Lab Report 1

Group members:
Jean Gonzalez, Jason Phan

Experiment time: 12:00PM-1:50PM

Lab room: PFT 2280

Platform:
OMAP-L138 eXperimenter Kit and Code Composer Studio

The contents of experiment:
Used Pycharm/VSCode to to perform peak detection and frequency analysis with FFT using sample accelerometer and gyroscope data.

The principle of experiment:
The FFT decomposes a time-domain signal into the frequency domain using an optimized version of the DFT using frequency bins, useful for detecting frequencies where motion is strongest. Peak detection finds local maxima to detect events of significant motion.

The code:

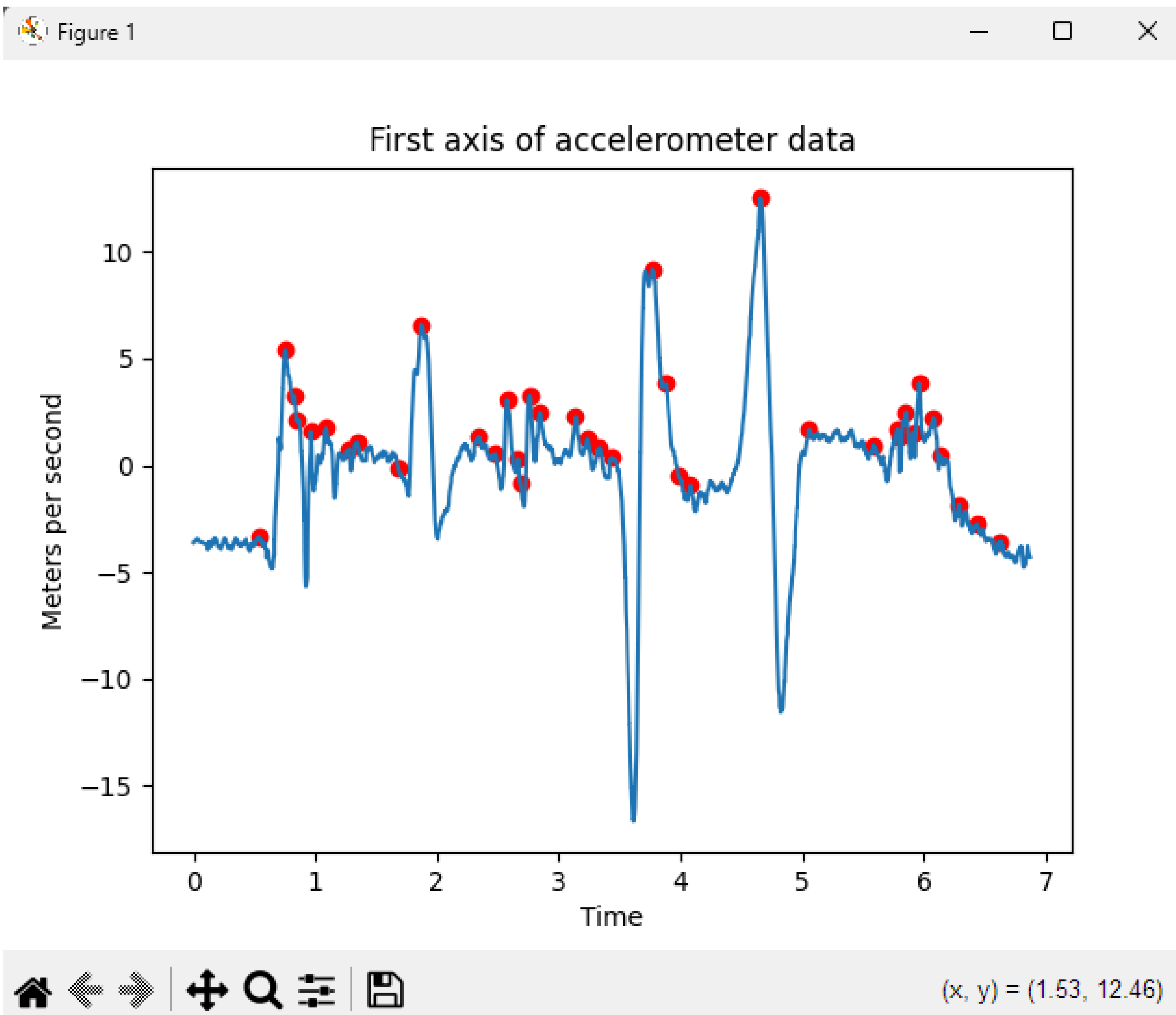
```
import numpy as np
import matplotlib.pyplot as plt

csv_filename = 'sample_sensor_data.csv'
data = np.genfromtxt(csv_filename, delimiter=',').T

timestamps = (data[0] - data[0, 0]) / 1000

accel_data = data[1:4]
gyro_data = data[4:-1]

#find all peaks above threshold and saves certain peaks within a tolerance
range
def peak_detection(t, sig, thresh, tol = 1):
   peaks = []
   max_val = -np.inf
   N = len(sig)
   for i in range(1,N-1):
       #save signal by checking neighboring signal
       if sig[i] > sig[i-1] and sig[i] >= sig[i+1]:
           if sig[i] > max_val:
               max_val = sig[i]
               position = t[i]
       #check if signal drop significantly
       if max_val > thresh and sig[i] < max_val - tol:
```

