

```
In [1]: import numpy as np
import matplotlib.pyplot as plt

print('Packages imported!')
```

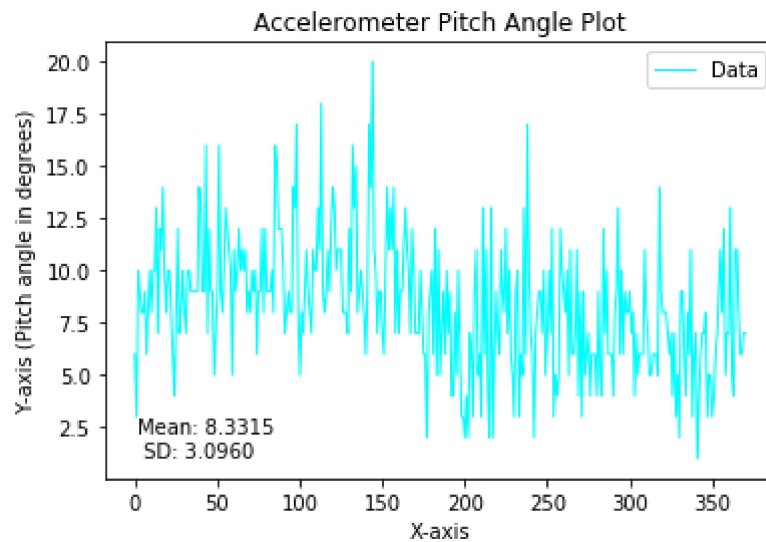
Packages imported!

```
In [2]: data = np.genfromtxt('imudata.txt', delimiter=' ')
print('Pitch angle - 5th Column of IMU Data')
print(data[:,4])
```

Pitch angle - 5th Column of IMU Data

```
[ 6.  3. 10.  9.  8.  8.  9.  6.  8. 10.  8. 10. 10. 13.  7. 12. 11. 14.
 10.  8. 10. 10.  8.  6.  4.  8. 12.  7.  8. 10.  8.  7. 10. 10.  9.  9.
  9.  9.  9. 14. 13.  9.  9. 16.  7. 12.  9.  9.  5.  7.  9. 16.  9.  8.
 10. 13. 12. 11. 10.  5. 11.  9. 10. 12. 10. 11. 10. 11.  8.  9.  8. 10.
  9. 10.  6.  9.  9. 12.  8. 12.  9.  9.  9. 10.  8. 16. 15. 12. 12. 12.
 10.  7.  8.  9.  8.  8. 14. 13. 17.  8.  5.  8.  7.  9. 11.  9.  8.  7.
 11. 10. 10. 13. 11. 18.  9.  8.  9. 11.  9. 12. 14. 13. 10. 11. 11. 11.
  8.  8.  8.  7. 12.  9. 16. 13. 15.  8.  9. 10.  9.  8.  6.  9. 17. 14.
 20. 11. 10.  7.  9.  9.  7.  6. 10. 14. 11. 13. 11. 14.  7. 11.  7.  9.
  9. 11. 13. 11. 10.  7. 12. 10.  7.  7.  7. 10.  8.  6.  6.  2.  8.  9.
 10.  6. 12.  5. 11.  5.  7.  9.  6. 10.  7.  9.  4.  4.  8.  5. 10.  6.
  3.  3.  2.  4.  2.  7.  6.  3.  9. 11.  5.  6.  3. 13.  6. 11.  4.  2.
 13.  2.  6.  9.  7.  6. 11.  9.  8. 12.  7. 10.  6.  5.  3.  9. 10.  3.
  6.  5. 13.  6. 17. 10.  7.  6.  2.  7.  8.  9.  9.  7. 10.  5.  7. 10.
  7. 12.  3.  5.  4.  5. 12. 10.  9.  8. 10.  5. 11.  6.  6.  7. 11.  4.
  9.  3.  9.  5.  6.  7.  4.  6.  6.  6.  4.  9.  5.  4. 12.  7. 10.  6.
  6.  6.  4.  8.  9. 13.  6. 10.  6.  9.  8.  8.  9.  7.  8.  4.  7.  5.
  6.  6.  6. 11.  8.  7.  5.  5.  6.  6.  6.  5. 14.  9.  8.  8.  8.  7.
  6.  7.  4.  7.  3.  5.  2.  9.  9.  6.  4.  8.  6. 11.  3.  6.  7.  1.
  4.  6.  7.  7.  8.  3.  5.  5.  3.  4.  6.  7.  9. 11.  8. 12.  5.  7.
  7. 13.  5.  4. 11. 11.  8.  6.  6.  7.  7.]
```

