AE 483
UAV Navigation and Control
Lab Manual

Lab 1

Vehicle Kinematics and Data Collection

1.1 Objectives

This lab is an introduction to working with the Hummingbird quadrotors. In this lab, you will:

- Program the Hummingbird quadrotors to collect onboard sensor data during flight.
- Understand how to communicate with the quadrotor over Xbee.
- Visualize collected flight data from both a world and body frame point-of-view by replaying the flight using animation.

1.2 Tasks

1.2.1 Program the Hummingbird quadrotors

The goal is to program the hummingbird quadrotor to transmit onboard sensor data during a short manual flight. This onboard sensor data will be collected in parallel with the motion capture system. In particular, you will collect the following sensor measurements:

- onboard sensors: 3-axis accelerometers, gyros, and magnetometer (compass) measure the following states of the vehicle p, q, r, a_x, a_y, a_z, where p, q, r are angular velocities in the body frame and a_x, a_y, a_z are accelerations in the body frame.
- motion capture: the motion capture system measures the following states of the rigid body defined by a set of motion capture markers: $x, y, z, \phi, \theta, \psi$, where x, y, z are positions of the centroid of the markers in the motion

capture world frame, and ϕ, θ, ψ are ZYX Euler angles specifying the orientation of the frame attached to the set of motion capture markers.

1.2.2 Communicate over Xbee

The goal of this task is to understand how data is sent to and from the quadrotor. In particular, Ascending Technologies has created a GUI to use with their AscTec Communication Interface which they call ACI Tool. You will read data from the IMU using Ascending Technologies' ACI tool in order to be introduced to this communication interface.

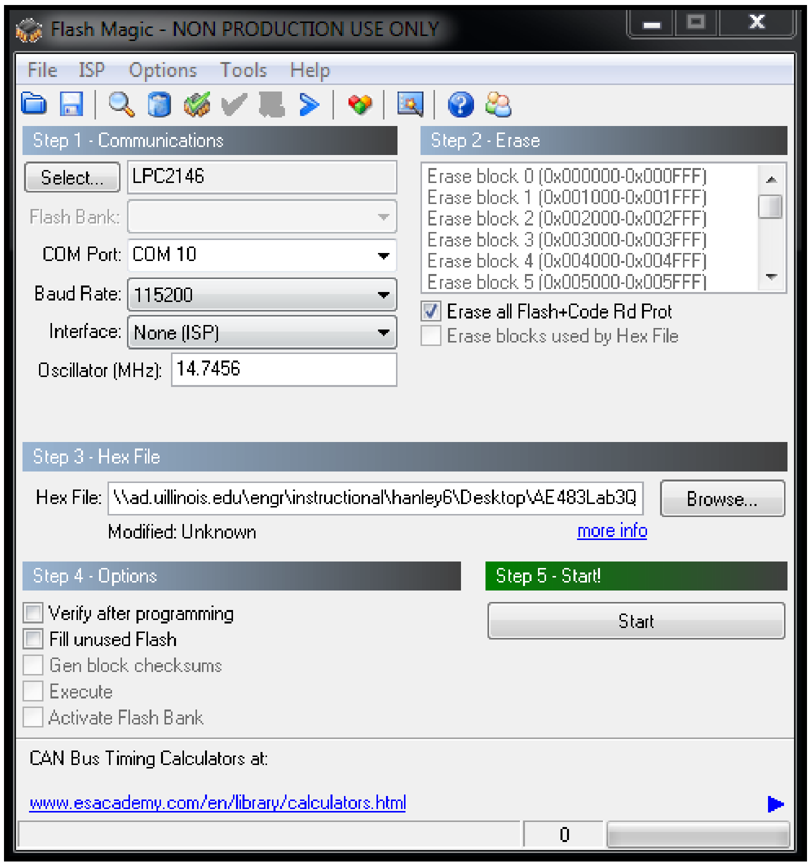
1.2.3 Visualize Flight Data

The goal of this task is to use your knowledge of rigid body transformations to visualize the flight data the you just collected. In particular, you will visualize the quadrotor's position and orientation with respect to the world frame, and visualize objects in the world from the point-of-view of the quadrotor. You will also integrate acceleration and angular velocity measurements to reconstruct position and orientation over time, and compare to measurements provided by the vehicle inertial measurement unit (IMU) and motion capture system.

