Autonomous Navigation of a Quadrotor Helicopter Using GPS and Vision Control

Group 1

December 10, 2009

Project Tasks

- Fly helicopter to a predetermined location using GPS feedback
- Take pictures at this location
- Fly a planned path along GPS coordinates
- Take pictures along the reference path
- Use GPS and camera feedback to visually servo to and land on a marked target

Practical Applications

- Any process involved with the discovery and inspection of small objects
- UAV refueling midflight
- Land mine detection by autonomous ground robots
- Landing of an AUV or parking an autonomous ground vehicle at a certain location based on object recognition

Quadrotor Specifications

- Weighs 1.25 kg
- 200 g maximum payload
- 23 minute battery life (hovering)
- 12 minute battery life (with max load)

Photo of the Ascending Technologies Hummingbird Autopilot Quadrocopter removed due to copyright restrictions.

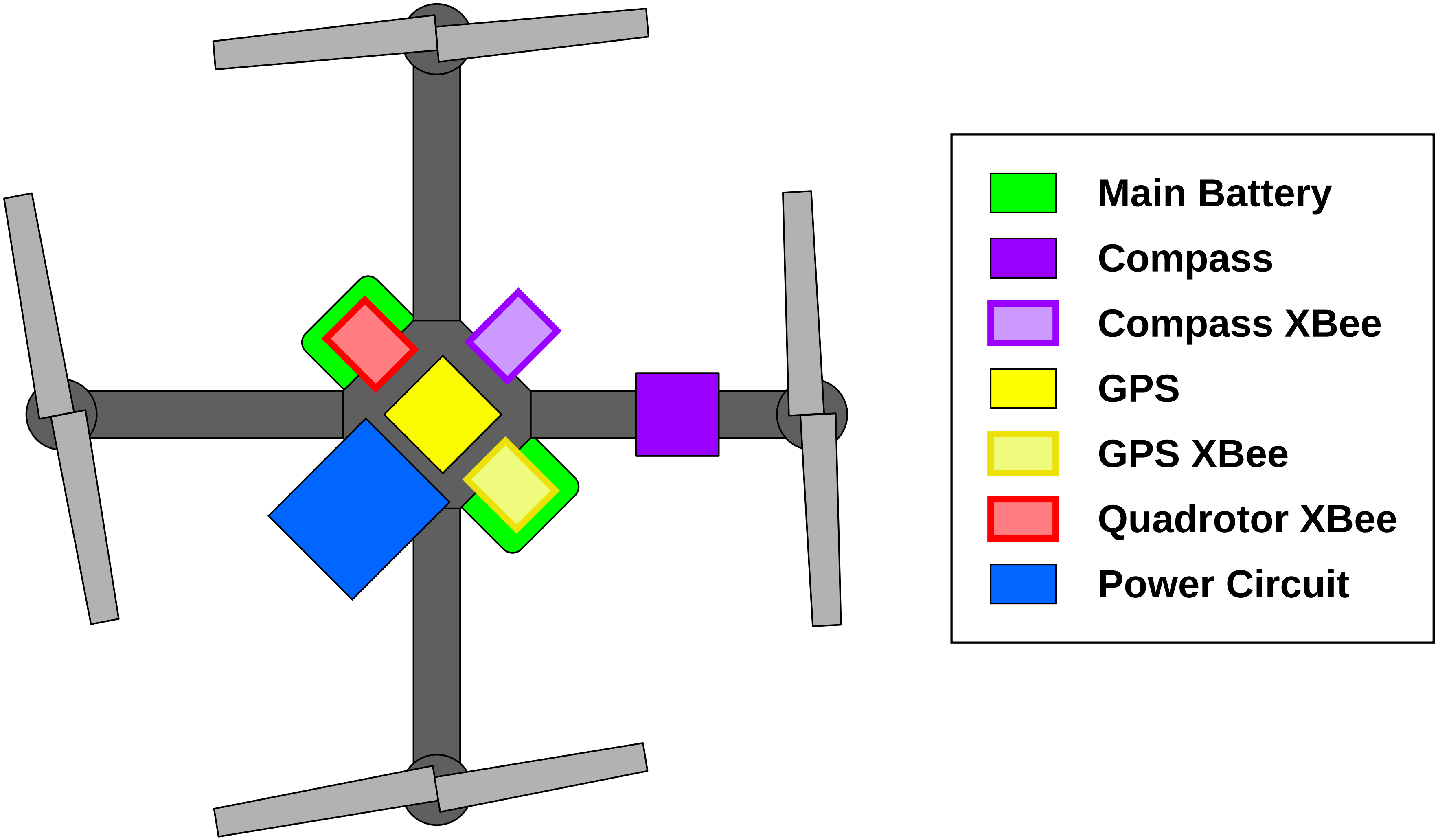
Quadrotor Dynamics

- Independent thrust, pitch, roll and yaw.
- Quadrotor able to make precise maneuvers.
- Can move one of two ways

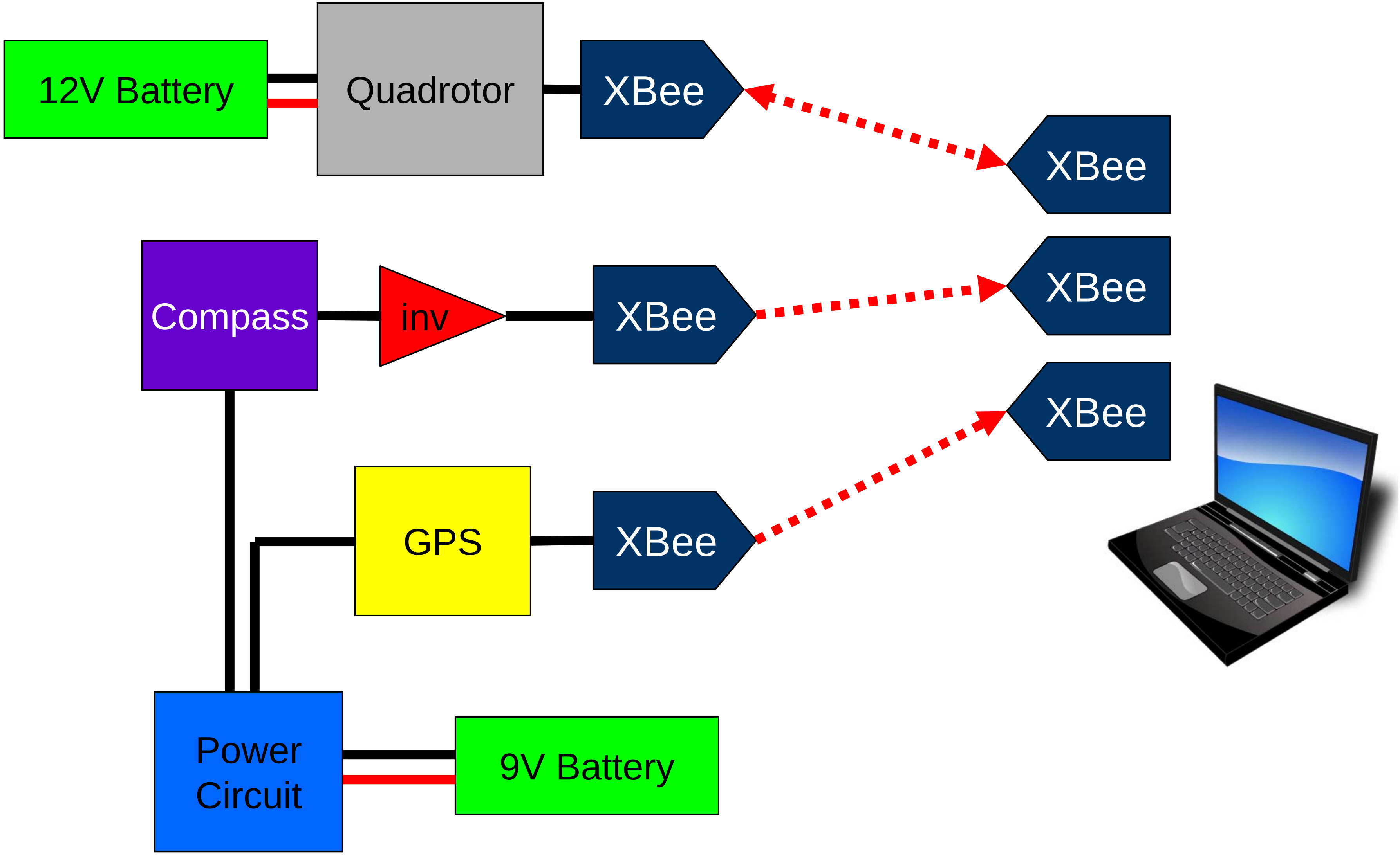
Pitch/Yaw　　　　　　　　　　　　Pitch/Roll

- Operates in random environment (wind)

