NAME   : PUSHKIN DUGAM

ROLL NO : B22ME052

SNS Coding Assignment Report

COLAB FILE LINK :

https://colab.research.google.com/drive/1m5U5Jsikb9qMchuZAj44p7ag8cdZjcMO?usp=sharing

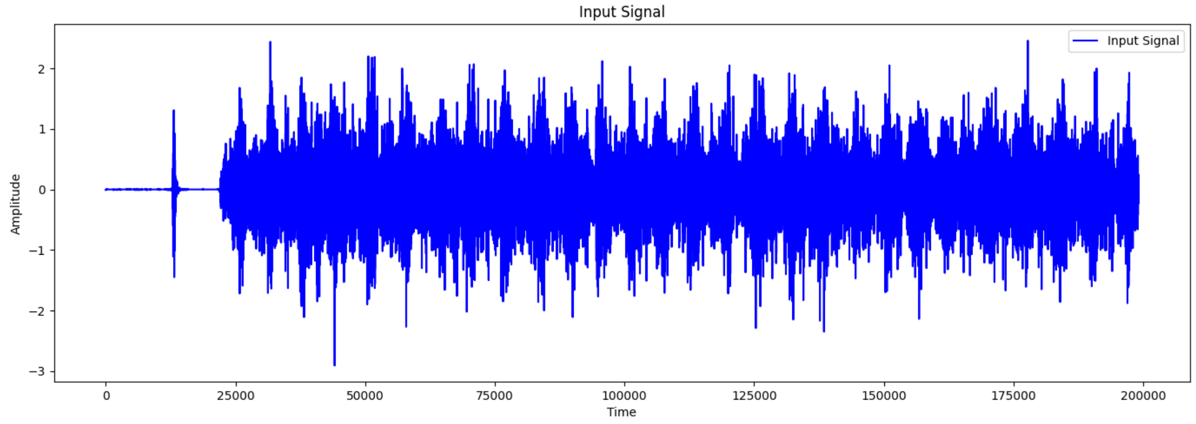
1] Getting and Reading Input and Output Signals

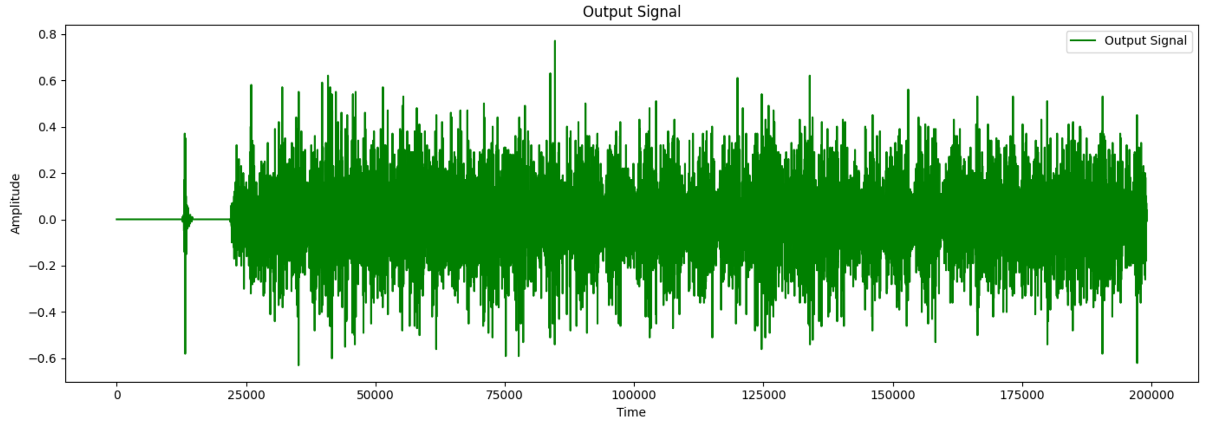
Import Libraries: The code imports necessary libraries including NumPy for numerical computation and SciPy for signal processing functions like correlate and convolve.

Read Signal Function: Defines a function read_signal to read a signal from a text file. It opens the file, reads the signal, and converts it into a NumPy array.

Read Input and Output Signals: Retrieves the file paths for input and output signals. Calls the read_signal function to read the signals from the respective files.

