AEE 323 ENGINEERING EXPERIMENTATION AND DATA ANALYSIS Group No : 10

Ayşe DİNÇ 20110111044
Melis YAZGAN 20110111011
Merve KARATAŞ 20110111006
Muhammed Berati KILIÇ 20110111028

January 19, 2024

Signal Processing Study: Application of Low Pass Filter for Noise Reduction

Final Report

1 Introduction

In the realm of signal processing, the accurate interpretation of time-domain data often necessitates the application of filtering techniques to mitigate unwanted noise.

To briefly talking about filters and their types in general: A filter is a circuit capable of passing (or amplifying) certain frequencies while attenuating other frequencies. Thus, a filter can extract important frequencies from signals that also contain undesirable or irrelevant frequencies.

Four Major Types of Filters

The four primary types of filters include the low-pass filter, the high-pass filter, the band-pass filter, and the notch filter (or the band-reject or band-stop filter). Take note, however, that the terms "low" and "high" do not refer to any absolute values of frequency, but rather, they are relative values with respect to the cutoff frequency.

1) LOW-PASS FILTER: A low-pass filter (LPF) is a circuit that only passes signals below its cutoff frequency while attenuating all signals above it. It is the complement of a high-pass filter, which only passes signals above its cutoff frequency and attenuates all signals below it. Low-pass filters can also be used in conjunction with high-pass filters to form bandpass, band-stop, and notch filters.

2) HIGH-PASS FILTER: A high-pass filter (HPF) is an electronic filter that passes signals with a frequency higher than a certain cutoff frequency and attenuates signals with frequencies lower than the cutoff frequency. The amount of attenuation for each frequency depends on the filter design.

3) BAND-PASS FILTER: A bandpass filter passes a range of frequencies while attenuating all frequencies outside of the band.

4) NOTCH FILTER: Notch filters are a type of band-stop filter that attenuate a very narrow set of frequencies, which can be created from a combination of low-pass and high-pass filters with cutoff frequencies very close to each other.

Figure 1 below gives a general idea of how each of these four filters works:

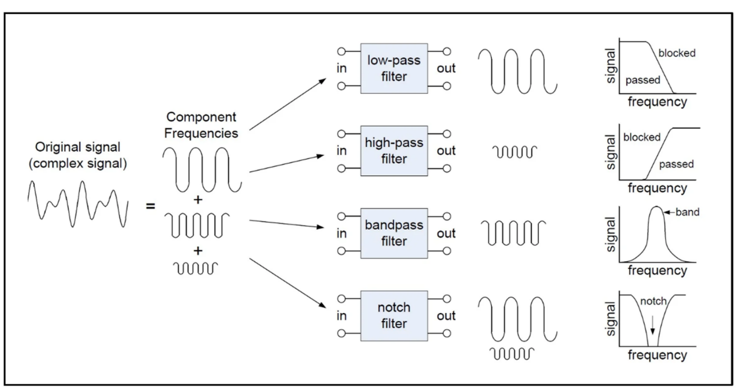


Figure 1: A basic depiction of the four major filter types.

Let's briefly talk about the usage areas of the low pass filter. Low-pass filters have applications such as anti-aliasing, reconstruction, and speech processing, and can be used in audio amplifiers, equalizers, and speakers.

Applications of Low-Pass Filters

Low-pass filters have a multitude of applications in various fields. Here are some common ones:

- Audio Processing: Low-pass filters are widely used in audio processing to remove high-frequency noise or to simulate the effect of distance (since high-frequency sounds diminish over distance).
- Data Communication: In data communication systems, these filters are used to limit the bandwidth of the transmitted signal, reducing interference.
- Image Processing: In digital image processing, low-pass filters are used to blur images or to reduce high-frequency noise.
- Electronics: They are used in electronic circuits to block high-frequency noise in power supplies.
- Antialiasing: They play a significant role in the antialiasing process during signal digitization to prevent high-frequency components from being misinterpreted.

This study revolves around original time-domain measurements, initially visualized to discern inherent patterns. However, the presence of noise prompted the utilization of a Low Pass Filter for noise reduction, preserving essential signal characteristics.

This report elucidates the rationale behind the adoption of the Low Pass Filter, highlighting its role in attenuating high-frequency noise while retaining the integrity of the original signal. The outcomes, presented in both time and filtered domains, showcase the effectiveness of this technique in enhancing the interpretability and fidelity of the original data.

