EECE 340 Project-Section 3.1 Filtering and System Design

Prepared by: Carl Wakim and Joseph Chahine

Introduction

In this section, we focus on designing and implementing a low-pass FIR filter to process a recorded voice signal. The filter is created using a windowed sinc function to limit the frequency content of the signal, aiming to remove high-frequency noise while preserving essential speech components. The filter is applied through convolution, and its performance is evaluated by observing its behaviour in both the time and frequency domains.

MATLAB file description

apply_lpf.m:

This function takes a signal and its sampling rate, designs a low-pass FIR filter with a 1300 Hz cutoff, and filters the signal. It uses a sinc function with a Hamming window to create the filter. The function also plots the filter's impulse and frequency responses, and returns the filtered signal.

test_filtered_signal.m**:**

This script loads a recorded voice signal and extracts 2 seconds from 3 to 5 seconds. It calls apply_lpf to filter this segment. Then, it plots the original and filtered signals in the time domain, and also compares their frequency content by plotting their spectra using the ftr function.

Choice of cutoff: The cutoff frequency was chosen as **1300 Hz** based on the spectral analysis of the recorded voice signal. The frequency spectrum showed that most important speech information lies below 1200 Hz, while frequencies above this range contained mainly weaker components or noise. Selecting a cutoff at 1300 Hz preserves key speech content while effectively attenuating unwanted high-frequency noise, ensuring clarity without losing essential information.

Figures description

Figure 1: Frequency Spectrum

The figure shows the normalized magnitude spectra of the voice signal **before and after filtering**, computed using the custom `ftr` function. The original spectrum (blue) shows energy distributed up to around **1600 Hz**, with several peaks between **200 Hz and 1200 Hz** and some higher-frequency noise. After filtering, the filtered spectrum (red) shows that frequencies above the cutoff at **1300 Hz** have been significantly attenuated, while preserving the dominant lower-frequency components below **1000–1200 Hz**. This demonstrates that the low-pass filter effectively reduced high-frequency noise while retaining the primary speech content.

