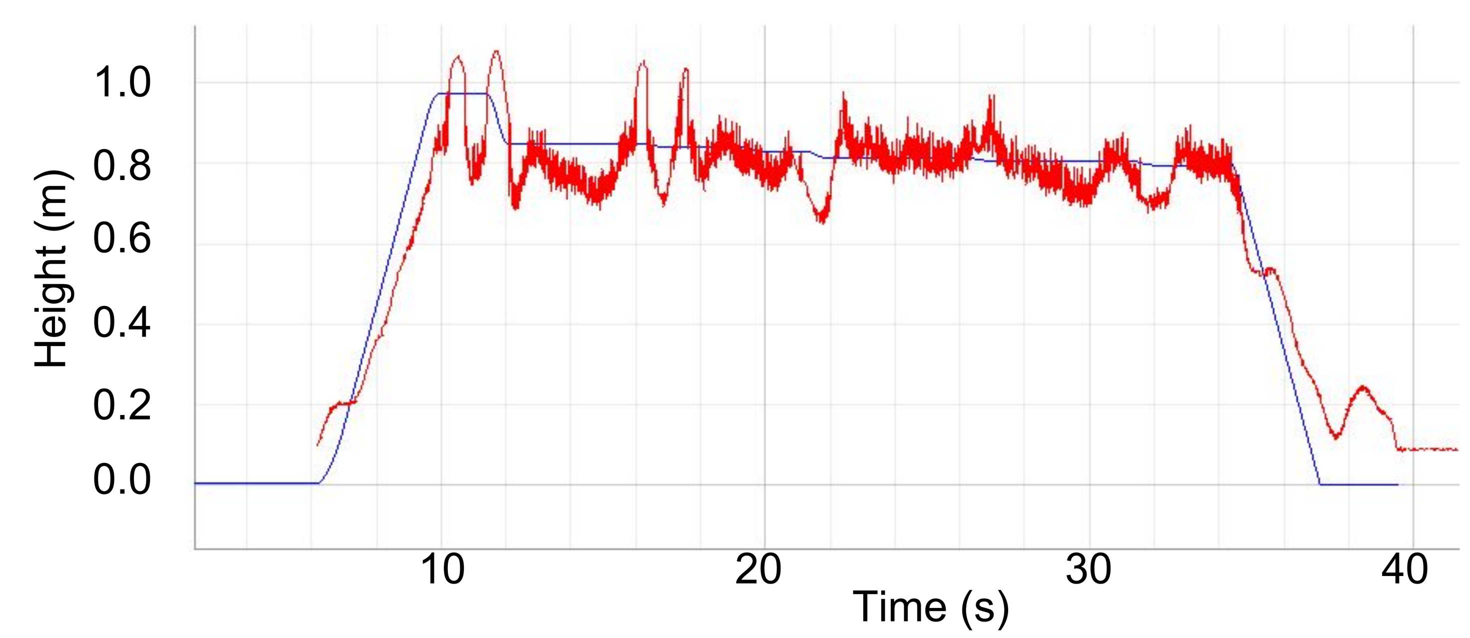
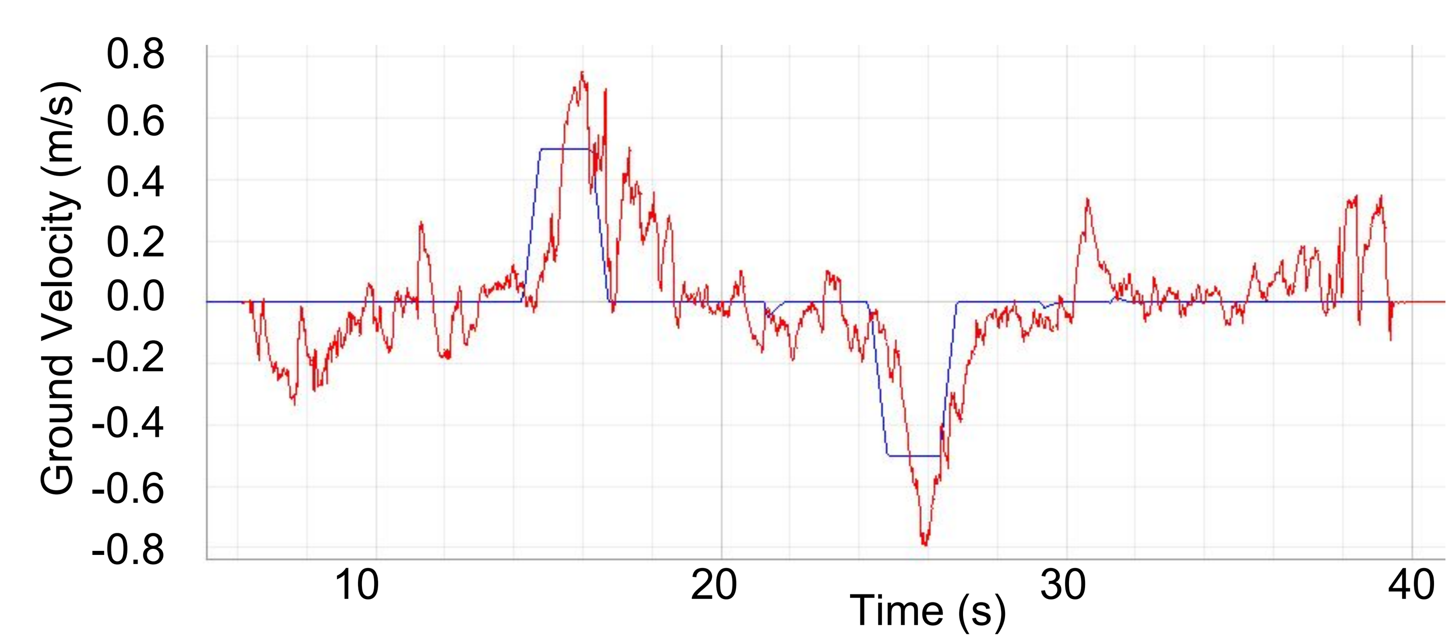


Figure 5. Measured velocity (red) versus commanded velocity (blue) in the x-direction during an autonomous flight



Acknowledgements

- Dr. Dickerson and Dr. Dallal for their support throughout the semester
- Dr. Mao for his mentorship and encouragement
- Jim Lyle, Bill Mcgahey, and Corey Weimann for the use of SERC resources
- Pitt's Robotics and Automation Society (RAS) for providing parts
- Pitt SSOE ECE Department for providing funding to RAS and Senior Design
- Pitt's 2016-17 and 2017-18 International Aerial Robotics Competition team whose previously developed software made the project feasible

References

- E. Kaufman, K. Caldwell, D. Lee, and T. Lee, "Design and development of a free-floating hexrotor uav for 6-dof maneuvers," in 2014 IEEE Aerospace Conference, pp. 1–10, March 2014
- D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3261–3266, May 2016.
- Pitt's International Aerial Robotics Competition team's code available at: github.com/Pitt-RAS/iarc7_common