

AE483: Lab #1

Sensor Data Collection and Analysis

T. Bretl

September 11, 2017

1 Goal

You will be working with the AscTec Hummingbird Quadrotor this semester. There are two sources of sensor data that you can use for control design and analysis:

- an inertial measurement unit (IMU) on-board the quadrotor that measures linear acceleration and angular velocity using accelerometers and gyroscopes;
- a motion capture system (mocap) off-board the quadrotor that measures position and orientation using cameras that track spherical markers.

As part of this lab, you will learn how to collect and visualize data from these two sources. You will show how these data are related and why it is often helpful to have both an IMU and mocap, rather than one or the other. Then, you will characterize an important limitation of mocap—that it provides accurate measurements of position and orientation only when the quadrotor is in view of several cameras at the same time. Your primary goal is to describe the region within which the quadrotor can fly while satisfying this constraint. (If your quadrotor exits this region in future weeks, it will almost certainly crash.)

2 On the first day

Your specific objective on this first day is to collect and visualize data from the quadrotor during a short manual flight. You will do the following things:

- Program the quadrotor to collect on-board sensor data (Section 2.1).
- Configure the network so that these on-board sensor data are transmitted wirelessly to the ground station (Section 2.2).
- Run the ground station so both on-board and off-board data are saved to a text file during your flight (Section 2.3).
- Load, parse, and plot these data using MATLAB (Section 2.4).

Section 2.5 has a summary of in-lab deliverables. **Make sure to save your flight data for later analysis.**

2.1 Program the quadrotor

Your first task is to program the quadrotor to collect on-board sensor data:

1. *Create a project in AscTec Eclipse (a software development environment).* Download these files from the course website:

```
AscTec_SDK_v3.0_Lab1.zip
AE483GroundStation_Lab1.zip
aci-tool-windows.zip
```

Run the AscTec Eclipse application from the start menu:

Start → AscTec ARM SDK → AscTec ARM SDK (Eclipse)

Create a workspace in your EWS home directory:

```
U:\Documents\workspace\
```

Import AscTec_SDK_v3.0_Lab1.zip in eclipse:

File → Import → General → Existing Projects into Workspace
→ Select Archive file

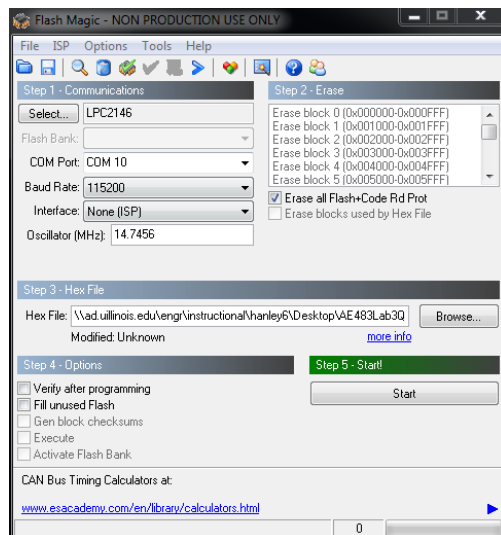
Browse to find AscTec_SDK_v3.0_Lab1.zip and make sure that the button “copy projects into workspace” is selected, then click “Finish”.

2. *Configure the build targets.* In Eclipse:

Project → Clean Project

Make sure there are no errors in the console window in Eclipse. If there were no errors, you should see a `main.hex` file in your workspace folder.¹

3. *Flash the main.hex file onto the quadrotor.* Turn the quadrotor on in programming mode and connect a USB cable to its programming port. Open “Device Manager” on your computer to find the COM port number for this USB cable. Open “Flash Magic Program” on your computer:



¹If you forgot the directory of the workspace folder, you can right-click the project and find it in properties.

