Indian Institute of Technology, Jodhpur

Signals and Systems (EEL 2010)

Coding Assignment

Group 20(Ramnani Nandini B22ME053)

Introduction:

Signal processing plays a crucial role in various fields such as telecommunications, audio processing, and biomedical engineering. The analysis of signals involves filtering, transformation, and correlation to extract meaningful information. In this report, we present a detailed analysis of signal processing techniques applied to input and output signals. The objective is to identify the best matching filter that closely resembles the output signal.

Methodology:

Two methods were employed for signal processing:

1. **Method 1 (Correlation with Butterworth Filters)**: Utilized built-in functions from the `scipy.signal` module for filter design and convolution. This method offers simplicity and convenience in applying standard filters such as low-pass, high-pass, and band-pass filters.
   a) **Butterworth Filter Application:**
   - The script defines functions to implement low-pass, high-pass, and band-pass filters using the `butter` and `lfilter` functions from `scipy.signal`.
   - These functions take the input signal, cutoff frequency, sampling frequency, and filter order as arguments.
   - The cutoff frequency determines the range of frequencies allowed to pass through the filter. Low-pass filters allow frequencies below the cutoff, high-pass allow frequencies above, and band-pass allow a specific range of frequencies.

   b) **Cross-Correlation Calculation:**
   - After applying each filter to the input signal, the script calculates the cross-correlation between the filtered signal and the output signal using the `correlate` function from `scipy.signal`.

- Cross-correlation measures the similarity between two signals as a function of a time lag. A high correlation value indicates a strong resemblance between the signals.

c) **Filter Selection:**
- The script identifies the filter that produces the highest correlation value with the output signal. This filter is considered the best match because it transforms the input signal in a way that most closely resembles the desired output.

2. **Method 2 (Convolution with Predefined Kernels)**: Directly convolved the input signal with manually defined filter kernels. This approach provides flexibility in designing custom filters tailored to specific signal characteristics.

a) **Predefined Filter Kernels:**
- The script defines filter kernels as NumPy arrays representing the weights applied to the input signal during convolution.
- Different kernel designs emphasize or attenuate specific frequency components. The provided script includes basic examples for low-pass, high-pass, and band-pass filters.

b) **Convolution Application:**
- The script defines a function to perform convolution between the input signal and each filter kernel using the `convolve` function from `scipy.signal`.
- Convolution is a mathematical operation that slides the kernel over the input signal, multiplying corresponding elements and summing the products.

c) **Correlation Calculation:**
- Similar to Method 1, the script calculates the correlation between each convolved signal and the output signal using the `correlate` function.

d) **Filter Selection:**
- As in Method 1, the filter that produces the maximum correlation with the output signal is chosen as the best match.

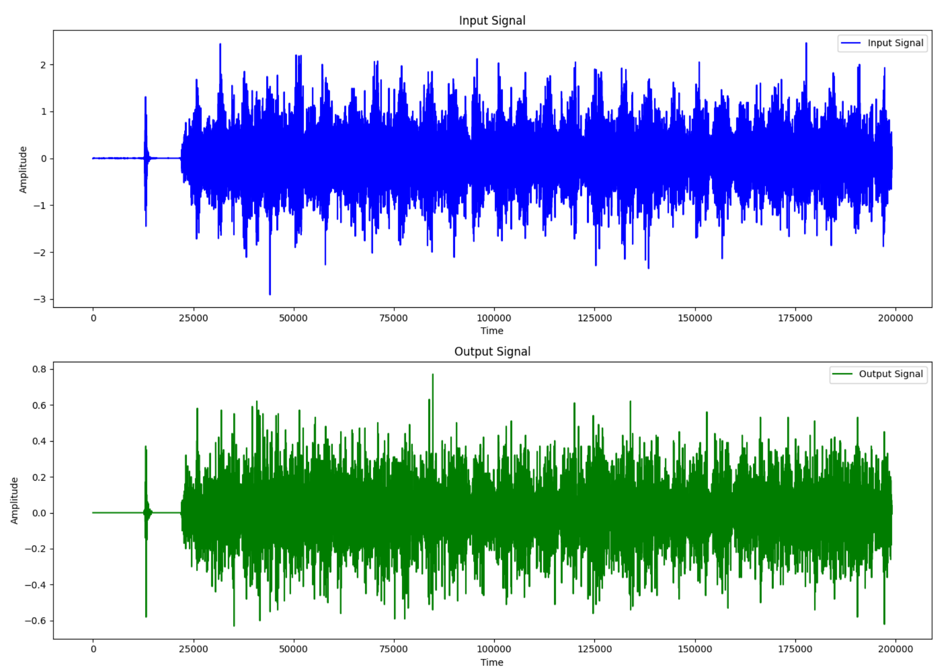
Implementation:

The Python script provided with this report incorporates both methods. It reads the input and output signals from text files, applies the filters, calculates correlations, and identifies the optimal filter.PLease refer to README.txt for further information.

