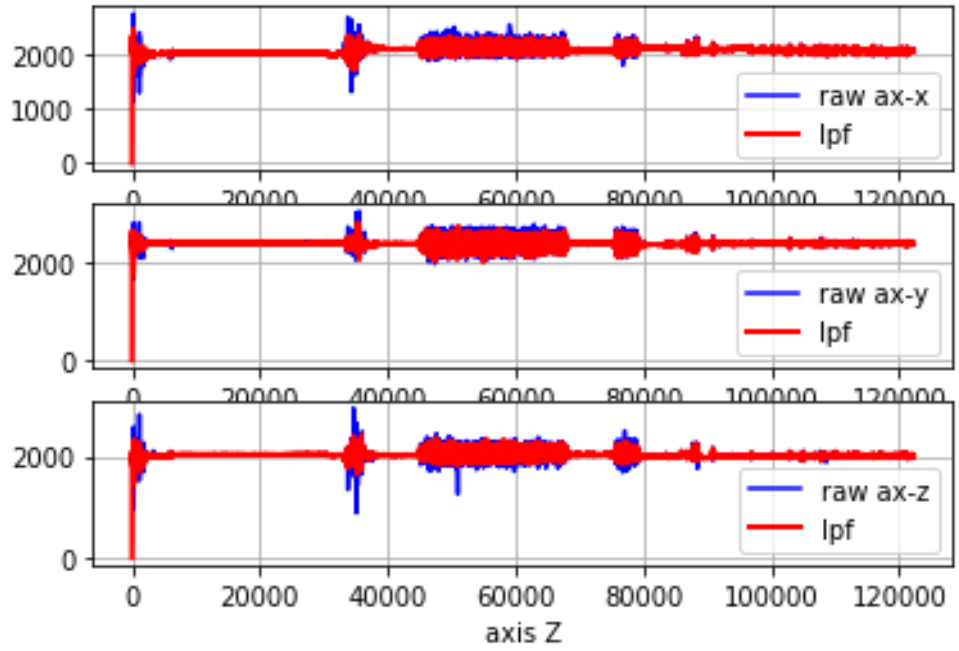
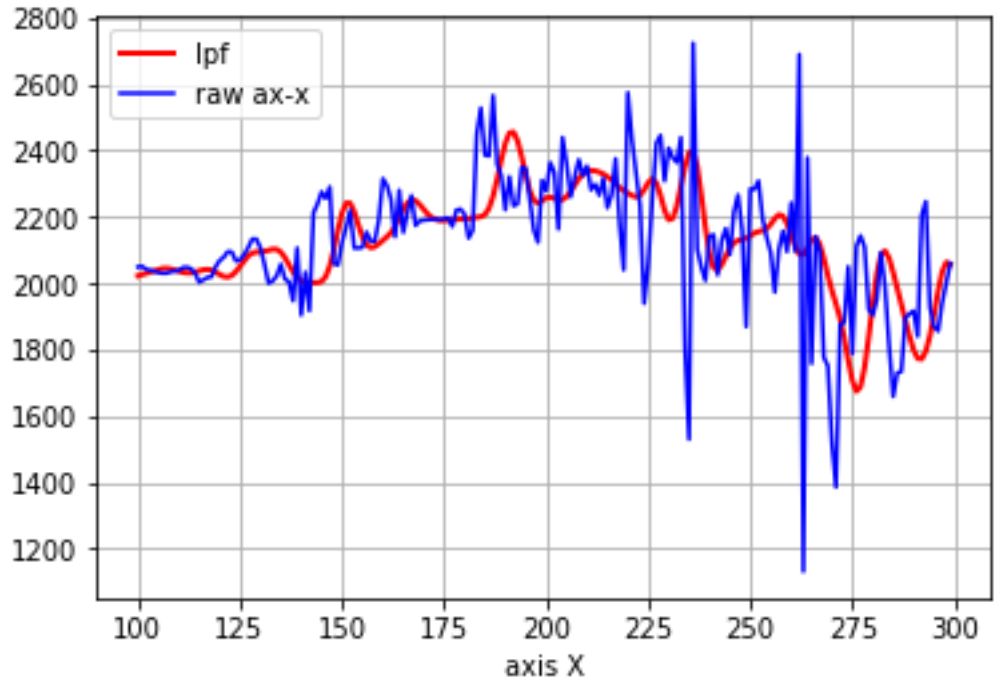
Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02