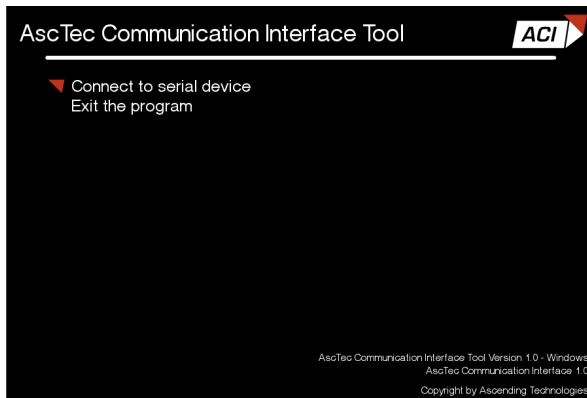
Select both the port number that you found and the `main.hex` file. Click **Start**.²

Let your TA know once you have successfully programmed (a.k.a., “flashed”) your quadrotor.

2.2 Configure the network

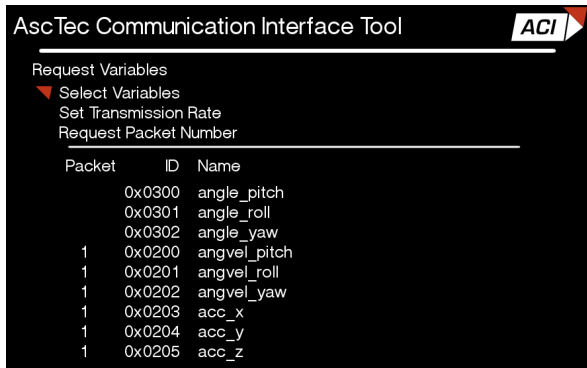
Your second task is to configure the network so that on-board sensor data are transmitted wirelessly to a ground station:

1. *Configure the quadrotor.* Turn the quadrotor on in running mode, and connect a USB cable from PC to the XBee wireless module.
2. *Download the ACI Tool.* On the course website, download `aci-tool-windows.zip` and unzip the folder on your U:/ drive.³
3. *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 this:



Select “connect to serial device” and enter the Device number as 0.

4. *Select variables and set transmission rate.* In the ACI Tool, choose the variables you want to receive wirelessly from the quadrotor. Assign the angular velocities and accelerations to packet number 1 as shown here:



²If you have any trouble, do the following. First, check that both switches on quadrotor indicate “Prog”. Second, check that every parameter in Flash Magic matches the screenshot except for the COM port number and the hex file directory. Note that LPC2146 can be found in the first folder, scrolling all the way up. Finally, even if everything is correct, you may still see “operation failed.” In that case, restart the quadrotor and unplug/plug the USB cable.

³It is possible for some files to be lost in this process. If anything goes wrong in subsequent steps, you can try repeating this step, so unzipping the files again.

Then, set the transmission rate. Choose $x = 100$, where the rate is defined as $(1000/x)$.⁴

Display the six variables you selected and show your TA the result. Then, **close the ACI Tool**. (You **must** close it before proceeding to run the ground station as described in Section 2.3.)

2.3 Run the ground station

Your third task is to run the ground station so that both on-board and off-board data are saved to a text file during your flight:

1. *Create a project in Visual Studio 2010 (another software development environment)*. Download and unzip the `AE483GroundStation` code from the AE 483 website. Right-click the solution file (with extension `.sln`) to open it with Visual Studio 2010.
2. *Set the IP address*. In `main.c`, change the IP address to your computer's IP address by modifying this code block:⁵

```
/*----- 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);
```

3. *Specify name of data file*. Change the location where data files are saved by modifying this code block in `main.c`:

```
- -
// 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");
}
}
```