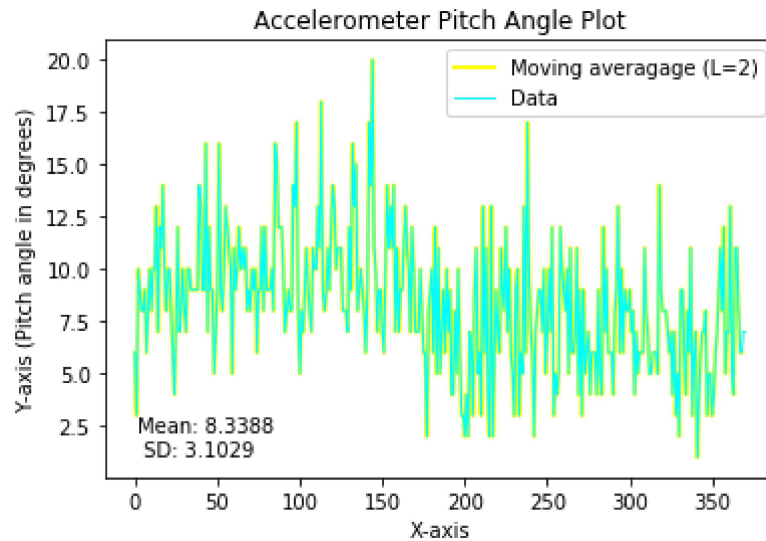
```
In [3]: plt.plot(data[:,4], label='Data', color='cyan', linewidth=1)
txt = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(data[:,4]), np.std(data[:,
4]))
plt.annotate(txt, xy=(0.05, 0.05), xycoords='axes fraction')
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()
```



```
In [4]: def moving_average(data, window=2):
dat = np.copy(data)
size = window-1
for i in range(dat.shape[0]-size):
    dat[i] = np.mean(dat[i:i+size])
return dat[0:dat.shape[0]-window]
```

```
In [5]: avg_2 = moving_average(data[:,4], 2)
txt_2 = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(avg_2), np.std(avg_2))
plt.annotate(txt_2, xy=(0.05, 0.05), xycoords='axes fraction')

plt.plot(avg_2, color='yellow', label='Moving averagage (L=2)', linewidth=2)
plt.plot(data[:,4], color='cyan', label='Data', linewidth=1)
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()
```

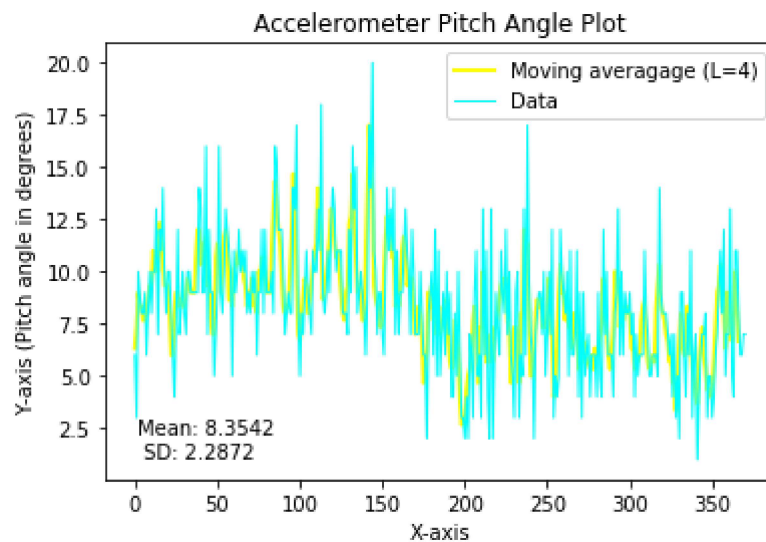


```

In [6]: avg_4 = moving_average(data[:,4], 4)
txt_4 = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(avg_4), np.std(avg_4))
plt.annotate(txt_4, xy=(0.05, 0.05), xycoords='axes fraction')

plt.plot(avg_4, color='yellow', label='Moving averagage (L=4)', linewidth=2)
plt.plot(data[:,4], color='cyan', label='Data', linewidth=1)
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()

```

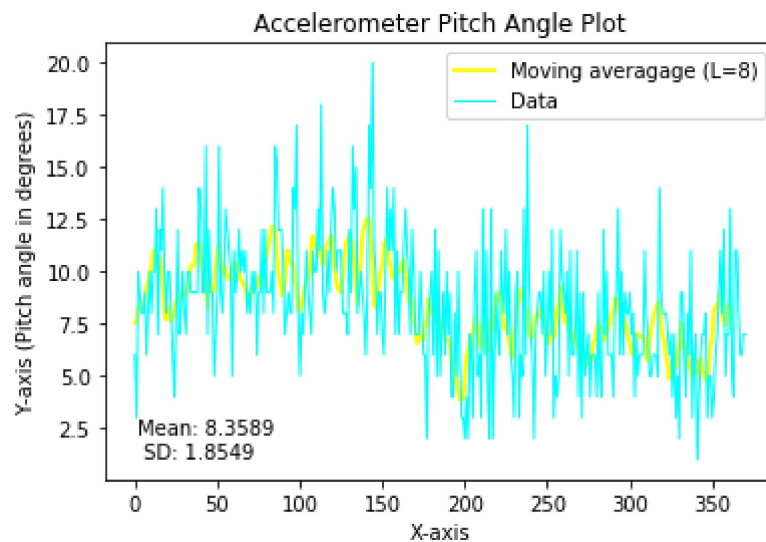


```

In [7]: avg_8 = moving_average(data[:,4], 8)
txt_8 = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(avg_8), np.std(avg_8))
plt.annotate(txt_8, xy=(0.05, 0.05), xycoords='axes fraction')

plt.plot(avg_8, color='yellow', label='Moving averagage (L=8)', linewidth=2)
plt.plot(data[:,4], color='cyan', label='Data', linewidth=1)
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()