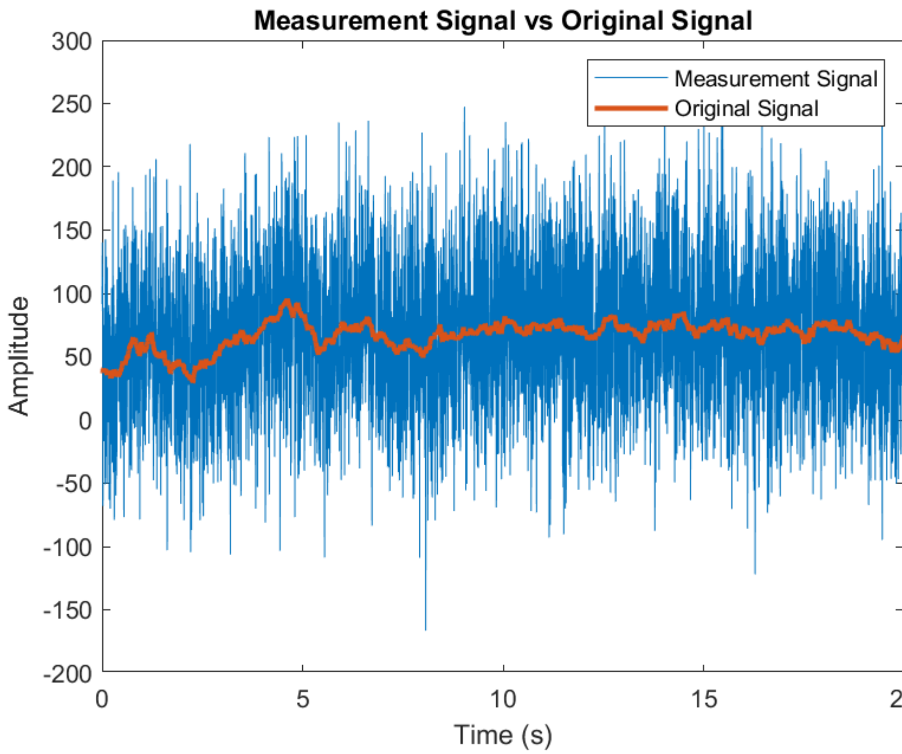


Figure 2: Measurement signal vs original signal in time domain

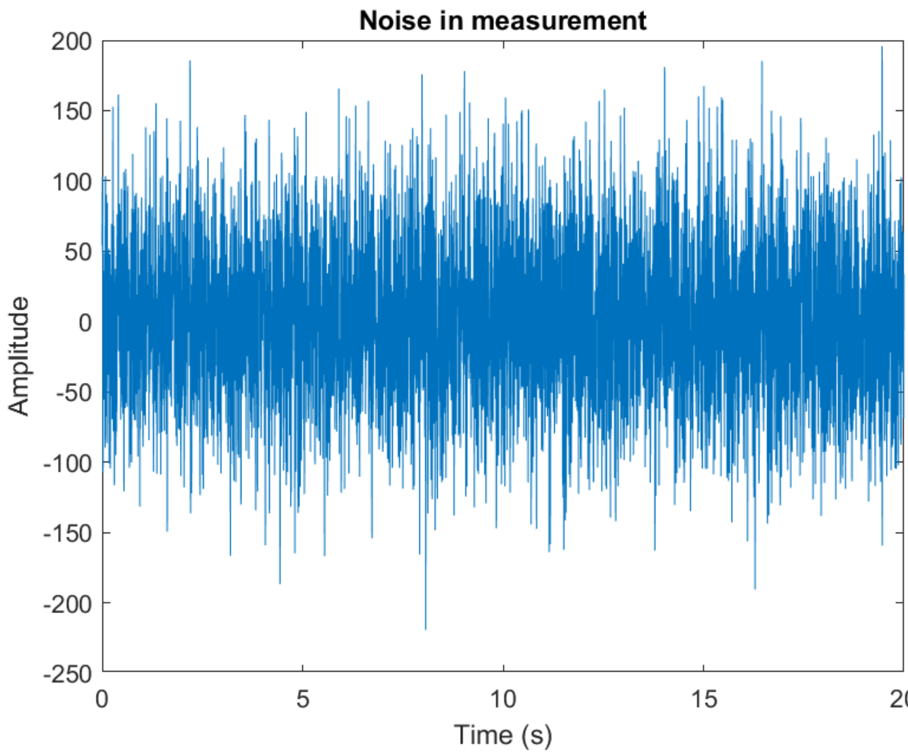


Figure 3: Given data noise in measurements

2 Methodology

Let's talk about what noise is ;

In data analysis, the term "noise" refers to meaningless, repetitive or misleading data in a data set. This data can result from the accuracy of the measuring instruments, the quality of the data sources or errors in the data collection process. "Noise" data can mislead the results of data analysis and lead to incorrect conclusions.