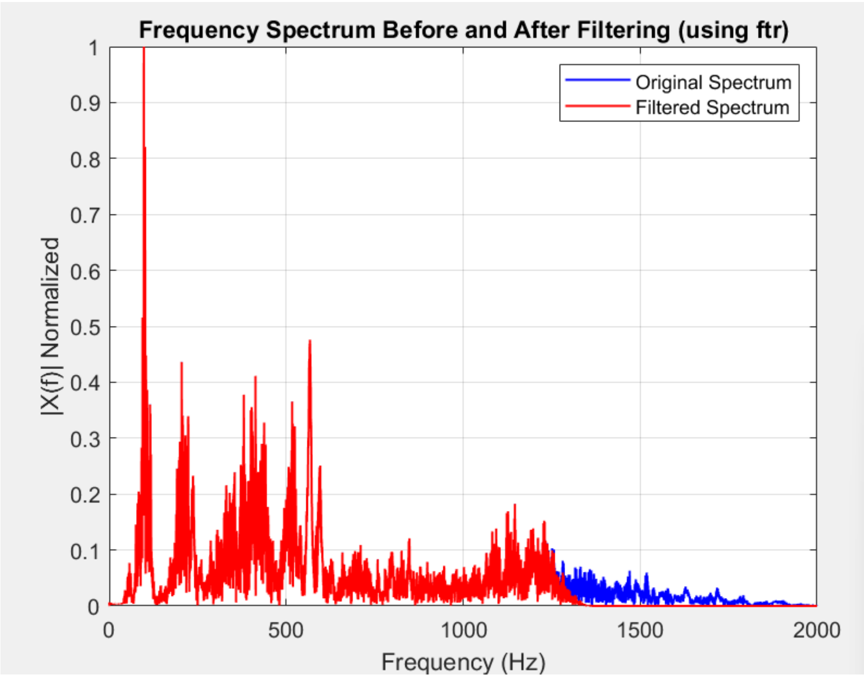
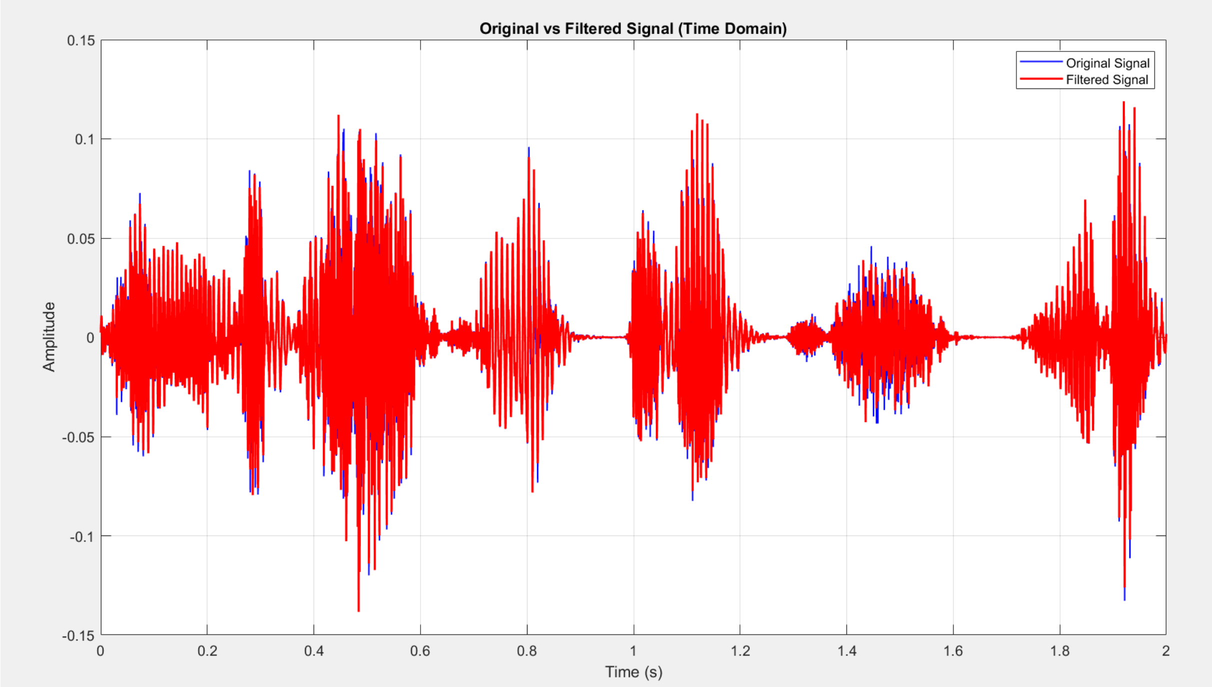


Figure 2: Signal in Time Domain

This figure shows a comparison between the original signal (blue) and the filtered signal (red) in the time domain over a 2-second segment. The filtered signal closely follows the overall shape of the original waveform with less high-frequency variation. This indicates that the low-pass filter preserved the main signal structure while reducing high-frequency noise and fine details. The alignment between both signals confirms that important low-frequency content was retained.



Additional demonstration:

We decrease the cutoff frequency even further to **500 Hz** to **amplify the effects of filtering**. This aggressive filtering is done primarily for **demonstration purposes**, as it **removes significant portions of the speech signal**. As a result, the reconstructed signal will realize even less reconstruction in regions where high-frequency content was present. The waveform of the reconstructed signal under this narrow bandwidth constraint looks like this:

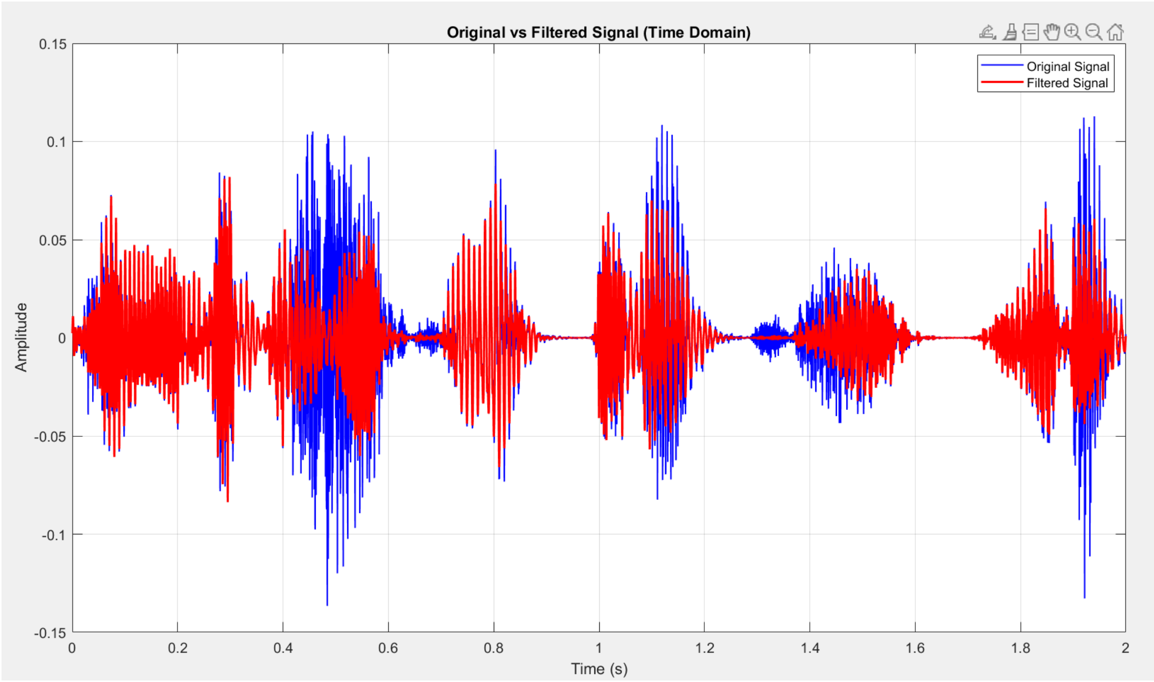


Figure 3: Frequency Response

This plot shows the magnitude frequency response of the designed low-pass FIR filter, computed using the custom ftr function. The response is nearly flat at a magnitude of 1 for frequencies below approximately 1300 Hz, confirming the filter passes low frequencies without attenuation. Beyond the cutoff, the response sharply drops toward zero, attenuating higher frequencies. The symmetry around 0 Hz is due to the linear-phase nature of the FIR filter. Since it resembles a rect function in the frequency domain, we expect to see a sinc function in the time domain as Impulse response.