## 1.Plot the raw data and compared to LPF and MVA Filter.

LPF

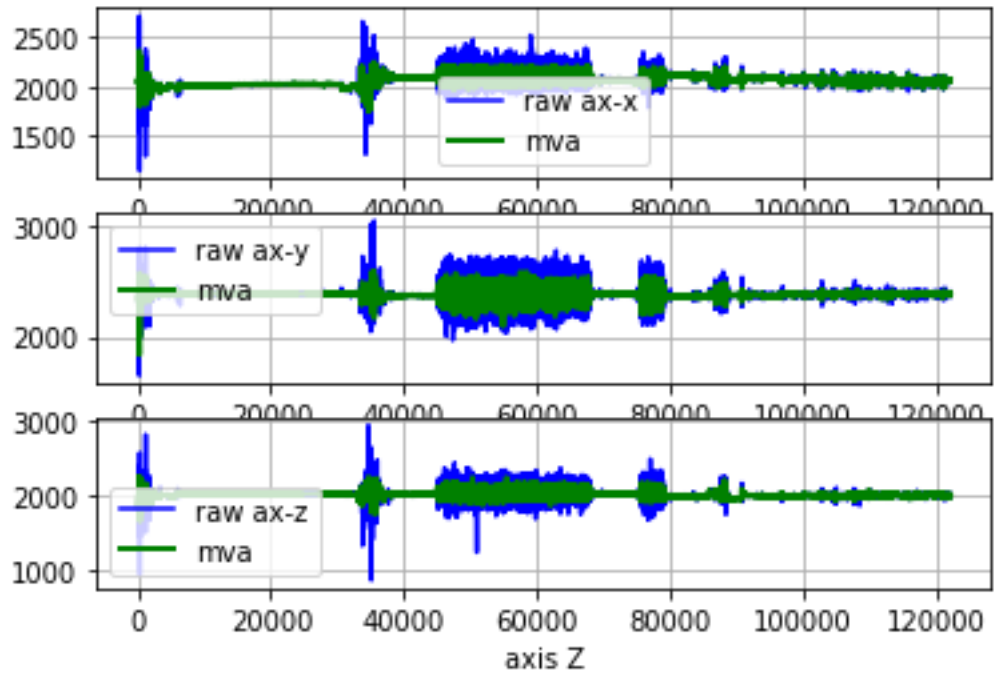



Karena data terlalu banyak dan sulit diamati, maka compare data akan mengambil 200 dari data dan untuk axis-x

