

Autonomous Navigation, Position Control, and Landing of a Quadrotor Using GPS and Vision

Group 1

Photo of the Ascending Technologies [Hummingbird Autopilot Quadcopter](#)
removed due to copyright restrictions.

September 29, 2009

2.017 Design of Electromechanical Robotic Systems
MIT Department of Mechanical Engineering

I. Objectives

The purpose of this project is to design and implement an integrated navigation system for the Ascending Technologies *Hummingbird* quadrotor that enables it to autonomously hold a position, fly through a fixed course, take pictures, and land on a marked location.

II. Project Description and Proposed Approach

A. Background

One of the biggest challenges facing unmanned aerial vehicles (UAVs) today is landing safely without human control. The ability to make precision landings is important for designing aircrafts that can inspect small targets or that need to re-fuel automatically. An error range of a few meters is unacceptable when landing a high speed vehicle. Generally, UAVs rely on a Global Positioning System (GPS) for movement since they tend to fly over obstacles and observe from a distance. However, UAVs designed for use in more claustrophobic environments (i.e. urban, forest) need a more reliable source of relative positioning information. Combining GPS with another type of control such as camera vision would provide UAVs with this extra precision.

B. Goals and Functional Requirements

There are four final goals of this mission: maintain a given position, navigate through a preset course, take pictures, and land on a marked target. Ultimately, we will collect GPS data during each part of the mission and plot them on three-dimensional coordinate axes.

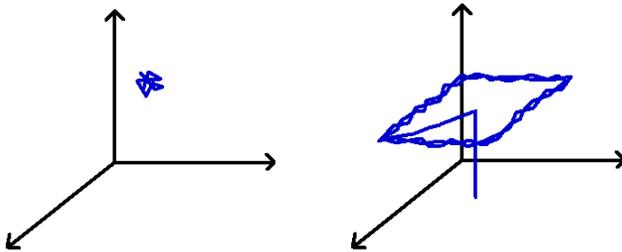
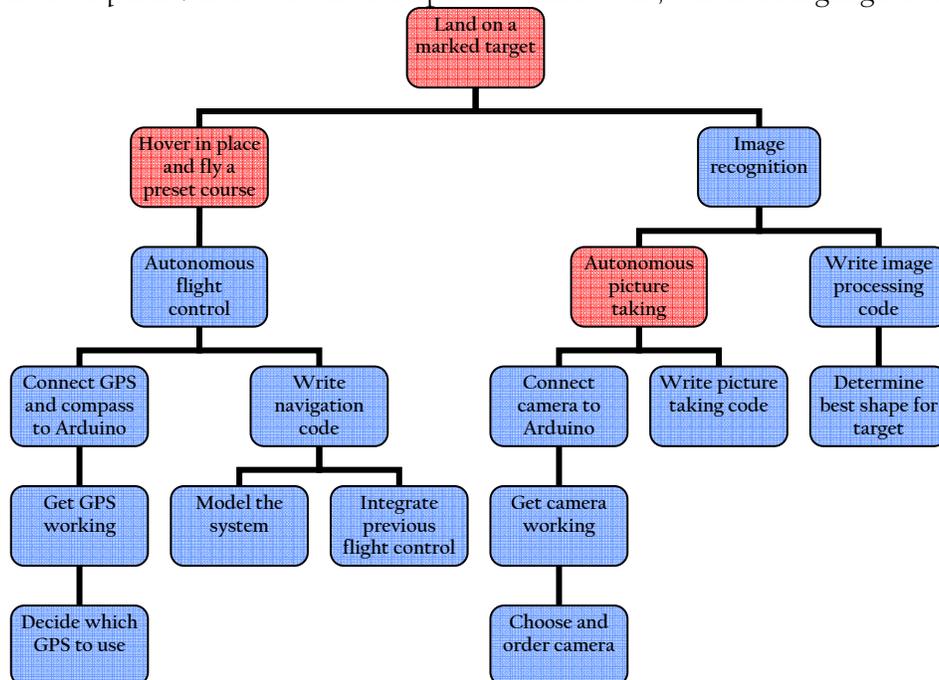


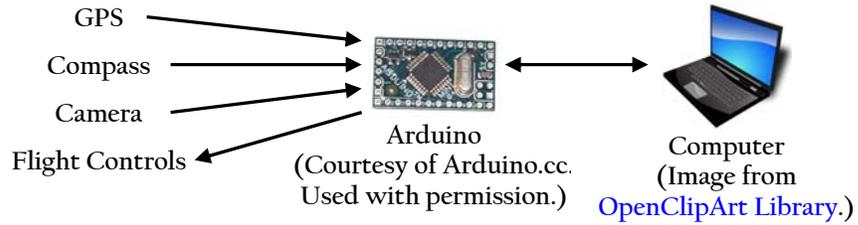
Figure A: The location of the quadrotor hovering in place.