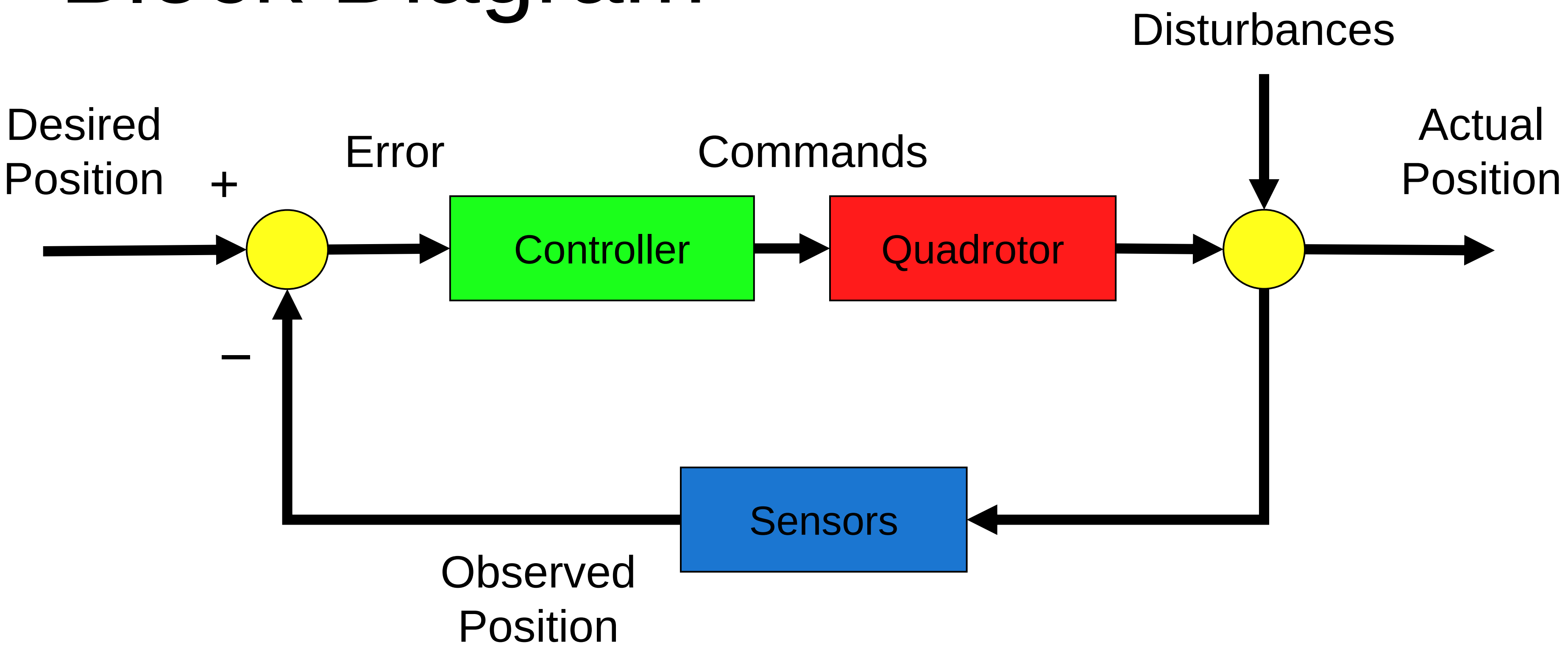
Current Hardware Layout



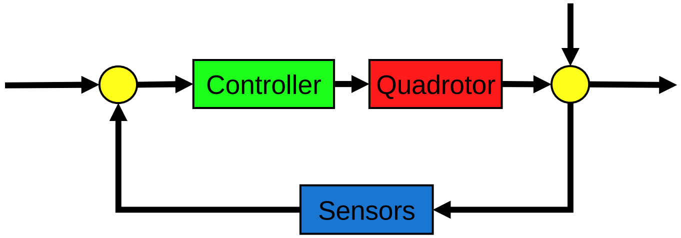
Electrical and Signal Flow Schematic



Block Diagram

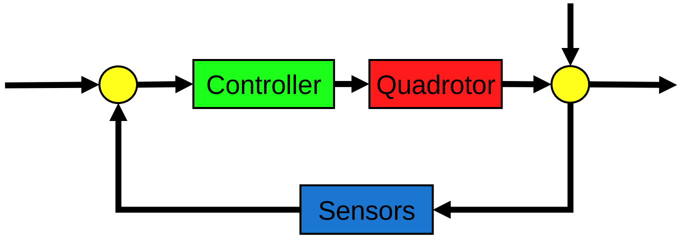


Controller:



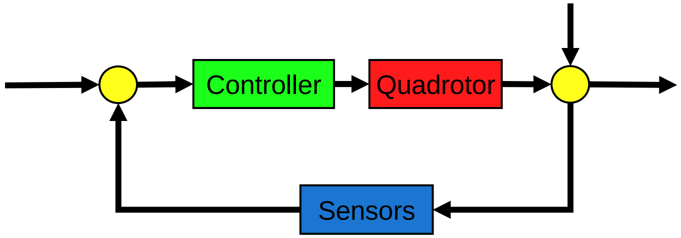
- Point and go control strategy
- Needs to be robust against sensor noise and wind gusts
- Written in MATLAB

Quadrotor:



- Controllable pitch, roll, thrust, and yaw-rate.
- Low drag
- Internal stabilizing controls

Sensors:

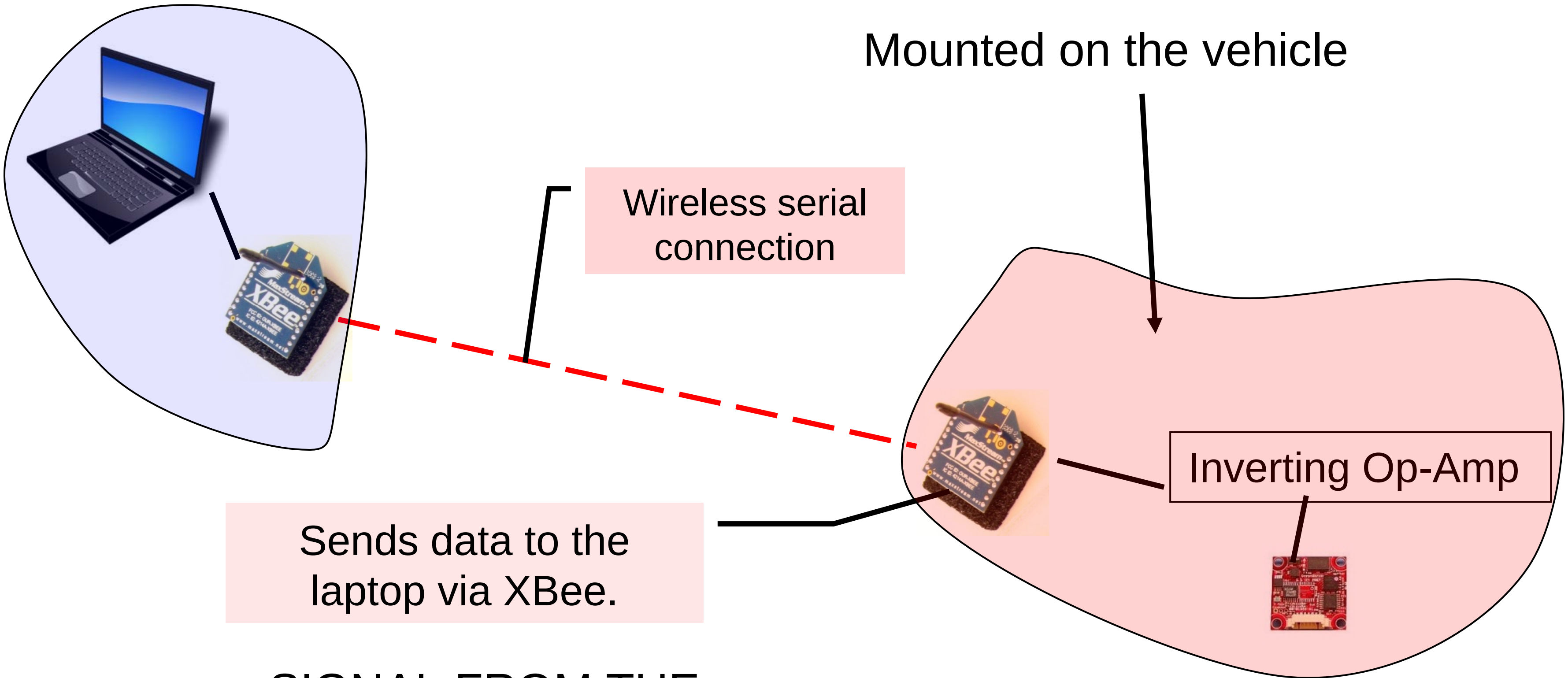


- GPS
- Compass
- Camera
- Quadrotor's internal sensors
- Communicate wirelessly

Compass, Power Circuit, and Anemometer

Student A

Compass



SIGNAL FROM THE COMPASS MUST BE INVERTED BEFORE IT IS FED TO THE XBEE

www.digi.com

www.ocean-server.com

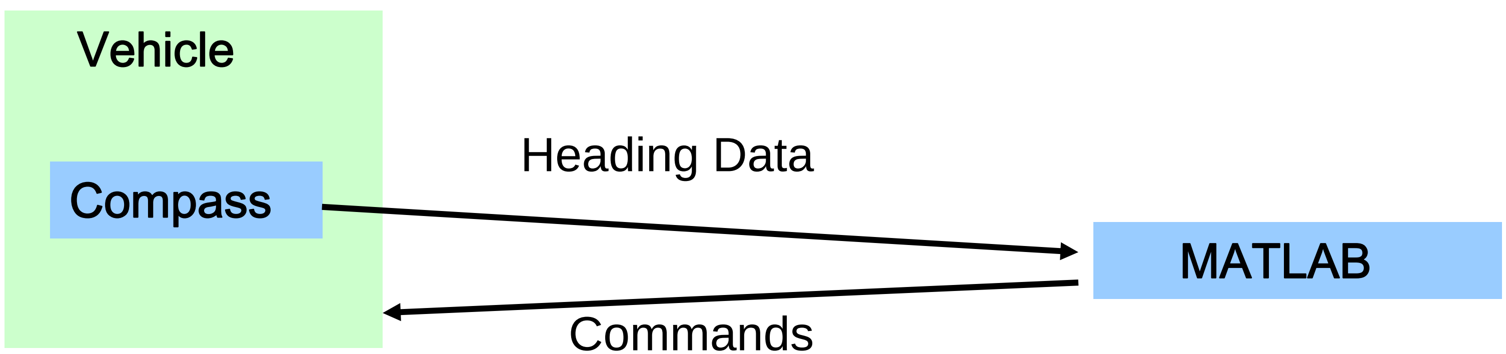
Images from the OpenClipArt Library and mangonha on Flickr.

Compass

The heading is saved directly to a variable in MATLAB. (Using a C++ mex function called “compmat”).

heading <1x100 double>

33	34	35	36	37
157.1000	133.6000	142.1000	151.2000	149.5000



Power Board Circuit

12 V in

5 V out

3 V out

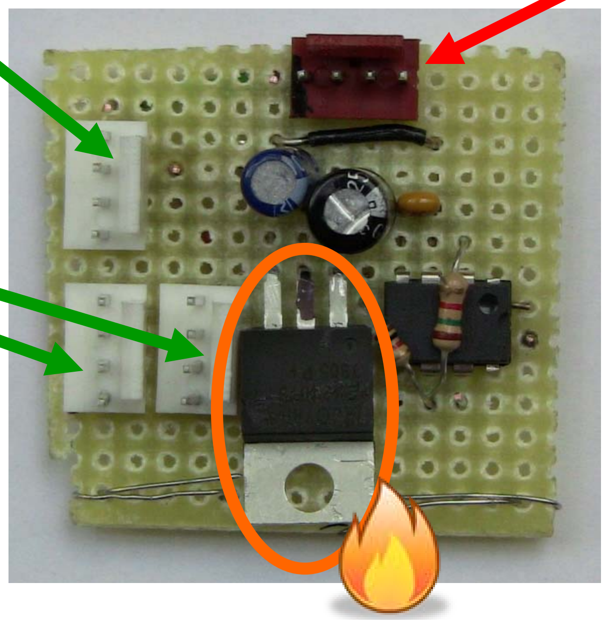
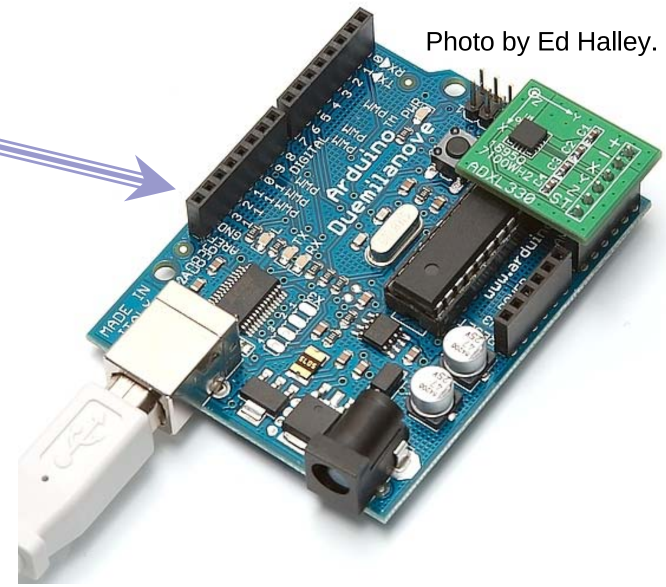


Photo by Ed Halley.