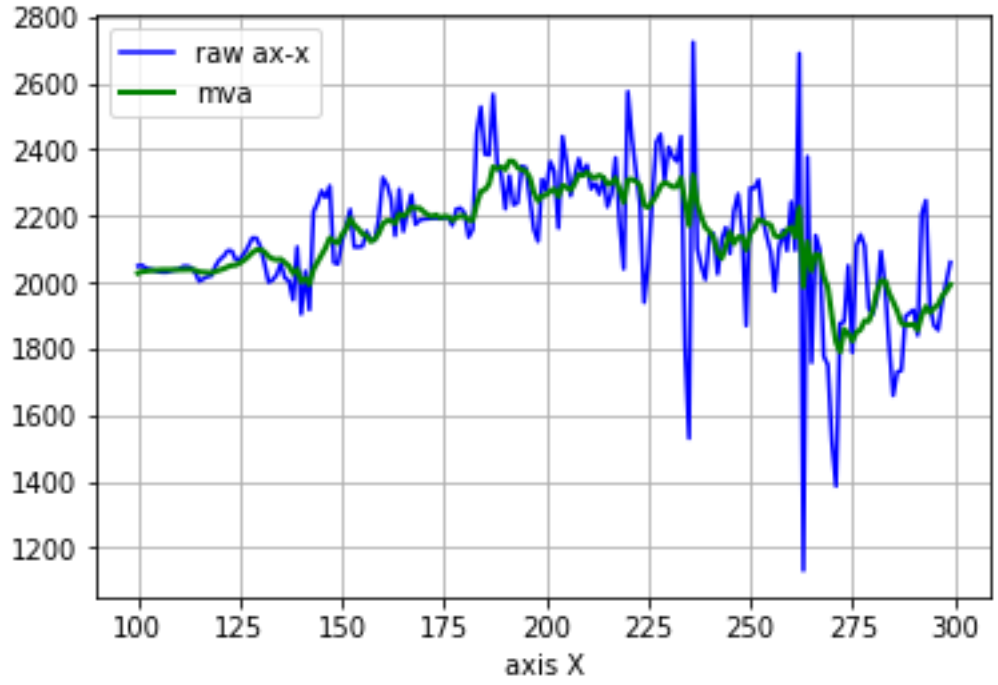


Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02

MVA

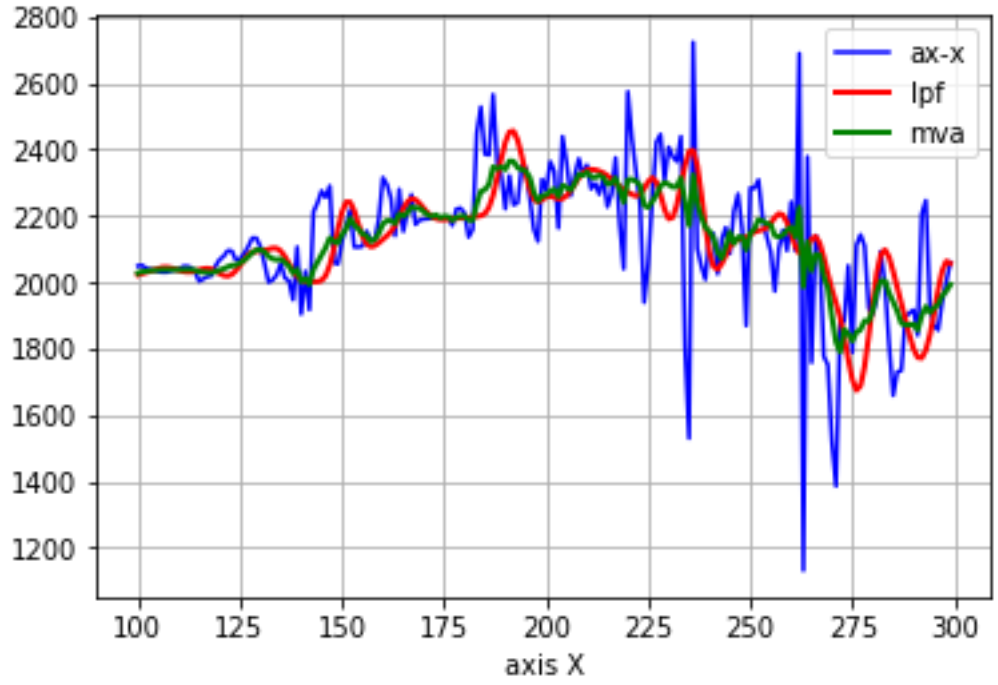



Karena data terlalu banyak dan sulit diamati, maka compare data akan mengambil 200 dari data dan untuk axis-x



Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02

Perbandingan MVA dan LPF pada raw data dalam 200 data pada axis x



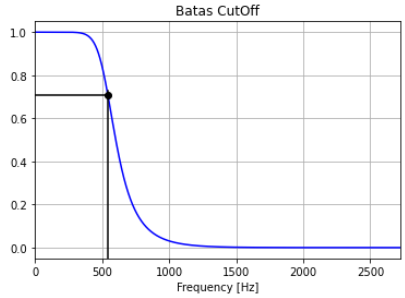
Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02

2. Which frequency cut-off (fc) do you use for LPF? How do you determine that fc? (please provide a plot of filtered data regarding the frequency cutt-offyou used)