2] Plot of Input and Output Signals



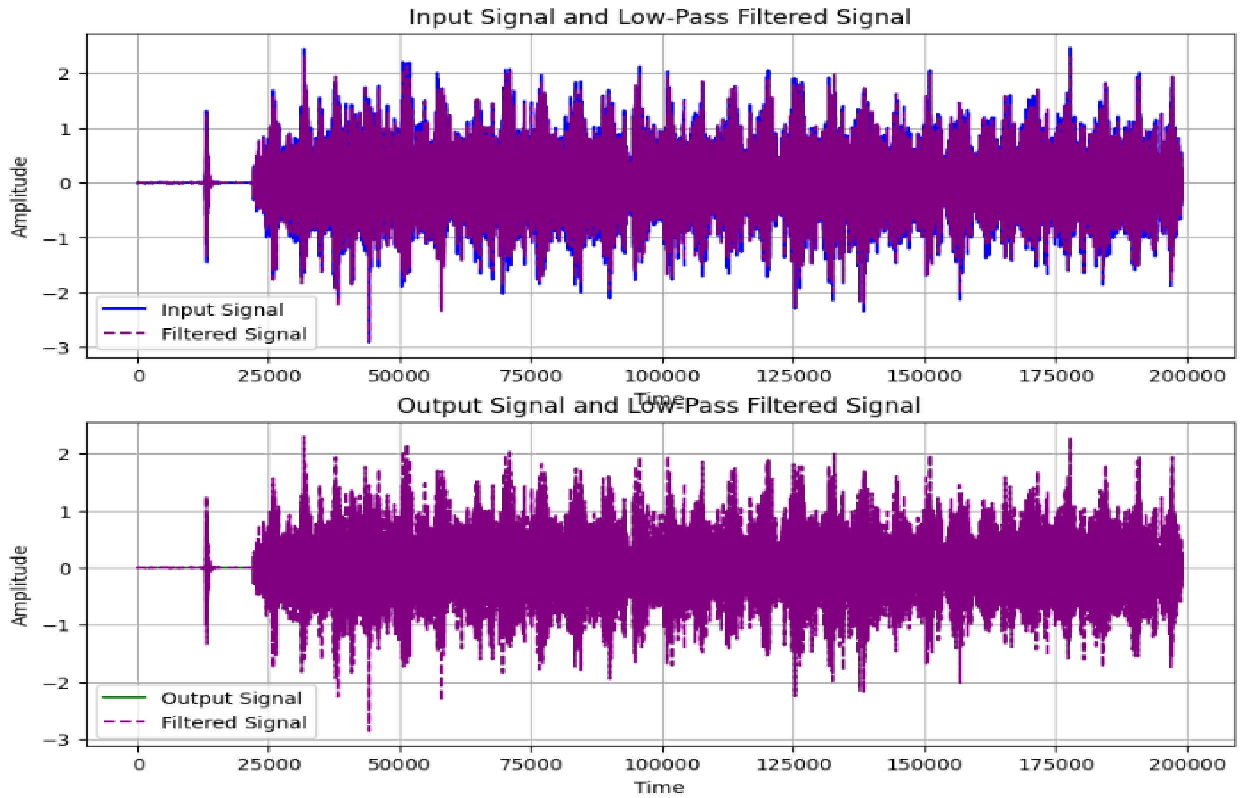


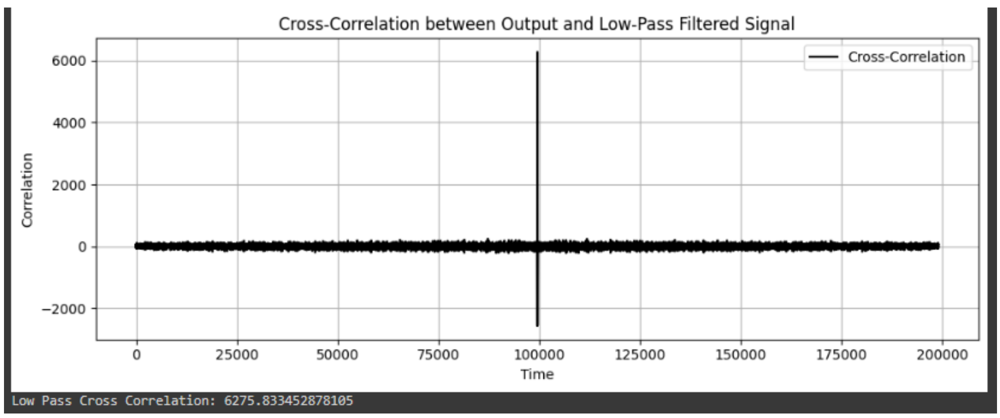
METHOD 1

1] Applied low pass filter to Input Signal & Cross Correlate with Output Signal

To design the low-pass filter, the function calculates the Nyquist frequency, which represents half of the sampling frequency. It then computes the normalized cutoff frequency by dividing the desired cutoff frequency by the Nyquist frequency. Utilizing the butter function from the SciPy library's signal module, the filter coefficients (denoted as b and a) are determined for a digital Butterworth filter with the specified cutoff frequency and filter order.

