

Yoseph Ayele Kebede

Dr. Mitachell

ENPM701 Assgnm #1

Moving Average

### Assignment #1

*Loading sensor data with Python/Numpy and plotting/analyzing data w/ Matplotlib  
Function: Moving Average*

#### Question #1

(Below is the commented code used to solve assignment)

+++++

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

#### # Question 1

##### # 1) Load imu data as string

```
imudata = np.genfromtxt("imudata.txt", dtype=str)
```

##### # 2) Plot the pitch angle raw data

##### # Obtain shape of data/array

```
size = imudata.shape
```

##### # Extract pitch angle raw data (5th column)

```
pitch = imudata[:,4].astype(np.int64)
```

##### # Create horizontal x axis for plot

```
x = np.linspace(0,size[0],num=size[0],endpoint=False,dtype=int)
```

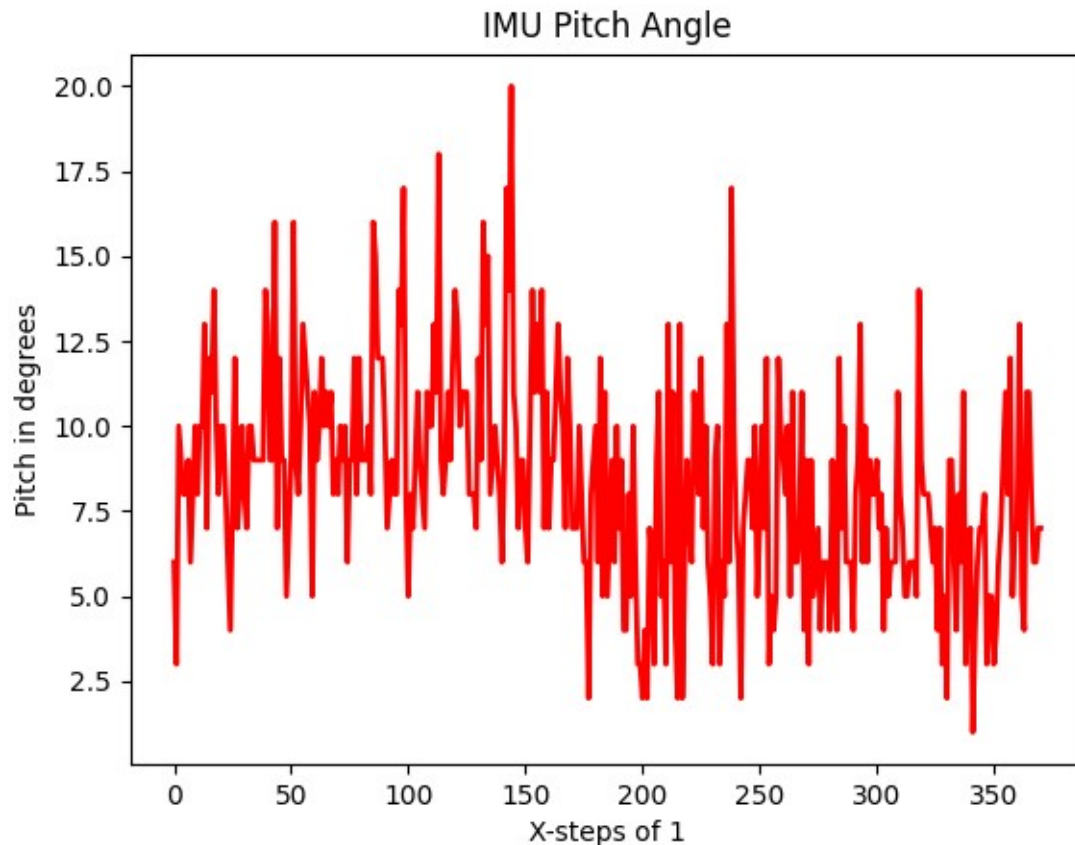
##### # Plot pitch angle with respect to horizontal step 0 - imusize

```
fig, ax1 = plt.subplots()
ax1.plot(x,pitch,ls='solid', color='red',linewidth=2, label='pitch-raw-data')
```

