

LAPORAN PROGRES PENGOLAHAN SINYAL DIGITAL



Analisa Transfusi Darah di EMG dengan Adaptive Filter

Nama : Sendy Prisma Nurferian

NRP : 5024211012

Kelas : B

Dosen : Bapak Dion Hayu Fardiantoro, S.T.,M.T

Pada tugas ini mahasiswa diberikan sebuah tugas untuk menelaah pemrosesan sebuah sinyal dan juga diharapkan para mahasiswa bisa memfilterisasi dari gelombang sinyal di sebuah database untuk menjadi sinyal yang sempurna atau setidaknya noise dalam sebuah gelombang sinyal tersebut menjadi kecil dan bisa dilihat. Maka dari itu saya lebih memilih ide dan juga analisa yang sesuai dan saya juga lebih memilih menganalisa transfusi darah karena ini bisa dijadikan sebagai gelombang dimana kita dapat mengetahui apakah darah orang tersebut mengalami tekanan darah tinggi atau normal. Karena menurut saya pada akhirnya sebuah alat medical akan membutuhkan metode ini nantinya, dan diharapkan bisa mempermudah pekerjaan seorang dokter kedepannya.

Lalu untuk analisa yang saya gunakan ini menggunakan metode adaptive filter. Mengapa? Karena hal ini sangat relevan hasilnya dengan EMG dan juga penerapan metode ini begitu akurat menurut saya. Dan selain itu, Ketika kita memfilter sebuah sinyal EMG menggunakan metode filter adaptif, tujuannya adalah untuk menghilangkan atau mengurangi noise dan gangguan yang terdapat dalam sinyal tersebut. Sinyal EMG sering kali rentan terhadap noise seperti interferensi elektromagnetik, noise lingkungan, atau noise yang dihasilkan oleh perangkat elektronik lainnya. Gangguan-gangguan ini dapat menyebabkan kesalahan atau distorsi dalam analisis sinyal EMG yang ingin dilakukan.

Metode filter adaptif menggunakan algoritma yang dapat menyesuaikan diri dengan sinyal input untuk secara otomatis menghilangkan komponen yang tidak diinginkan. Dalam hal ini, metode Least Mean Squares (LMS) digunakan dengan menggunakan fungsi `lmsfilter` dari library `scipy.signal`. Keuntungan utama dari filter adaptif adalah kemampuannya untuk secara dinamis menyesuaikan respons filter terhadap perubahan dalam sinyal input. Dalam konteks sinyal EMG, filter adaptif dapat mengikuti perubahan dalam karakteristik sinyal EMG yang terjadi selama aktivitas otot atau perubahan kondisi pasien. Dengan demikian, filter adaptif dapat membantu dalam memperoleh sinyal EMG yang lebih bersih dan akurat, yang memudahkan analisis dan interpretasi lebih lanjut. Lalu untuk sampel dapat diakses menggunakan pada link berikut:

<https://www.kaggle.com/datasets/whenamancodes/blood-transfusion-dataset>

Berikut untuk programnya dalam python:

1. Import library yang diperlukan:

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

numpy (diimpor sebagai np): Library untuk operasi numerik dan komputasi array.

pandas (diimpor sebagai pd): Library untuk manipulasi dan analisis data tabular.

matplotlib.pyplot (diimpor sebagai plt): Library untuk visualisasi data dalam bentuk grafik.

2. Membaca file csv yang kita dapatkan dan kita jadikan database

```
data = pd.read_csv('transfusion.csv')
```

fungsi dari fungsi diatas Membaca data dari file CSV 'transfusion.csv' dan menyimpannya ke dalam objek DataFrame dengan nama data.

3. Mengambil sinyal data EMG untuk transfusi darah normal

```
normal_data = data[data['label'] == 'Normal BT']['signal'].values
```

akan tetapi karena kita sudah menemukan dan juga menentukan maka normal data harus diatur sesuai dengan label dan berapa normal transfusi nya. Maka akan menjadi seperti berikut:

```
normal_data = data["Monetary (c.c. blood)"]
```

Atau lebih jelasnya fungsi ini akan mengambil kolom 'signal' yang sesuai dengan transfusi darah normal dari data DataFrame dan menyimpannya ke dalam array normal_data.