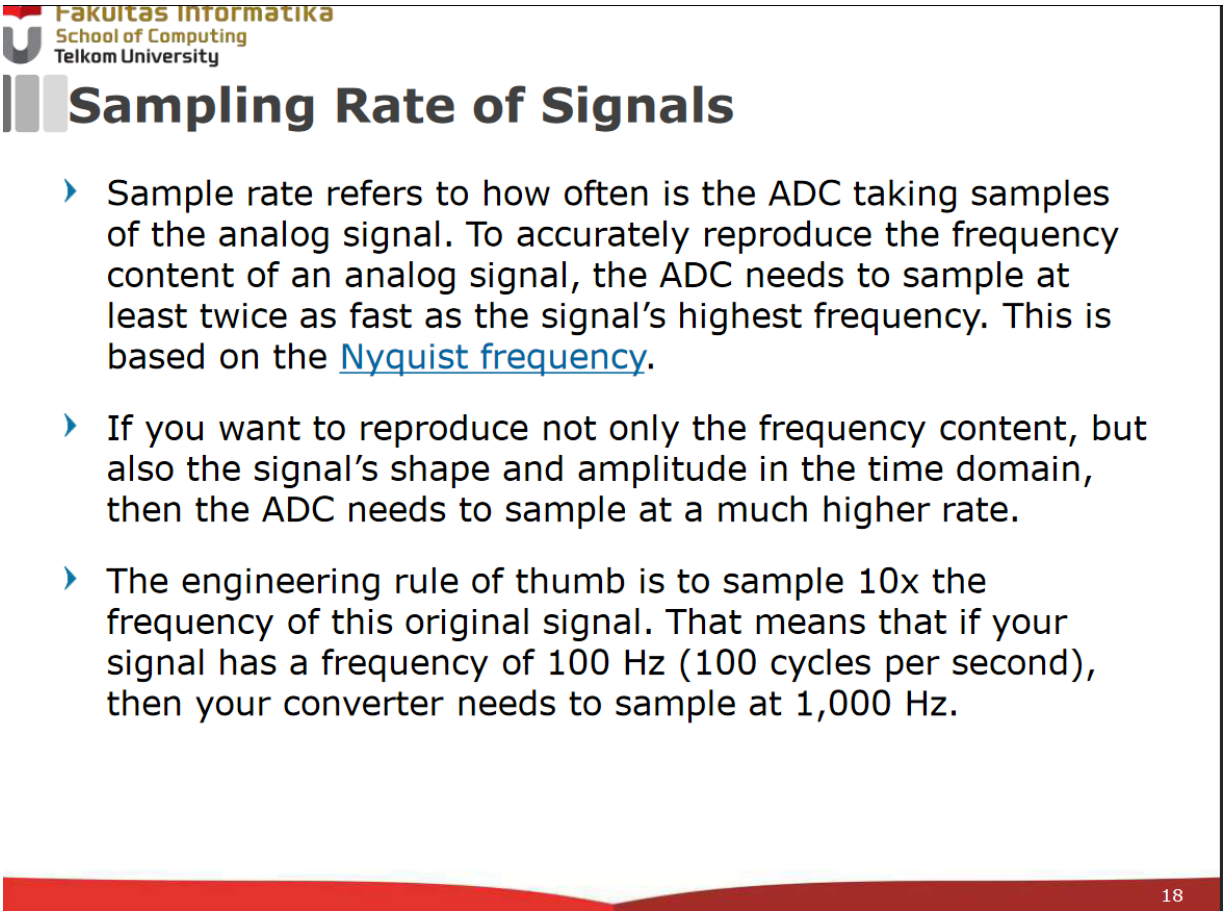
```
data = read_csv('4.csv', header=0, index_col=0)
data.describe()
```

| | 2045 | 2326 | 1833 | 1 |
|---|---|---|---|---|
| count | 122200.000000 | 122200.000000 | 122200.000000 | 122200.000000 |
| mean | 2053.151244 | 2377.315663 | 2024.088453 | 3.904984 |
| std | 51.148140 | 56.854621 | 46.606461 | 2.313783 |
| min | 1132.000000 | 1649.000000 | 878.000000 | 0.000000 |
| 25% | 2019.000000 | 2369.000000 | 2007.000000 | 1.000000 |
| 50% | 2053.000000 | 2380.000000 | 2025.000000 | 4.000000 |
| 75% | 2081.000000 | 2387.000000 | 2034.000000 | 7.000000 |
| max | 2723.000000 | 3046.000000 | 2954.000000 | 7.000000 |

```
fs = 5446.0        # sample rate
fc = 544.6         # cutoff
```




Untuk axis x cutoff yang digunakan adalah 544.6 Hz , angka tersebut didapat dari nilai sinyal original yang merupakan 1/10 sampel . Sampel sendiri diambil dari 2 kali nilai sinyal tertinggi pada data.