##### # 3) Label the axes, title, legend

```
ax1.set(title="IMU Pitch Angle",
        ylabel="Pitch in degrees",
        xlabel="X-steps of 1")
```

```
plt.show()
```



```
plt.savefig('pitch-plot.png')
plt.close()
```

#### # 4) Write moving average function

```
def movave(pitch, window):
```

```
    # Create a list to save average of kernel
```

```
    # as it strides over list
```

```
    tempave = []
```

```
    # Loop the kernel through list in a nested
```

```
    # loop
```

```
    for i in range(pitch.size):
```

```
        # Break if window steps outside of list
```

```
        if (i+window) >= (pitch.size-1) and (pitch.size-1-i>window//2):
```

```
            # Take average with the valid kernel cells in list
```

```
            k = pitch.size-1-i
```

```
            # Slice list based on new kernell
```

```
            tmp = pitch[i:i+k]
```

```
    elif i==0:
```

```
        # Ignore the kernell cells outside list
```

```
        j = window//2
```

```
        # Slice list based on new kernell
```

```
        tmp = pitch[i:i+j]
```

```
    elif i>0 and i<=(window//2):
```

**# Slice list based on new kernell**

tmp = pitch[0:i+window//2]

else:

**# slice the list based on window size**

tmp = pitch[i:i+window]

**# Take average of sliced array**

tmp\_avg = np.average(tmp)

**# Save average obtained from kernel**

tempave.append(tmp\_avg)

return tempave

**# Set kernel size to perform move average**

**# Node needs to be odd rememeber**

krn = [2, 4, 8, 16, 64, 128]

colors=['b','g','r','k','m','c']

**# Counter**

idx=0

**# 5) plot averaged data over raw data**

for k in krn:

**# Call movave method and save new list for k window size**

movave\_lst = movave(pitch,k)

**# print(movave\_lst)**

mave = np.array(movave\_lst)

**# 6) Calculate mean and std of average data**

mn = np.average(movave\_lst)

std = np.std(movave\_lst)

**# Printing mean and std to the terminal**

print("The mean of the averaged data is: ", mn)

print("The standard deviation of the averaged data is: ", std)

**# Label moving average kernel**

kstr = str(k)

**# Plot curve**

fig2, ax = plt.subplots()

ax.plot(x,pitch,ls='solid', color='red',linewidth=2, label='pitch-raw-data')

ax.plot(x,movave\_lst,ls='solid', color=colors[idx],linewidth=2, label='ave-data, k='+kstr)