Subsequently, the designed filter is applied to the input signal using the lfilter function from the same module. This function takes the filter coefficients and the input signal as arguments and returns the resulting low pass filtered signal.

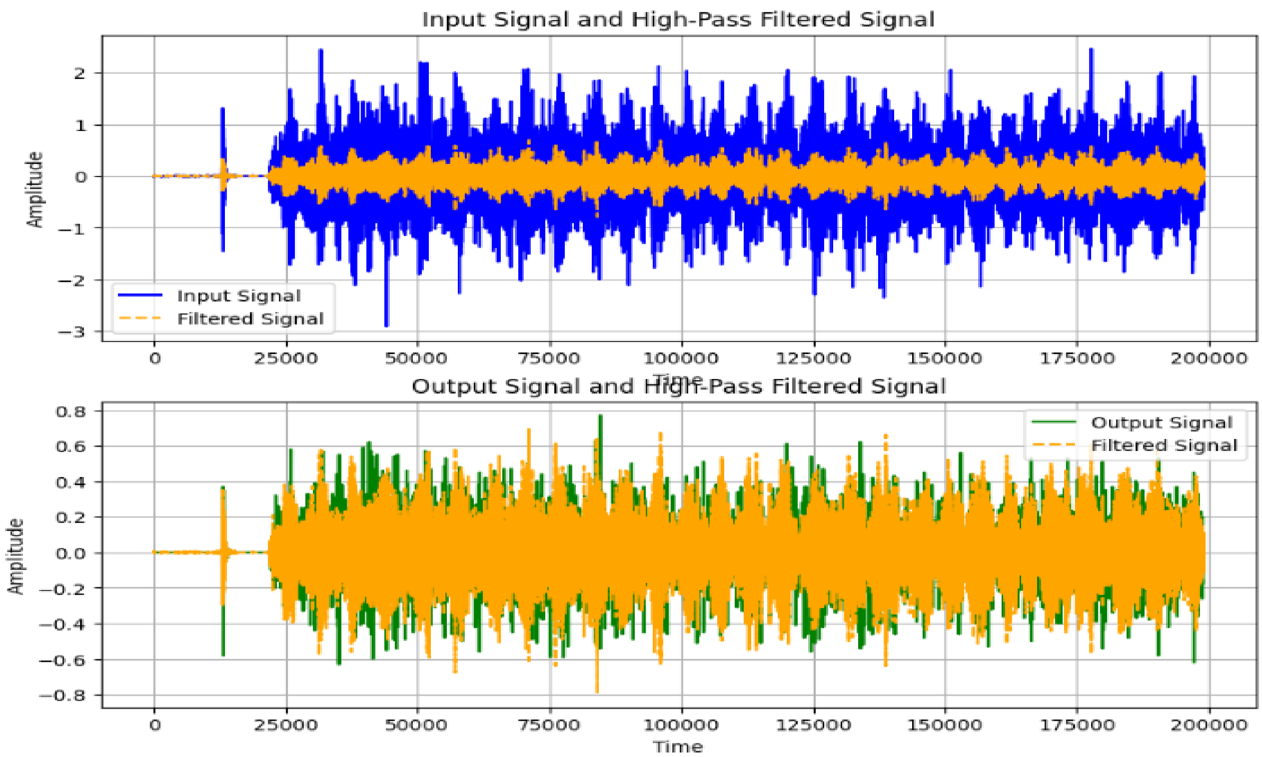


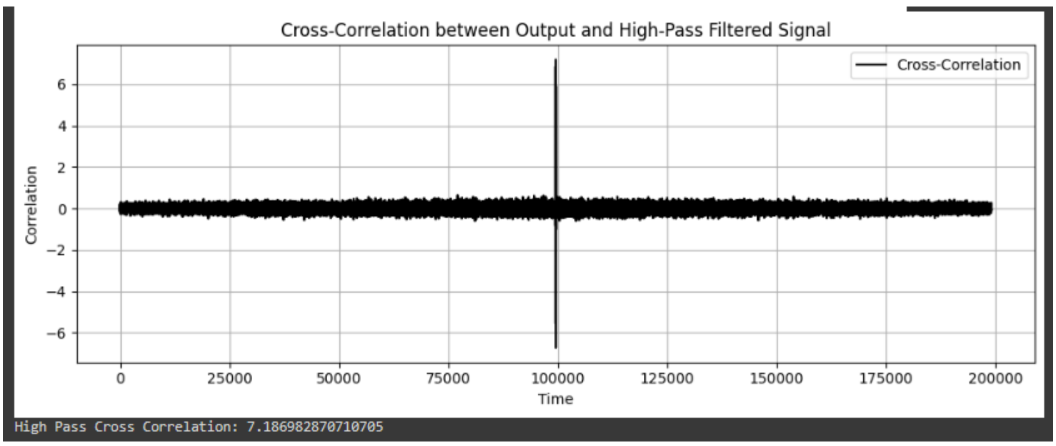


2] Applied high pass filter to Input Signal & Cross Correlate with Output Signal

To implement the high-pass filter, the function follows a similar procedure as for the low-pass filter. Initially, it calculates the Nyquist frequency, representing half of the sampling frequency, and then computes the normalized cutoff frequency by dividing the desired cutoff frequency by the Nyquist frequency. Employing the butter function from SciPy's signal module, it determines the filter coefficients (b and a) for a digital Butterworth filter with the specified cutoff frequency and filter order.

Once the filter coefficients are obtained, the function applies the filter to the input signal using the lfilter function from the same module. This function convolves the input signal with the filter coefficients and returns the resulting high pass filtered signal.

