University of Washington Bothell
Electrical Engineering Program

BEE 341 Discrete-Time Linear Systems

Lab 1: Elementary Music Synthesis

1. Purpose:

The purpose of this lab is to study how to generate signals of different frequencies and durations mathematically and simulate them in MATLAB. You will gain some understanding of the physical meaning of the signals you construct by using audio playback. We also hope that you have fun!

For this lab, you will need to use a PC with a sound card installed in it. You will need speakers or a set of headphones to listen to your musical creation.

2. Background

In this section, we explore how to use simple tones to compose a segment of music. By using tones of various frequencies, you will construct the first few bars of the Scarborough Fair song. In addition, you will work on improving the perceived quality of the sound your created.

Each musical note can be simply represented by a sinusoid whose frequency depends on the note pitch. **In this lab, you will use a sampling rate of 8 kHz.**

A continuous signal of an amplitude 'A' and of a single frequency f, in Hertz, is defined using a sinusoidal function as follows:

$$x(t) = A\cos(2\pi \times f \times t)$$

The signal x(t) should be discretized to be generated and simulated by MATLAB. The signal is discretized by sampling at uniform period T_s, i.e., by drawing samples at time instants $t = nT_s$ where n is an integer.

$$X(t)|_{t=nTs} = x(nT_s) = A\cos(2\pi \times f \times nT_s) = A\cos(2\pi \times f/fs \times n)$$

By dropping Ts, we write the discrete time signal as

$$x[n] = A\cos(2\pi \times f/fs \times n)$$

Where f_s is the sampling frequency or rate, and is obtained by taking the reciprocal of T_s ($f_s=1/T_s$).

There are seven natural notes: A, B, C, D, E, F and G. After G, we begin again with A. Music is written on a "staff" consisting of five lines with four spaces between the lines. The notes on the staff are written in alphabetical order, the first line is E as shown in Figure 1. Notes can extend above and below the staff. When they do, ledger lines are added

Figure 1. Musical Notes

Musical notes are arranged in groups of twelve notes called *octaves*. The notes that we'll be using for Scarborough Fair are in the octave containing frequencies from 220 Hz to 440 Hz. The twelve notes in each octave are logarithmically spaced in frequency, with each note being of a frequency $2^{1/12}$ Times the frequency of the note of lower frequency.

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

Table 1 shows the ordering of notes in the octave to be used to synthesize the opening of Scarborough Fair, as well as the fundamental frequencies for these notes.

Table 1. Notes in the 220 – 440 Hz octave

Note	**Frequency**
A	220
$A^{\#}$, B^{b}	$220\times2^{1/12}$
B	$220\times2^{2/12}$
C	$220\times2^{3/12}$
$C^{\#}$, D^{b}	$220\times2^{4/12}$
D	$220\times2^{5/12}$
$D^{\#}$, E^{b}	$220\times2^{6/12}$
E	$220\times2^{7/12}$
F	$220\times2^{8/12}$
$F^{\#}$, G^{b}	$220\times2^{9/12}$
G	$220\times2^{10/12}$
$G^{\#}$, A^{b}	$220\times2^{11/12}$
A	440

A musical score is essentially a plot of frequencies (notes) on the vertical scale versus time (measures) on the horizontal scale. The musical sequence of notes for the piece you will synthesize is given in Figure 2.

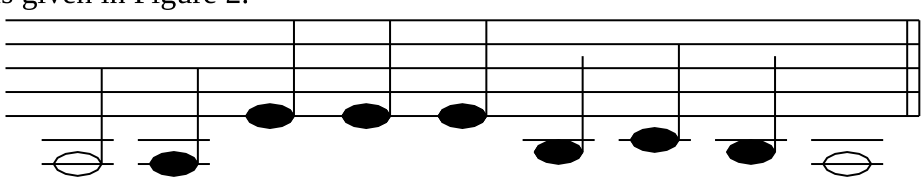
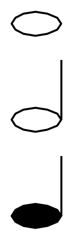


Figure 2. Musical Score for Scarborough Fair.

The following discussion identifies how musical scores can be mapped to tones of specific pitch .

3. Note Frequency

In the simplest case, each note may be represented by a burst of samples of a sinusoid followed by a shorter period of silence (samples of zeros, which are a pause). The pauses allow us to distinguish between separate notes of the same pitch. The duration of each note burst is determined by whether the note is a whole note, half note, fourth note, or an eight note (see Figure 3). Obviously, a fourth note has twice the duration of an eighth note, and so on.



whole note. it lasts 4 counts

half note. it lasts 2 counts

fourth note. it lasts 1 count

Figure 3. Types of notes. <u>In this Lab, use a duration of 4,000 samples</u> <u>for 1 count.</u>

Therefore, your whole notes should be four times the duration of your fourth notes. The short pause you use to follow each note should be of the same duration regardless of the length of the note. Longer periods of silence that are part of the musical score are indicated by one of more rest symbols. There are no rest symbols in the score you are given.