**# Label the axes, title, legend**

ax.set(title="Averaged IMU Pitch Angle with 1D Kernel size "+kstr+"-pt",

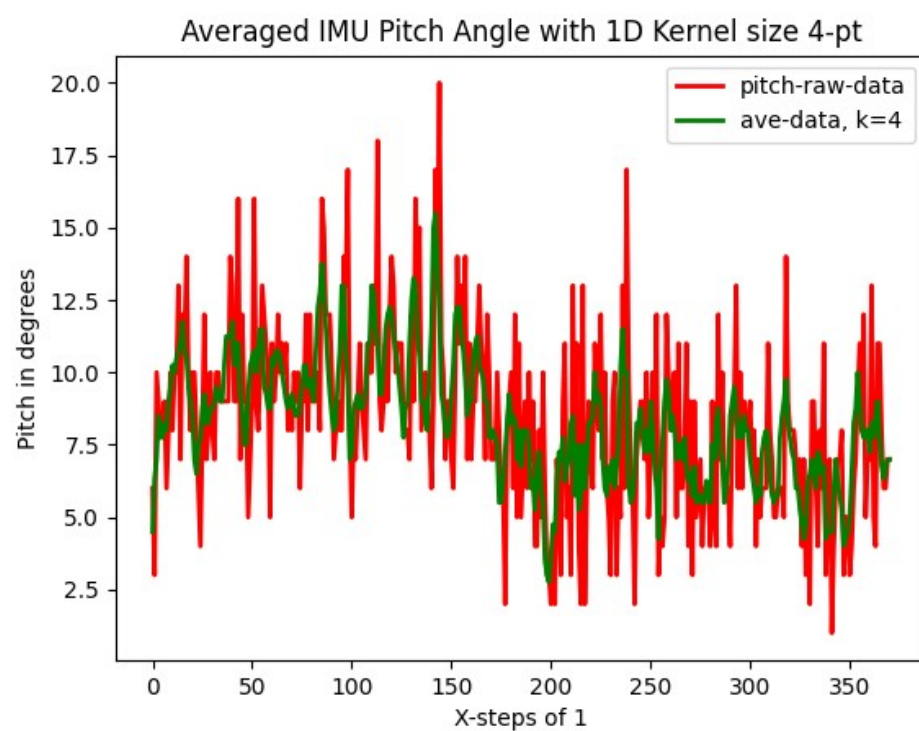
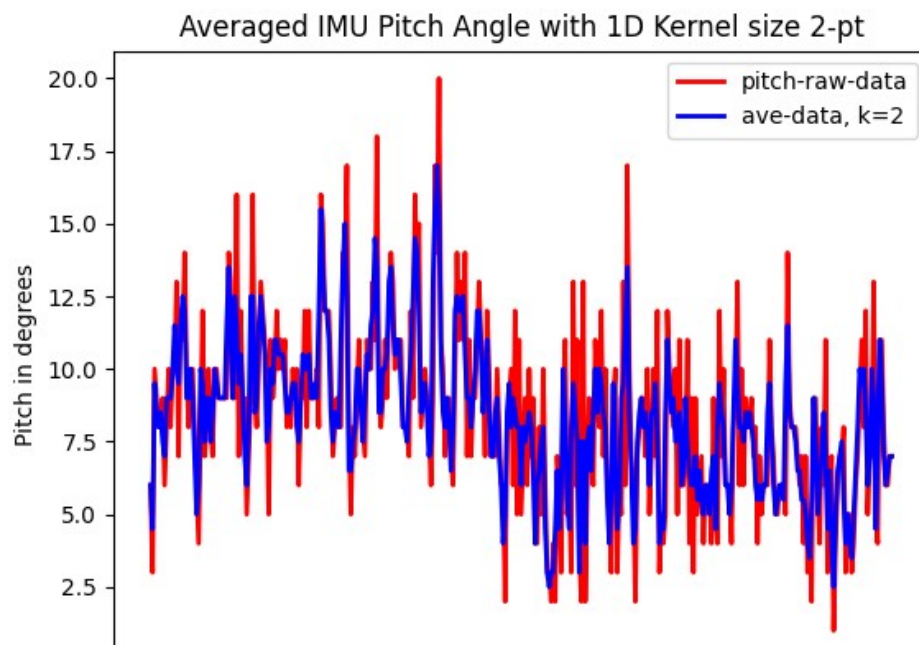
```
ylabel="Pitch in degrees",  
xlabel="X-steps of 1")
```

```
ax.legend(loc='best')
```

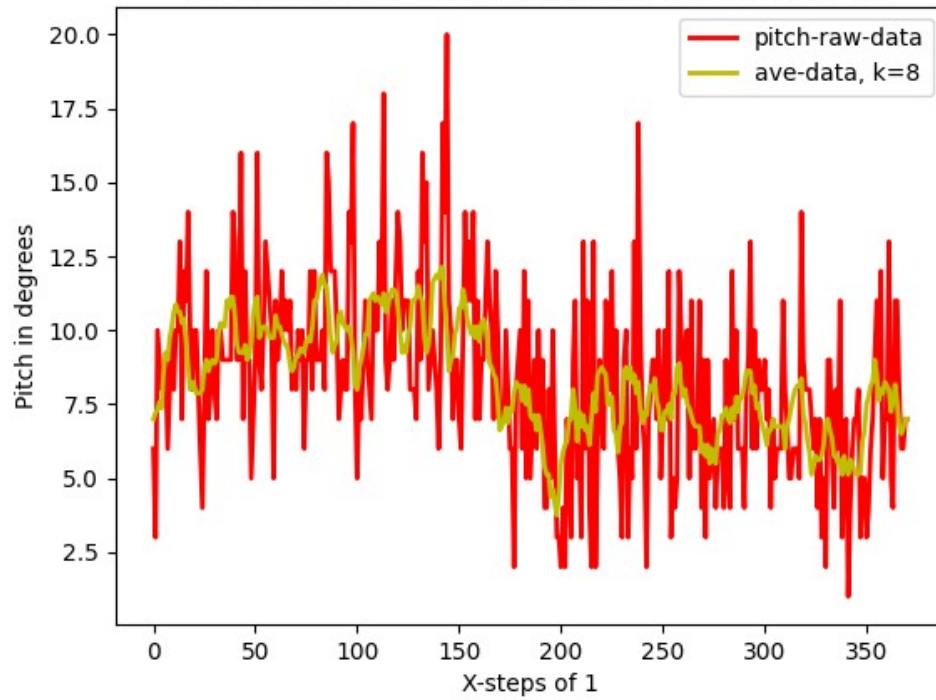
```
plt.savefig('movave-%d-pt-plot.png'% k)  
plt.show()  
plt.close()
```

```
# increment counter for color list
```