Noise in measurement data can stem from various sources, and identifying the specific cause can be crucial for accurate interpretation and analysis. Here are several common reasons why noise may be present in measurement data:

Instrumentation Noise:

The measurement instrument itself can introduce noise. Electronic instruments, sensors, or transducers may have inherent limitations and imperfections, leading to variations in the readings. Electromagnetic Interference (EMI) and Radio-Frequency Interference (RFI):

External electromagnetic fields or radio-frequency signals can interfere with the measurement equipment, especially if it is not adequately shielded. This interference can manifest as noise in the collected data. Environmental Factors:

Environmental conditions, such as temperature fluctuations, humidity, or vibrations, can affect the accuracy of measurements. Changes in these conditions may introduce noise into the data. Sensor Characteristics:

The sensor's design and characteristics can contribute to noise. For instance, sensors may have a finite resolution or may be sensitive to certain environmental conditions. Analog-to-Digital Converter (ADC) Noise:

When converting analog signals to digital data, the analog-to-digital converter may introduce quantization noise. This noise is associated with the finite precision of the ADC. Signal Transmission:

Noise can be introduced during the transmission of signals from the sensor to the measurement instrument. Poor-quality cables, long transmission distances, or improper grounding can contribute to signal degradation. Power Supply Issues:

Fluctuations or noise in the power supply to the measurement equipment can affect its performance. Inadequate power regulation may lead to variations in the collected data. Human or Environmental Interference:

Human factors, such as movement or interference by nearby electronic devices, can introduce noise during measurements. Additionally, nearby equipment emitting electromagnetic signals can impact measurements. System Instabilities:

Instabilities in the measurement system, such as jitter in timing components or instability in control systems, can contribute to noise. Sensor Aging or Wear:

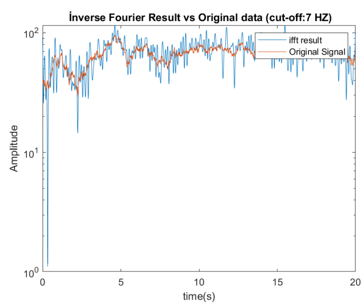
Over time, sensors may degrade or experience wear, leading to changes in their performance characteristics and introducing noise into the measurements.

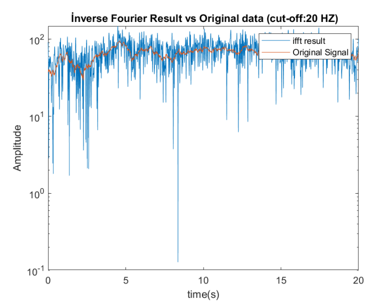
How Does A Low-Pass Filter Work?

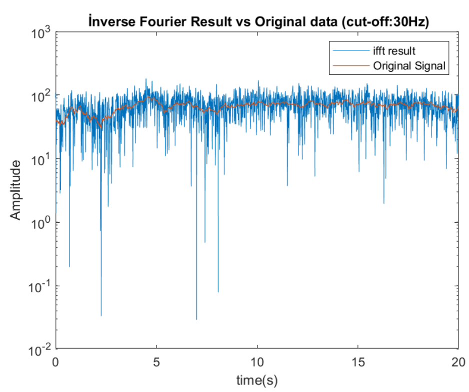
It works by using resistors and capacitors to generate a frequency-dependent impedance. The filter has a low resistance below the cutoff frequency, allowing low-frequency sounds to pass through relatively undisturbed. Above the cut-off, the impedance rises, causing the signal's higher-frequency components to be attenuated. As a result, the output is smoother and filtered, with less high-frequency noise or interference. The cutoff frequency is an important parameter since it dictates where the switch from pass to attenuate takes place. The sharpness of the transition is affected by the kind and arrangement of low-pass filters. Higher-order filters have steeper cutoffs, allowing for more exact control over the frequency content of the filtered signal. Low-pass filters are commonly employed in applications such as audio systems to eliminate undesirable noise or high-frequency interference while preserving important low-frequency information.

Filter Design

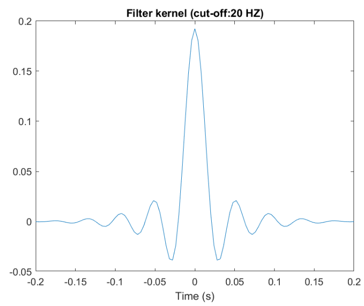
If we need to talk about the process of designing the filter, the method we use is the comparison of values since we do not have any information about the signal characteristics. First of all, a cut-off frequency value of 20 Hz was tried. Then, for comparison, cut-off frequencies of 7 Hz and 30 Hz were tried. There are images of this comparison below.