```

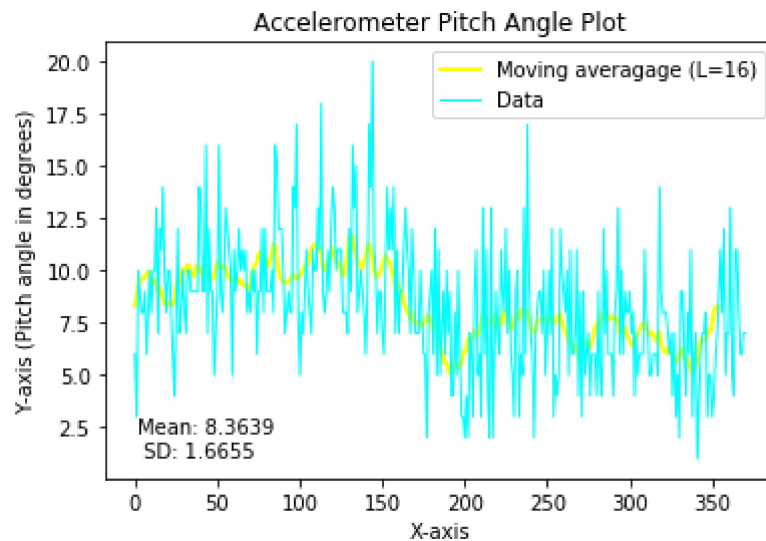


```

In [8]: avg_16 = moving_average(data[:,4], 16)
txt_16 = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(avg_16), np.std(avg_16))
plt.annotate(txt_16, xy=(0.05, 0.05), xycoords='axes fraction')

plt.plot(avg_16, color='yellow', label='Moving averagage (L=16)', linewidth=2)
plt.plot(data[:,4], color='cyan', label='Data', linewidth=1)
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()

```

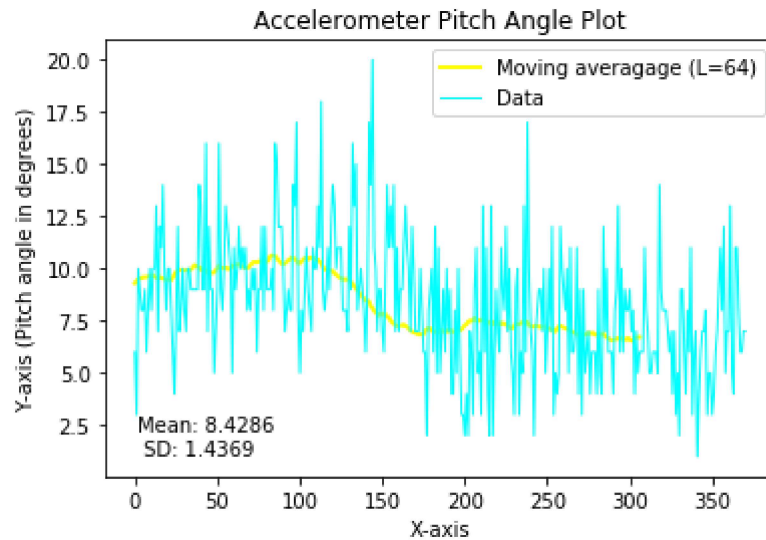


```

In [9]: avg_64 = moving_average(data[:,4], 64)
txt_64 = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(avg_64), np.std(avg_64
))
plt.annotate(txt_64, xy=(0.05, 0.05), xycoords='axes fraction')

plt.plot(avg_64, color='yellow', label='Moving averagage (L=64)', linewidth=2)
plt.plot(data[:,4], color='cyan', label='Data', linewidth=1)
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()

```



```

In [10]: avg_128 = moving_average(data[:,4], 128)
txt_128 = 'Mean: {0:.4f} \n SD: {1:.4f}'.format(np.mean(avg_128), np.std(avg_128))
plt.annotate(txt_128, xy=(0.05, 0.05), xycoords='axes fraction')

plt.plot(avg_128, color='yellow', label='Moving averagage (L=128)', linewidth=2)
plt.plot(data[:,4], color='cyan', label='Data', linewidth=1)
plt.xlabel('X-axis')
plt.ylabel('Y-axis (Pitch angle in degrees)')
plt.title('Accelerometer Pitch Angle Plot')
plt.legend()
plt.show()

```