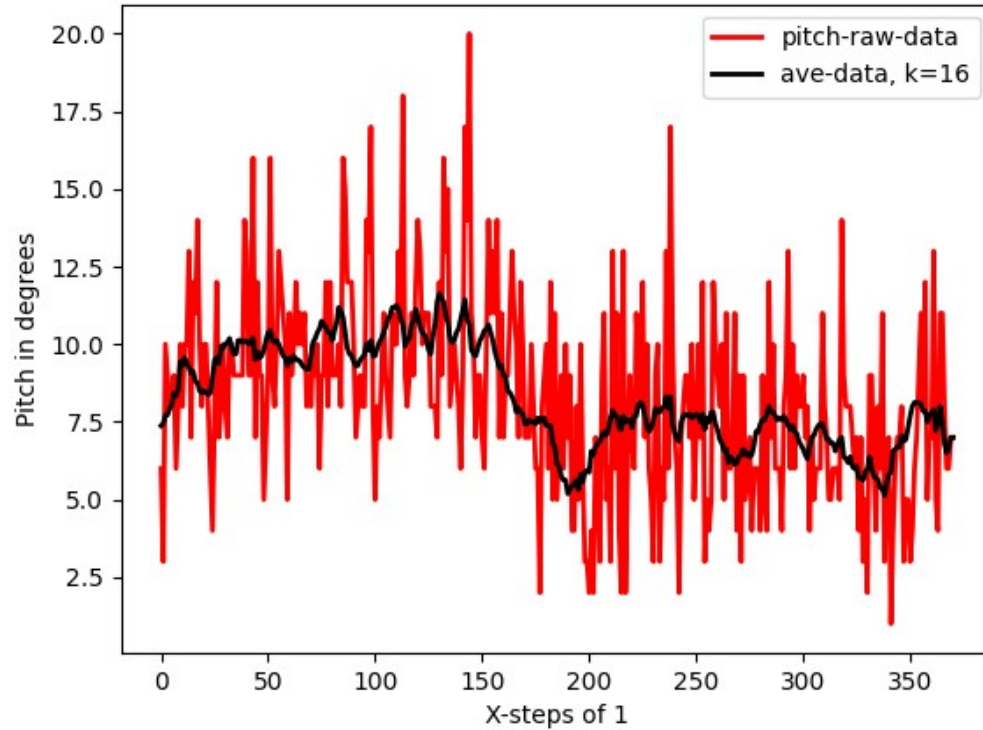
```
idx+=1
```

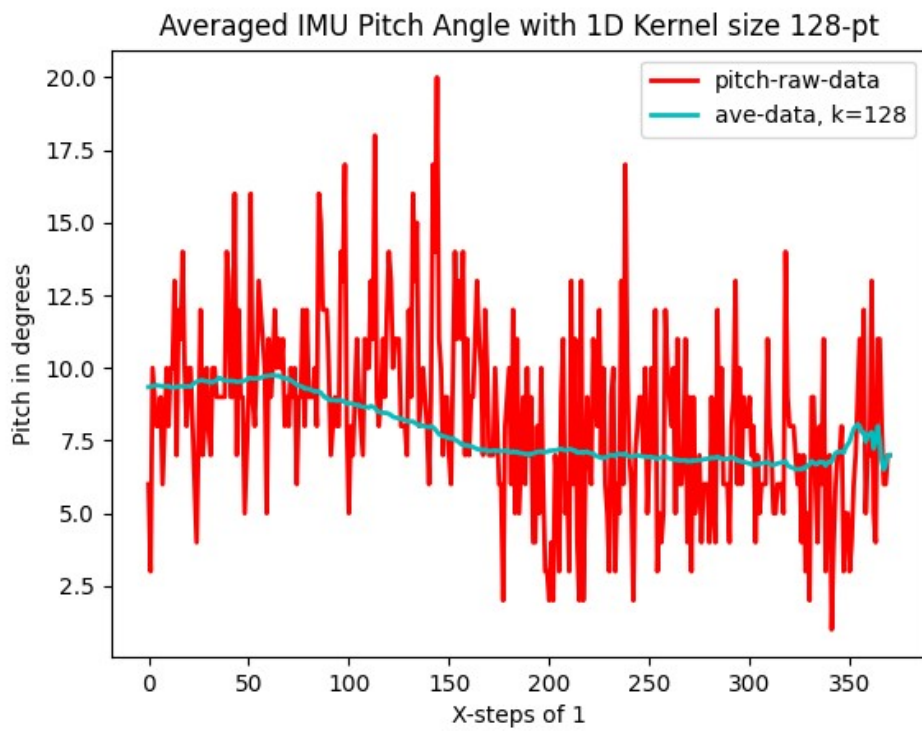
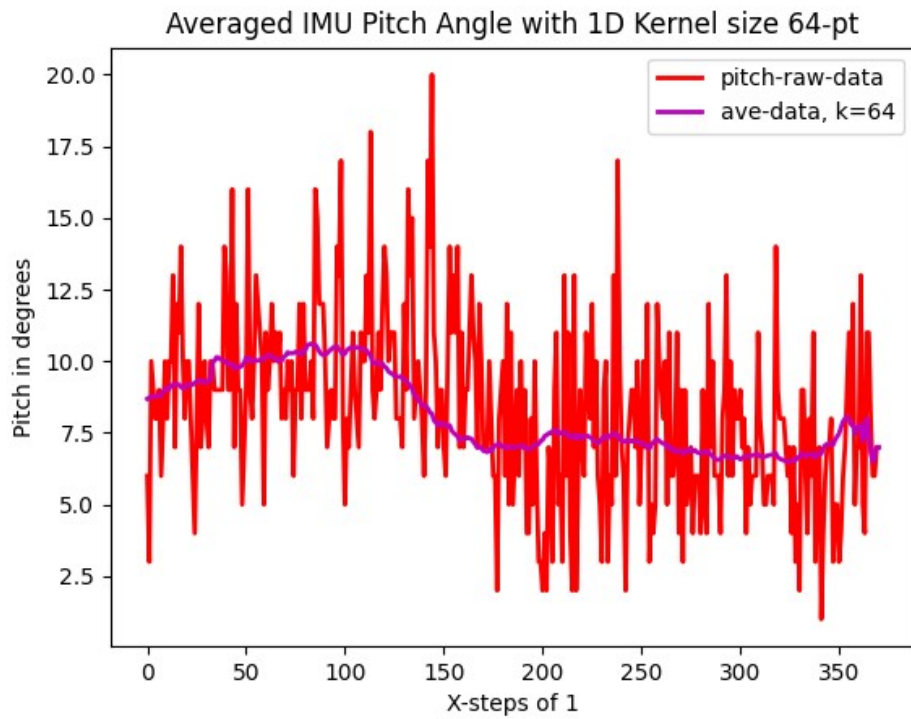


Averaged IMU Pitch Angle with 1D Kernel size 8-pt



Averaged IMU Pitch Angle with 1D Kernel size 16-pt





+++++

**The End**