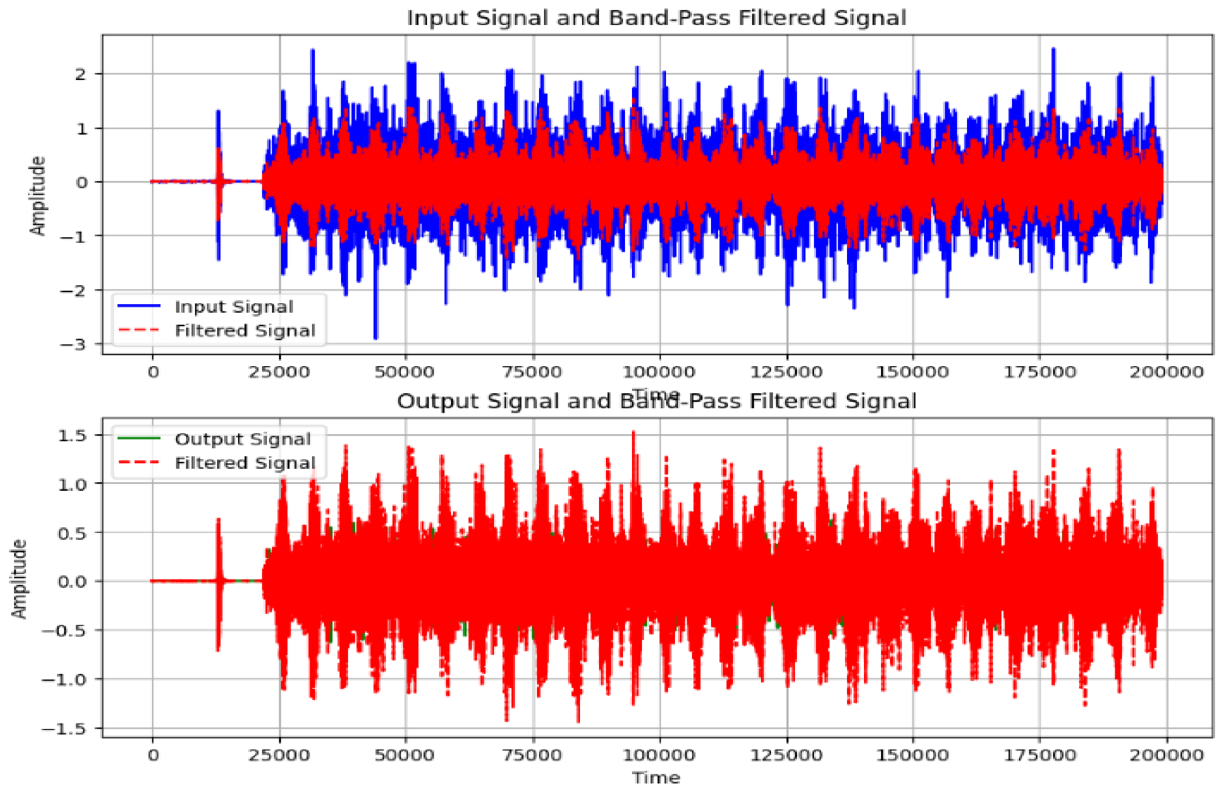


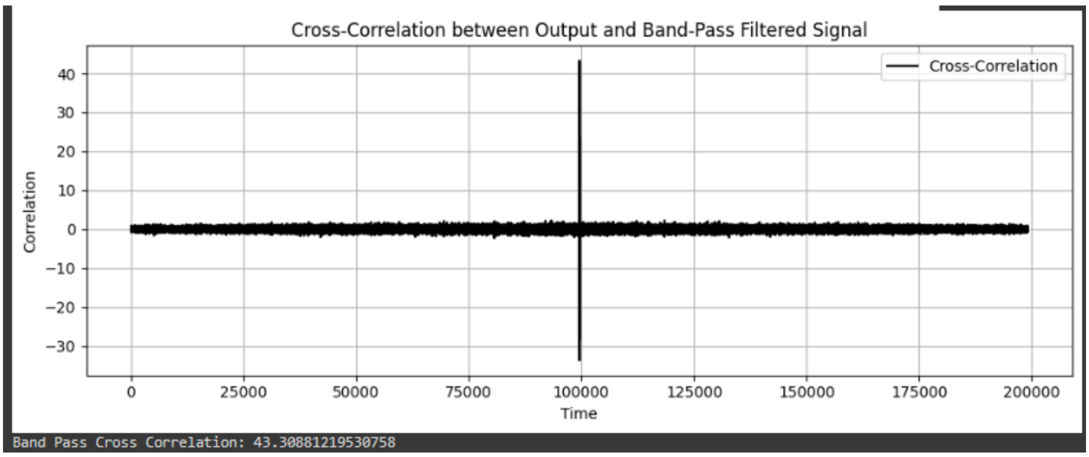


3] Applied Band pass filter to Input Signal & Cross Correlate with Output Signal

To implement the band-pass filter, the function follows a similar procedure as for the low-pass filter. Initially…………

Once the filter coefficients are obtained, the function applies the filter to the input signal using the lfilter function from the same module. This function convolves the input signal with the filter coefficients and returns the resulting filtered signal. The band-pass filter selectively allows a range of frequencies, defined by the specified cutoff frequencies, to pass through while attenuating frequencies outside this range.





4] Then I have found out the max and mean correlation of filtered Signal and Identified which the best filtered signal matches the output signal

(max Correlation)- If you're interested in identifying the peak correlation between two signals, you should use the maximum value. This is useful when you want to find the point of highest similarity or alignment between the two signals.

```
Max Correlation results:
Low Pass: 6267.101640000003
High Pass: 3205.5027000000005
Band Pass: 2472.1218000000003
```

```
Analysis Results:
Best matching filter: Low Pass
```

(mean Correlation)- If you're interested in the overall strength of correlation between two signals across their entire length, you can use the mean value. This provides a measure of the average correlation between the two signals.

```
Mean Correlation results:
Low Pass: 0.006234604278961365
High Pass: 0.0031330747939135794
Band Pass: -0.00621787639337507
Analysis Results:
Best matching filter based on mean correlation: Low Pass
```

In many cases, especially when comparing filtered and output signals in signal processing tasks, you might be interested in both the peak correlation (to find the best alignment) and the overall strength of correlation. Therefore, it's common to compute both the maximum and mean correlations for a comprehensive analysis.