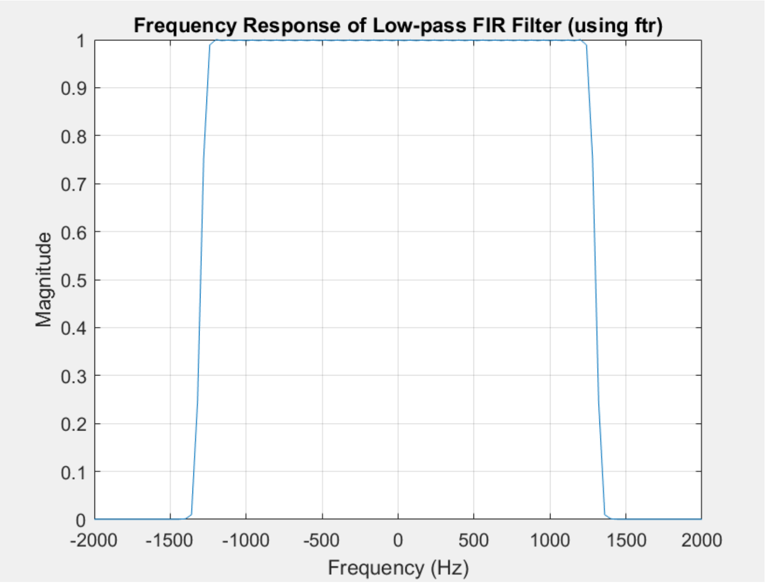
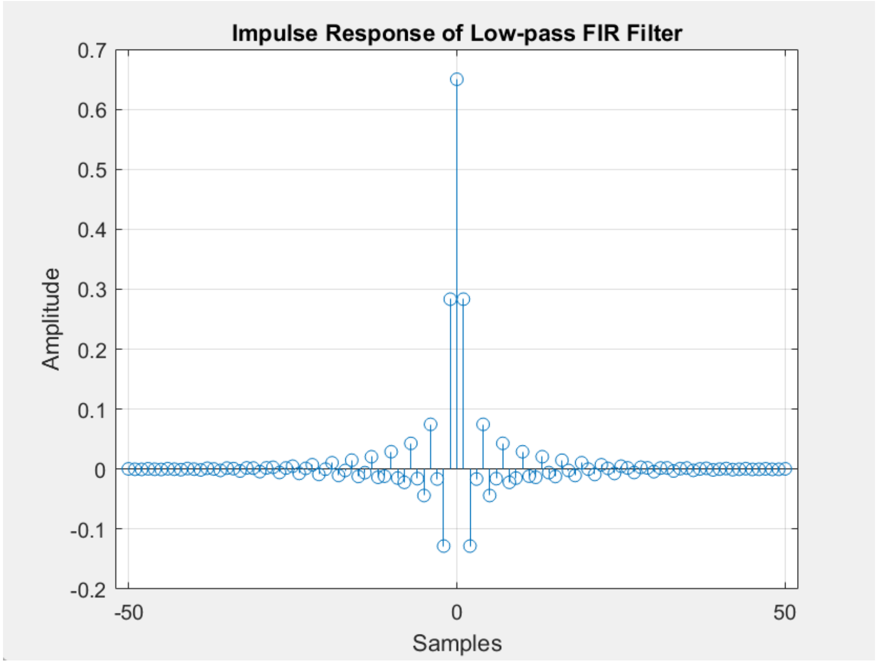


Figure 4: Impulse Response

This figure displays the impulse response of the low-pass FIR filter. The impulse response is a symmetric, windowed sinc function (just as we expected) centered at zero characteristic of a linear-phase FIR filter. The central peak corresponds to the main weight of the filter, while the decaying sidelobes show the sinc function’s oscillatory behavior. The finite duration and smooth shape result from applying a Hamming window to truncate the ideal infinite sinc.



Stability and Causality:

Impulse Response Analysis

The impulse response h[n] of the designed FIR low-pass filter includes negative values and is symmetric around its center. The presence of negative indices indicate that the filter is **not causal**, as causal systems require the impulse response to be zero for all n<0. However, the filter is designed with a finite number of coefficients (a finite impulse response), meaning h[n] is nonzero only for a limited number of values. Because of this, the impulse response is **absolutely summable**, which ensures that the system is **BIBO stable**. Therefore, the filter is **stable but non-causal.**

If we want to look at it from a Z-transform point of vue, we need to look at H(z) in the z domain. It is calculated by the familiar formula:

H(z)=$\sum_{-N}^{N} h[n].z^{-n}$

Where N is the filter order. In total we have 2N+1 taps (samples) so we already know from the formula that it will be stable because N is a finite number therefore we have finite number of terms, meaning that $\sum_{-\infty}^{+\infty} |h[n]| < \infty$ (BIBO stability). Now looking at H(z) it will simply be a summation of the values of h[n] at different n values:

h[-50].z^50 + h[-49].z^49 + … + h[0]+h[1]z^-1 + … + h[50]z^-50

since for negative value of n, h[n]$\neq 0$, then it is not causal