1.3 Procedure

1.3.1 Program the Hummingbird quadrotor

1. Check that you have the AscTec SDK Eclipse on your machine. Download the `AscTec_SDK_v3.0_Lab1.zip, AE483GroundStation_Lab1.zip` and `aci-tool-windows.zip` from the course website. Run the AscTec Eclipse application from the start menu: Start, AscTec ARM SDK, AscTec ARM SDK (Eclipse). Create a workspace under your EWS home directory, for example, `U:\Documents\workspace\`. Import `AscTec_SDK_v3.0_Lab1.zip`: In Eclipse, select "File → Import → General → Existing Projects into Workspace → Select Archive file". Then browse to find `AscTec_SDK_v3.0_Lab1.zip` and and make sure that the button "copy projects into workspace" is selected, then click "Finish".
2. Configure the build targets. In Eclipse select "Project → Clean Project". Make sure there are no errors at the Console window in Eclipse. If there were no errors, you should see a main.hex file in your workspace folder.
3. Next, you have to flash the main.hex file on to the vehicle. Turn the vehicle on in programing mode and connect a USB cable to the vehicle's programming port. Check Device Manager to obtain the correct COM Port number. Navigate to the Flash Magic program (shown in Figure 1.1) and select the



LPC2146
Flash Bank:
COM Port: COM 10
Baud Rate: 115200
Interface: None (ISP)
Oscillator (MHz): 14.7456

Erase block 0 (0x000000-0x000FFF)
Erase block 1 (0x001000-0x001FFF)
Erase block 2 (0x002000-0x002FFF)
Erase block 3 (0x003000-0x003FFF)
Erase block 4 (0x004000-0x004FFF)
Erase block 5 (0x005000-0x005FFF)

[✓] Erase all Flash+Code Rd Prot
[ ] Erase blocks used by Hex File

Hex File: \\ad.uillinois.edu\engr\instructional\hanley6\Desktop\AE483Lab3Q
Modified: Unknown

[ ] Verify after programming
[ ] Fill unused Flash
[ ] Gen block checksums
[ ] Execute
[ ] Activate Flash Bank

CAN Bus Timing Calculators at:
www.esacademy.com/en/library/calculators.html

0

Figure 1.1: Interface for Flash Magic Program

correct COM port (can be found in your computer's Device Manager) and the Hex file. Then select `Start`. Let your TA know once you have successfully flashed your quadrotor.

1.3.2 Communicate over Xbee

1. Download the ACI Tool. On the course website, download `aci-tool-windows.zip` and unzip the folder on your `U:/` drive.
2. Open and run the ACI Tool. Open the `aci-tool-windows` folder and run the executable file. A GUI should pop up that looks like Figure 1.2 below. Enter the Device number as 0.
3. In the ACI Tool, request variables that you would like to receive from the quadrotor over the Xbee. Assign the angular velocities and accelerations to packet number 1 as shown in Figure 1.3. Next, set the transmission rate. Choose $x = 100$, where the rate is defined as $(1000/x)$.

