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

- Main Battery
- Compass
- Compass XBee
- GPS
- GPS XBee
- Quadrotor XBee
- Power Circuit
Electrical and Signal Flow Schematic

12V Battery → Quadrotor → XBee

Compass → inv → XBee

GPS → XBee

Power Circuit → 9V Battery

9V Battery → XBee
Block Diagram

Desired Position + Error → Controller → Commands → Quadrotor → Actual Position

- Observed Position → Sensors → -

Disturbances
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

Sends data to the laptop via XBee.

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

5 V out

3 V out

12 V in

Big, but more reliable

Photo by Ed Halley.
Anemometer

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

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

Vehicle tries to maintain position

Path of the vehicle

Wind direction is assumed to vary within this angle

Desired Position
Rough Simulation

Command: Move to (10,-10).

Drifting away from the path

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)

- GPS Only ± 15m
- WAAS Enabled GPS ± 5m

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

Time (s)

Heading (degrees)
Results

- Steady state error
  - Gains too low
  - Steady disturbance from the string
- 30 second settling time
Arduino Microcontroller and Communication

Student D
Mega Arduino

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

Courtesy of Arduino.cc. Used with permission.
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
T Packets

- Centroid of tracked data
- Bounding box coordinates
- Indicates a color tracking data packet
- Number of pixels that match the tracked color
- Confidence

T 84 132 4 1 172 250 255 12
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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worked independently—not the most effective</td>
<td>Weekly group meetings</td>
</tr>
<tr>
<td>Strategy depended on all hardware components working</td>
<td>Design a simpler, more independent system</td>
</tr>
<tr>
<td>Inexperience</td>
<td>Take more advantage of our resources</td>
</tr>
</tbody>
</table>
Questions?