Big, but more reliable

Anemometer

- A small AC generator. The output voltage increases linearly to the wind speed.

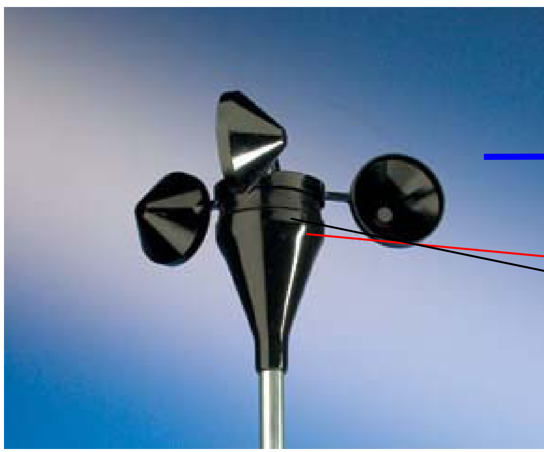
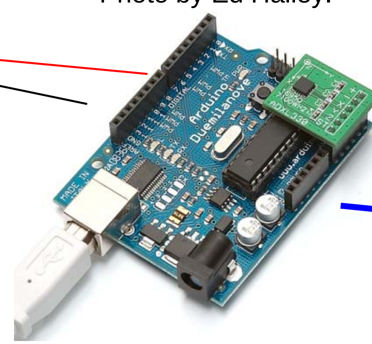
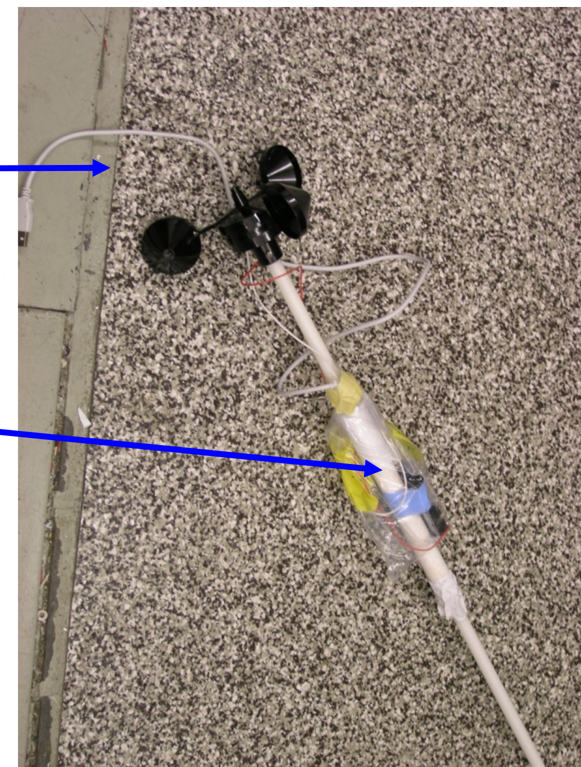


Photo by Ed Halley.



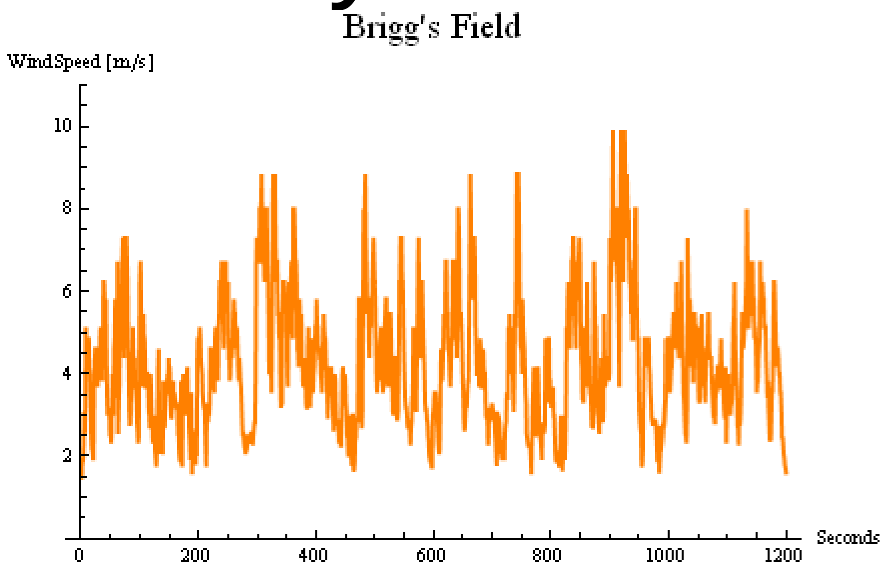


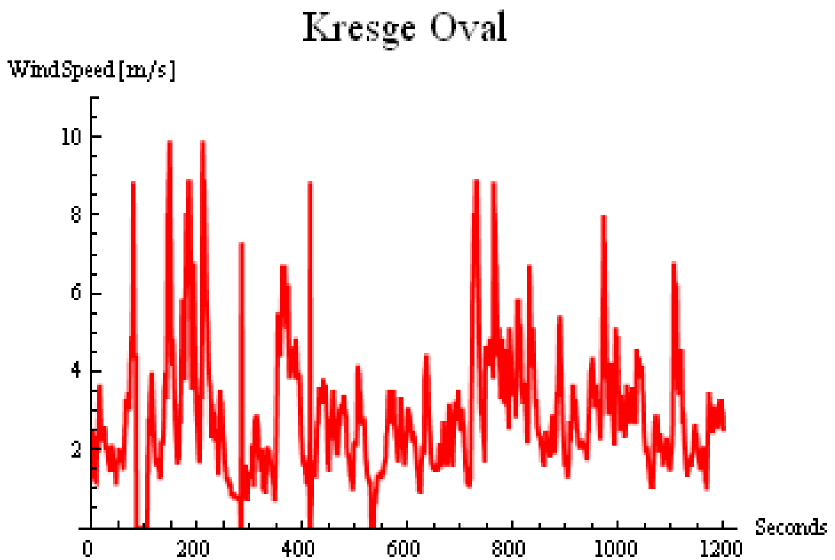
Coded the Arduino so it could read the analog signal

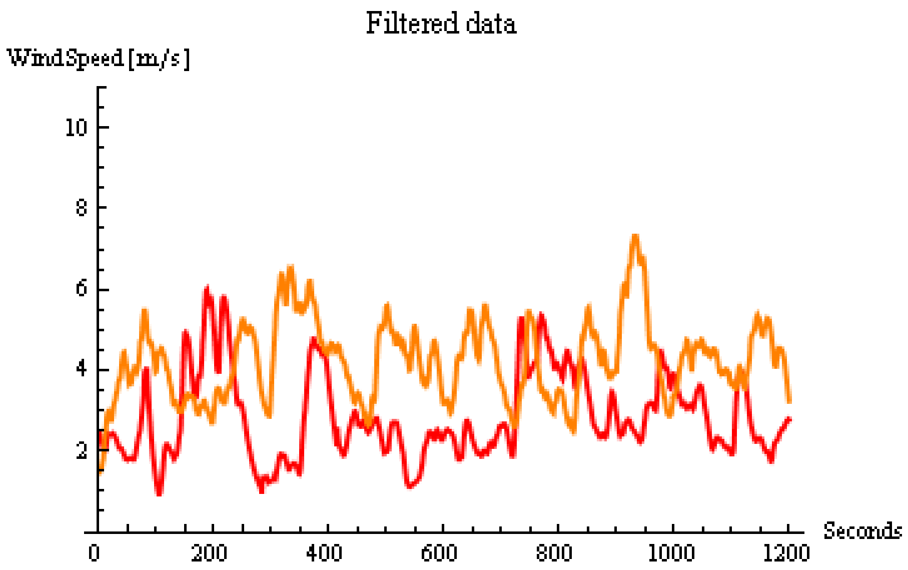
www.nrgsystems.com

www.arduino.cc

Windy Environment

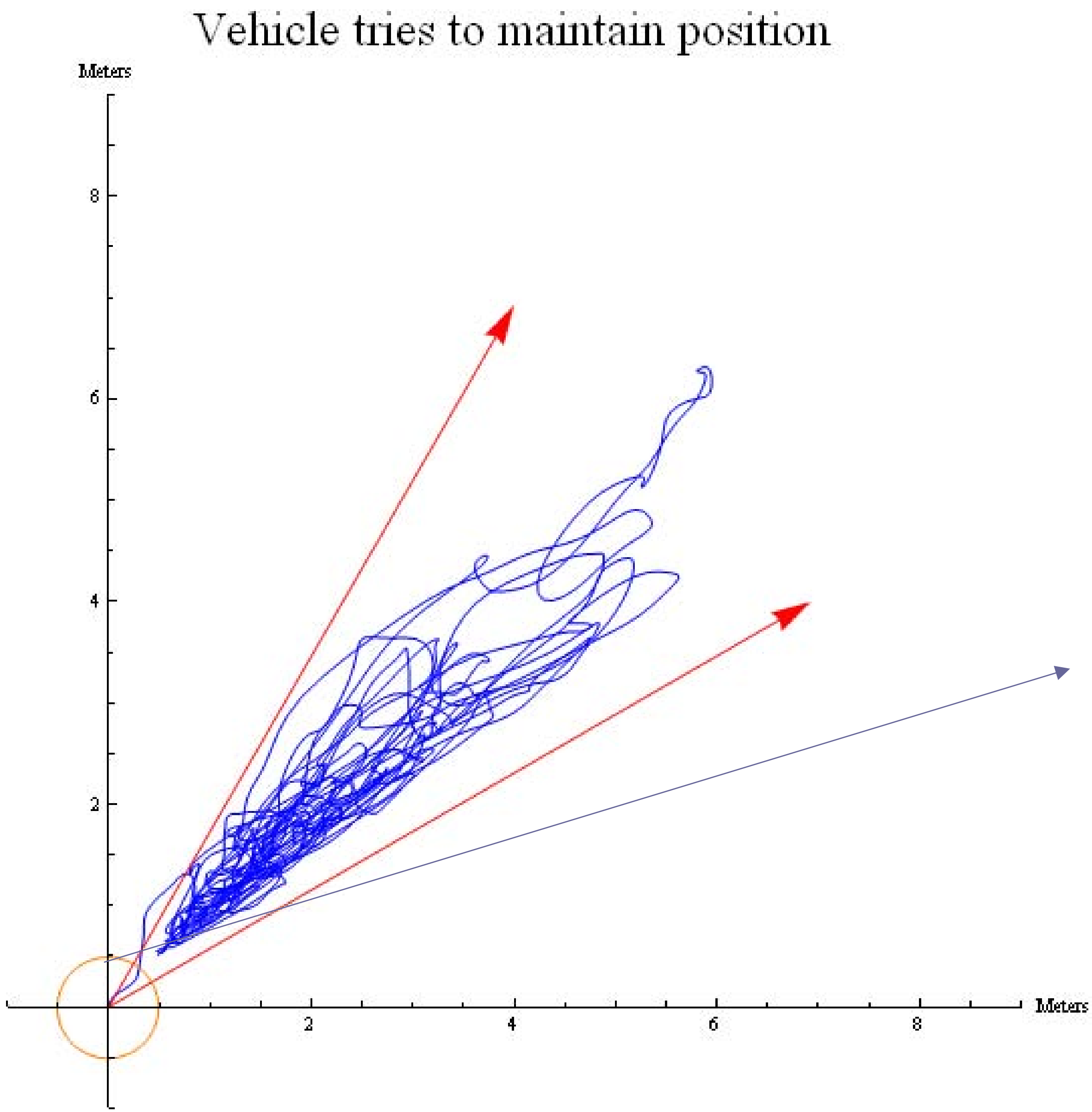






The data was collected on a windy day, within the same hour.