Results

The script generates the following outputs:

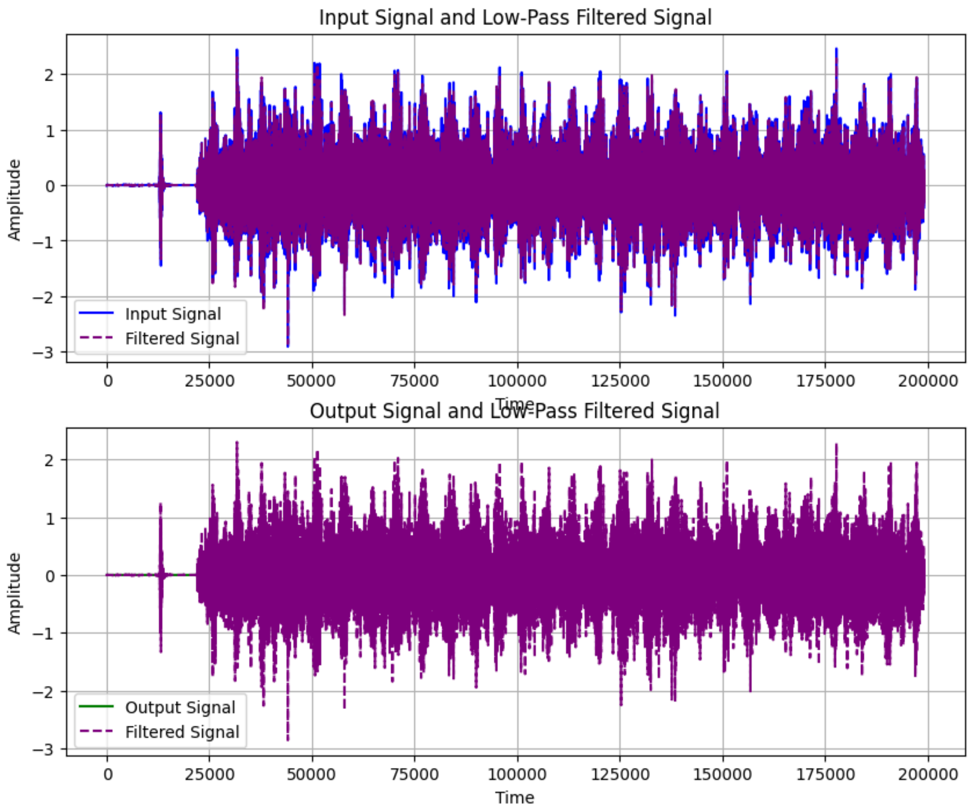
- **Plots :** Visualizations of the input, output, and filtered signals can be implemented using `matplotlib.pyplot`. These plots aid in understanding the filtering effects



Input and output signal plot(AMplitude vs Time)

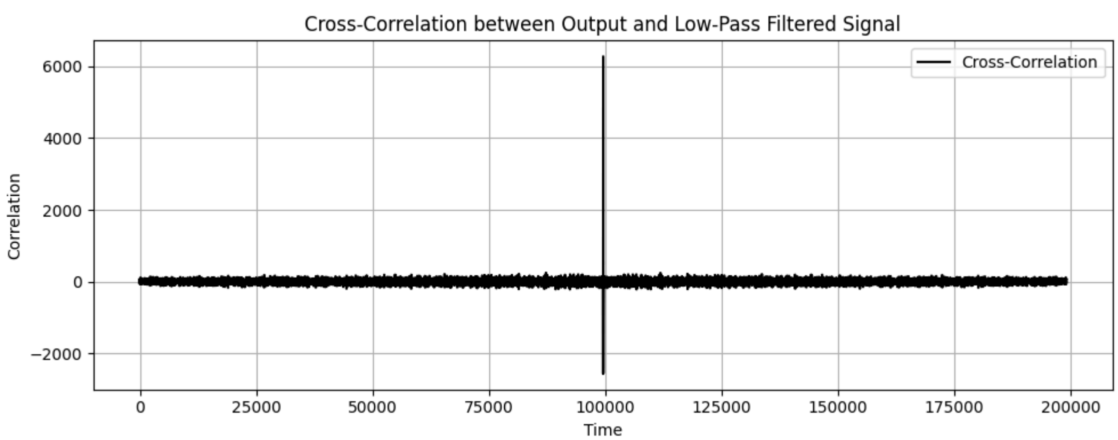
Method 1:

Apply Low Pass Filter to Input Signal and Cross Correlate with Output Signal



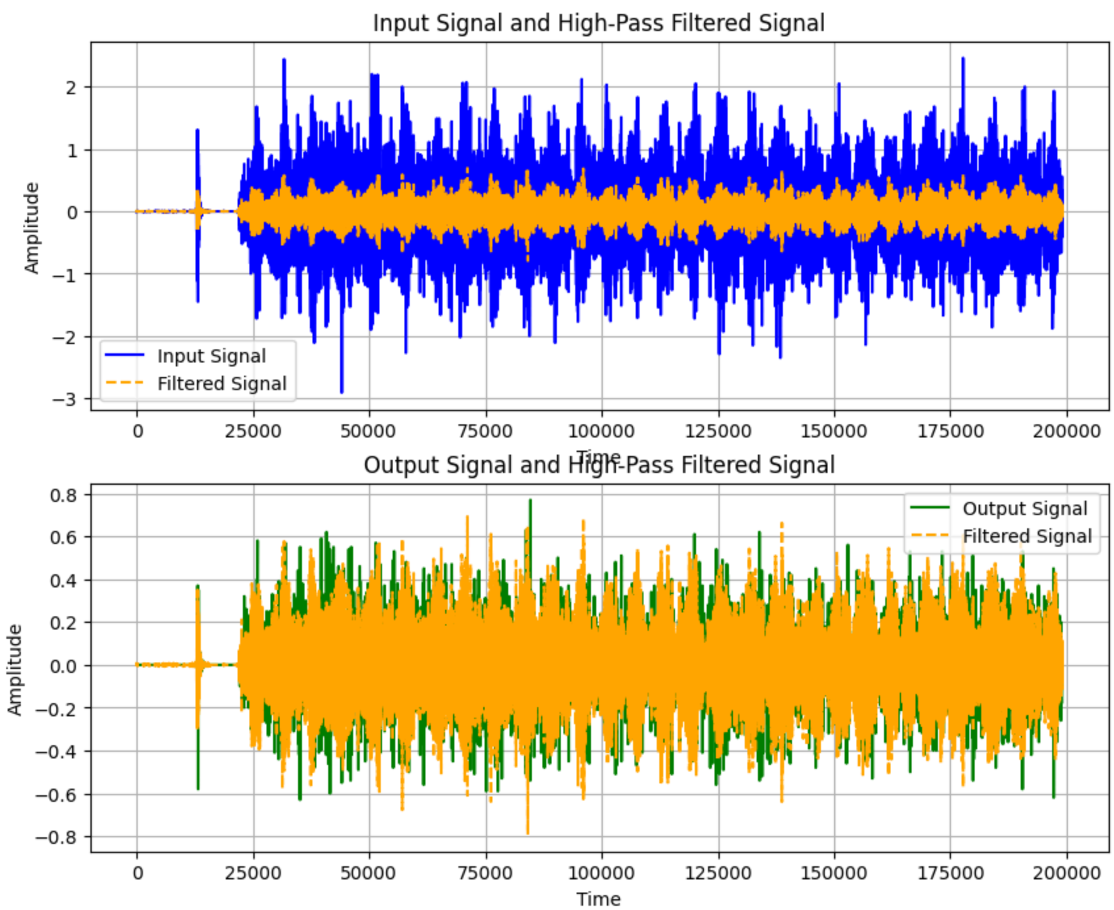
i) Input Signal and Low Pass Filtered Signal