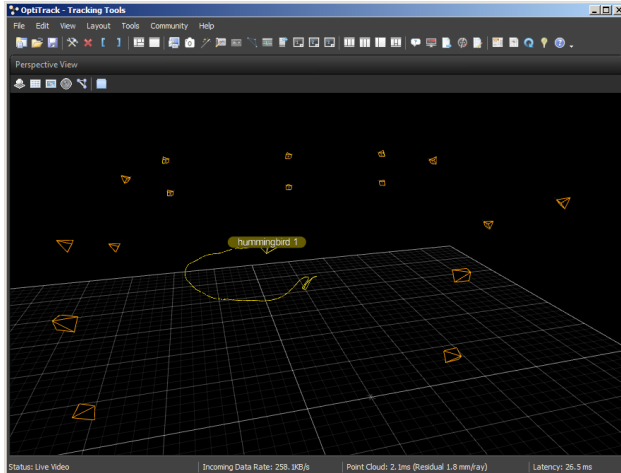
Select **Build Solution** from the **Build** dropdown menu. Make sure the code successfully compiles.⁶

4. *Start mocap*. Open the motion capture interface application on the TA's computer and make sure that your quadrotor appears:

⁴Only the value for packet 1 is used in this lab, but you should still set the transmission rate $x = 100$ for all packets. Make sure you choose 1 when asked for a packet number—this one contains the six variables you want.

⁵There is another computer on the shelf, with a paper attached. Please use the PC-IP written in the paper as your computer's IP address.

⁶Keep "filename.txt" unchanged. Make sure that the two directories are exactly the same except for "filename.txt". Always use "\\\" instead of "\" in directories.



In particular, make sure the ID of your quadrotor in the mocap interface matches the ID of your quadrotor in `main.c`:

```
#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

//Mokosiss 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;
// . . . . .
```

5. *Start wireless transmission.* Plug in your quadrotor's XBee wireless module (the thing it uses to wirelessly transmit data) 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, click **Enter** to record data. ⁷
6. *Go!* Fly the quadrotor around the room (see attached flight guidelines). **Record video of its flight (e.g., with a smartphone) for comparison with your flight data.**

Show your TA the text file that was generated with data from your flight.

2.4 Load, parse, and plot data

Your fourth task is to load, parse, and plot the flight data, using MATLAB:

1. *Load and parse.* Open the text file produced by the ground station in a text editor (or in Microsoft Excel). You will see that the data in this file are comma-separated. Each line in the

⁷In case you fail to check all boxes but click Allow Access, go to Windows firewall to manually change the inbound rules.

file has a row of data. The first row tells you what is contained in each column (`Flight Time`, `accel_x`, and so forth). Open the text file in MATLAB. Organize and rename it as you see fit. Save the result as a `.mat` file for later analysis.

2. *Plot.* Plot the position of the quadrotor (i.e., of its center of mass) as a function of time.

Show your plot to the TA when you have finished.

2.5 Summary of in-lab deliverables

Confirm that you have done the following things in lab on the first day:

1. Show the TA that you flashed the quadrotor with code downloaded from the course website.
2. Show the TA that you received IMU data through the ACI tool.
3. Show the TA that flight data was saved to a text file.
4. Show the TA that you can plot the position of the quadrotor as a function of time.

Please come back to the lab during office hours to do anything you did not finish.

3 In between the first and second day

In between the first and second day, you will do further analysis to show how on-board and off-board sensor data are related. In particular, please do everything in Sections 3.1-3.3 before arriving on the second day of lab and be prepared to discuss your results with your TA.

3.1 Show that it is not a good idea to rely only on IMU data

Please do the following:

- You have measurements of angular velocity in the coordinates of a frame attached to the quadrotor (we will call this *the body frame*). Numerically integrate these measurements (e.g., using Euler's method) to obtain ZYX Euler Angles as functions of time. Plot these angles together with the yaw, pitch, and roll angles that are provided by mocap. In particular, create three separate figures—one for yaw, pitch, and roll. In each figure, plot two curves—one from integration and one from mocap.

Be prepared (at the start of the second day in lab) to show these plots to your TA and to discuss any differences you see between your integrated results and the mocap measurements. Should you rely only on IMU data?