Rough Simulation

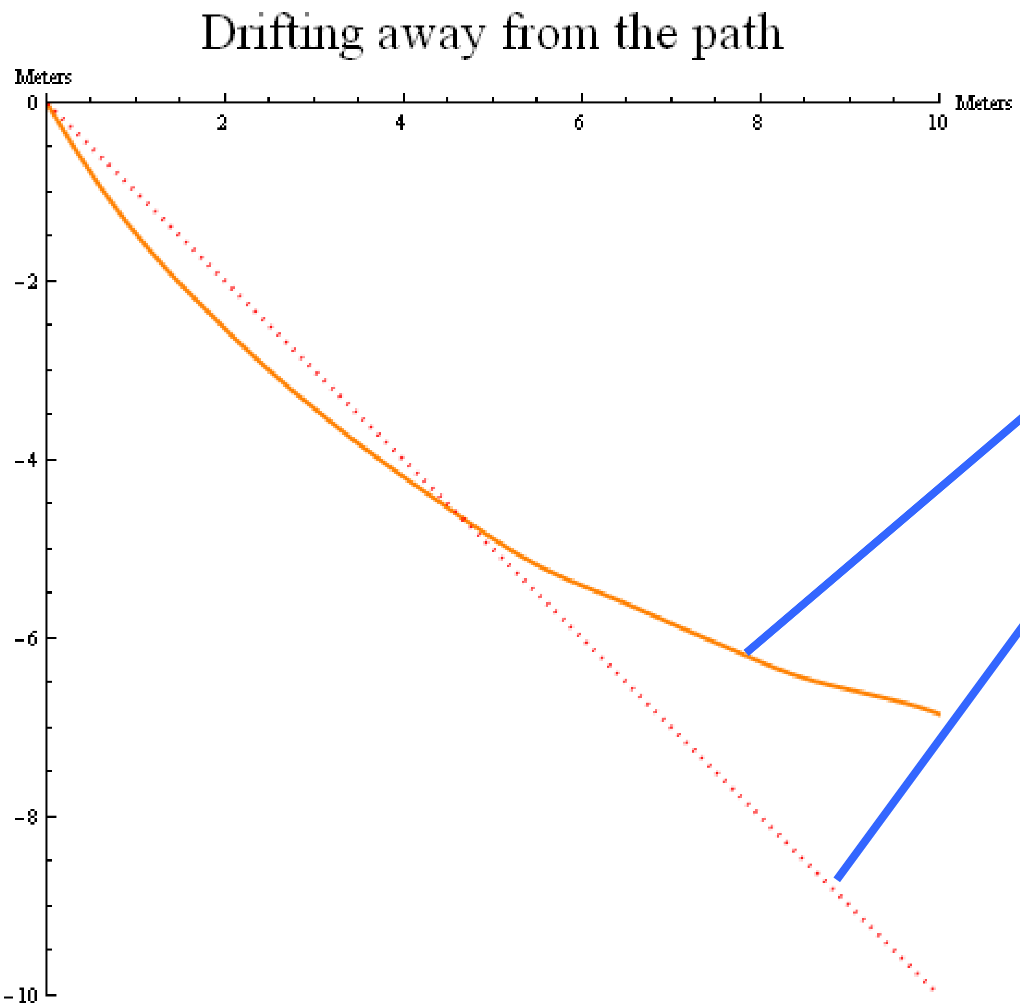


Path of the vehicle

Wind direction is assumed to vary within this angle

Desired Position

Rough Simulation



Command: Move to (10,-10).

Actual path due to wind

Ideal path with no wind

The effect is quite significant –

This should be considered if we want to land on a small target

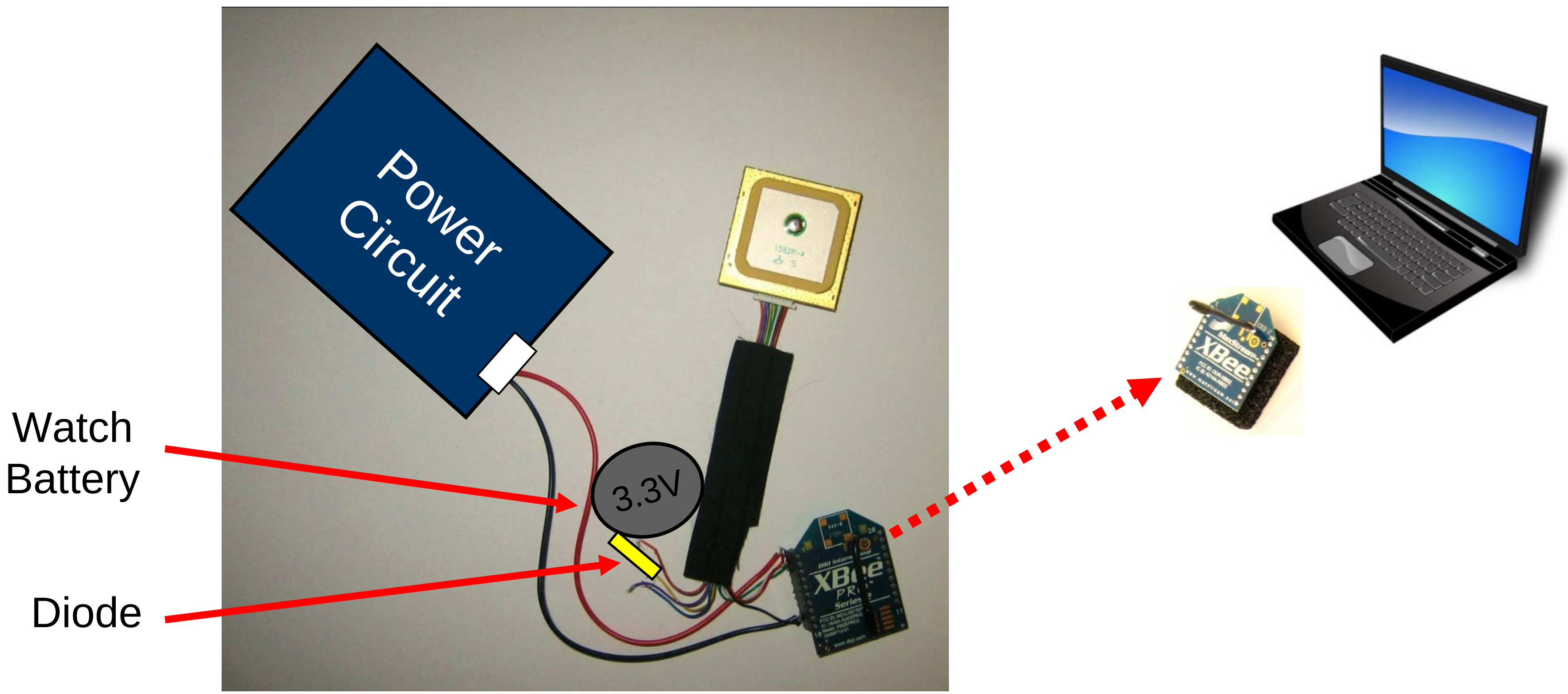
GPS Hardware and Integration

Students C and E

GPS Hardware

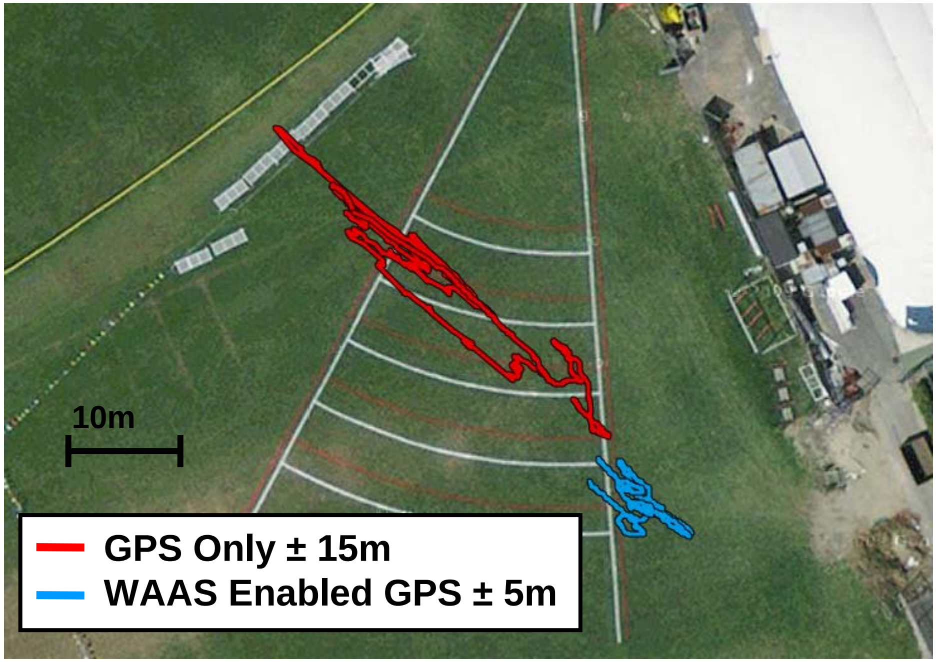
- XBee Communication
- Back-up Battery
  - Retains configuration settings
- Quadrotor Integration
  - 5V from power circuit

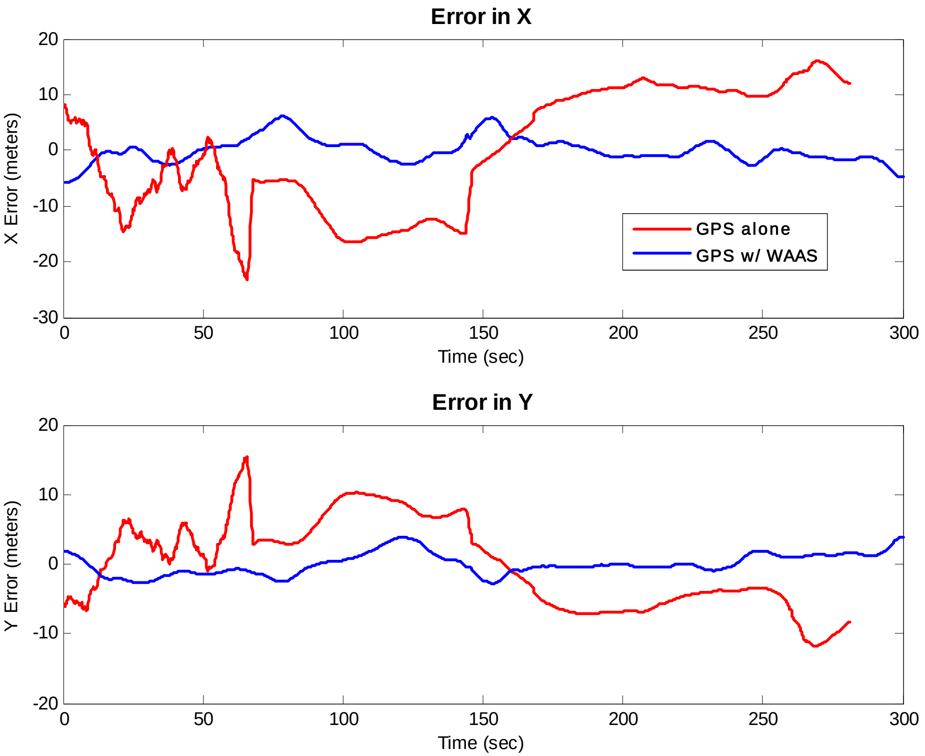
Current GPS Set-up



Images from the [OpenClipArt Library]() and [mangonha]() on Flickr.