ii) Output Signal and Low Pass Filtered Signal



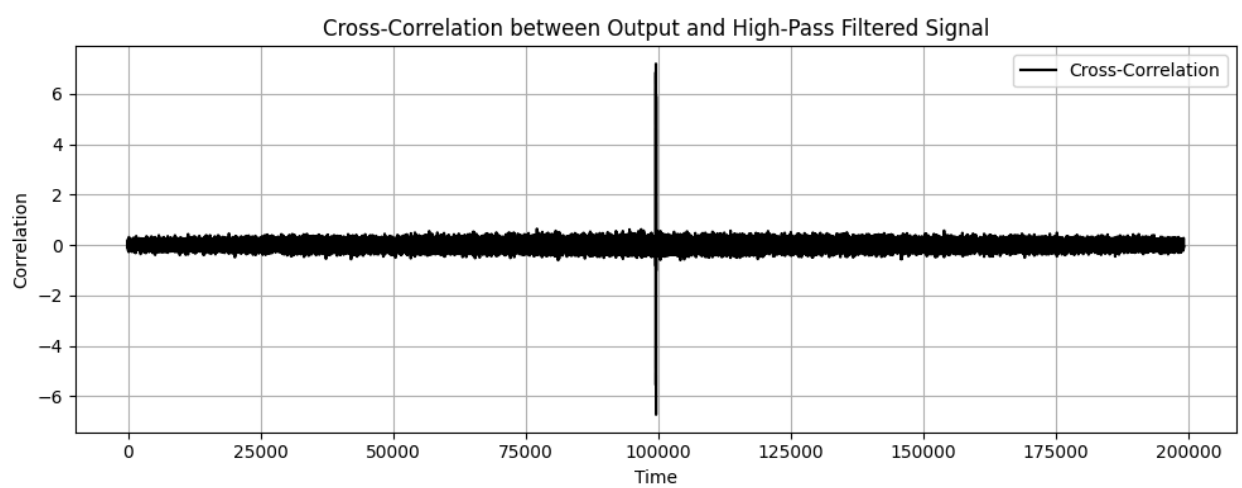
Cross-Correlation between output and low pass filtered signal

Apply High Pass Filter to Input Signal and Cross Correlate with Output Signal



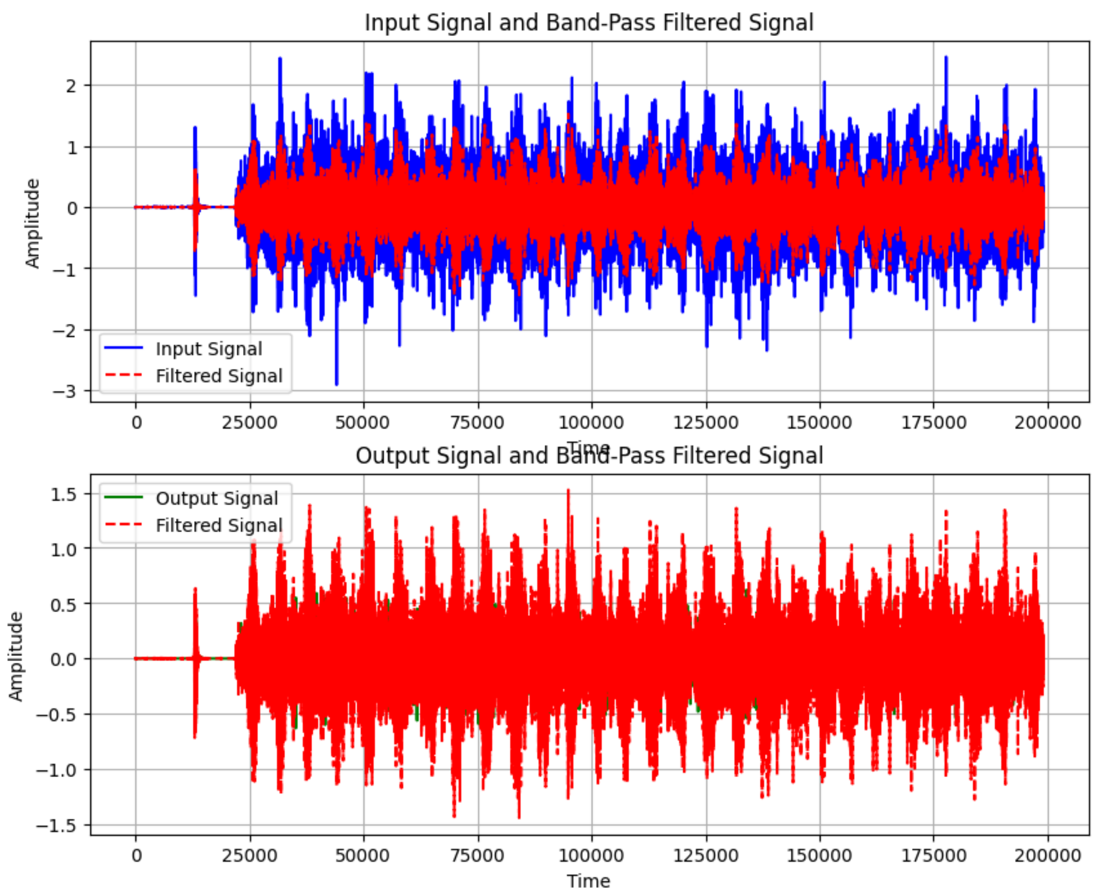
i) Input Signal and High-Pass Filtered Signal