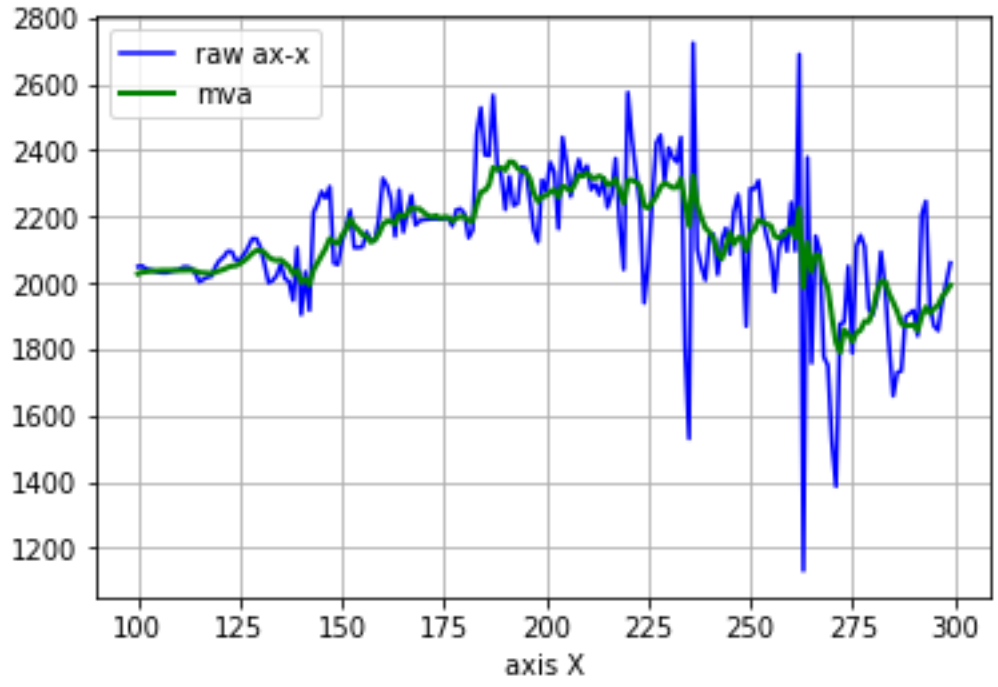
Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02

3.What is the windows size you used in MVA ? (please select 3 different windows size, and then plot and compared the results). Which windows size does give smooth data plot? Why?

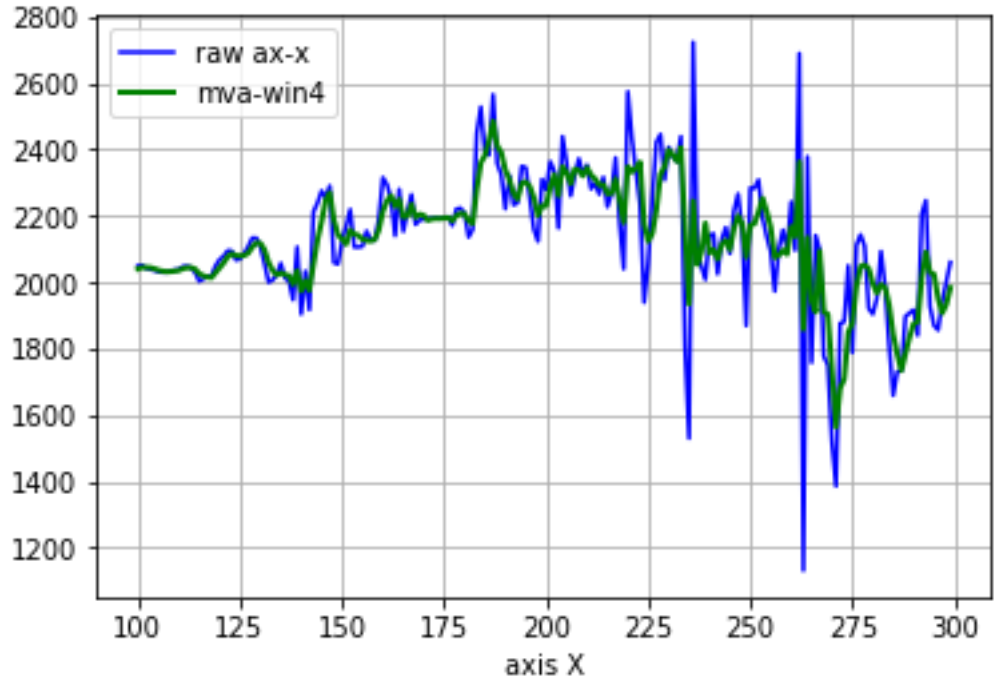
```
###MVA
window = 10
def moving_average(_data,window):
  result = []
  data = _data
  for a in range(window):
    data = np.insert(data,0,data[0])
  for i in range(window,len(data)):
    x = data[i]
    for j in range(window):
      x += data[i-j]
    avg = x/(window+1)
    result.append(avg)

  return result
```

