BELMONT UNIVERSITY, CURB COLLEGE OF ENTERTAINMENT AND MUSIC BUSINESS

AET 5420 Homework 3

Audio Signal Processing

Due: April 1, 2024

1 SYNTHESIZING PINK NOISE

Audio engineers use different types of noise as test signals. Each type of noise is characterized by the amplitude of different frequencies contained in the signal. White noise has (the probability of) equal amplitude at all frequencies. The frequency spectrum of white noise appears flat across the bandwidth of the signal.

Pink noise is a different type of test signal. It has (the probability of) equal amplitude in each octave. This means the amplitude of the signal between 100-200 Hz is equal to the amplitude of the signal between 10,000-20,000 Hz. As a result, the frequency spectrum of pink noise appears to decrease for higher frequencies.

Pink noise can be created by filtering white noise. The filter should be used to decrease the amplitude of the white noise by 3 dB in each octave compared to the previous octave.

1.1 PROBLEM

For this problem, you will create a **script** (m-file) to synthesize a pink noise signal.

- Name the script - prob2.m
- Start by synthesizing a white noise signal
  - Fs - 48000 Hz
  - Duration - 5 seconds
- Create a filter

  - *fir2* - filter design function
  - Use a loop to determine the normalized frequencies of each octave up to Nyquist and the linear gain at each frequency
  - Store the frequencies and gains in separate arrays to use with the fir2 function

- Convolve the white noise with the filter to create pink noise

- Plot the magnitude (amplitude only, not phase) response using the DFT of the signal
  - Use a decibel scale for the vertical axis
  - Use a logarithmic scale for the horizontal axis - *semilogx*

- Save the pink noise to a wav file as part of your script

2 Band-pass and Band-stop Filters

A band-pass filter is a type of spectral processor in which low frequencies and high frequencies outside of a frequency band are reduced in amplitude. The frequencies within the band are not changed in amplitude and pass through the system without being attenuated.

A band-stop filter is a type of spectral processor in which the frequencies within a frequency band are reduced in amplitude. The low frequencies and high frequencies outside of the band are not changed in amplitude and pass through the system without being attenuated.

Both the band-pass and band-stop filters can be created by combining simple low-pass (LPF) and high-pass filters (HPF). In the case of the band-pass filter, it can be created by combining the LPF and HPF in series, shown in Fig. 2.1. In the case of the band-stop filter, it can be created by combining the LPF and HPF in parallel, shown in Fig. 2.2.

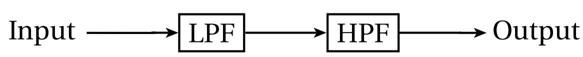

Figure 2.1: Block Diagram of Band-pass Filter

2.1 Problem

The purpose of this problem is to experiment with the creation of filters. Create and save two MATLAB **functions** and an m-file **script** used to analyze the functions.

- Name the first function - bandPass.m

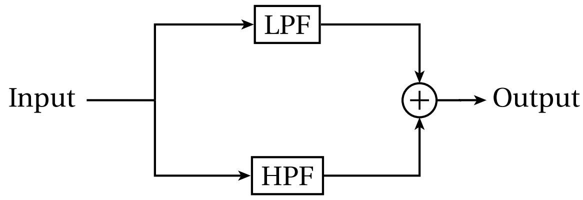

Figure 2.2: Block Diagram of Band-stop Filter

- The output variable of the function should be a array representing the processed audio file
- There should be three (3) input variables
  * input : column vector of a sound file to be processed
  * lowFreq : lower cut-off frequency (0 to 20 kHz)
  * highFreq : higher cut-off frequency (0 to 20 kHz)
- Create a LPF based on the **higher** cut-off frequency.
- Create a HPF based on the **lower** cut-off frequency.
- Process the input signal sequentially through the LPF and HPF.
  * Assign the result to the output variable

- Name the second function - bandStop.m
  - The output variable of the function should be a array representing the processed audio file
  - There should be three (3) input variables
    * input : column vector of a sound file to be processed
    * lowFreq : lower cut-off frequency (0 to 20 kHz)
    * highFreq : higher cut-off frequency (0 to 20 kHz)
  - Create a LPF based on the **lower** cut-off frequency.
  - Create a HPF based on the **higher** cut-off frequency.
  - Process the input signal in parallel through the LPF and HPF.
    * Assign the result to the output variable

- Create a test script to analyze your functions named - problem2.m
  - In this script you will be calling your functions to analyze their performance
  - At the beginning of the script, create three (3) variables
    * impulse : column vector of an impulse signal used to analyze your functions
    * lowFreq : lower cut-off frequency (0 to 20 kHz)
    * highFreq : higher cut-off frequency (0 to 20 kHz)

- Process the impulse signal using your bandPass.m function
    - Plot the result using **freqz(output)**
- Process the impulse signal using your bandStop.m function
    - Plot the result using **freqz(output)**

Remember to add comments to your code to explain what each command is accomplishing.

3 Spectral Analysis of Systems

Given the following system, analyze its spectral characteristics with the frequency response and pole-zero plot. Refer to the following difference equation:

$$y[n] = (0.45) * x[n] + (-0.5) * x[n-1] + (0.05) * x[n-2]$$

Draw the block diagram for this difference equation:

Take the $\mathbb{Z}$ - transform:

Re-write the result in the form of a transfer function $\frac{Y[z]}{X[z]} = H[z]$:

Draw the magnitude response for this system (use Matlab). What type of filter is this?

Re-write the numerator in the form of a characteristic equation. Solve for z.

Plot the result for the numerator as ‘zeros’ within the unit circle on the complex plane.

4 Submission

To submit your homework, create a single zip file that contains the MATLAB function, along with the included sound file and GUI. Name the zip file: xxx_AET5420_HW3.zip, where *xxx* is your last name. Email the zip file to: eric.tarr@belmont.edu by 3:00 pm on April 1st.