RESULT – LOW PASS

METHOD 2

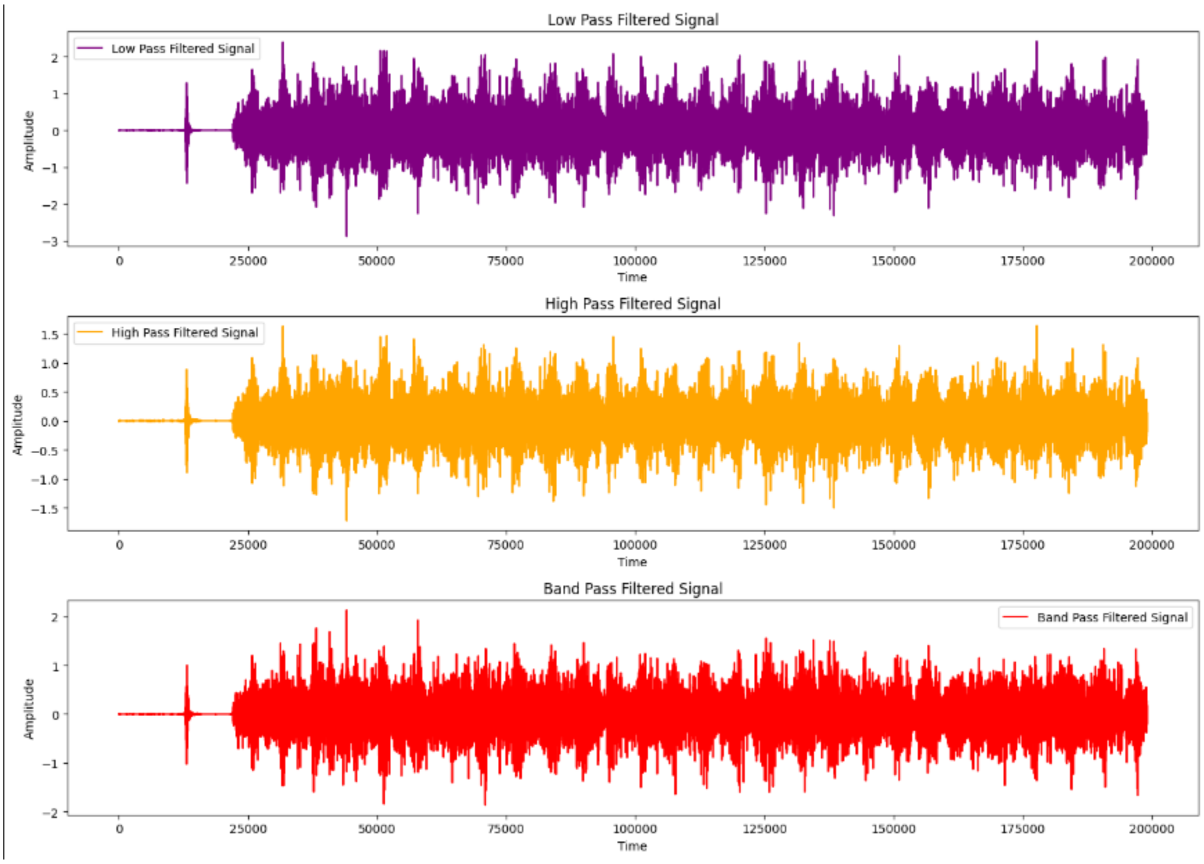
1] Define Filter Kernels and Convolve Input Signal with Filters

Three different types of filters are defined: a low-pass filter, a high-pass filter, and a band-pass filter.

Each filter is represented by a kernel, which defines the convolution operation to be applied to the input signal. The low-pass filter kernel (hlp) consists of values [0.1, 0.8, 0.1], indicating that it allows low-frequency components of the signal to pass through while attenuating high-frequency components. Similarly, the high-pass filter kernel (hhp) has values [-1, 2.5, -1], enabling it to pass high-frequency components while suppressing low-frequency ones. The band-pass filter kernel (hbp) is designed to filter a specific frequency band, with values [-1, 1, -1, 1, -1] indicating alternating positive and negative weights to capture frequency components within the desired band.

These filters are then applied to the input signal using the apply_filter function, resulting in three separate filtered signals representing the low-pass, high-pass, and band-pass components of the input signal.

2] Plotting Low Pass , High Pass & Band Pass Filtered Signals



<u>3] Computed Max Correlation between filtered signal and output signal</u>

```
Max Correlation results:
Low Pass: 6267.101640000003
High Pass: 3205.5027000000005
Band Pass: 2472.1218000000003
```

```
Analysis Results:
Best matching filter: Low Pass
```

RESULT – LOW PASS

EXPLAINATION

Correlating the filtered signals with the output signal allows us to assess how closely each filtered signal resembles the output signal. Correlation measures the similarity between two signals by calculating the degree of linear relationship between them. In this context, if a filtered signal closely matches the output signal, it indicates that the filtering process applied to the input signal was effective in producing an output similar to the observed one.

By comparing the correlation values of different filtered signals with the output signal, we can determine which type of filter was likely applied to the input signal. The filter that produces the highest correlation value with the output signal is considered the best match. This is because a higher correlation value indicates a stronger similarity between the filtered signal and the output signal, suggesting that the characteristics of the filter align well with the transformation applied to the input signal.

Therefore, by analyzing the correlation values, we can infer the type of filter (e.g., low-pass, high-pass, or band-pass) that was applied to the input signal, enabling us to identify the unknown filter used in the signal processing pipeline.