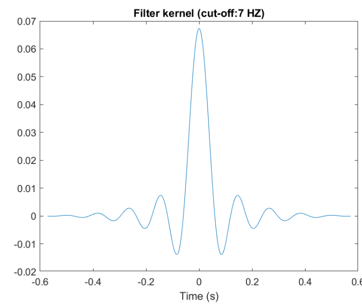


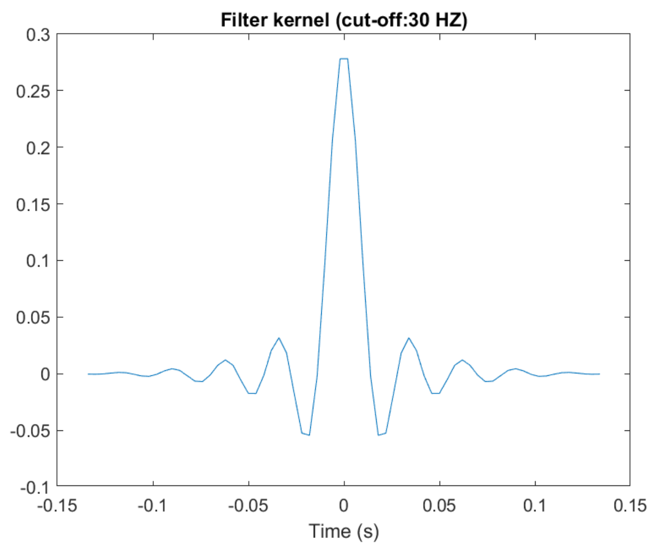




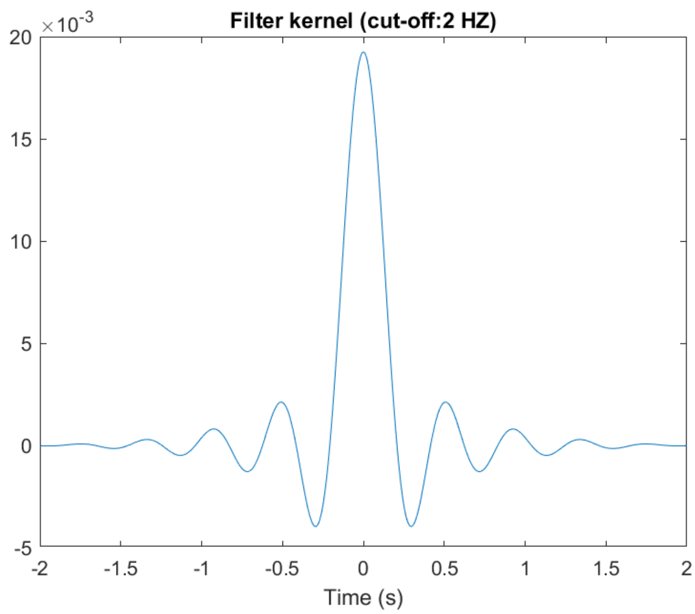
After comparing the graphs given above, it is seen that the error is less at low frequencies. Among the frequencies we tried, the frequency value of 2 Hz gave us the lowest error. The Kernel comparisons given below for given values.



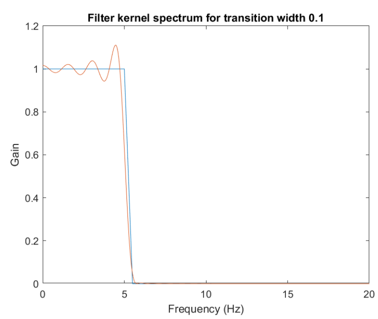


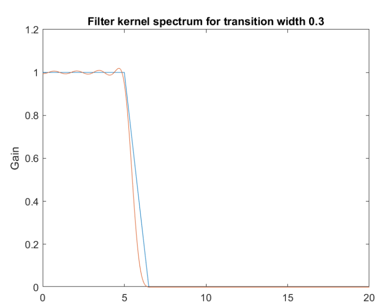


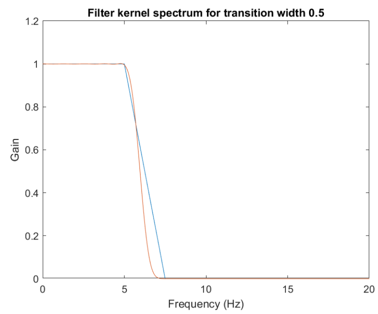
As a result of the comparison of the kernel graphics, we observed that the kernel graphics had sharper lines as the frequency increased, so we thought it would be more appropriate to choose smaller frequencies. Afterwards, we tried 1 Hz, but our code gave an error during our 1 Hz attempt, so we tried 2 Hz. As a result, our optimal choice was 2 Hz.



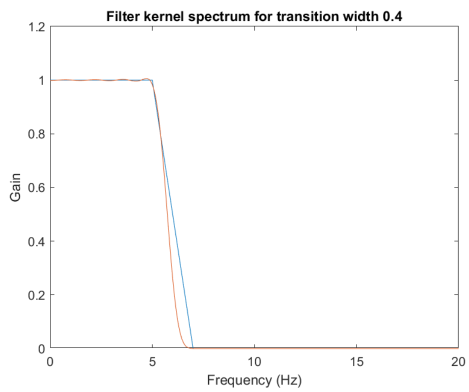
The values we tried for the transition width are 0.1, 0.3, 0.4 and 0.5, respectively. Images of these experiments are given below.