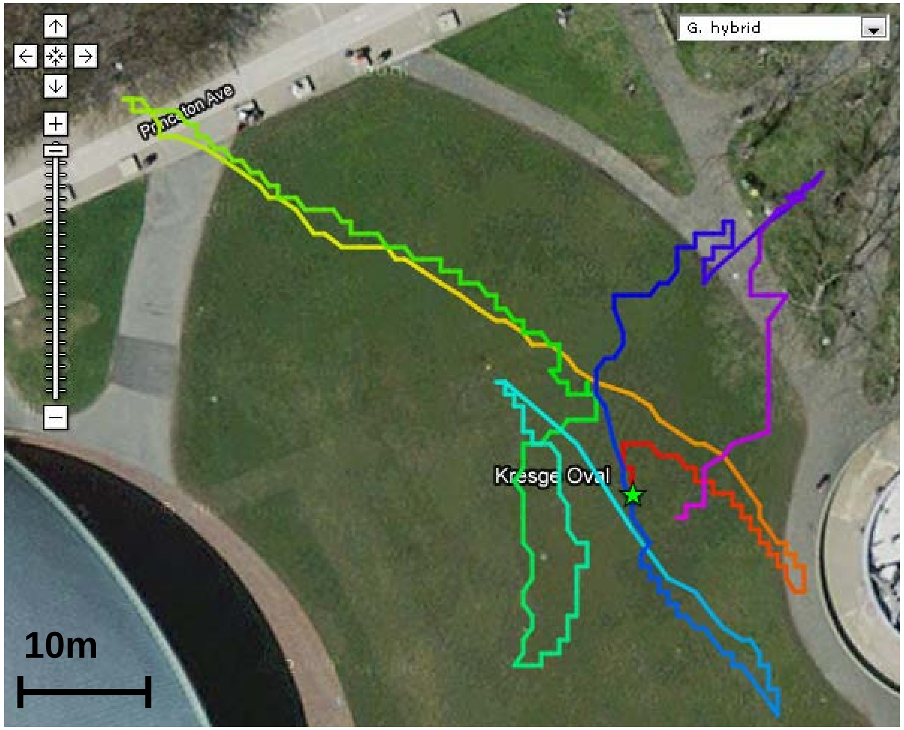
GPS Accuracy Tests (Stationary)

