BIOM1010: Engineering in Medicine and Biology
Tutorial: Estimating IMU orientation

Stephen Redmond, Michael Stevens, Socrates Dokos

7 September 2018

1 Preparation

1.1 Get access to MATLAB

- Before you attend the class, if you have not already done so, it is important you watch the YouTube videos describing the very basics of the MATLAB programming environment. There is about 1 hour 20 mins of video and some quizzes.
  - These can be found via our Moodle module (for session T2-2018) where you can also attempt simple quizzes to test your knowledge, and find other tutorial material.
  - Or you can link directly to them in this YouTube playlist.
  - If you recall, there is also a YouTube video on the basics of MATLAB in the Week 4 section of BIOM1010 in Moodle.
- You can download Matlab software here. Note that MATLAB is already installed on the lab computers in room 518, but if you prefer to practice before the tutorial, you can install MATLAB on your own computer by downloading the software.
- Alternatively, you can also run MATLAB remotely on your computer through UNSW myAccess.
- Ask a tutor for help if you cannot access MATLAB.

1.2 Install Physics Toolbox Suite

- Install the Physics Toolbox Sensor Suite app on your Apple iOS or Android phone:
  - Apple: http://tinyurl.com/y8a5l2gd
  - Android: http://tinyurl.com/yc2qsutc

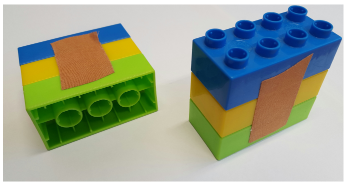
2 Tasks

2.1 Rotating around an axis by an angle

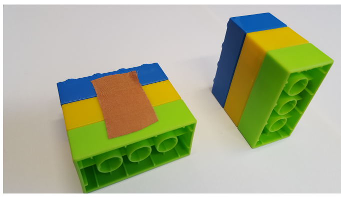
- To get a feel for performing rotations using quaternions, use the MATLAB function provided on Moodle called `sampleRotationsBIOM1010.m` to perform rotations of the world coordinate frame my some angle, θ (called `theta` in the code), about some vector, $\mathbf{u} = (u_x, u_y, u_z)$. For example, you can rotate by $\frac{2\pi}{3}$ about a unit vector in the direction of $\mathbf{u} = (1, 4, 3)$ using the code below. The result will print out in the MATLAB command window.

```
sampleRotationsBIOM1010([1 4 3],2*pi/3)
```

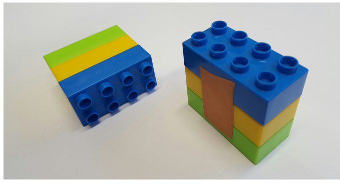
- Consider what axis and by what angle you should rotate the Lego object on the left hand side of each the three images below in order to obtain the attitude shown on the right of each image. Perform this rotation using the `sampleRotationsBIOM1010.m` function and confirm whether you were correct or not.



(a)

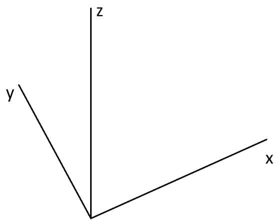


(b)



(c)

- It is important to pick coordinate system before you begin. We recommend a right hand coordinate system shown below, with the positive $\mathbf{x}$ pointing to the right of the image and horizontal to the ground, positive $\mathbf{y}$ perpendicular to $\mathbf{x}$ but also horizontal to the ground, and positive $\mathbf{z}$ pointing up.



3 Fusing sensor data from smartphone to estimate attitude

3.1 Download and import pre-recorded data

We have recorded some data on a Samsung S6 smartphone using the Physics Toolbox Sensor Suite app. The recorded data was exported from the phone to a comma separated variable (csv) file named `raw-StephenRotatingPhone.csv`

- Please download and open this `raw-StephenRotatingPhone.csv` file from Moodle and open it using Microsoft Excel, or some other text file editor.
  - The units of the measurements are not included in the exported file, but we are confident they are:
    - Time: s
    - Accelerometer: G = 9.81 m s^{-2}
    - Gyroscope: rad s^{-1}
    - Magnetometer: μT
- Import the data into the MATLAB workspace.