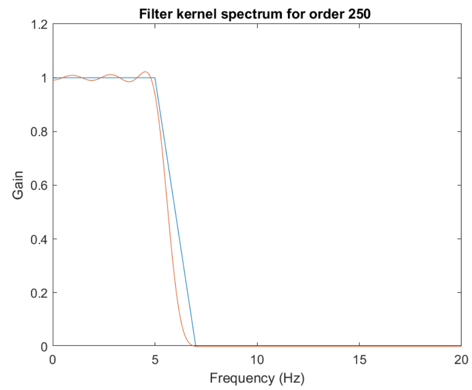


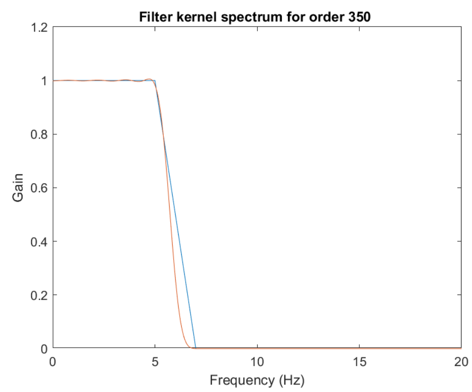


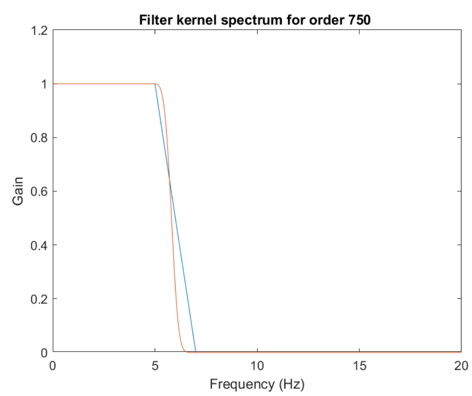
As a result of comparing the graphs, we observed that different values gives different fluctuations. It was decided that the value that provided the best transition was 0.4.



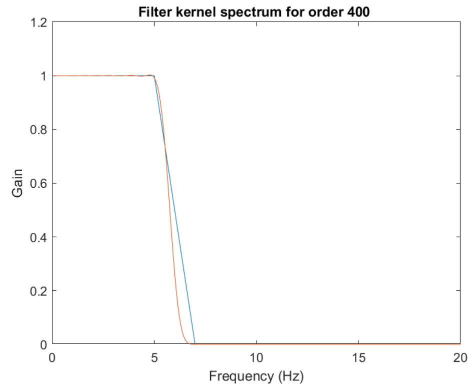
The values we use for order comparisons are 250, 350, 400 and 750. Images of these values are given below.







As a result of comparing the graphs, we observed that different values gives different attenuations. It was decided that the value that provided the best order was 400.



Filtering Process

First of all, the given data, that is, the measurement signal and original signal, were examined. It was determined that the data provided did not match each other. Therefore, it was determined that the measurement data had disturbances, that is, noise. Afterwards, the given data were compared and the noise graph was obtained by subtracting original data from the measurement data.

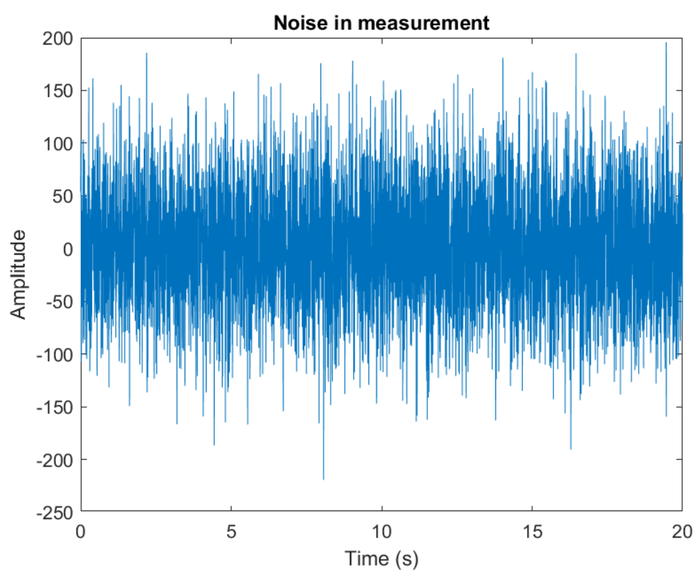


Figure 4: Noise graph at time domain

Then, the measurement signal was converted from time domain to frequency domain with the help of Fourier transform and a power-frequency graph was obtained.

