Lab 4: Inertial Measurement Units

1 Learning Objective

In this lab, you are going to:

1. Learn about the 3D accelerometer and rate gyroscope[1] sensors
2. Compute steps using the accelerometer sensor readings

2 Components Needed

- Raspberry Pi 3
- MPU 6050 IMU chip
- Connecting wires

3 Lab Activities

3.1 Preparation

You'll need to enable your Raspberry Pi's I2C interface in order to communicate with the MPU 6050. Start by running the command "`sudo raspi-config`". In the menu that appears, choose option 5, "Interfacing Options". Then choose option P5, "I2C", then pick the option to enable it.

With I2C enabled, you can now connect the IMU to the Pi. The necessary pins are highlighted on the right in cyan. Connect the pins as follows (Pi <-> IMU):

- 3V3 (1) <-> VIN (1)
- Ground (6) <-> GND (3)
- SDA1 (3) <-> SDA (5)
- SCL1 (5) <-> SCL (4)

Be sure to use relatively long wires for these connections, since you'll be moving around with the sensor throughout the lab 4.

You *MUST* solder the chip to the pins using soldering wires if the chip has no pins. If you don't know how to solder, your TA can show you.

[1] A "rate gyroscope" measures the *rate* of rotation about its axis, in contrast with a regular gyroscope

3.2 Reading Raw Sensor Data

You will need Internet connection to first install the Python bindings for the IMU by running:

```
sudo pip3 install adafruit-circuitpython-mpu6050
```

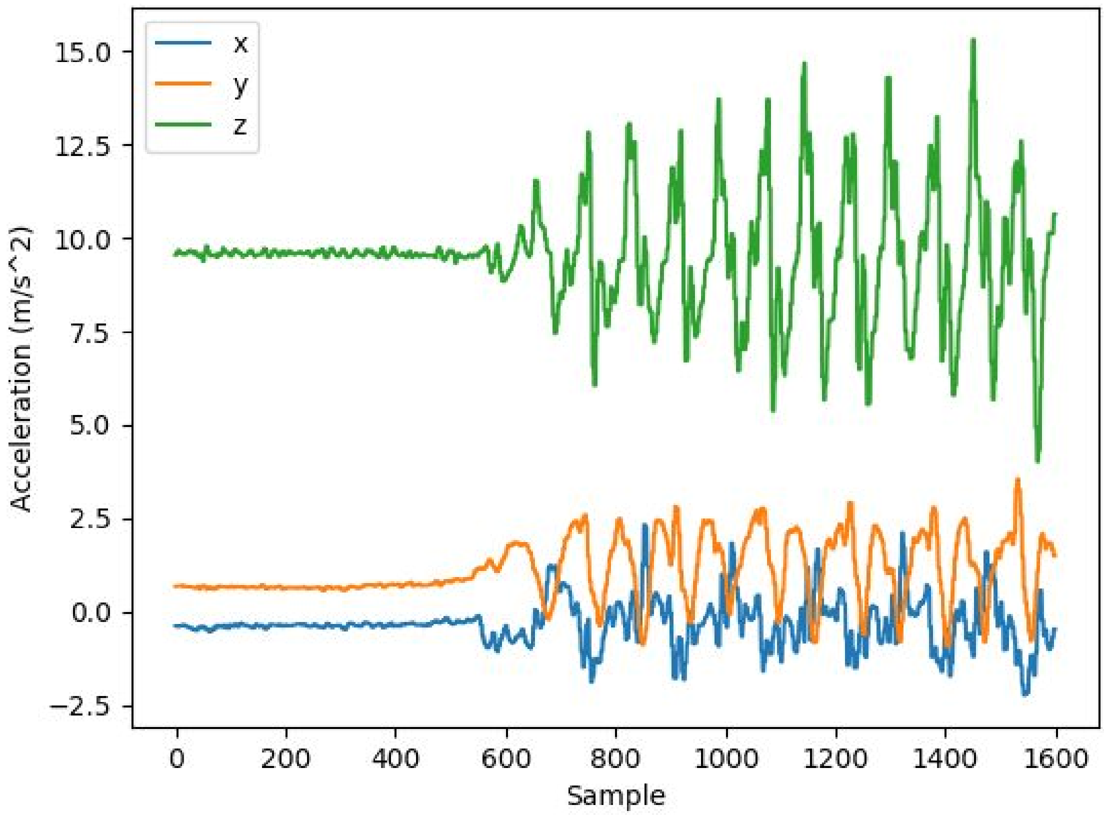
IMPORTANT: This library is only supported by Python 3. This means you need to use pip3 to install a package, and use `python3` to execute a python file.

Then create a new test Python script using the template:

```
import board
import busio
import adafruit_mpu6050

# perf_counter is more precise than time() for dt calculation
from time import sleep, perf_counter

i2c = busio.I2C(board.SCL, board.SDA)
mpu = adafruit_mpu6050.MPU6050(i2c)

# Write a loop to poll each sensor and print its axis values
while True:
    print("Acceleration: X:%.2f, Y: %.2f, Z: %.2f m/s^2" %
(mpu.acceleration))
    print("Gyro X:%.2f, Y: %.2f, Z: %.2f degrees/s" % (mpu.gyro))
    print("")
    sleep(0.1)
```

3.3 Detecting and Counting Steps Using Acceleration Signal

Human walking behavior presents strong periodic patterns. This periodicity is also illustrated in both velocity and acceleration, usually in all 3 axis directions. For example, someone’s foot will have a velocity close to 0 when it’s touching the ground, and in contrast, will have a peak velocity when it’s at the highest point above the ground. As you have learned in introductory physics, there is a differential relationship between acceleration and velocity. In the case above, the acceleration of someone’s foot would be close to 0 when the velocity is at a local max value (foot at highest point), and the acceleration would reach a local peak/valley when the velocity approaches zero (foot hitting/leaving ground). This periodicity exist as we keep walking. If we keep the sensor relatively still to our body, then the periodicity will be captured by the sensor, enabling us to detect and count the steps we have taken.



3-axis acceleration signal collected before and during walking

In this part of the lab, you will use the acceleration sensor to count the steps you have walked in real time. You need to walk for **50 steps**, record your walking as video, and show the output of your step counting program at the same time using screen recording. Your program should update the counting result as you walk.

It would be easier if you hold the sensor relatively still to your body, so that its measured acceleration is more close to your body acceleration. You may remove the acceleration component introduced by gravity, or choose to keep it. It will be beneficial to the step counting accuracy if you apply noise filters to remove the measurement noise. You need to do some experiments to improve your algorithm for a higher accuracy. Keep trying and see if you can reach an accuracy of 90% or even higher!

4 Questions to Explore

1. How frequently can the sensor be polled? Is this fast enough for this lab? Why or why not?
2. Did you apply filter(s) to sensor readings? If yes, how did you use the filter(s)? If no, why?
3. Explain your step counting algorithm and its accuracy.

5 What to Turn In

- Answers to the questions (4)
- Attach the code at the end of the lab report (1)
- Video showing both your walking and program output (5)