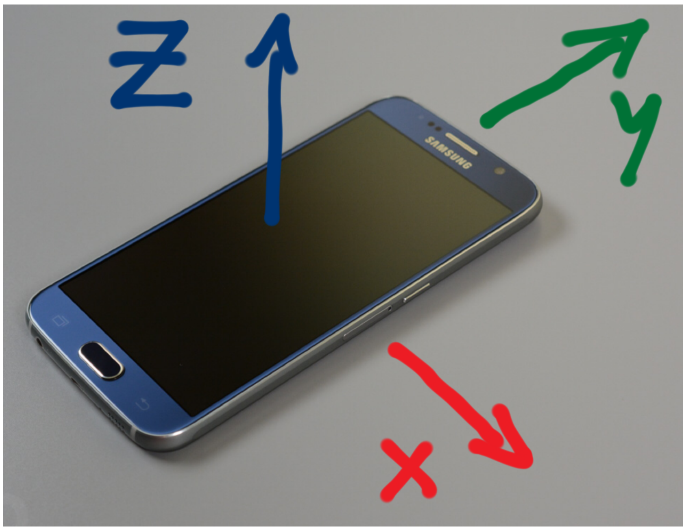


Figure 1: Samsung S6 showing positive sense of sensor axes for accelerometer, gyroscope and magnetometer.

- Ensure that all data and MATLAB files you are using are in the same directory, and that this is the current directory which MATLAB is looking at. Ask for help if you don't know what this means.
- Run the following command to import the data into a structure; type it into the command window yourself, as the inverted commas will not paste as the correct characters:

```
IMU = importdata('raw-StephenRotatingPhone.csv')
```

3.2 Process the IMU data

Next we want to run the MATLAB code provided, which will fuse the IMU sensors signals to estimate the phone's attitude. Perform the following tasks:

- Put data into intermediate variables so it's easier for us to read the code.
  - Run the following commands:

```
time = IMU.data(:,1);
Acc = 9.81*IMU.data(:,2:4);
Gyro = IMU.data(:,5:7);
Mag = IMU.data(:,8:10);
```

- Now we will put the data through the code to fuse the signals.
  - Download from Moodle the MATLAB file called `mainAHRS.m`, place it in the current directory with your other files.
  - Run the following command:

```
[AHRSresult] = mainAHRS(Acc,Mag,Gyro,time);
```

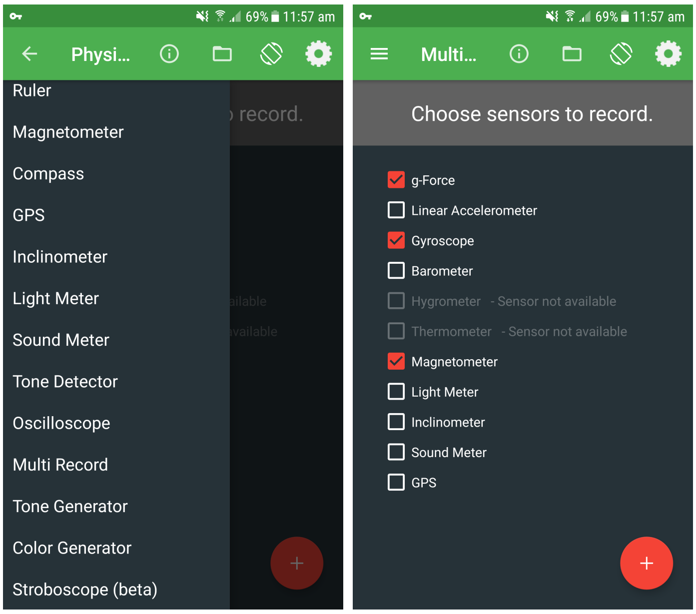
- In order to visualise the result, we will create a video which shows a simple animation of the phone's attitude.
  - Download from Moodle the MATLAB file called `videoOrientations.m`, place it in the current directory with your other files.
  - Run the command below. This will use the default codec to create a video; most likely a `.avi` file.
    * When you run the code, it will create the figure and then pause. It is waiting for you to resize/maximise the window and then press any key on the keyboard, after which it will continue with creating the video.
    * Note, the video is 22.207 seconds long, but creating the video file may take a few minutes, so use the time to ask your group mates how their life is going.

```
videoOrientations(AHRSresult,'myVideoOfPhoneAttitude')
```

- When the video is ready, open it with a video player and view it. We would recommend VLC media player if you don't already have one installed.