4. Membuat fungsi untuk filterisasi atau smoothing sinyal dari database awal.

```
def smooth_signal(signal, window_size):
    smoothed_signal = np.convolve(signal, np.ones(window_size)/window_size, mode='same')
    return smoothed_signal
```

Penjelasan : **def smooth_signal(signal, window_size)** berfungsi untuk mendefinisikan fungsi `smooth_signal` yang digunakan untuk melakukan pemulusan pada sinyal. Selanjutnya pada **smoothed_signal = np.convolve(signal, np.ones(window_size)/window_size, mode='same')** berfungsi untuk memuluskan atau memfilter sinyal dengan menggunakan operasi konvolusi. Menggunakan kernel berbentuk array yang terdiri dari $1/\text{window_size}$ dan mode 'same' untuk memastikan panjang sinyal hasil sama dengan sinyal input.

5. Menentukan ukuran window untuk filterisasi agar nantinya sinyal yang tertampil bisa sesuai dan juga bisa terlihat dari hasil analisisnya.

```
window_size = 10
```

Disini saya memilih 10 karena menurut saya sudah cukup untuk menggambarkan gelombang sinyalnya.

6. Buat fungsi yang memanggil dari sinyal yang awal dan yang sudah di filter agar proses dari pemfilteran bisa terlaksana.

```
smoothed_signal = smooth_signal(normal_data, window_size)
```

Memanggil fungsi `smooth_signal` untuk memuluskan sinyal `normal_data`.

7. Membuat plot grafik untuk menampilkan sinyal asli dan sinyal yang sudah dipuluskan.

plt.figure(figsize=(12, 6)): Membuat objek figure dengan ukuran 12x6 inci untuk menampung plot.

plt.plot(normal_data, label='Original Signal'): Menampilkan plot sinyal asli dengan label 'Original Signal'.

plt.plot(smoothed_signal, label='Smoothed Signal'): Menampilkan plot sinyal yang sudah diputuskan dengan label 'Smoothed Signal'.

plt.xlabel('Sample'): Memberikan label sumbu x dengan teks 'Sample'.

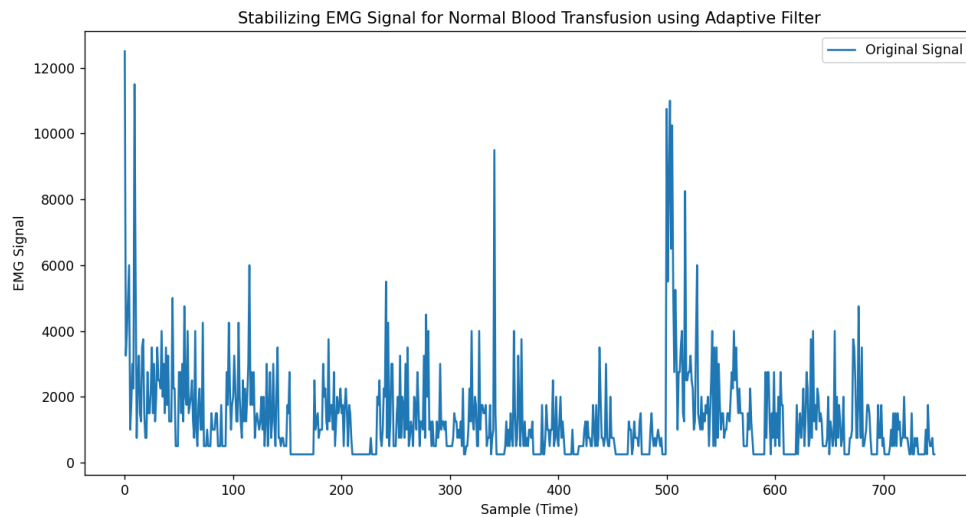
plt.ylabel('EMG Signal'): Memberikan label sumbu y dengan teks 'EMG Signal'.

plt.title('Stabilizing EMG Signal for Normal Blood Transfusion using Adaptive Filter'): Memberikan judul grafik.

