Digital Signal Processing-Laboratory

Music Signal Analysis & Processing

1. Introduction

Musical tones are sounds with a fixed pitch produced by sounding objects vibrating regularly, such as 1(Do), 2(Re), 3(Mi) in music. A string of musical tones arranged in order of pitch is the scale. A scale ordered by increasing pitch is an ascending scale, such as 1(Do)2(Re)3(Mi) 4(Fa)5(So)6(La)7(Ti). Tones are composed of sinusoidal signals of different frequencies. The simplest mathematical model is $cos(2\pi ft)$. Chords are generally composed of 3 (or more than 3) musical notes with a certain pattern, such as $x(t) = cos(2\pi f_1 t) + cos(2\pi f_2 t) + cos(2\pi f_3 t)$.

Follow the step-by-step instructions below and using your DSP knowledge learned in class, design and implement MATLAB programs to analyze and process the music. You are required to submit the following documents before the end of this semester:

- The speech file after processing.
- All the MATLAB file you have used.
- A report with plots, your observations and understanding.

2. Frequency-domain Analysis

a) In MATLAB, read the music file (C major ascending scale) using "[x,Fs]=audioread('Tones.wav');", which returns the sampled data x and a sample rate for that data, Fs. Use MATLAB function "sound(x,Fs)" to play the speech.

b) Compute and plot the magnitude spectrum of the signal using DFT. Analyze the frequencies corresponding to each tone of the C major scale.

c) Use DFT to analyze the chord 'Chord.wav' and identifies the tones composed in this chord.

Note	C_4	$C^{\#}_4/D^{b}_4$	D_4	…	F_4	$F^{\#}_4/G^{b}_4$	G_4	$G^{\#}_4/A^{b}_4$	A_4
Frequency	261.63	277.18	293.66	…	349.23	369.99	392.00	415.30	440.00

3. Filter Designs

The file 'Tones_currupted.wav' is corrupted by high frequency pitches. Please following the following procedures to design filters and remove the interference.

- Design an IIR filter to reject the interference signal. Sketch your desirable $H(z = e^{j\omega})$ and obtain the coefficients a and b, your filter coefficients.
- Design an FIR filter using window method to reject the interference signal. Sketch your desirable $H_d(\omega)$ and obtain $h[n]$, your filter coefficients.
- Using convolution algorithm to process the corrupted signal with your designed filter. Then truncate to get the valid (central) portion of the output signal.

- Design an FIR filter using frequency sampling method to reject the interference signal. Sketch your desirable $H_d(\omega)$ and obtain $h[n]$, your filter coefficients.
- Compare the above three filters obtained using IIR and FIR methods. Choose one of them to process the corrupted music signal.
- Use “sound(x,Fs)” to listen to your result. Then, output your result using audiowrite('Tones_recovered.wav', x, Fs);

Useful MATLAB functions:

- Y = fft(X, N): returns the N-point DFT. If no value is specified, Y is the same size as X.
- X = ifft(Y, N): returns the N-point inverse Fourier transform of Y by padding Y with trailing zeros to length N.
- Y = fftshift(X): rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array. If X is a vector, then fftshift(X) swaps the left and right halves of X.