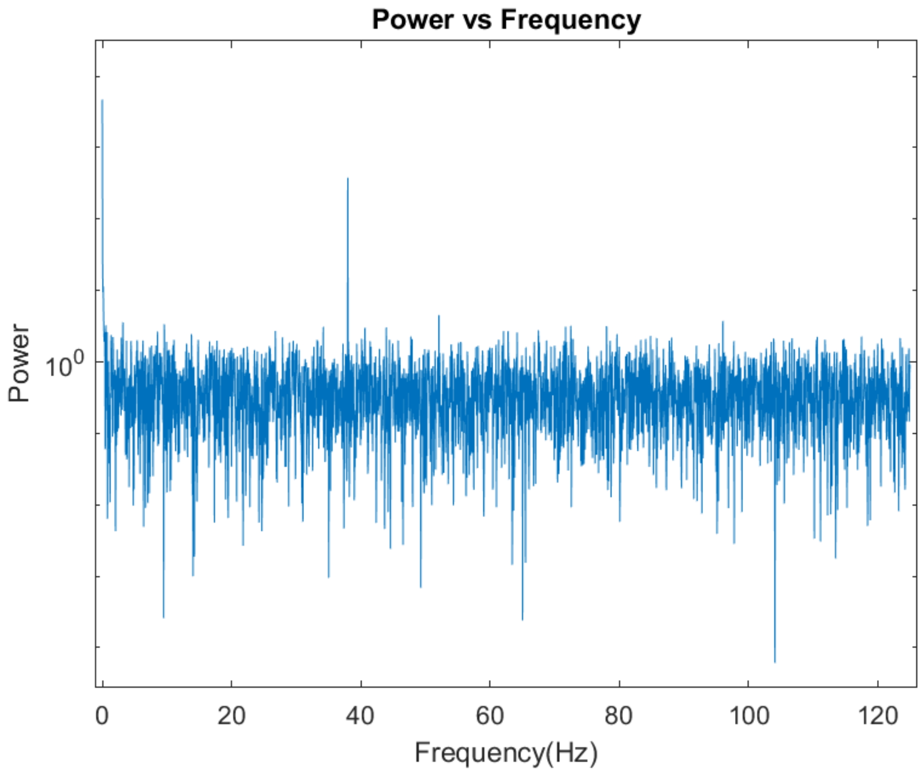


Figure 5: Measurement signal in frequency domain

This is our filter kernel in the time domain in the graph above, in Figure 6. Figure 7 below shows the behavior of our filter kernel in the frequency domain.

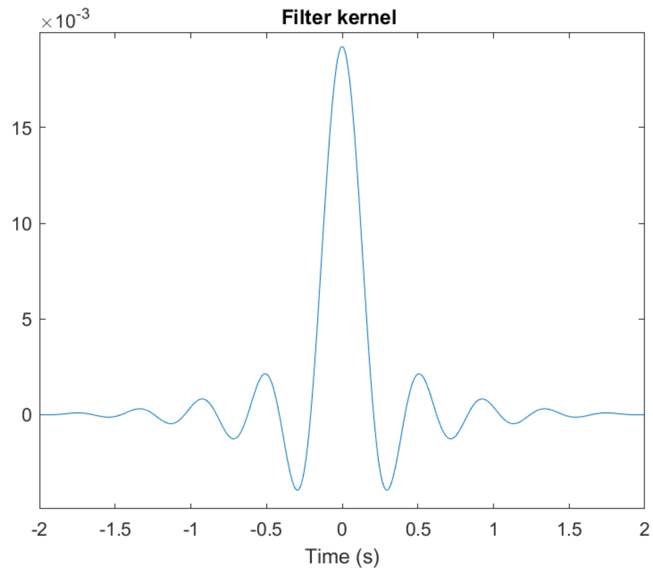


Figure 6: Filter Kernel in time domain

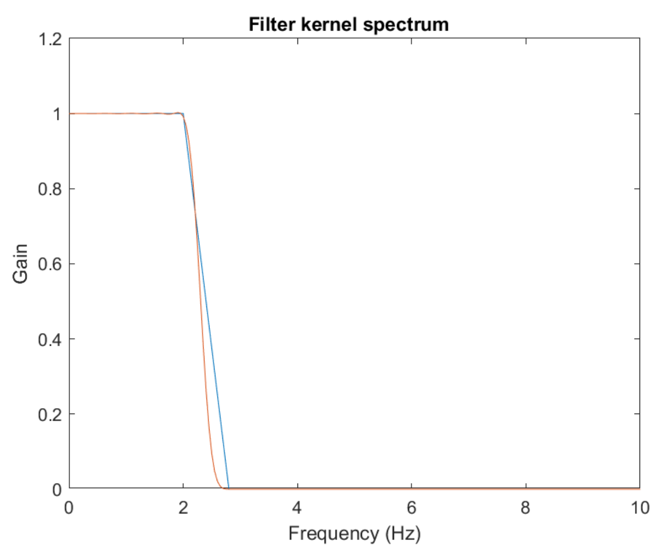


Figure 7: Filter Kernel Spectrum frequency domain

The graph in figure 8 is a graph comparing measurement signal and filtered signal.