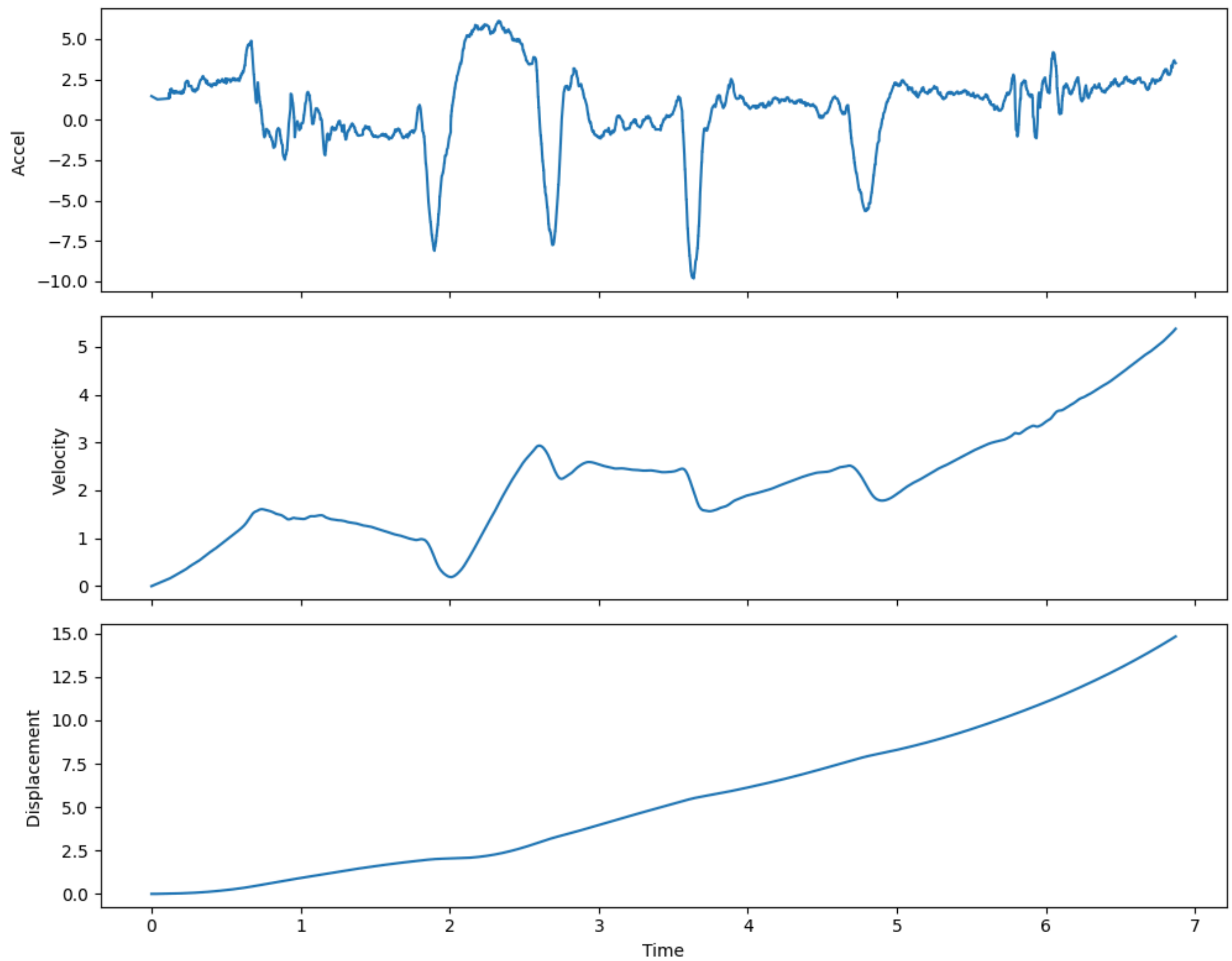
```
          peaks.append((position, max_val))
          max_val = -np.inf

   return np.array(peaks)


max_peaks = peak_detection(timestamps, accel_data[0], thresh = -15)

plt.plot(timestamps, accel_data[0])
plt.title("First axis of accelerometer data")
plt.xlabel("Time")
plt.ylabel("Meters per second")
plt.scatter(max_peaks[:, 0], max_peaks[:, 1], color='red')
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft, fftfreq

csv_filename = 'sample_sensor_data.csv'
data = np.genfromtxt(csv_filename, delimiter=',').T

timestamps = (data[0] - data[0, 0]) / 1000

accel_data = data[1:4]
gyro_data = data[4:-1]
dt = np.diff(timestamps,prepend=timestamps[0]) #time intervals
velocity = np.cumsum(accel_data*dt,axis=1) #integration of acceleration
displacement = np.cumsum(velocity*dt,axis=1) #integration of velocity
axis = 1




fig, axes = plt.subplots(3, 1, sharex=True, figsize=(10, 8))
axes[0].plot(timestamps, accel_data[axis])
axes[0].set_ylabel('Accel ')

axes[1].plot(timestamps, velocity[axis])
axes[1].set_ylabel('Velocity ')

axes[2].plot(timestamps, displacement[axis])
axes[2].set_ylabel('Displacement ')

axes[-1].set_xlabel('Time')
plt.tight_layout()
plt.show()

fig,axes = plt.subplots(3,1,figsize=(10,8))
```

