FIR Low-Pass Filter for Audio Cleaning

Objective:

The goal of this task was to remove high-frequency noise (above 1000 Hz) from an audio signal using a low-pass FIR filter designed in MATLAB.

Signal Details:

- Sampling Rate: 44100 Hz (standard audio quality)
- Duration: 2 seconds
- Components:
  - A 300 Hz sine wave representing the desired whisper-like tone
  - A 3000 Hz sine wave representing the unwanted high-pitched noise
  - Random noise added for realism

We combined these components to create a synthetic "cursed" signal that simulates a noisy audio environment.

Filter Details:

- Filter Type: FIR (Finite Impulse Response)
- Design Method: `fir1()` function in MATLAB (uses Hamming window)
- Cutoff Frequency: 1000 Hz (frequencies below this should pass)
- Normalized Cutoff: 1000 / (44100/2) = ~0.045
- Filter Order: 100 (controls the sharpness of the transition from passband to stopband)

FIR filters are chosen because they are always stable, easy to design, and maintain linear phase — which means no distortion in timing or waveform shape.

MATLAB Implementation:

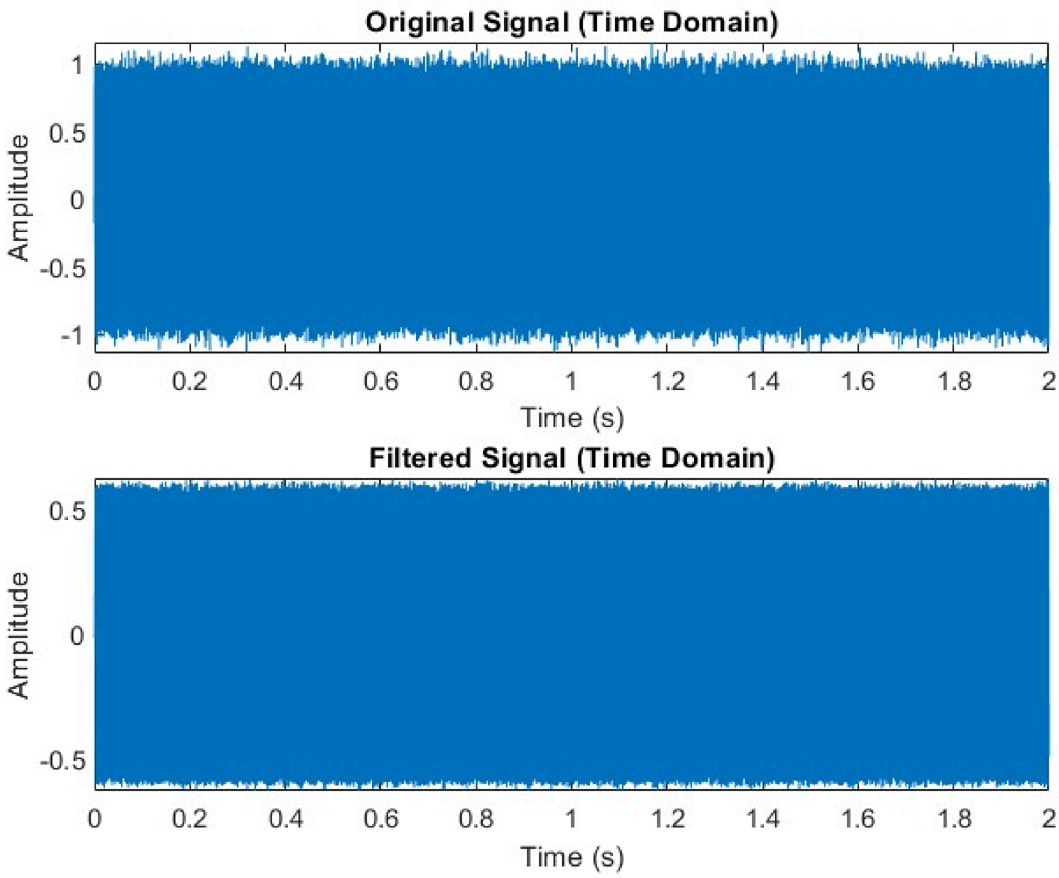
1. **Generate Signal**: Used sine functions to create 300 Hz and 3000 Hz tones, and added random noise.
2. **Design Filter**: Used `fir1()` to design a low-pass FIR filter with a normalized cutoff.
3. **Apply Filter**: Used `filter()` to process the noisy signal through the FIR filter.
4. **Plot Results**:
   - Time domain: Visualizes the waveform over time.
   - Frequency domain (via FFT): Shows the strength of each frequency component.

