University of Toronto

Department of Electrical and Computer Engineering

ECE 356S – System and Signal Analysis II 1999

Exp 4: DIGITAL FILTER DESIGN FOR AUDIO SIGNAL PROCESSING

1 Purpose

In this experiment you'll learn about audio signal processing in MATLAB, and also apply digital filter design techniques for that application.

2 Introduction

Frequencies in Music

In music, twelve consecutive notes constitute an octave. The twelve notes in each octave are logarithmically spaced in frequency, with the frequency of each note being $2^{1/12}$ times the frequency of the note just below it. For example,

Note	Frequency (Hz)
A	110
A$\sharp$, B$\flat$	$110 \times 2^{1/12}$
B	$110 \times 2^{2/12}$
C	$110 \times 2^{3/12}$
.	.
.	.
.	.
G$\sharp$, A$\flat$	$110 \times 2^{11/12}$

Thus, a 1-octave pitch shift upwards corresponds to a doubling of the frequencies of the notes in the original octave.

Sound in MATLAB

There are several formats for storing sound files on a computer. We'll use the .wav format, which MATLAB can read. There are various sound handling capabilities in MATLAB. For this lab the two functions *wavread*( ) and *sound*( ) are used. They are described as follows:

> $[y, Fs]$ = *wavread*('wavefile') loads the .wav format file *wavefile*, returning the sampled data in variable y and the sample rate in variable Fs.
>
> *sound*(y) sends the signal in vector y to the speaker. On the UGSPARC machines, the signal is played at the default sampling rate of 8192 Hz.

3 Procedure

Perform the lab on a UGSPARC (or a computer of your choice) with speakers. (Most have speakers. Make sure yours does.)

1. Download the file *lab4data.mat* from the course web site.

   This file is in MATLAB's .mat format. From MATLAB, load the file (type *load lab4data*). Display the saved variables (type *who*). You'll see that two variables, *scale* and *sum*, have been stored.

2. Play *scale* using *sound*. You should hear a series of five musical notes. Write a MATLAB program to generate the same result. Hint: Display the DFT of the signal to see the frequencies.

3. Play *sum*. You should hear the chord consisting of the five notes simultaneously.

4. Using the Kaiser window technique, design a filter that ideally when applied to *sum* will output the first note alone.

5. Download the file *noisy_speech.wav* from the same web site. Load it into MATLAB and play it. You should hear some speech mixed with noise. Design a filter to filter out the noise and thereby improve the quality of the speech. For reference, the speech without noise is *clean_speech.wav*.

4 Your Report

Your report should contain all programs. Discuss the performance of your filters.