Figure B: The path of the quadrotor following a preset square pattern, centering itself over a marked target to take pictures, and making a precision landing.

Our projected plan can be summarized in an “objectives tree” as seen below. Many functional requirements such as getting the sensors working and writing programming code to implement them into the quadrotor system are required before we can accomplish the final tasks, which are highlighted in red.

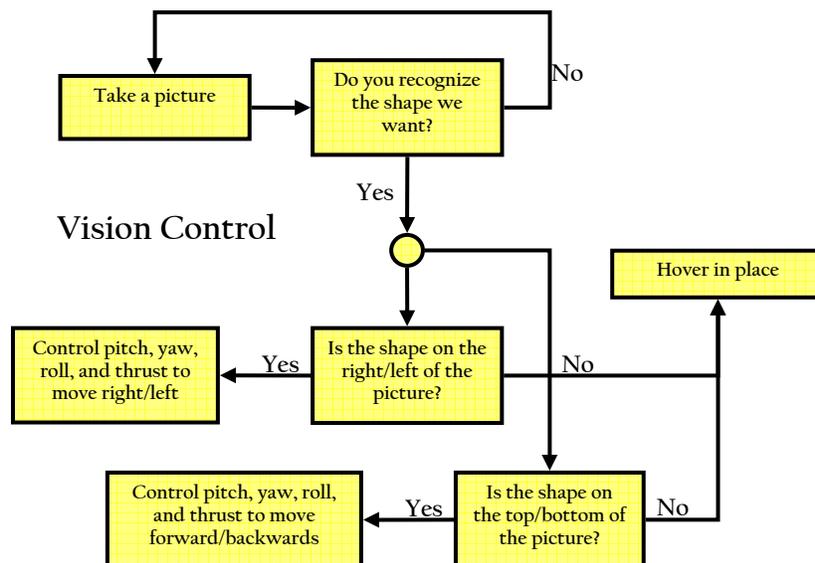
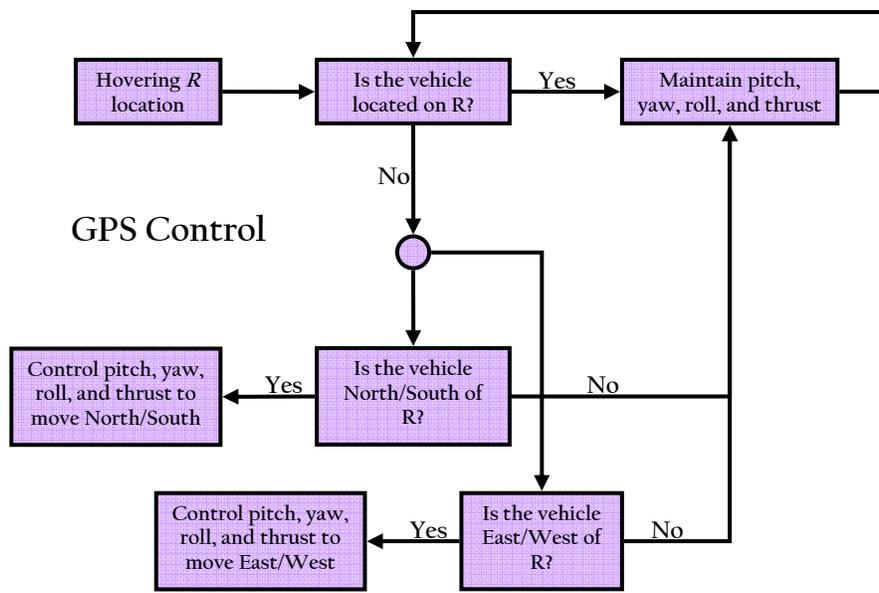


All of the quadrotor's sensors currently have separate connections that communicate with the computer. One of our goals for this project is to link the sensors through an Arduino board, which will then communicate with the computer. This will allow us to have a quadrotor that is completely autonomous and capable of flying on its own.



C. Functional Block Diagrams

Two of the most important aspects of this mission are the programs to control the flight of the quadrotor using GPS coordinates and using images taken from the camera. The following two diagrams show the steps we will take to write these codes and how we can use them to control the movement of the quadrotor.



III. Project Plan

A. Tasks

Although everyone expressed interest in being involved with multiple areas of the project, we decided that it would be beneficial to have one main person responsible for each section to guarantee that each task gets accomplished in the end. We will form subgroups on a week-by-week basis.

Software:

Student A will combine the previous flight control code with the on-board sensors' code and generate a new navigation code. He will also brief the rest of the group on the thoughts behind the code to avoid conceptual errors.