3.2 Show that it is not a good idea to rely only on mocap data

Please do the following:

- You have measurements of ZYX Euler Angles from mocap. Numerically differentiate these measurements (e.g., using a finite difference) to obtain angular velocity in the coordinates of the body frame as a function of time. Plot this result together with the measurements of angular velocity that are provided by the IMU. In particular, create three separate figures, one for each component of angular velocity, and plot two curves on each figure (one from differentiation, one from IMU data).

Be prepared (at the start of the second day in lab) to show these plots to your TA and to discuss any differences you see between your differentiated results and the IMU measurements. Should you rely only on mocap data?

3.3 Visualize the flight

Show a 3D animation of the quadrotor’s flight in MATLAB, given the measurements of position and orientation in the coordinates of the room frame from mocap. You can do this by adding a small amount of code to `lab1_drawquad.m`, available on the course website. In particular, you will:

- Add code to parse the flight data.
- Add code to define a frame—we will call it *the display frame*—that has the same origin as the room frame and that can be obtained by rotating the room frame 180° about the x_{room} axis. This frame makes it easier to plot things in MATLAB (which likes “z up” and not “z down”).
- Add code to apply coordinate transformation. You are given points that describe the shape of the quadrotor in the coordinates of the body frame. You are also given points that describe the shape of the room (e.g., the position of each camera) in the coordinates of the room frame. You need to describe all of these points in the coordinates of the room frame. Here is a common pattern for coordinate transformation in MATLAB:

```
1 % p_in1 is a 3xN matrix that describes a set of points, one in each column,  
2 %   in the coordinates of frame 1  
3 %  
4 % p_in0 is a 3xN matrix that will describe these same points, one in each  
5 %   column, in the coordinates of frame 0  
6 for i = 1:size(p_in1,2)  
7     p_in0(:,i) = o_lin0 + R_lin0*p_in1(:,i);  
8 end
```

Be prepared (at the start of the second day in lab) to show the resulting animation to your TA and to discuss any differences you see between it and the video you took with your phone.

4 On the second day

Your specific objective on this second day is to identify and describe the region within which the quadrotor must remain in order for mocap to provide accurate measurements of position and orientation. We will call this region the *workspace* of the motion capture system. To convince yourself that this workspace is bounded, move the quadrotor slowly away from the center of the room, and watch it disappear from the motion capture interface application.

4.1 Before lab

You have learned how to setup and fly the quadrotor (and are free, from now on, to do so on your own—if you follow the flight guidelines in the attached document). You have also learned how to collect sensor data and how to visualize these data with MATLAB. Before you come to lab on the second day, you should design an experiment (e.g., a flight that moves back and forth, up and down, in spirals, etc.) that will allow you to identify the mocap workspace.

4.2 During lab

Implement the experiment that you designed. Show the TA:

- the text file with your flight data;
- a plot of position as a function of time.

You should verify, in particular, that your plot disappears at certain times—these are times when your quadrotor flew outside the mocap workspace (or when tracking was otherwise lost). **Make sure to save your flight data for later analysis.** As before, we suggest you also record a video of your flight (e.g., with a smartphone).

5 Report

Write a report that uses the flight data collected on the second day of lab to identify, describe, and visualize the mocap workspace. Your report will have at least four parts:

- *Goal.* Say what you wanted to achieve.
- *Method of approach.* Describe the experiment that you implemented on the second day of lab.
- *Results.* Visualize the mocap workspace.
- *Discussion.* State at least one source of error. Describe at least one way in which your results could be improved. Propose at least one strategy for keeping the quadrotor inside the mocap workspace (as you've characterized it).

You are encouraged to go beyond these requirements. Your report must be submitted as a PDF with single-space text no larger than 12-point font and with 1-inch margins. You are strongly encouraged, but not required, to use \LaTeX to prepare your report. It is expected that your PDF document will include a minimum of four pages of text and a minimum of two pages of figures. However, you are welcome to write more. You must submit your report no later than 11:59PM on Friday, September 29. Submission details will be posted to piazza.