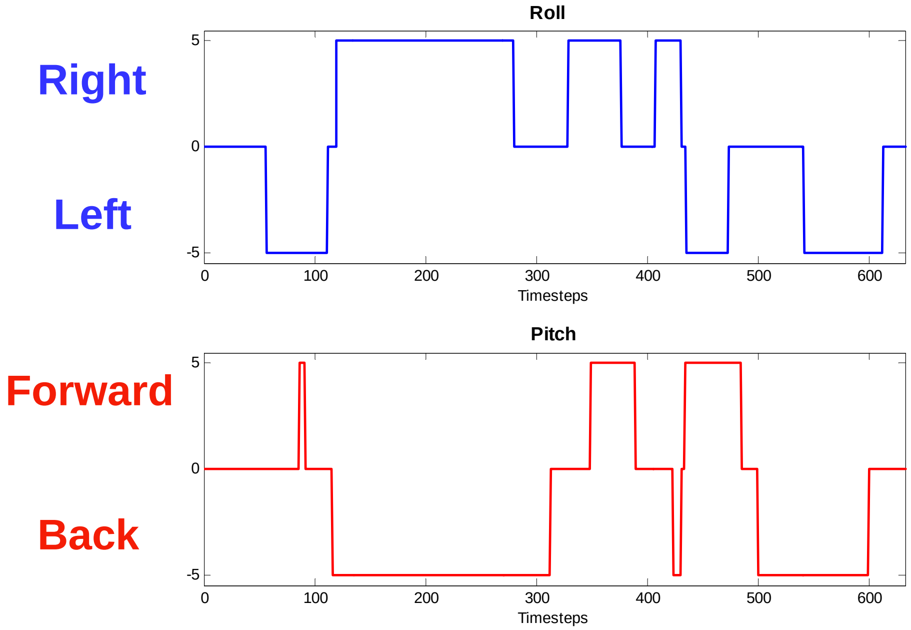




http://www.gpsvisualizer.com/map?output_home

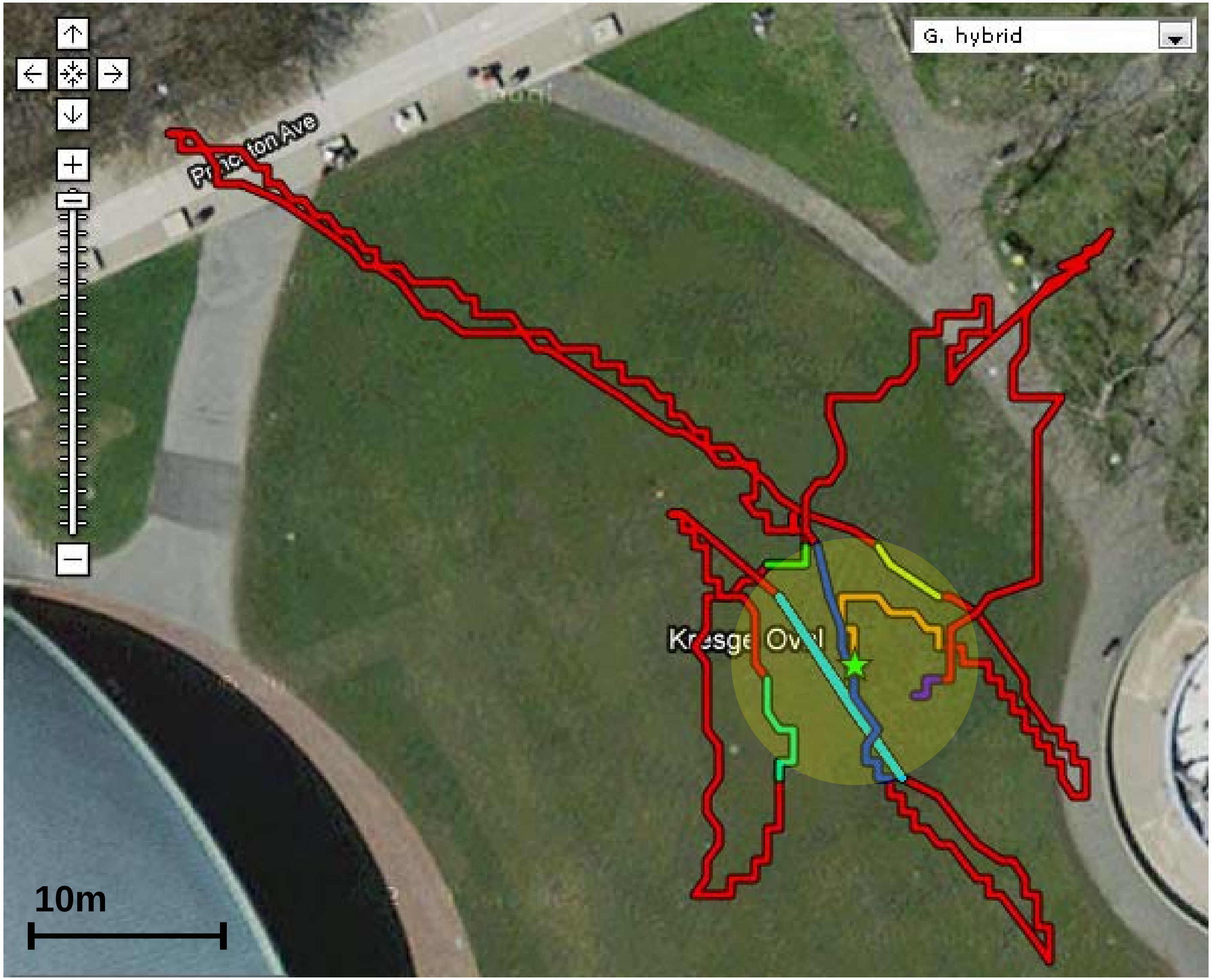
Walk with GPS and Mock Controller





http://www.gpsvisualizer.com/map?output_home

Proposed Region for Switch to Vision Control

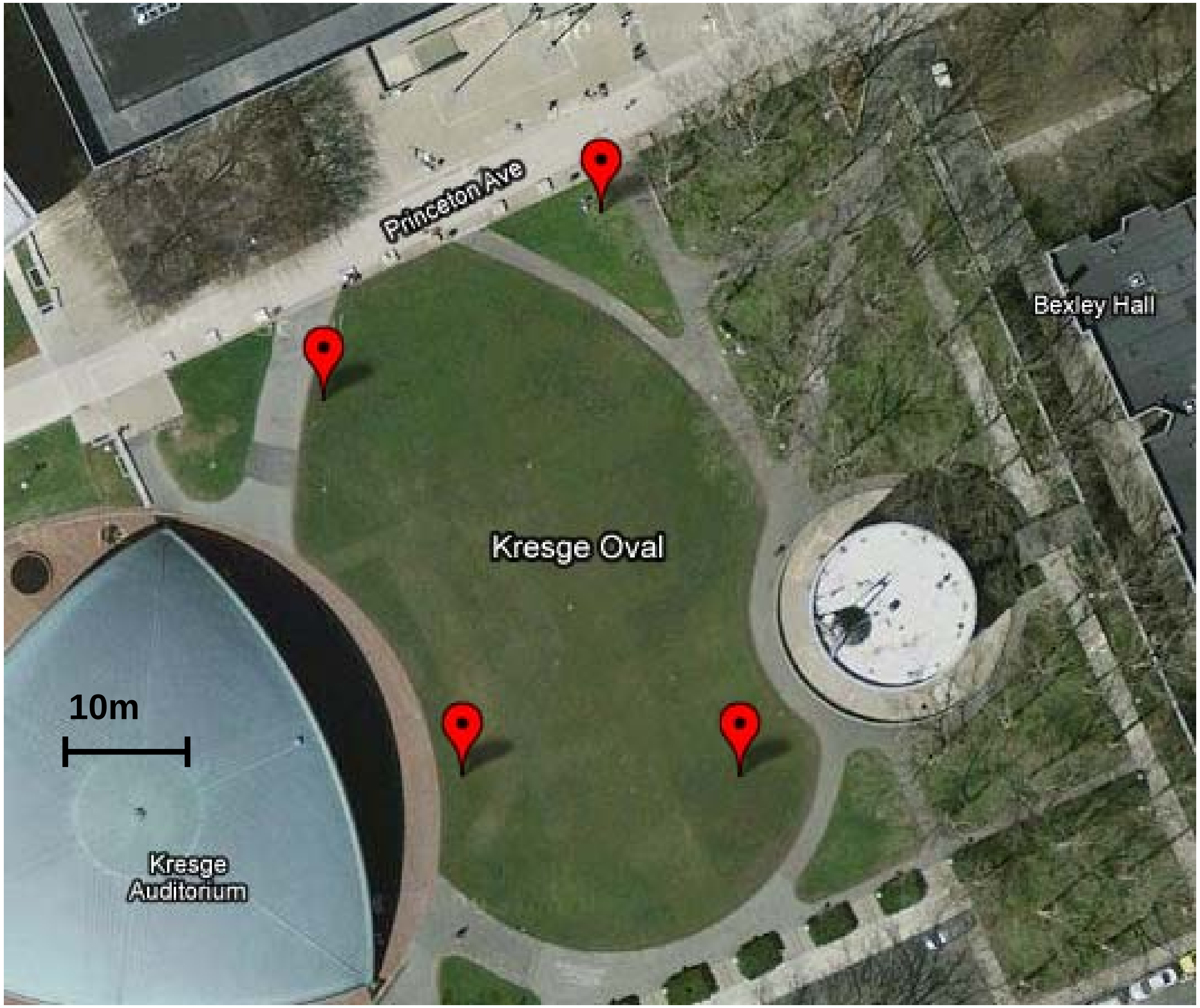


http://www.gpsvisualizer.com/map?output_home

Waypoint Testing

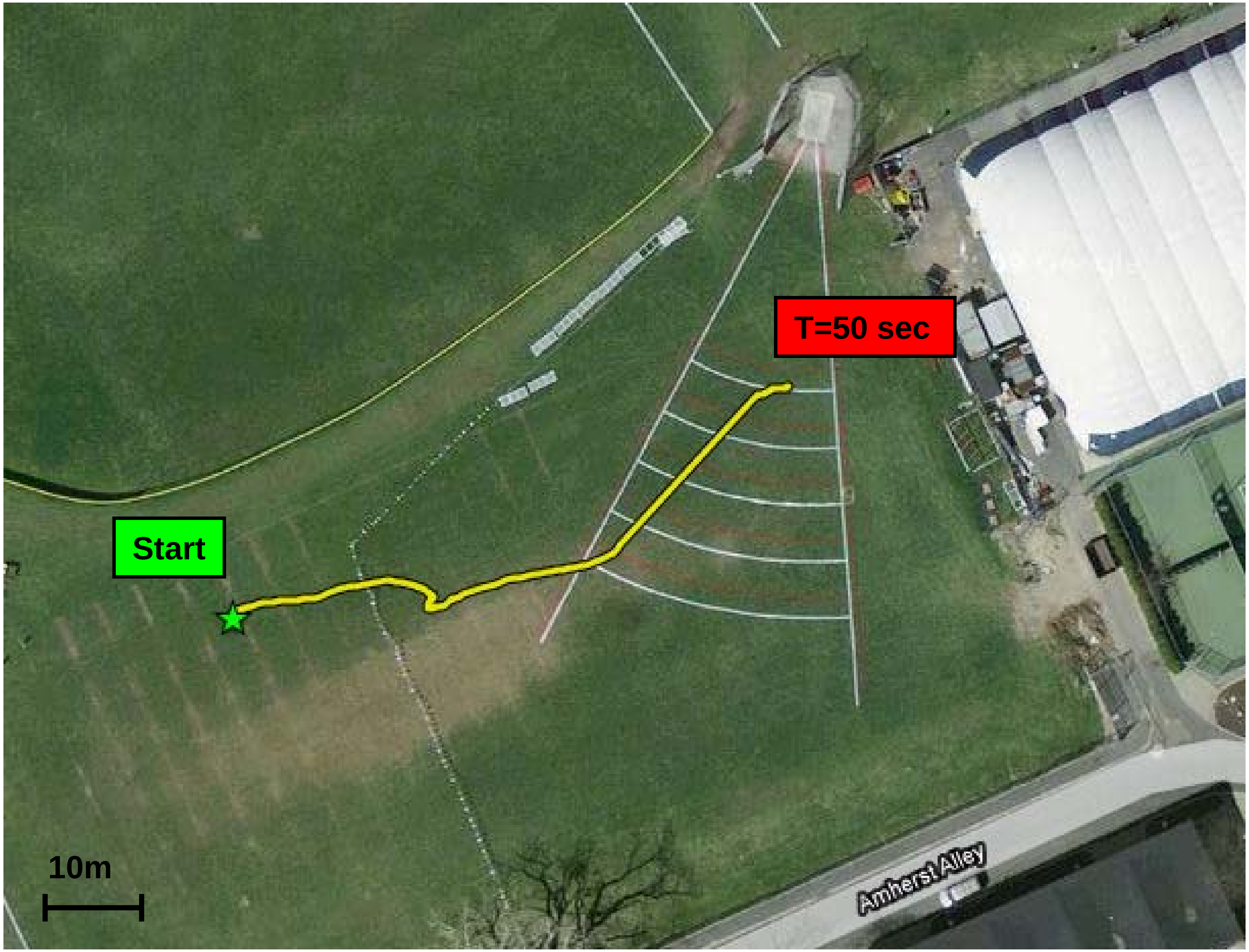
- Combines compass and GPS
  - Adds heading
- Use GPS to pick waypoints
- Walk quadrotor by following commands from controller
- Check for arrival at destinations

Waypoint Testing



http://www.gpsvisualizer.com/map?output_home

GPS on Flying Quadrotor



http://www.gpsvisualizer.com/map?output_home

Accomplished:

- ☑ Read GPS signal through XBee communication
- ☑ Maintain GPS settings using a watch battery
- ☑ Integrate GPS hardware with Quadrotor
- ☑ Transmit GPS signal from Quadrotor
- ☑ Send GPS data to flight controller

The Next Step:

- ❑ Control quadrotor with GPS feedback

Control System

Student B

Deliverables

- ☑ Demonstrate closed loop control on a LTI model of the quadrotor
- ☐ Demonstrate closed loop control of the quadrotor

Control System Design

Strategy: Point and go

- Heading is set initially and is static
- Controlled variables
  - Yaw rate: points at the target
  - Roll: keeps on line to target
  - Pitch: determines speed forward or backwards
  - Thrust: offsets gravity and brings rotor to correct height
- Measured variables
  - Heading: compass
  - Latitude, Longitude positions: GPS
  - Height: internal pressure sensor
- PD control

Model Assumptions

- Linear Time Invariant
- Small angle pitch and roll (less than 5 deg)
- Max, Min thrust = 1.25mg and 0.75mg
- Rate of system: 4 Hz
- Added random noise to position data:
  - +/- 5m Gaussian error in X,Y
  - +/- 1m Gaussian error in height
  - +/- 10 Gaussian error degree for heading

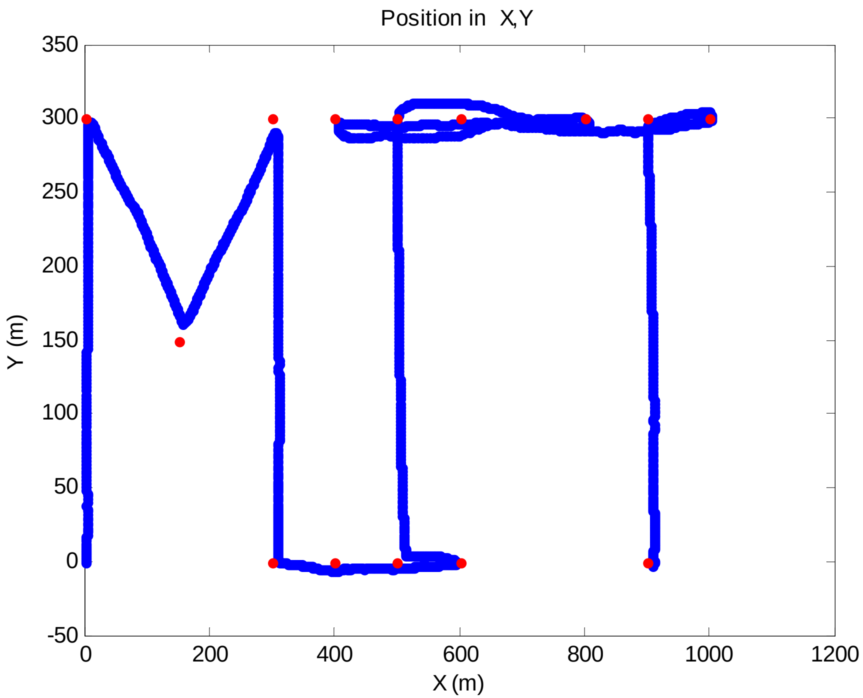
Features

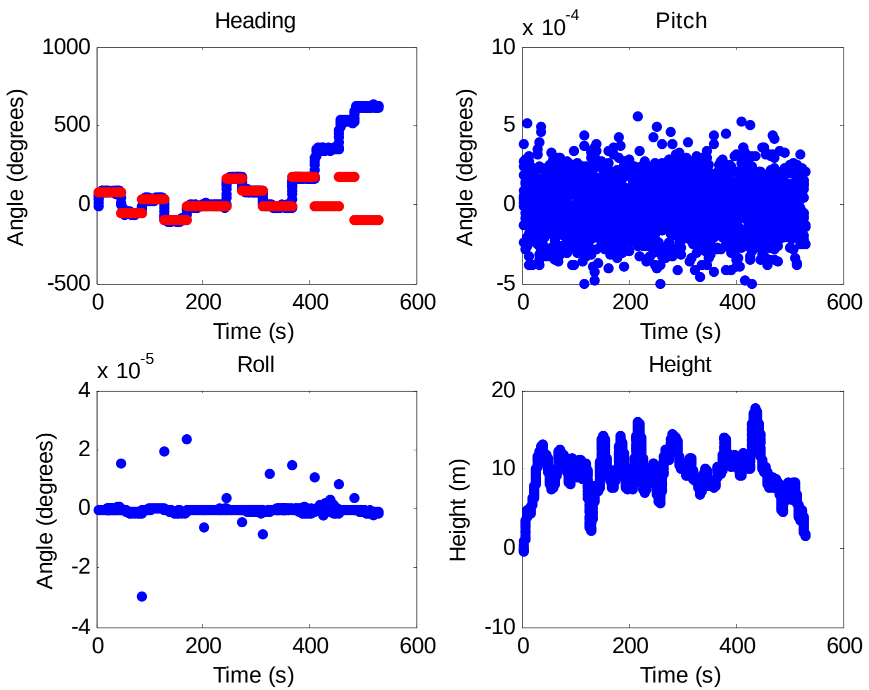
- Simulation mode and communication mode
- Waypoints enabled
- Mid-run user-activated terminate
- Mid-run user-activated hover toggle
- Flight data written to a file
- Recalculates route when overshoots