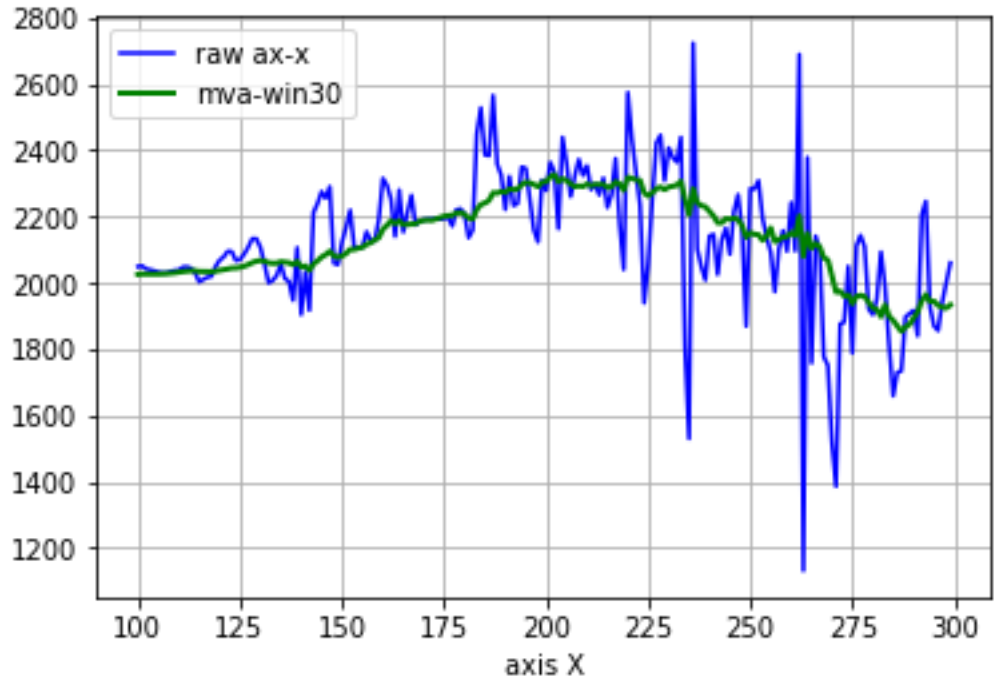
Untuk window yang saya gunakan adalah 10 frequensi data ke belakang karena dianggap lebih smooth dimana tidak ada data pending yang terlewat, dan noise tidak terlalu banyak, namun saya juga mencoba beberapa window size lain yaitu 4 dan 30 . berikut adalah window size 10



Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02




Untuk window berukuran 4 data sebelum nya dirasa masih banyak noise yang ditangkap seperti pada freq waktu ke 250 sampai 275.




Dan untuk window berukuran 30 kebelakang, sepertinya banyak data penting yang terlewat dan freq setelah di filter terasa seperti minim pergerakan.

Nama : M Fadhil
NIM : 1301180263
Kelas : IF-41-GAB02

## 4.Please attach your listing program for LPF and MVA.

Github : https://github.com/fadhilfadhil/IOT_Assigment1_IF-41-GAB02_1301180263

Drive : https://drive.google.com/drive/folders/1gsBtnHeTX_8Mn6gYxCD5NqIEAi0M0wPO?usp=sharing

Referensi:

LPF

https://gist.github.com/junzis/e06eca03747fc194e322

https://medium.com/analytics-vidhya/how-to-filter-noise-with-a-low-pass-filter-python-885223e5e9b7

MVA

https://lms.telkomuniversity.ac.id/pluginfile.php/1756901/mod_resource/content/1/Slide%204%20AI%20IoT%2020202.pdf ,pada halaman 38