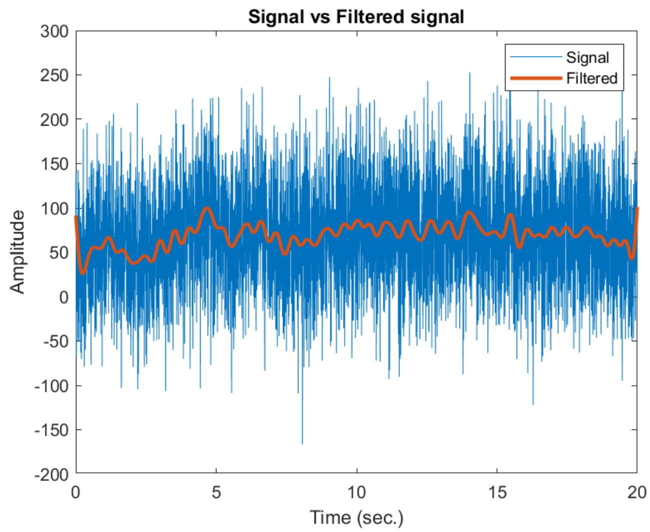


Figure 8: Measurement signal vs Filtered signal in time domain

Figure 9 shows the comparison of the filtered measurement signal and the measurement signal in the frequency domain for the part after cut-off.The part after the cut-off shows how the signal decreases.

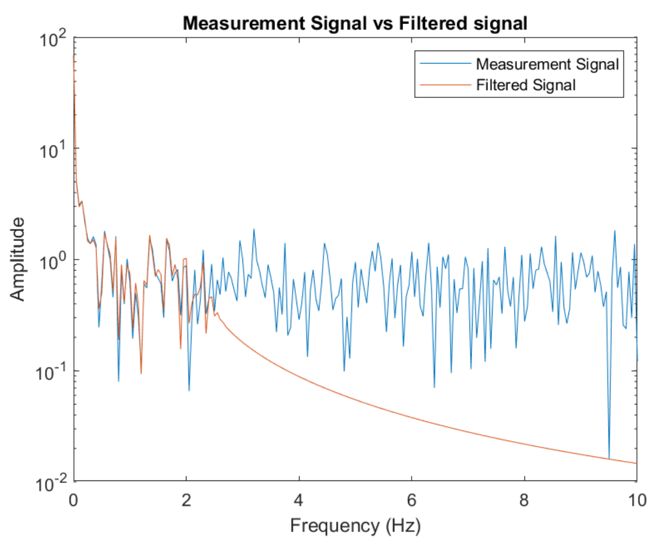


Figure 9: Measurement signal vs filtered signal in frequency domain

In the graph in figure 10, the comparison of filtered data and original data was made in the time domain.

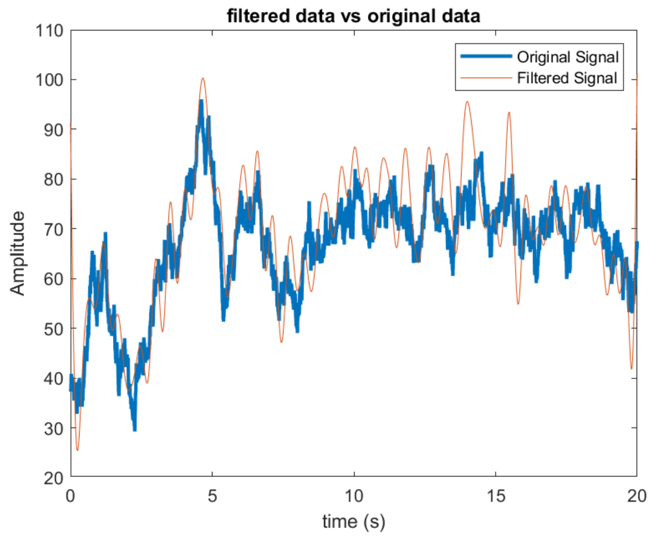


Figure 10: Filtered signal vs original data in time domain

The ifft-applied version of the filtered and fft-applied signal shown in figure 9 was compared with the original data in time domain in Fig 11

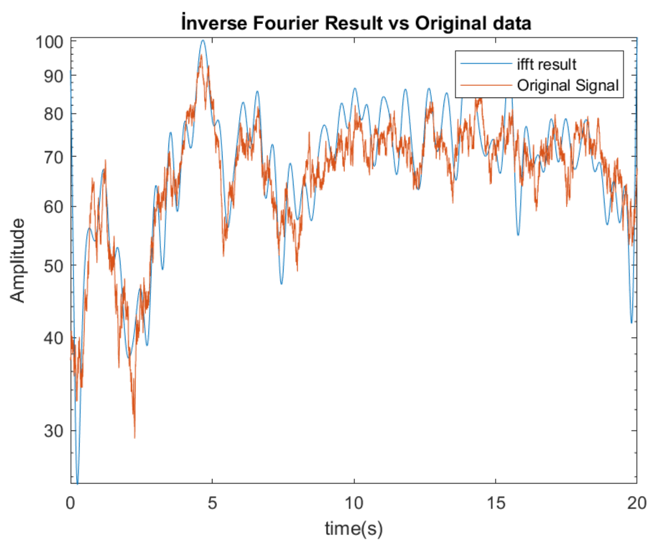


Figure 11: Inverse Fourier Result vs Original Data in time domain

3 Results

As we see in figure 11, ifft results do not exactly match the original data. Thus, in figure 12, we see that the ifft applied signal was plotted by taking the difference between the original data given to us. The reason why the filtered signal differs from the original signal is that signals below a certain frequency value are received. When we apply ifft to the filtered signal, we only apply it to signals below the frequency value we choose. Therefore, information loss, that is, an error, occurs between the original and filtered data.