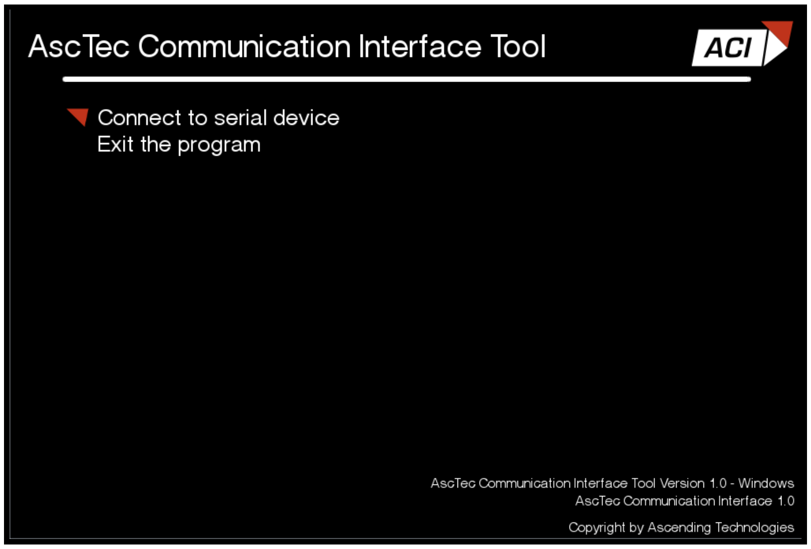
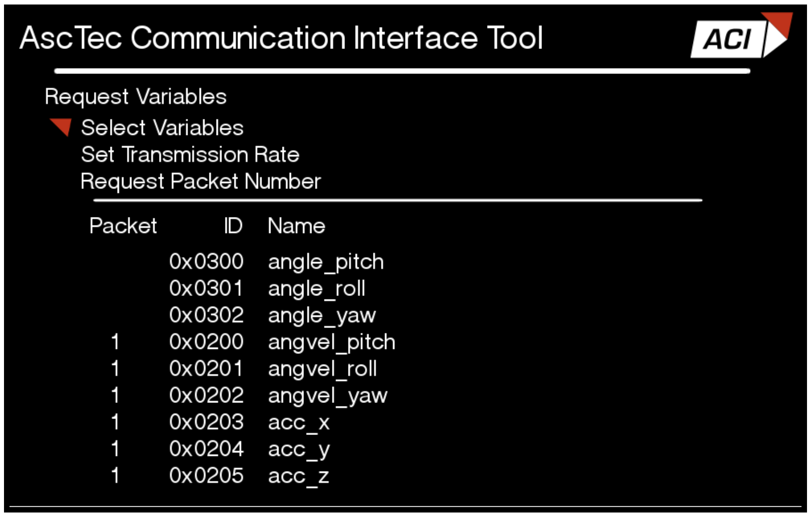

AscTec Communication Interface Tool
ACI
Connect to serial device
Exit the program
AscTec Communication Interface Tool Version 1.0 - Windows
AscTec Communication Interface 1.0
Copyright by Ascending Technologies

Figure 1.2: ACI Tool GUI



Packet	ID	Name
	0x0300	angle_pitch
	0x0301	angle_roll
	0x0302	angle_yaw
1	0x0200	angvel_pitch
1	0x0201	angvel_roll
1	0x0202	angvel_yaw
1	0x0203	acc_x
1	0x0204	acc_y
1	0x0205	acc_z

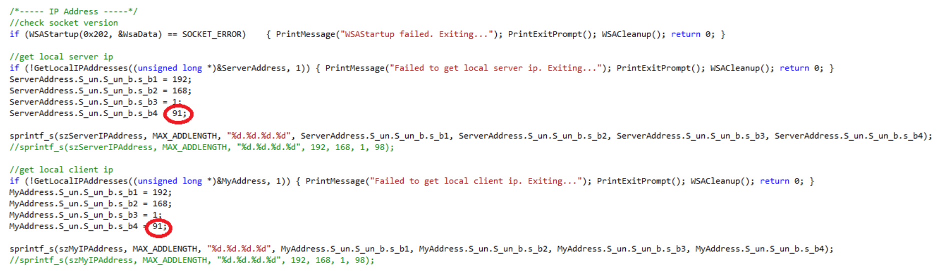
Figure 1.3: ACI Tool Request Variables

4. Display the six variables you selected above. Show your TA the result.

1.3.3 Visualize Flight Data

1. Download and unzip the `AE483GroundStation` code from the AE 483 website. Open the solution file (**.sln** extension) in Visual Studio **2010** [Right-click and Open with Visual Studio 2010].
2. In main.c, change the IP to your PC IP. Specifically change the code block