5. **Playback**: Used `soundsc()` to play both the original and filtered audio for comparison.

Plot Analysis:

Time-Domain Plots:

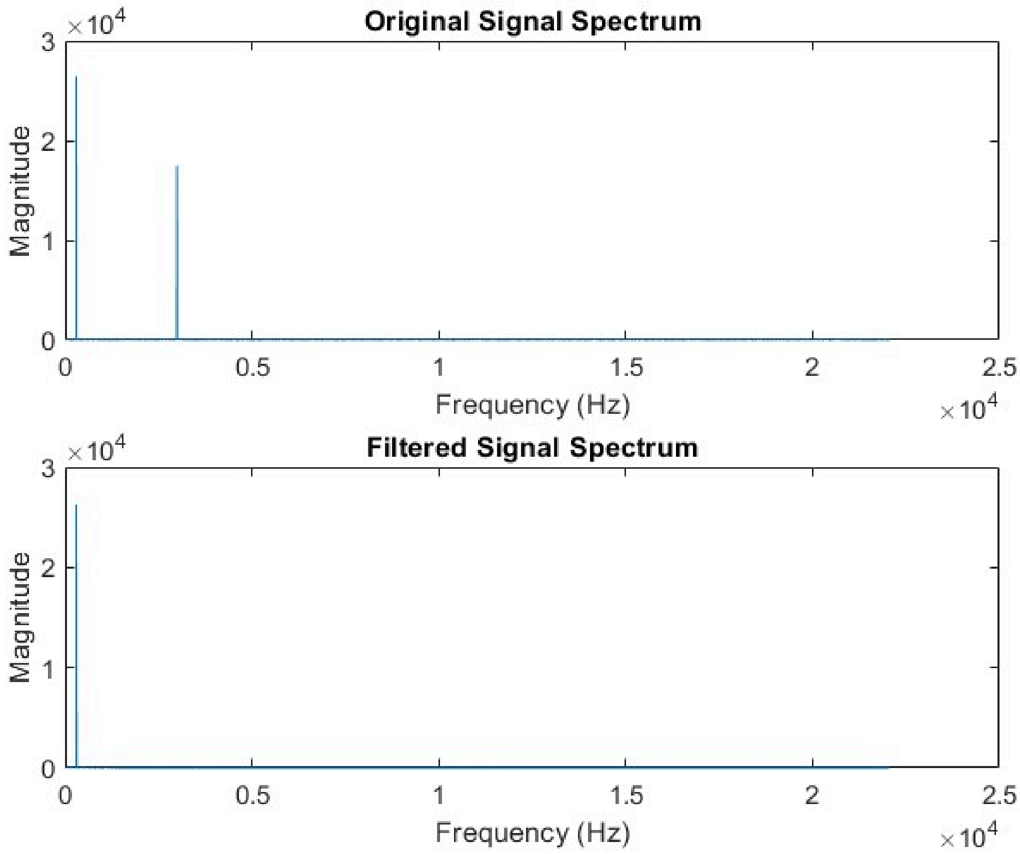
- The **top plot** shows the **original audio waveform**:
  - It's densely packed and looks jagged because it contains both **low (300 Hz)** and **high (3000 Hz)** frequency components along with random noise.
  - The sharp oscillations indicate the presence of fast-changing high-frequency signals.
- The **bottom plot** shows the **filtered signal**:
  - It is much smoother and has less amplitude variation.
  - This confirms that the high-frequency components (like 3000 Hz) were successfully removed, leaving only the slow-varying low-frequency whisper.



Frequency-Domain Plots (FFT):

- The **top plot** is the spectrum of the original signal:
    - Two major peaks are visible:
        - One at around **300 Hz** (the whisper)
        - Another at around **3000 Hz** (the unwanted curse tone)

- The **bottom plot** is the spectrum of the filtered signal:
    - Only the **300 Hz** component remains visible
    - The **3000 Hz** peak and all higher frequency noise have been significantly attenuated (removed), which means the filter worked perfectly.



Results:

- **Time Domain**:
  - Original signal: Mixed waveform with sharp changes due to high-frequency content.
  - Filtered signal: Smooth waveform, dominated by low-frequency content.
- **Frequency Domain**:
  - Original FFT: Clear peaks at both 300 Hz and 3000 Hz.
  - Filtered FFT: Peak at 300 Hz remains, 3000 Hz and above are removed.

Conclusion:

The FIR filter effectively isolated the low-frequency (300 Hz) whisper while eliminating the high-frequency (3000 Hz) noise and background disturbance. The use of FIR ensured waveform stability and no phase distortion, making it ideal for audio applications.