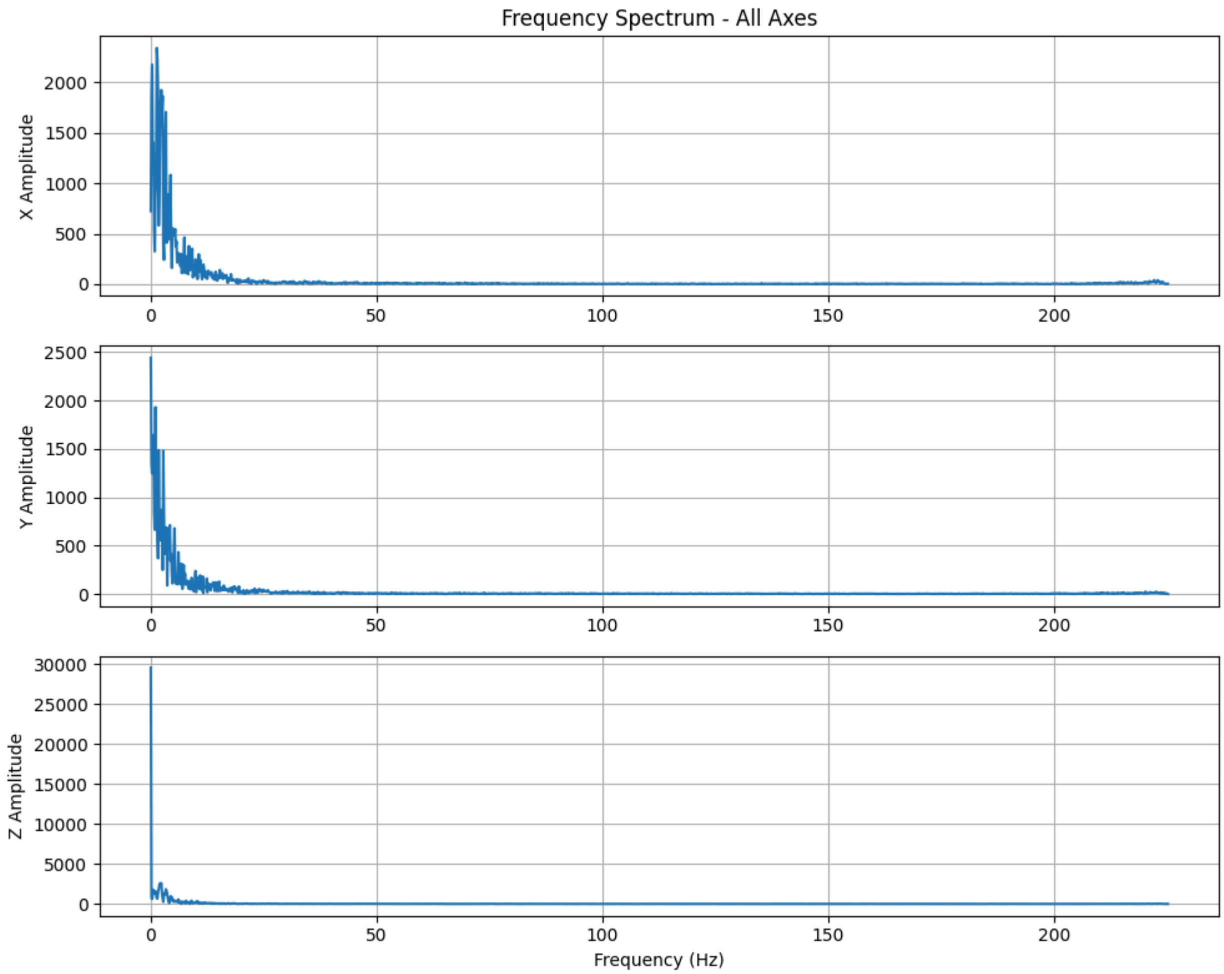
```
labels = ['X','Y','Z']
N=accel_data.shape[1] #label for 3 axes
sampling_rate = 1/ np.mean(np.diff(timestamps)) #sampling freq(Hz)
freqs = fftfreq(N, d=1/sampling_rate) #frequency bins
#plot all fourier transform on 3 axes
for i in range(3):
   fft_magnitude = np.abs(fft(accel_data[i])) #compute FFT
   axes[i].plot(freqs[:N // 2], fft_magnitude[:N // 2]) #plot positive freq
   axes[i].set_ylabel(f'{labels[i]} Amplitude')
   axes[i].grid(True)
axes[0].set_title('Frequency Spectrum - All Axes')
axes[-1].set_xlabel('Frequency (Hz)')
plt.tight_layout()
plt.show()
```

The experiment results:





Accel
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Velocity
5
4
3
2
1
0
Displacement
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
1
2
3
4
5
6
7
Time



The problems you met in the experiment:

The understanding of the experiment:
Peak detection demonstrates events of significant motion at certain time intervals. Since accelerometer data is 3-D, data is often decomposed into three 3-D subplots to visualize varying data in all directions. It can be shown through numerical integration that other useful quantities can be derived such as velocity and displacement to conclude that drifting is common with increasing data points as time increases. Additionally, FFT of the data shows the frequency where significant motion is concentrated.