3.3 Record your own data and process it

- Using the Physics Toolbox Suite app, record your own data. Instructions for doing this will be slightly different for the Apple and Android versions. The main difference is that for the Apple version, it is not possible to simultaneously record from the g-Force, Gyroscope, and Magnetometer sensors.
  - From the menu, select the Multi Record option (Android only).
  - Select the g-Force, Gyroscope, and Magnetometer sensors to record from. Note that for the Apple version, you have the option of selecting only one of these at a time.
    * For the Android, do not select the Linear Accelerometer sensor, as this signal has had the gravity information used to correct the device inclination removed from it.
  - Press the big red circle with the '+' symbol to start.
    * You will not see a graph of the signals, but they are being recorded.
    * Perhaps record about 15-20 seconds of data, otherwise it will take a long time to create the video later.
    * You should try rotating the phone about different axes, shaking/translating it, and both shaking/translating and rotating at the same time, to see how this influences the attitude estimates.
  - Press the red circle again to stop the recording when you're done.
  - Give the recorded file a name and then email it to yourself when prompted by the app.
- You can now repeat the steps in section 3.2 above to create a video animation of this recording.



3.4 Investigate how correction gains affect the result (Android only)

Finally, we are going to fiddle with the correction gains inside the code. Remember from the lecture, given an estimate of the phone's attitude, the following steps are performed. The gyroscope is used to rotate the phone around the angular velocity vector (as we believe it to be in the world frame given our current estimate of the phone's attitude) to estimate the next orientation.

We then look at whether the acceleration vector in the world frame is pointing up. If it is not, we will rotate the phone so the acceleration vector moves towards pointing upwards. But we do not make it point upwards in one correction, as we do not trust the accelerometer is only measuring gravity - it could be measuring inertial accelerations also. So we move by a fraction of the angular error - this fraction is related to the `muAcc` variable in the MATLAB code of the `mainAHRS.m` file. Moving by small fractions of the angular error helps to average out the inertial accelerations over time.

Similarly, we expect the xy component of the magnetometer reading in the world frame (at least the direction we believe the xy component of the magnetometer reading is pointing in the world frame given our current estimate of the phone's attitude) to point north, which we say is along the x direction. If there is an angular error, we correct this by some fraction (related to the `muMag` variable in the `mainAHRS.m` file).

In this part we will investigate the effect of meddling with these correction fractions to see how it affects the result. You might consider making a recording which involves several (maybe 5 or more) complete rotations around various phone axes.

3.4.1 Turn off the accelerometer and magnetometer corrections

- In the `mainAHRS.m` file on lines 68 and 69, set the following values:

```
muMag = 0.0*dt; % Correction rate for magnetometer alignment.
muAcc = 0.0*dt; % Correction rate for accelerometer alignment.
```

- What is the effect on the pitch/roll (inclination) and yaw (deviation from north) errors?

3.4.2 Only turn off the accelerometer correction

- In the mainAHRS.m file on lines 68 and 69, set the following values:

```
muMag = 0.01*dt; % Correction rate for magnetometer alignment.
muAcc = 0.0*dt; % Correction rate for accelerometer alignment.
```

- What is the effect on the pitch/roll (inclination) error? Does this also affect the yaw? What is the reasoning behind your answer?

3.4.3 Only turn off the magnetometer correction

- In the mainAHRS.m file on lines 68 and 69, set the following values:

```
muMag = 0.0*dt; % Correction rate for magnetometer alignment.
muAcc = 0.2*dt; % Correction rate for accelerometer alignment.
```

- What is the effect on the yaw (deviation from north) error? Does this also affect the pitch/roll? What is the reasoning behind your answer?

4 Walking Activity

- Select the Gyroscope sensor only, and record yourself walking a few metres in a straight line, then turning around and walking back to your starting position.
- Import the data into MATLAB, and plot the x, y, and z components of your angular velocity in rad s^{-1}. Include a legend (HINT: Use MATLAB's `legend` command), and provide appropriate axes labels.
- Save your plot as a PDF file (use File → Save As... from the figure menu), and submit via Moodle.