ii) Output Signal and High-Pass Filtered Signal



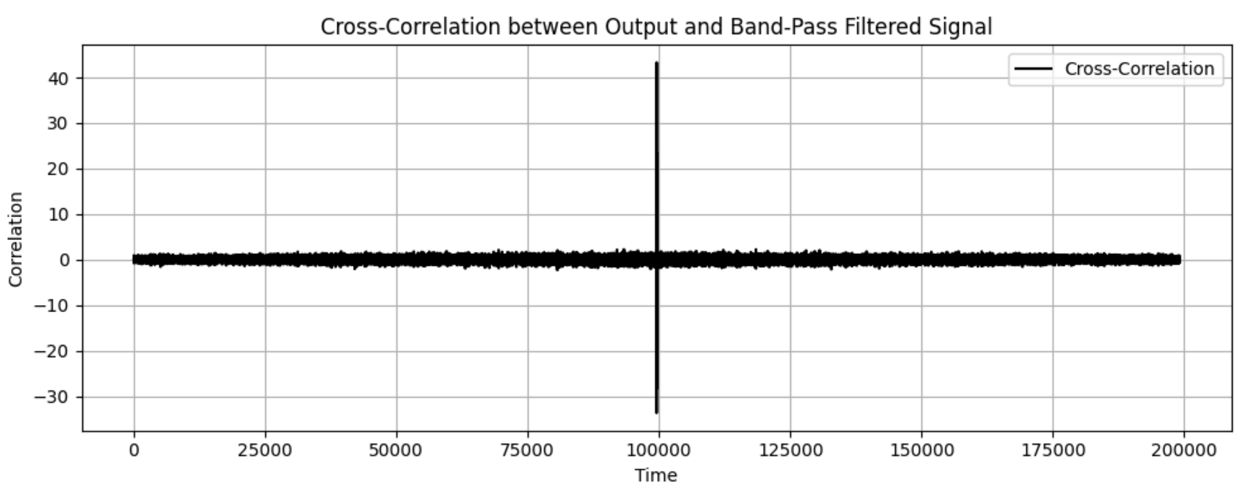
Cross-Correlation between Output and High-Pass Filtered Signal

Apply Band Pass Filter to Input Signal and Cross Correlate with Output Signal



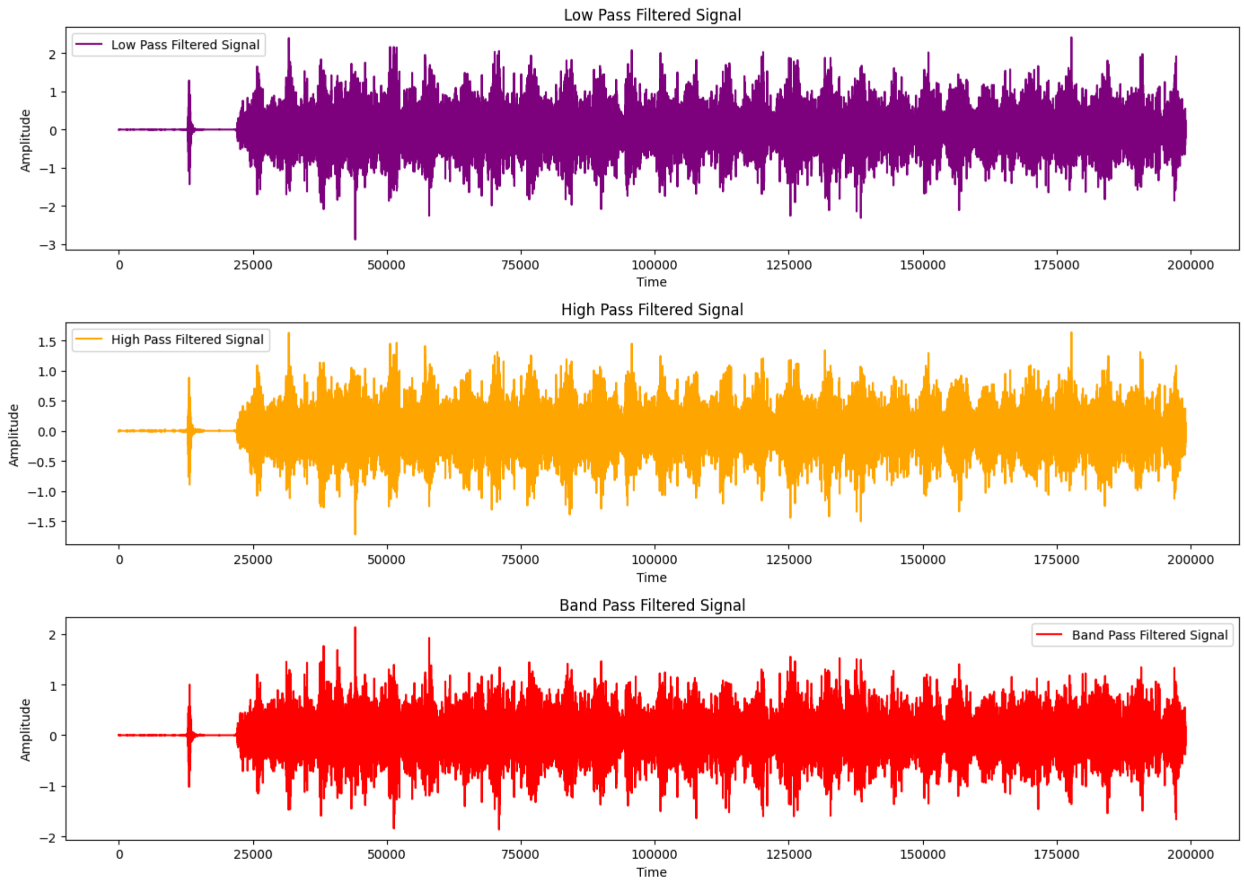
i) `Input Signal and Band-Pass Filtered Signal`