Modeling and Controls:

Student B will design the controls of the new navigation system by choosing a good feedback controller based on our system's models.

Logistics and Communications:

Student C will make sure the team stays on schedule and that the different subgroups hardware and software will not conflict or prevent the others from working.

Arduino:

Student D will make sure that the sensors ultimately get linked to the computer through an Arduino rather than through separate serial links.

GPS and Additional Sensors:

Student E will test the GPS and the additional sensors (i.e. compass, pressure sensor) and implement it into our navigation system.

Camera and Image Processing:

Student F is responsible for getting the camera to take pictures and working with the code so that the camera can recognize an object and determine the quadrotor's relative location to that object.

B. Model

We already have a preliminary model for the dynamics and kinematics of the quadrotor system. We found six partial differential equations to describe the motion of the aircraft.

Assumptions:

1. The center of mass is located where the arms meet
2. The four thrusts are always parallel
3. "Flapping" does not occur in the rotor blades
4. The vehicle is rigid
5. All arms are the same length
6. Ignore drag

ψ : Yaw	ϕ : Roll	F : Thrust force
θ : Pitch	M : Moment	l : Arm length

$$\ddot{X}_{\square} = \frac{\left(\sum_{i=1}^4 F_i\right) \cdot (\cos \phi \cdot \sin \theta \cdot \cos \psi + \sin \phi \cdot \sin \psi)}{m \square}$$

$$\ddot{Y}_{\square} = \frac{\left(\sum_{i=1}^4 F_i\right) \cdot (\sin \phi \cdot \sin \theta \cdot \cos \psi - \cos \phi \cdot \sin \psi)}{m \square}$$

$$\ddot{Z}_{\square} = \frac{\left(\sum_{i=1}^4 F_i\right) \cdot (\cos \phi \cdot \cos \psi) - mg \square}{m \square}$$

$$\ddot{\theta} = \frac{l \cdot (-F_1 - F_2 + F_3 - F_4)}{I_{x \square}}$$

$$\ddot{\psi} = \frac{l \cdot (-F_1 + F_2 + F_3 - F_4)}{I_{y \square}}$$

$$\ddot{\phi} = \frac{(M_1 - M_2 + M_3 - M_4)}{I_{z \square}}$$

C. Required Sensors, Equipment, and Resources

CMUCam2 with OV7620 Camera Module

The CMUCam is both a camera and image processor built into one compact device, which is ideal for this project because we need to analyze pictures taken during flight to tell the quadrotor where to move next. Having a built-in image processor allows us to skip the extra step of sending data back to a computer for analysis.

Arduino Mini Board

Although the *Hummingbird* quadrotor is very robust and can carry a fairly heavy payload, it would still be helpful to use a smaller Arduino and leave space for additional sensors. The smaller Arduino should still have enough processing power for this mission. In the chance that it is not enough, we can switch back to the Arduino we have already been working with.

D. Risks and Prevention

Even though the blades of the quadrotor are designed to be “flexible and harmless,” we will only fly in open, uncrowded areas and implement an emergency landing mechanism to avoid any possible injury to team members, bystanders, or the system itself. The quadrotor will also be tethered to the ground to prevent it from running into obstacles as well as prevent it from leaving the designated flying area.

E. Backup Plan

For most hardware failures, we can replace the damaged parts or, worst case scenario, salvage the remaining parts. In the event of a catastrophic failure, where the quadrotor becomes damaged beyond repair or we are unable to retrieve it, we can switch to using the regular helicopter. It would set us back a great deal because it would introduce the additional obstacles of decreased stability and lower payload, but we would still be able to continue the mission.

F. Schedule

Week	Date	Objective
1	Oct. 5	- Read through and understand previous flight control code - Test previous flight control code with the quadrotor - Learn how to fly the quadrotor safely
2	Oct. 12	- Experiment with camera, GPS, and their codes
3	Oct. 19	- Get the quadrotor flying autonomously (dead reckoning)
4	Oct. 26	- Get the quadrotor flying autonomously with GPS - Take pictures during flight
5	Nov. 2	- Get the quadrotor to hover autonomously - Experiment with image recognition
	Nov. 5	Milestone Presentations *Have pictures taken from helicopter
6	Nov. 9	- Implement vision controlled autonomous flight - Use vision to center the quadrotor over a marked target
7	Nov. 16	- Improve the accuracy of the quadrotor's positioning
8	Nov. 23	- Get the quadrotor to land on a marked target
8	Nov. 30	- Perform final tests - Last minute debugging
10	Dec. 7	- Analyze data - Work on PowerPoint for final presentation
	Dec. 10	Final presentations

MIT OpenCourseWare
<http://ocw.mit.edu>

2.017J Design of Electromechanical Robotic Systems
Fall 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.