Filter Design Project

CSE 3313 – 001 Fall 2020

Due: Monday, Dec 7

Submission Format: .zip file

Description

In this project, you will be designing a filter to remove unwanted noise from an audio file. You will accomplish with the paper design of a Butterworth lowpass filter. The designed filter will then be implemented in MATLAB and used to filter the audio and compare to the unfiltered audio.

1. **Audio Frequency Analysis**

   a. Use the *audioread()* MATLAB command to provide the sampling frequency Fs of the file and read the contents of 'noisyaudio.wav' into an array.
   b. Use your code from the previous project, or the MATLAB *fft()* command to find the DFT of the audio sample.
   c. Form a frequency axis for plotting the DFT. It should have the same number of elements as the input audio sample and span from -Fs/2 to Fs/2.
   d. Plot the magnitude (abs) of the DFT vs frequency with $\omega = 0$ at the center of the plot. This can be accomplished using the *fftshift()* command. The plot should show energy in the speech frequencies from about 100 Hz to 2 kHz. The plot should also show significant high-frequency noise from about 2.5 kHz to 5.5 kHz. The goal will be to design a filter that will remove this high-frequency noise occurring above 2.5 kHz, while having minimal impact on the speech frequencies up to around 2 KHz.
   e. Plot the normalized log plot of the DFT, where the max value of the DFT is 0 dB.

2. **Filter Design**

   f. Based on the DFT of the signal, decide on a digital frequency ω_p for the end of the passband and a digital frequency ω_s for the beginning of the stop band.
   g. Assume no more than 1dB attenuation in the passband. Based on 1e above, decide on the appropriate minimum attenuation for the stopband so that the noise is attenuated sufficiently.
   h. Use the Butterworth filter design procedure in the lecture for bilinear transformation to design an analog filter according to the specification on the digital filter that you found in 2a and 2b above. This will involve finding the filter order, N, and the Butterworth cutoff frequency, Ω_c. Describe where you are meeting filter requirements and where you are exceeding filter requirements in choosing Ω_c.
   i. Find the equivalent ω_c for the digital filter based on 2h above.

j. Plot the logarithmic gain of the frequency response of your analog filter using the equation below.

$$|H_a(s)| = 20\log_{10}\left(\sqrt{\frac{1}{1+\left(\frac{j\Omega}{j\Omega_c}\right)^{2N}}}\right)$$

3. Filter Implementation

Manual filter implementation would normally involve finding $H(s)$ for the analog filter designed in step 2, converting this to $H(z)$ using the bilinear transformation, and implementing this transfer function using a difference equation or signal flow graph. For higher-order filters, this can take a lot of work on paper! Luckily, we now have computers to do all of this for us.

a. The MATLAB *butter()* command will design a digital filter for us based on our desired digital cutoff frequency. Use the MATLAB *butter()* command to find the numerator and denominator (b and a) polynomial coefficients for a digital Butterworth lowpass filter with the final cutoff frequency Ω_c and order N you found in step 2. In this command, the cutoff frequency is specified as a percentage of $F_s/2$. So, if your F_s is 2000 Hz and you want your cutoff frequency to be 500 Hz, you would use:

$$\mathrm{Wn} = \frac{\Omega_c}{\frac{F_s}{2}} = 0.5$$

in the *butter()* command for cutoff frequency.

b. Use the a and b filter coefficients you found in 3a and the MATLAB *filter()* command to create a filtered version of the original noisy audio sample.
c. Calculate the DFT if the filtered audio sample and plot the magnitude.
d. Compare the unfiltered audio and filtered audio DFT plots. Did the filter reduce the noise in the stop band frequencies and keep the desired speech in the passband frequencies?
e. Use the *sound()* MATLAB command to listen to the original and filtered audio. What difference do you hear?
f. Write the new filtered audio file to 'filteredaudio.wav' using the *audiowrite()* MATLAB command.
g. BONUS POINTS: What movie is the audio from?

4. Report

Write a report to summarize your design, show your plots, and describe your findings. Upload a .zip file with your report, MATLAB code, audio files, and plots.