Lab 3

Sensor Calibration

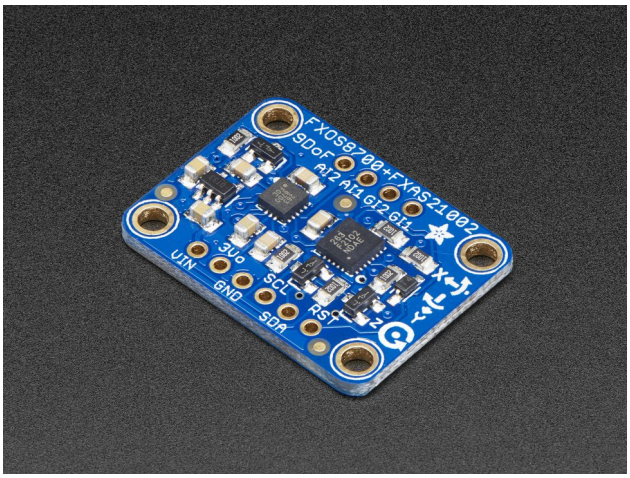

Figure 1: Adafruit inertial measurement unit

In this lab you will calibrate the inertial measurement unit (IMU)[1] on your roomba. We are using Adafruit's NXP IMU. The board consists of two separate ICs:

FXOS8700 3-Axis Accelerometer/Magnetometer

- 2-3.6V Supply
- ± 2/4/8 g adjustable acceleration range
- ± 1200 μT magnetic sensor range
- Output data rates (ODR) from 1.563 Hz to 800 Hz
- 14-bit ADC resolution for acceleration measurements
- 16-bit ADC resolution for magnetic measurements

FXAS21002 3-Axis Gyroscope

- 2-3.6V Supply
- ± 250/500/1000/2000 deg/s configurable range
- Output Data Rates (ODR) from 12.5 to 800 Hz
- 16-bit digital output resolution
- 192 bytes FIFO buffer (32 X/Y/Z samples)

You should learn or gain experience with:

[1] https://en.wikipedia.org/wiki/Inertial_measurement_unit

- Collect raw data from a robotic sensor
- Understand that most sensors are inaccurate or useless without proper calibration
- Learn a very simple technique to calibrate an IMU

Task 1: Get Data

The first part of the lab we will gather data. Using the provided code `bagit`, you will save a bunch of data. `bagit` will save the sensor data to a json[2] file.

```
#!/usr/bin/env python

from __future__ import print_function, division
import nxp_imu
import time

if __name__ == "__main__":
    imu = nxp_imu.IMU('imu.json')
    bag = bagit.Bag(['imu'])

    for i in range(1000):
        a,m,g = imu.get()          # grab data
        bag.push('imu', (a,m,g))   # save data
        time.sleep(1/20)           # grab data at 20 Hz

    bag.close()
    print('Done ...')
```

When you are capturing the data, you will do it 2 different ways:

1. The first way is to help determine biases, so we need to roll the roomba around and exercise the axes of the sensor. **WARNING:** If you drop the roomba, you automatically fail the lab. There are not enough robots available if we start damaging them. Please make sure you have a good grip at all times and no horse play!
2. Start off holding the roomba level. Slowly rotate 360 degrees stopping for a few seconds every 90 degrees. This second data set is to check our biases are correct. Remember, the biases you calculate are only good the robot you develop them for, they are not transferable to another robot.

After you have save the data successfully, open the file and take a look at the data.

```
cat imu_1.json
```

You will notice that the data is all text. Since text is generally not efficient, a better way to store lots of data would be to use a binary form of json (bson[3]) with some method of compress to reduce the data file size.

Note: We will use `bagit` again later when working with the Roomba to record sensor and camera data.

Task 2: Determine Calibration

Now open a new `jupyter notebook` and input the first set of data. Use the template provided and determine the biases of the IMU.

[2]https://en.wikipedia.org/wiki/JSON
[3]https://en.wikipedia.org/wiki/BSON

Task 3: Correct for Biases

Once you have the biases, apply them to the second set of data and see if you get what you expect. Also plot the second set of data without the calibration data so see what an uncalibrated IMU gives you ... are the results noticeably bad?

Turn In

When you are done, print out and turn in your `jupyter notebook` showing *Task 2* and *Task 3*.