Future Work

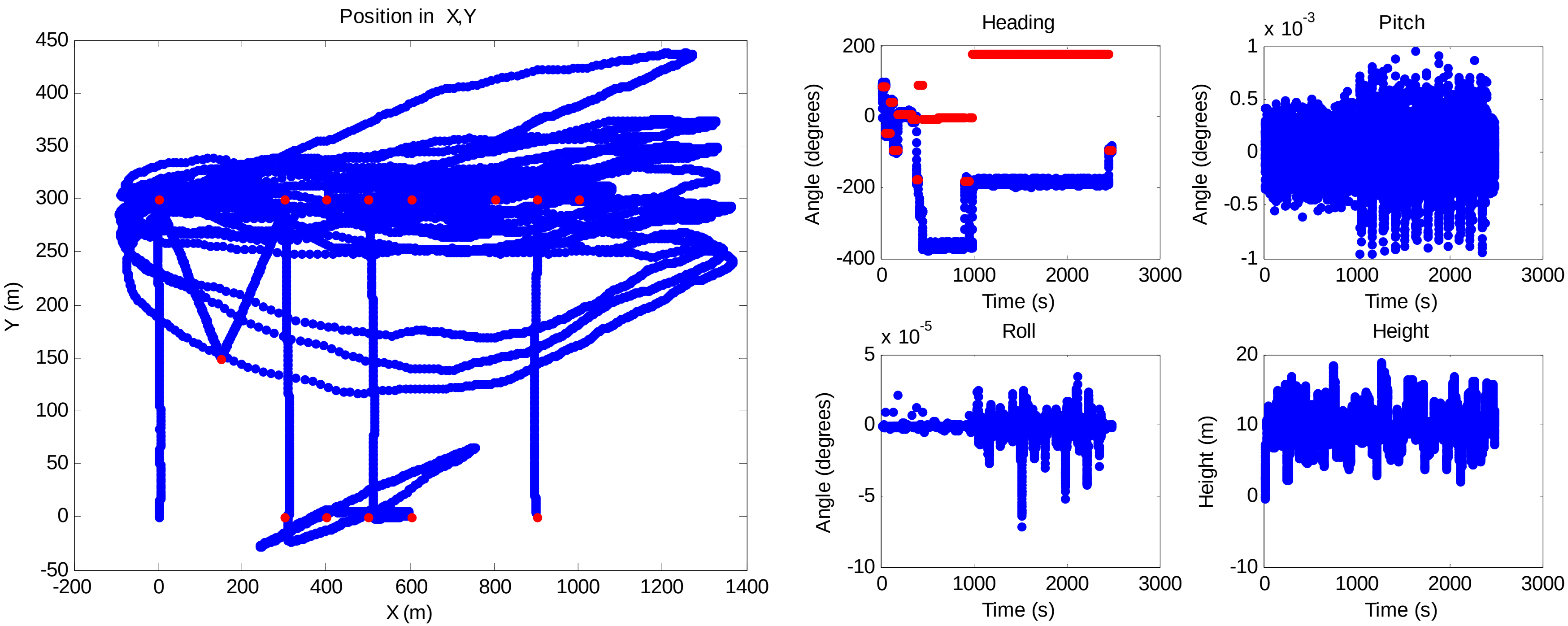
- Design against bad data-packets
- Tune gains for the real quadrotor

Simulator Performance





Simulator Performance



To Do Differently

- Use a Cartesian coordinate control system instead of radial
- Start with a control system that only uses GPS

Yaw Test

Set-up:

- Disturbed Quadrotor manually
- String contributed a restoring force
- Internal controls prevented fast movements

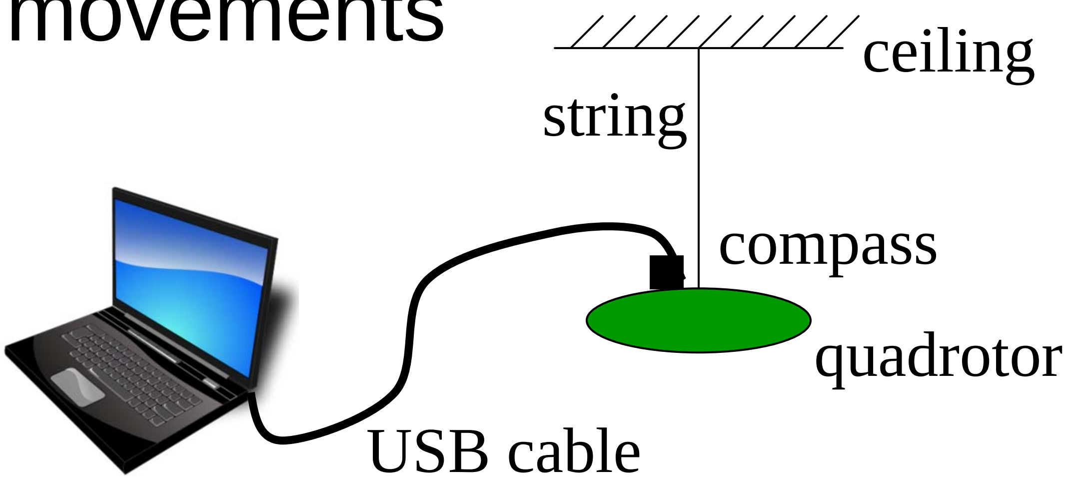
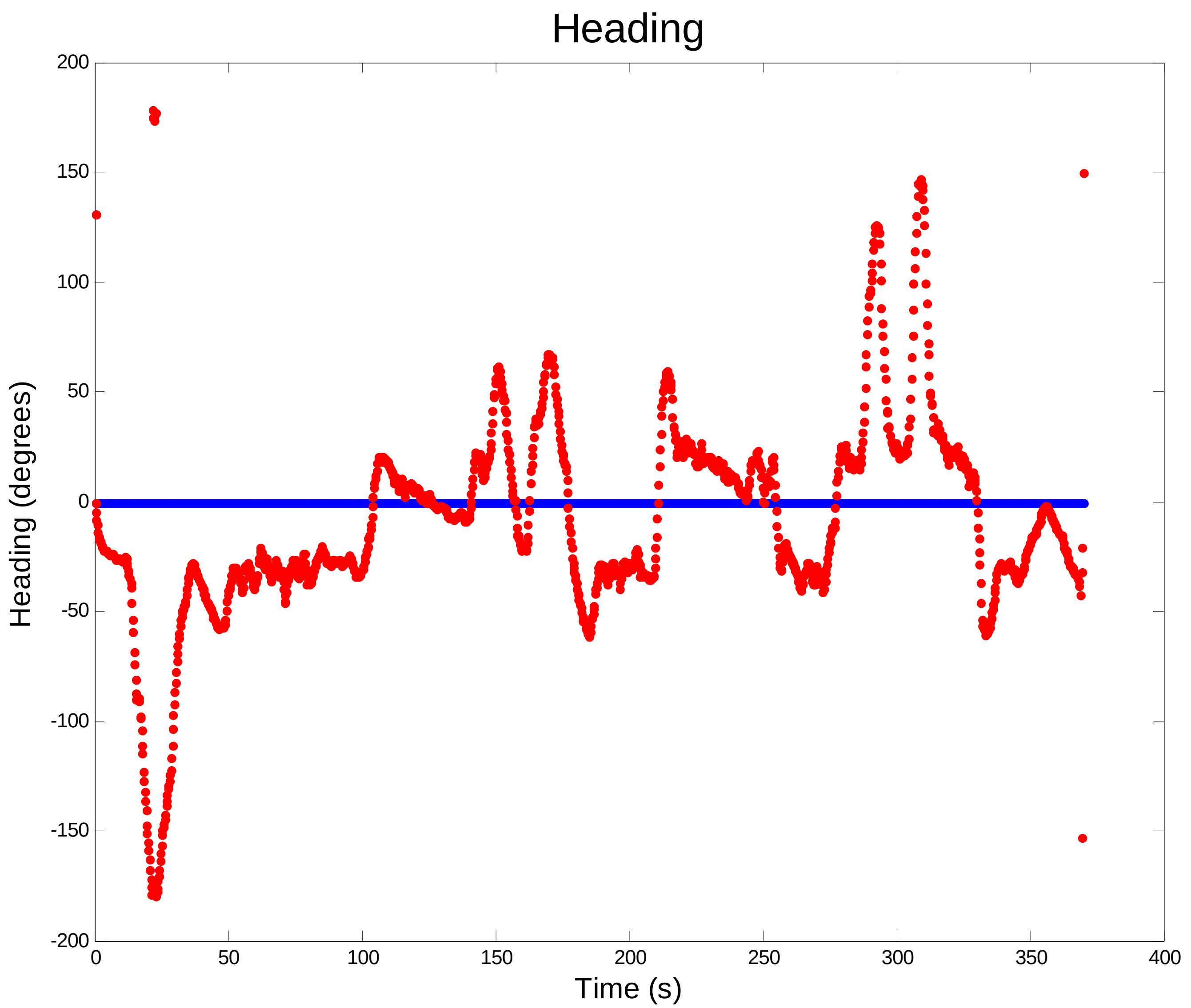


Image from the OpenClipArt Library.

Data



Heading
200
150
100
50
0
-50
-100
-150
-200
Heading (degrees)
0
50
100
150
200
250
300
350
400
Time (s)

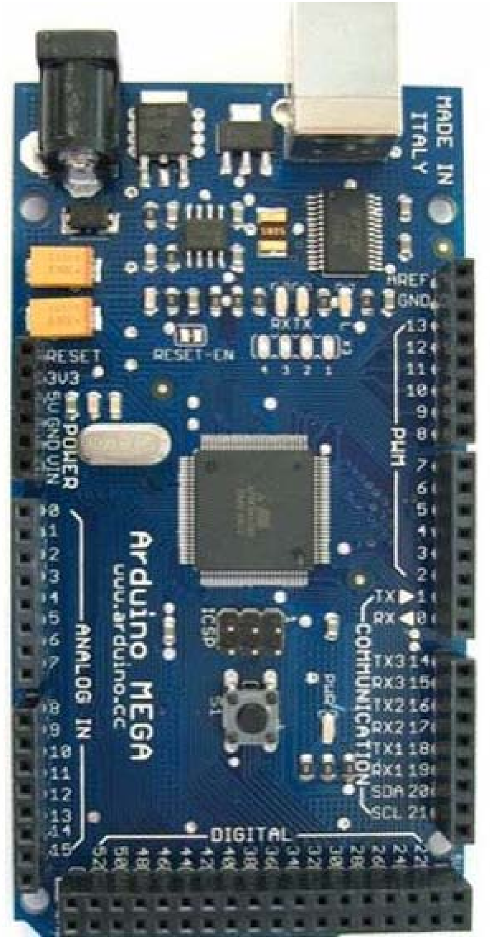
Results

- Steady state error
  - Gains too low
  - Steady disturbance from the string
- 30 second settling time

Arduino Microcontroller and Communication

Student D

Mega Arduino



Courtesy of Arduino.cc. Used with permission.