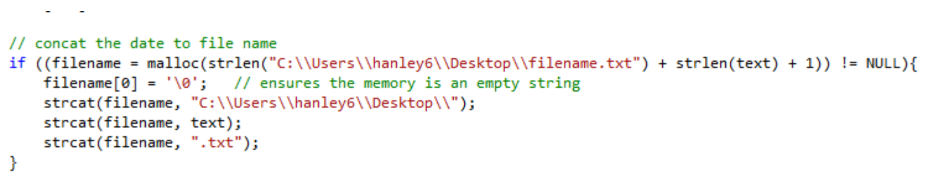
shown in 1.4.



```
/*----- IP Address -----*/
//check socket version
if (WSAStartup(0x202, &WsaData) == SOCKET_ERROR)    { PrintMessage("WSAStartup failed. Exiting..."); PrintExitPrompt(); WSACleanup(); return 0; }

//get local server ip
if (!GetLocalIPAddresses((unsigned long *)&ServerAddress, 1)) { PrintMessage("Failed to get local server ip. Exiting..."); PrintExitPrompt(); WSACleanup(); return 0; }
ServerAddress.S_un.S_un_b.s_b1 = 192;
ServerAddress.S_un.S_un_b.s_b2 = 168;
ServerAddress.S_un.S_un_b.s_b3 = 1;
ServerAddress.S_un.S_un_b.s_b4 = 91;

sprintf_s(szServerIPAddress, MAX_ADDLENGTH, "%d.%d.%d.%d", ServerAddress.S_un.S_un_b.s_b1, ServerAddress.S_un.S_un_b.s_b2, ServerAddress.S_un.S_un_b.s_b3, ServerAddress.S_un.S_un_b.s_b4);
//sprintf_s(szServerIPAddress, MAX_ADDLENGTH, "%d.%d.%d.%d", 192, 168, 1, 98);

//get local client ip
if (!GetLocalIPAddresses((unsigned long *)&MyAddress, 1)) { PrintMessage("Failed to get local client ip. Exiting..."); PrintExitPrompt(); WSACleanup(); return 0; }
MyAddress.S_un.S_un_b.s_b1 = 192;
MyAddress.S_un.S_un_b.s_b2 = 168;
MyAddress.S_un.S_un_b.s_b3 = 1;
MyAddress.S_un.S_un_b.s_b4 = 91;

sprintf_s(szMyIPAddress, MAX_ADDLENGTH, "%d.%d.%d.%d", MyAddress.S_un.S_un_b.s_b1, MyAddress.S_un.S_un_b.s_b2, MyAddress.S_un.S_un_b.s_b3, MyAddress.S_un.S_un_b.s_b4);
//sprintf_s(szMyIPAddress, MAX_ADDLENGTH, "%d.%d.%d.%d", 192, 168, 1, 98);
```

Figure 1.4: Code in main.c to set IP

3. Change the location where data files are saved (see Figure 1.5). Navigate to the `Build` dropdown and select `Build Solution`. Make sure the code successfully compiles.



```
// concat the date to file name
if ((filename = malloc(strlen("C:\\Users\\hanley6\\Desktop\\filename.txt") + strlen(text) + 1)) != NULL){
    filename[0] = '\0';   // ensures the memory is an empty string
    strcat(filename, "C:\\Users\\hanley6\\Desktop\\");
    strcat(filename, text);
    strcat(filename, ".txt");
}
```

Figure 1.5: Change the location where data files are saved.

4. Ensure the motion capture system identifies your quadrotor (see Figure 1.6 for picture of the motion capture interface), and make sure the quadrotor ID in your groundstation code matches the ID for your quadrotor in the motion capture system (see Figure 1.7).
5. Plug in your quadrotor's Xbee and with the quadrotor on, navigate to `Debug` and select `Start Without Debugging`. When prompted by Windows Firewall, **check all boxes** and click `Allow Access`. On the command prompt that opens up, `Hit Enter` to record data. Proceed to fly the quadrotor around the room.
6. Load the text file produced by the groundstation into MATLAB.
7. Plot the time history of the position of the center of mass of the quadrotor using data from the motion capture system.
8. Plot the time history of the roll, pitch, and yaw angles from integrating the onboard sensor information and from the motion capture system.
9. Show the results to the TA when you have finished.

