AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

i-CREDIT HOURS ENGINEERING PROGRAMS

COMPUTER ENGINEERING AND SOFTWARE SYSTEMS PROGRAM

ECE 251 : Signals and Systems Fundamentals
Course Project Description
Fall 2022

1 Objectives:

- Become familiar with Matlab / GNU-Octave.
- Use Matlab / GNU-Octave to deal with signals in time and frequency domain.
- Use Matlab / GNU-Octave to design Butterworth low-pass and high-pass filters.

2 Introduction:

- The musical notes are sinusoidal waves whose frequencies are defined by the following equation

$$f_n = f_0 \cdot \alpha^n \tag{1}$$

where n is an integer, $f_0 = 440$ Hz, and $\alpha = 2^{(1/12)}$.

- The frequency of the musical note DO is $f_{(-9)}$, that is to substitute for $n = -9$ in equation (1)

$$f_{(-9)} = 440 \times 2^{(-9/12)} = 261.6256 \text{ Hz}$$

- The Musical notes in a C-Major musical scale are (DO, RE, MI, FA, SOL, LA, TI, DO). The frequencies of these musical notes are defined by equation (1) for the following values of the integer n

$$n_{\text{C-Major}} = [\underbrace{-9}_{DO}, \underbrace{-7}_{RE}, \underbrace{-5}_{MI}, \underbrace{-4}_{FA}, \underbrace{-2}_{SOL}, \underbrace{0}_{LA}, \underbrace{2}_{TI}, \underbrace{3}_{DO}] \tag{2}$$

Note that $n = -9$ corresponds to DO, where as $n = 3$ corresponds to another DO with a higher frequency. It goes the same way for all musical notes.

3 Steps:

1. (4%) Generate four signals $x_1(t)$, $x_2(t)$, $x_3(t)$ and $x_4(t)$ which correspond to the four musical notes DO, RE, MI and FA. Let the time duration of each musical note be half a second.

$$x_1(t) = \cos\left(2\pi f_{(-9)} t\right)$$
$$x_2(t) = \cos\left(2\pi f_{(-7)} t\right)$$
$$x_3(t) = \cos\left(2\pi f_{(-5)} t\right)$$
$$x_4(t) = \cos\left(2\pi f_{(-4)} t\right)$$

What is an appropriate sampling frequency f_s in this case?

2. (4%) Create a signal $x(t)$ which corresponds to sequentially playing the musical notes (DO, RE, MI, FA) which you have created in the previous step. Store the generated signal $x(t)$ as an audio file with extension (*.wav)

3. (4%) Plot the signal $x(t)$ versus time t.
4. (4%) Compute the energy of the signal $x(t)$.
5. (4%) Compute the frequency spectrum $X(f)$ of this signal.
6. (4%) Plot the magnitude of $X(f)$ in the frequency range $-f_s/2 \leqslant f \leqslant f_s/2$, where f_s is the sampling frequency.
7. (4%) Compute the Energy of the signal $x(t)$ from its frequency spectrum $X(f)$, and hence you can verify Parseval's theorem.
8. (4%) Design a Butterworth low-pass filter with filter order 20 such that when the signal $x(t)$ is applied to this filter, the output does NOT contain the MI and FA musical nodes. What is the cut-off frequency of this filter?
9. (4%) Plot the magnitude and phase response of the Butterworth LPF you've designed.
10. (4%) Apply the signal $x(t)$ to this Butterworth LPF and let's denote the output signal as $y_1(t)$.
11. (4%) Store the generated signal $y_1(t)$ as an audio file with extension (*.wav)
12. (4%) Plot the signal $y_1(t)$ versus time t.
13. (4%) Compute the energy of the signal $y_1(t)$.
14. (4%) Compute the frequency spectrum $Y_1(f)$ of this signal.
15. (4%) Plot the magnitude of $Y_1(f)$ in the frequency range $-f_s/2 \leqslant f \leqslant f_s/2$.
16. (4%) Compute the Energy of the signal $y_1(t)$ from its frequency spectrum $Y_1(f)$, and hence you can verify Parseval's theorem.
17. (4%) Design a Butterworth high-pass filter with filter order 20 such that when the signal $x(t)$ is applied to this filter, the output does NOT contain the DO and RE musical nodes. What is the cut-off frequency of this filter?
18. (4%) Plot the magnitude and phase response of the Butterworth HPF you've designed.
19. (4%) Apply the signal $x(t)$ to this Butterworth HPF and let's denote the output signal as $y_2(t)$.
20. (4%) Store the generated signal $y_2(t)$ as an audio file with extension (*.wav)
21. (4%) Plot the signal $y_2(t)$ versus time t.
22. (4%) Compute the energy of the signal $y_2(t)$.
23. (4%) Compute the frequency spectrum $Y_2(f)$ of this signal.
24. (4%) Plot the magnitude of $Y_2(f)$ in the frequency range $-f_s/2 \leqslant f \leqslant f_s/2$.

25. (4%) Compute the Energy of the signal $y_2(t)$ from its frequency spectrum $Y_2(f)$, and hence you can verify Parseval's theorem.

4 Useful Matlab / GNU-Octave Commands:

1. buttord, butter
2. zp2sos, sosfilt
3. freqz
4. fft, fftshift
5. audioread, audiowrite

1. Each **group of** 4/5 **students** should work together and submit one report.
2. Please prepare one compressed file that includes the following items:
   (a) Your Matlab / GNU-Octave codes (*.m files).
   (b) A report (pdf files Only) that includes your output waveform, the energy values to be computed, plots of the filtered signal, etc.
   (c) In your report make sure to clearly indicate the contribution of each member of the group.
   (d) The audio files generated by your code.
3. Project should be submitted on LMS before 11 : 59 PM on December 24^{th} 2022.

Good Luck