- On board control
- No need for XBees
- Successful compass communication
- Successful camera communication

Other Work

- Helped with CMUCam communication
- Failed to successfully communicate with quadrotor
- Helped others with programming

Next Steps

- Continue work with Mega Arduino
- Read more about serial communication

What Could've Been Done Differently

- Research more early on
- Better use of available resources (mentors)

CMUCam and Image Processing

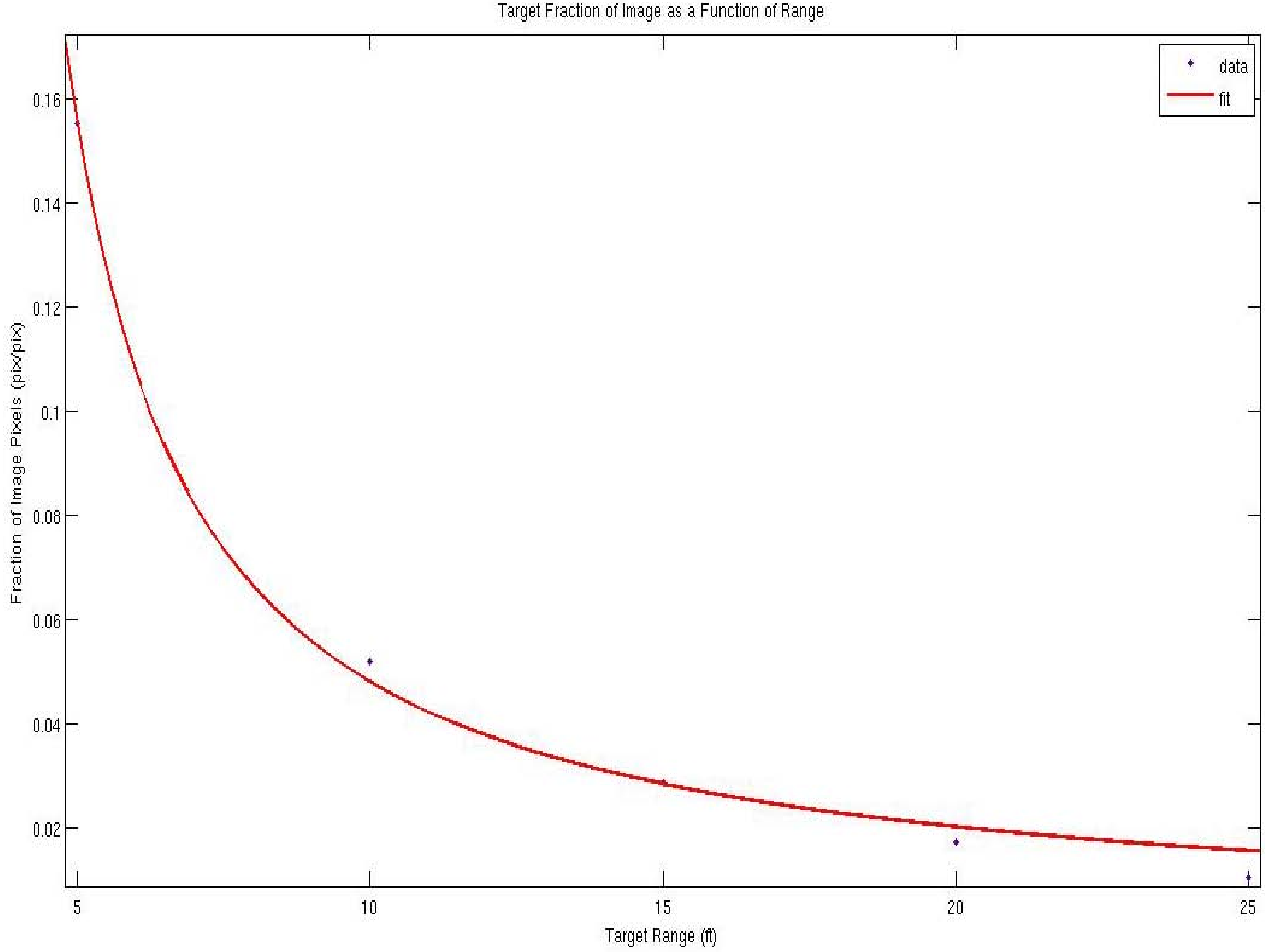
Student F

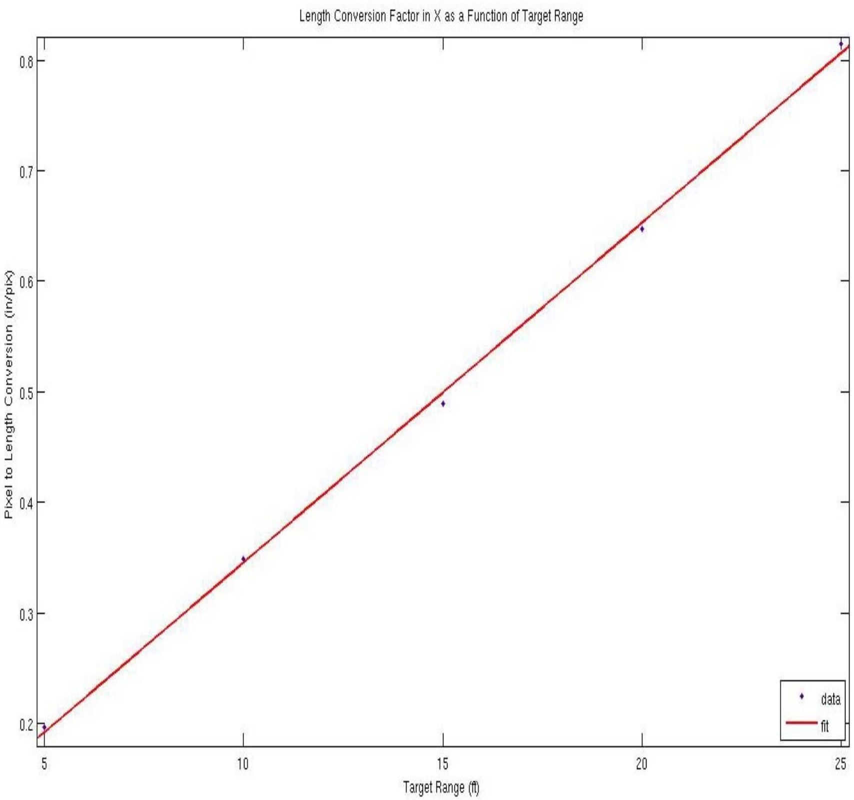
Image Processing Goals

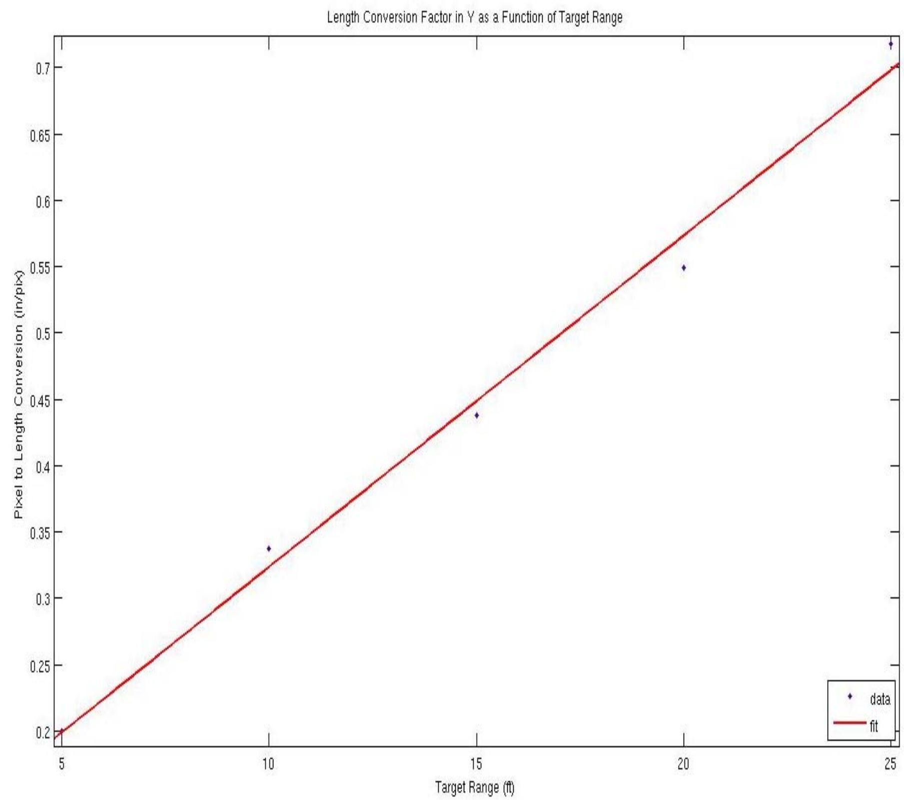
- Take pictures of a predetermined location and also along a reference flight path
- Track a landing target at a known location
- Visually servo to the target using feedback from the image

Finding Distances in X, Y, and Z

- Find target size as fraction of pixels in the image at known ranges
- With a target of known size, we can find a parameter that converts pixels to distance at a known range


Target Fraction of Image as a Function of Range
data
fit
Fraction of Image Pixels (pix/pix)
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
5
10
15
20
25
Target Range (ft)


Length Conversion Factor in X as a Function of Target Range
Pixel to Length Conversion (in/pix)
0.2
0.3
0.4
0.5
0.6
0.7
0.8
5
10
15
20
25
Target Range (ft)
data
fit


Length Conversion Factor in Y as a Function of Target Range
Pixel to Length Conversion (in/pix)
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
5
10
15
20
25
Target Range (ft)
data
fit

T Packets

Centroid of
tracked data

Bounding box
coordinates

T 84 132 4 1 172 250 255 12

Indicates a color
tracking data packet

Number of pixels that match
the tracked color

Confidence

What Went Wrong

- Neither CMUCam is ideal for our mission
- Wireless communication never really worked
  - XBee drops too many packets
- GPS waypoint tracking did not work
- Ran out of time

What We Could Have Done Differently

- Problem
  - Worked independently—not the most effective
- Solution
  - Weekly group meetings
- Problem
  - Strategy depended on all hardware components working
- Solution
  - Design a simpler, more independent system
- Problem
  - Inexperience
- Solution
  - Take more advantage of our resources

Questions?