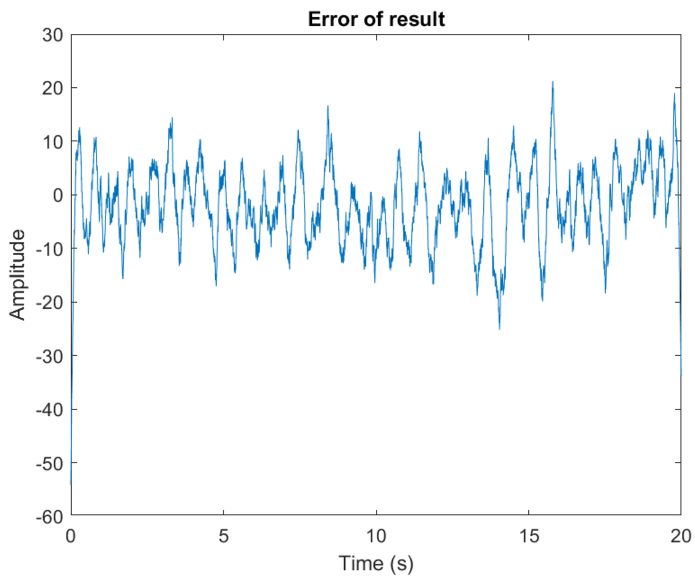


Figure 12: Error between the original and filtered data

4 References

[1]Paul Horowitz and Winfield Hill, 1980, "The Art of Electronics".

[2]T.H. Lee, 1998, "Active Filter Design".

[3]Nick Davis, 2023, "An Introduction to Filters".

5 Appendix

load('GroupData$_1 0.mat'$');

figure plot(timevec, measurement); hold on; plot(timevec, original, 'LineWidth', 2); xlabel('Time (s)'); ylabel('Amplitude'); title('Measurement Signal vs Original Signal'); legend('Measurement Signal','Original Signal');

noise=measurement-original;

meanoforig= mean(original);

figure plot(timevec,noise); xlabel('Time (s)'); ylabel('Amplitude'); title('Noise in measurement');

fs=1/mean(diff(timevec)); points= length(timevec);

hz=linspace(0,fs/2,floor(points/2)+1);

fftofmeasurement=fft(measurement)/points;

amp=abs(fftofmeasurement); amp(2:length(hz))=2*amp(2:length(hz));

pow=abs(fft(measurement)/points).^{2};

figure semilogy(hz,pow(1:length(hz))); xlabel('Frequency(Hz)'); ylabel('Power'); title('Power vs Frequency')

nyq=fs/2;

fcutoff = 2 ; transw = 0.4; order = round(8 * fs / fcutoff); shape = [1 1 0 0]; frex = [0 fcutoff fcutoff+fcutoff*transw nyq] / (nyq);

filtkern = firls(order, frex, [1 1 0 0]);

filtkernX = abs(fft(filtkern, points)).^{2};

figure plot((-order / 2: order / 2) / fs, filtkern) xlabel('Time (s)') title('Filter kernel ') figure plot(frex * fs / 2, [1 1 0 0]) hold on; plot(hz, filtkernX(1:length(hz))) set(gca, 'xlim', [0 10]) xlabel('Frequency (Hz)'), ylabel('Gain') title('Filter kernel spectrum ')

yFilt = filtfilt(filtkern, 1, measurement);

figure plot(timevec, measurement); hold on; plot(timevec, yFilt, 'LineWidth', 2); legend('Signal', 'Filtered') xlabel('Time (sec.)'), ylabel('Amplitude') title('Signal vs Filtered signal')

fftoffiltered = fft(yFilt) / points ;

powerfftmeas = abs(fft(measurement) / points); powerfftoffiltered = abs(fft(yFilt) / points);

figure semilogy(hz, powerfftmeas(1:length(hz))); hold on; semilogy(hz, powerfftoffiltered(1:length(hz))); set(gca, 'xlim', [0 10]); xlabel('Frequency (Hz)'), ylabel('Amplitude') title('Measurement Signal vs Filtered signal') legend('Measurement Signal','Filtered Signal'); fftoforiginal= fft(original) / points ;

figure plot(timevec, original, 'linew', 2); hold on; plot(timevec, yFilt); xlabel('time (s)'), ylabel('Amplitude') title('filtered data vs original data') legend('Original Signal','Filtered Signal');

invfftfilty=ifft(fft(yFilt));

figure semilogy(timevec,invfftfilty); hold on; plot(timevec,original); xlabel('time(s)'), ylabel('Amplitude') title(' İnverse Fourier Result vs Original data') legend('ifft result','Original Signal');

errorflt= original-invfftfilty; figure plot(timevec,errorflt); xlabel('Time (s)'); ylabel('Amplitude'); title('Error of result');