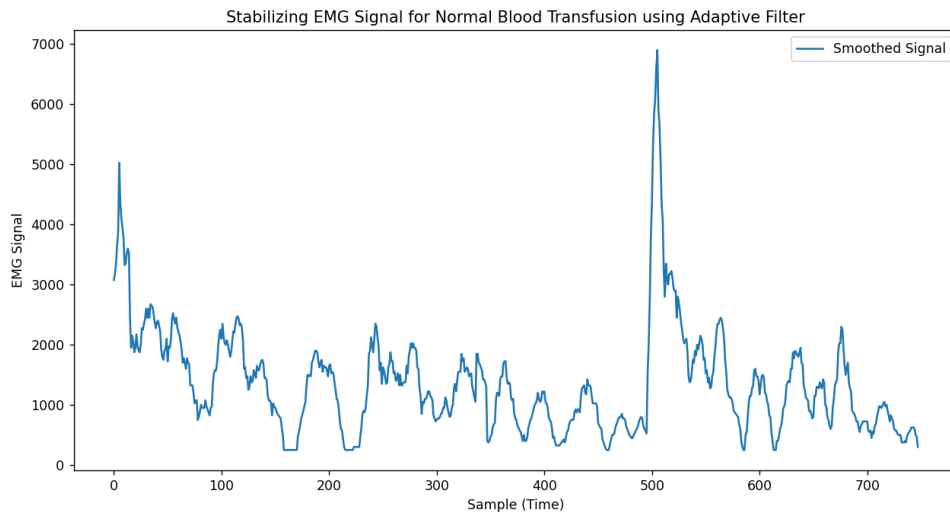
plt.legend(): Menampilkan legenda untuk membedakan plot sinyal asli dan sinyal yang sudah diputuskan.

plt.show(): Menampilkan grafik yang telah dibuat

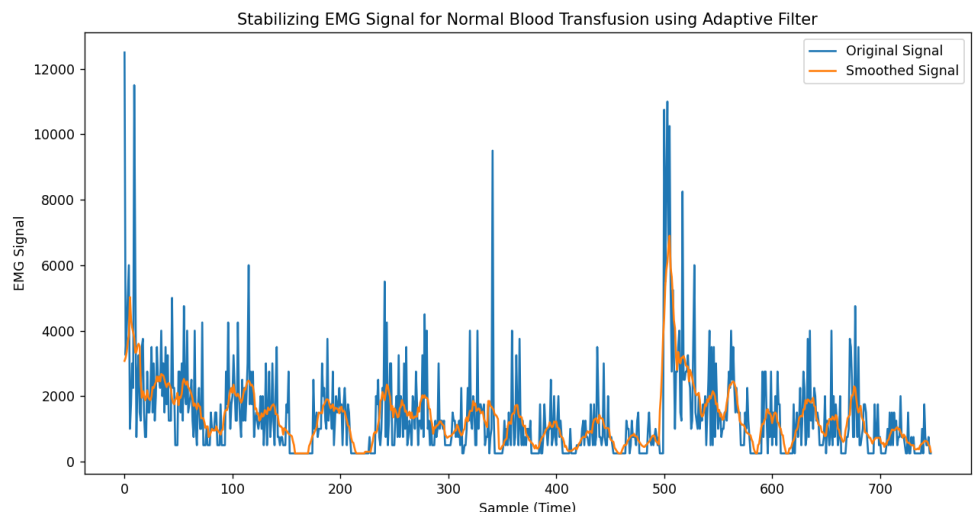
Gambar proses sinyal pada awalnya tanpa di filter:



Gambar proses sinyal saat di filter dengan sifat adaptive filter :



Gambar proses sinyal sebelum dan sesudah di filter:



+