For example, a whole note of ‘A’ followed with a pause of 0.0625 sec can be generated as follows:

```
fa =220 ; % the ‘A’ note has frequency of 220Hz
fs = 8000; % smapling frequency is 8000 samples/sec
n= 0: 15999; % n defines duration, a whole note is 4 counts (4 × 4000 samples)
pause = 0.0625*8000; % pause number of samples in the pause of 0.0625 sec

x=[cos(2*pi*fa/fs*n) zeros(1,pause)]; % the signal is extended by zeros for the pause
```

Note that A-G only yields seven notes; the additional notes are due to changes in pitch called sharps (denoted by the symbol #) or flats (denoted by the symbol *b*) that follows a given note. A sharp increases the pitch by $2^{1/12}$ and a flat decreases the pitch by $2^{1/12}$.There are no sharp or flat symbols in the music you are going to create.

In the musical score in Figure 2, the first half note and fourth note are both A. The next three fourth notes are all E and so on. You can get the fundamental frequencies for these notes from Table 1.

4. Improving perceived quality

There are many ways of improving the perceived quality of a synthesized sound. Here you will learn about two methods: varying the note volume and overlapping individual tones.

4.1 Volume variations

Typically, when a note is played, the volume rises quickly from zero and then decays over time, depending on how hard the key is struck and how long it is depressed. The variation of the volume over time is divided into four segments: Attack, Decay, Sustain, and Release (ADSR). For a given note, volume changes can be achieved by multiplying a sinusoid by another function called a windowing function. An example of such function is shown in Figure 4. You may take A = 5%, D=10%, S=70% and R=15% of the length of a note. The level of A goes from 0 to 1 and the level of D goes from 1 to 0.8.

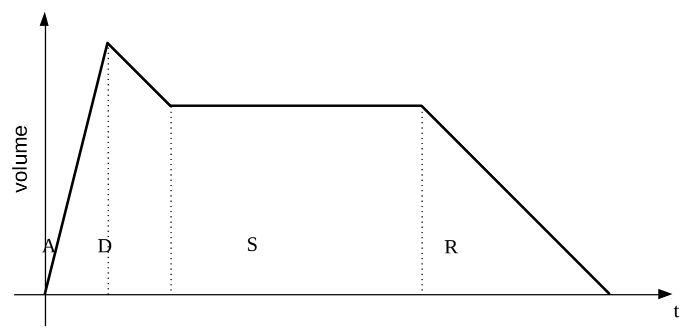


Figure 4. An ADSR Envelope.

4.2 Overlapping tones

Another improvement in perceived quality can be achieved by overlapping some notes as done by advanced piano players. As the volume of one note is decaying, another note is played. Mathematically, this can be accomplished by allowing the time regions occupied by subsequent sinusoids to overlap, hence removing the pause. This will yield a much smoother, less staccato-sounding piece.

5. Assignment

1. What is the minimum sampling rate required to digitize a speech signal whose spectrum ranges from 0 - 3000Hz? (5 points)

2. What is the standard sampling rate used for CD quality music? (*Hint: search online for the information*) (5 points)

3. Synthesize the piece appearing in Figure 2 using only information from Sections 2 and 3. Use a pause of 0.0625 seconds.

- Play it back using the sound(*x*,*fs*)command in Matlab. (Type ***help sound*** for more information. Please specify the sampling rate = 8k Hz in the playback.)

- Discuss the quality of the music you heard. (5 points)

- Turn in a listing of your MATLAB code. (25 points)

4. Improve the quality of the sound with a volume window function (see Section 4.1 above). Try concatenating different function to model ADSR.

- Were you able to improve the sound quality? (5 points)

- Save the modified music synthesis in a new MATLAB file. Also, turn in a listing of this code. (25 points)

Hint: You may create the ADSR function as one vector by concatenating the samples of its each part (A, D, S and R). The samples of each part can be conveniently generated using the MATLAB function ***linspace*(*start, end, number of points*)**.

3. As explained in Section 4.2, allow the decaying notes (i.e. with the windowing function) to overlap slightly in time. The size of the overlap between two consecutive ADSR functions can be 100 samples.

- Were you able to improve the sound quality? (5 points)

- Save the modified music synthesis in a new MATLAB file. Also, turn in a listing of this .m-file. (25 points)

Hint: Here is an example of how to concatenate two vectors with an overlap. Assume we have vector x and y given by,

```
x=[ 1 2 3 4 5 6];
y=[7 8 9 10 11];
```

and we want to concatenate them with an overlap of two samples. The first step is to extend both vectors, by pre and post padding them by zeros, to a length of the final concatenated vector. Note that the length of the final vector is equal to length(x) +length(y) – length(overlap). Therefore, we should extend vector x by **post** padding it with zeros of length equal to length(y)-length(overlap); similarly, we should extend vector y by **pre** padding it with zeros of length equal to length(x)-length(overlap). Then, add the two extended vectors. Here is a MATLAB code which performs the steps explained:

```
x = [ 1 2 3 4 5 6];
y = [7 8 9 10 11];
overlap = 2;
x1 = [x, zeros(1, length(y)-overlap)]; % post zero-padded x
y = [zeros(1,length(x)-overlap), y] ; %pre zero padded y
z =  x1 +y;
```

Also it could be done in one line as:

```
z = [ x(1:end-overlap), x(end-overlap+1:end)+y(1:overlap),  y(overlap+1:end)];
```