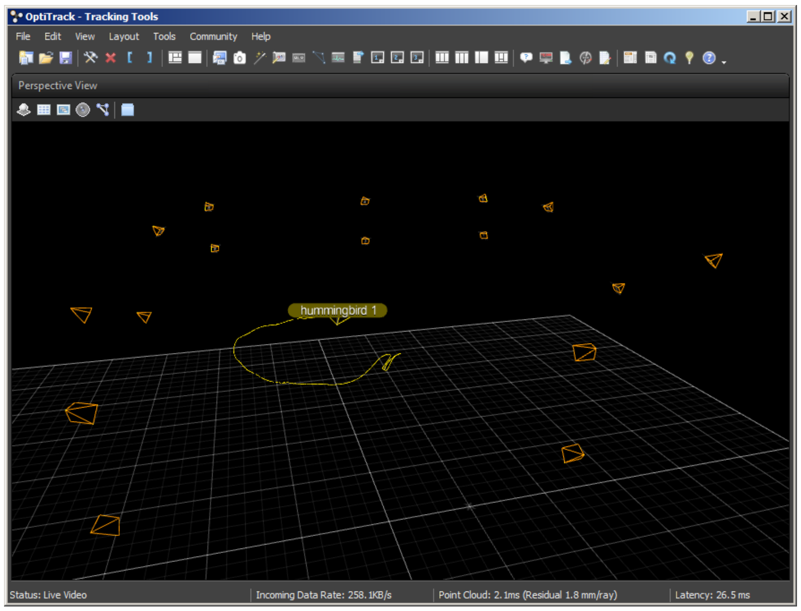
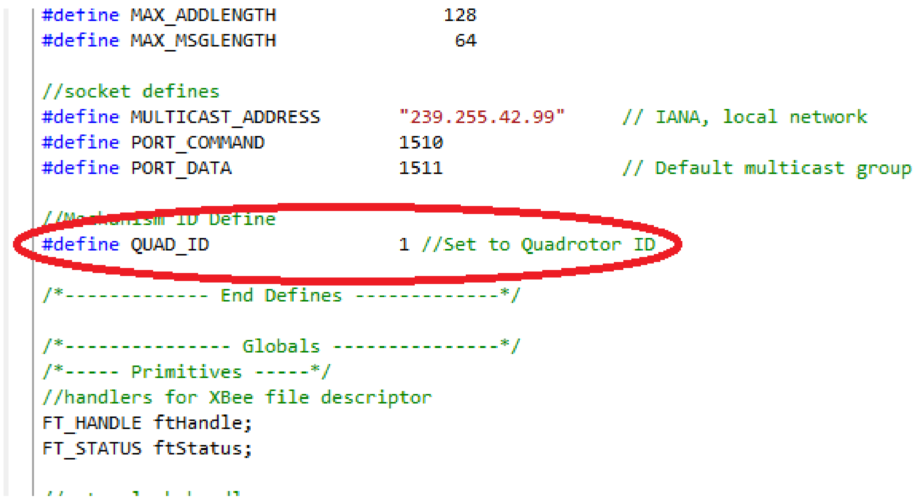

OptiTrack - Tracking Tools
File Edit View Layout Tools Community Help
Perspective View
hummingbird 1
Status: Live Video
Incoming Data Rate: 258.1KB/s
Point Cloud: 2.1ms (Residual 1.8 mm/ray)
Latency: 26.5 ms

Figure 1.6: Example of the motion capture system tracking the hummingbird vehicle.



```
#define MAX_ADDLENGTH               128
#define MAX_MSGLENGTH                64

//socket defines
#define MULTICAST_ADDRESS       "239.255.42.99"     // IANA, local network
#define PORT_COMMAND            1510
#define PORT_DATA               1511                // Default multicast group

//Mechanism ID Define
#define QUAD_ID                 1 //Set to Quadrotor ID

/*------------- End Defines -------------*/

/*--------------- Globals ---------------*/
/*----- Primitives -----*/
//handlers for XBee file descriptor
FT_HANDLE ftHandle;
FT_STATUS ftStatus;
```

Figure 1.7: Change quadrotor ID in ground station to match the motion capture system.

1.4 Checkoff Points

1. Show the TA that you have flashed the quadrotor with the C code you downloaded from the course website.

2. Show the TA that you can receive IMU data through the ACI Tool.
3. Show the TA your MATLAB code and plots you made when visualizing flight data above.

1.5 Report

Write a lab report which answers the following problems.