ii)`Output Signal and Band-Pass Filtered Signal`



`Cross-Correlation between Output and Band-Pass Filtered Signal`

Method 2:



i) Low Pass Filtered Signal

ii) High Pass Filtered Signal

iii) Band Pass Filtered Signal

- **<u>Correlation Values:</u>** The script prints the correlation coefficient obtained for each filter with the output signal.

  Method 1:

  Correlation results:

  - Low Pass: 6275.833452878105
  - High Pass: 7.186982870710705
  - Band Pass: 43.30881219530758

Mean Correlation results:

- Low Pass: 0.006234604278961365
- High Pass: 0.0031330747939135794
- Band Pass: -0.00621787639337507
- Best matching filter based on mean correlation: Low Pass

Method 2:
Correlation results:

- Low Pass: 6267.101640000003
- High Pass: 3205.5027000000005
- Band Pass: 2472.1218000000003

Mean Correlation results:

- Low Pass: 0.006234604278961365
- High Pass: 0.0031330747939135794
- Band Pass: -0.00621787639337507
- Best matching filter based on mean correlation: Low Pass

- **<u>Best Matching Filter:</u>**

  Method 1:Low Pass

  Method 2:Low Pass

The low pass filter which has the highest correlation value is considered the best match because it transforms the input signal in a way that most closely aligns with the desired output. The correlation coefficient quantifies this similarity. A value closer to 1 indicates a stronger resemblance between the filtered and output signals.

<u>Implementation Considerations:</u>

- **Filter Design:** The effectiveness of both methods relies on the chosen filter types (Butterworth in Method 1, predefined kernels in Method 2). In some cases, more sophisticated filter design techniques might be necessary depending on the specific signal characteristics and desired outcome.

- **Filter Order:** The filter order (a parameter in Method 1) influences the filter's sharpness. A higher order leads to a sharper transition band between retained and attenuated frequencies but can also introduce ripple in the passband. The script might require adjustments to the filter order based on the application.
- **Predefined Kernels:** The predefined kernels in Method 2 offer a simple approach but may not be optimal for all scenarios. Experimentation with different kernel designs or using custom filter design tools might be necessary for complex filtering tasks.

Conclusion:

We explored two methods for filter selection: correlation with Butterworth filters and convolution with predefined kernels. Both methods leverage correlation to identify the filter that transforms the input signal in a way that most closely resembles the desired output signal. By our analysis the low pass filter which has the highest correlation value is considered the best match because it transforms the input signal in a way that most closely aligns with the desired output. The choice of method depends on the specific application and the level of control required over the filter design. By understanding the principles behind these methods and their limitations, engineers can make informed decisions when selecting filters for various signal processing tasks.