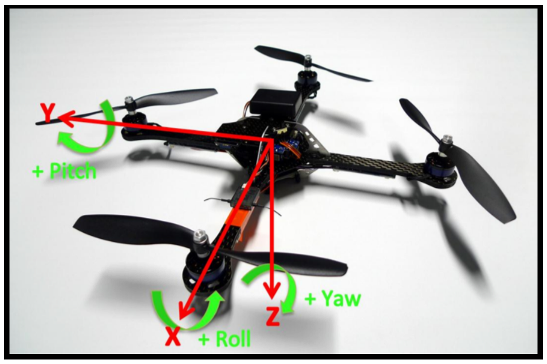
Problem 1.1. Using the flight data collected in lab, integrate angular velocities and translational accelerations.

(a) Use Euler's method of numerical integration to integrate measured angular velocities and obtain Euler angles as functions of time. Plot these together with the angles that are provided by mocap. For example, create three separate figures, one for roll, pitch, and yaw. In each figure, plot three curves representing data obtained from motion capture and your integration results.

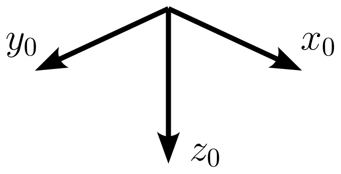
(b) User Euler's method of numerical integration to integrate the accelerations a_x, a_y, a_z from the vehicle sensor measurements and obtain velocity v_x, v_y, v_z. Integrate again to get position x, y, z. Plot your integration results against the data provided by motion capture. Again, we suggest creating three separate figures, one for x, y, and z. In each figure, plot two curves represening data measured by motion capture and your integration results. Discuss any differences you see between your integrated results and measured data.

Problem 1.2. Draw (in MATLAB) a 3D picture of the quad-rotor, given position and orientation from mocap, as viewed from the space frame. To complete this problem, you will be adding a small amount of code to the script `lab1_drawquad.m`, available on the course website. In particular, to animate the motion of the rigid body, you just need to update each vertex p of the rigid body. In the script, each column of the matrix `p1` is a vertex written in the coordinates of frame 1 (the body frame). Your goal is to generate the matrix `p0`, each column of which is the same vertex written in the coordinates of frame 0 (the world frame). Then, to display the result in MATLAB, you will generate the matrix `p2`, each column of which is the same vertex written in the coordinates of frame 2 (the MATLAB display frame). See Figure 1.8 for a definition of the three coordinate frames in this problem. The world frame and display frame share the same origin and the same x-axis. Note that you should be able to compute `p0` from `p1` using only *one* matrix multiplication – this is one of the many ways that matrix notation makes computation easier. Turn in your MATLAB code, which does animation, by email.

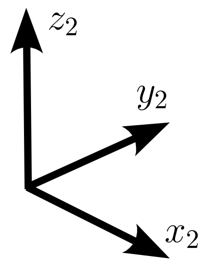
Problem 1.3. Draw (in MATLAB) a 3D picture of the world as viewed from the body frame. To complete this problem, you will be adding a small amount of code



(a) Frame 1 – body frame.



(b) Frame 0 – world frame.



(c) Frame 2 – MATLAB display frame.

Figure 1.8: Coordinate frames considered in Problems 2 and 3.

to the script `lab1_bodypov.m`, available on the course website. In particular, to animate the motion of the world, you just need to update each vertex w of the objects. These world objects, already defined in the code, include the floor and the positions of the motion capture cameras. In the script, each column of the matrix `w0` is a vertex written in the coordinates of frame 0 (the world frame). Your goal is to generate the matrix `w1`, each column of which is the same vertex written in the coordinates of frame 1 (the body frame). Then, to display the result in MATLAB, you will generate the matrix `w2`, where again each column is the same vertex written in the coordinates of frame 2 (the MATLAB display frame). Note that in this problem, frame 1 and frame 2 share the same origin and x-axis. Note that you should be able to compute `w1` from `w0` using only *one* matrix multiplication – this is one of the many ways that matrix notation makes computation easier. Turn in your MATLAB script, which does animation, by email.

Note: Your report should carefully describe anything that went "wrong" with the experiments, or any mismatch between the data and what you saw with your own eyes. It should be **written with your lab group**, and submitted by email to `ae483.fall2016@gmail.com` with the subject line "Lab 1 Report" and the attached report should be named with the last names of all group members "*Lastname1-Lastname2-Lastname3*-Lab-1.pdf".