BILKENT UNIVERSITY ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT

EEE - 321: Signals and Systems
Lab Assignment 1

Please carefully study this assingment before coming to the laboratory. You may begin working on it or even complete it if you wish, but you do not have to. There will be short quizzes both at the beginning and end of the lab session; these may contain conceptual, analytical, and Matlab-based questions. Within one week, complete the assignment in the form of a report and turn it in to the assistant. Some of the exercises will be performed by hand and others by using Matlab. What you should include in your report is indicated within the exercises.

Part 1

This part contains several basic Matlab exercises. The goal is to briefly remind you about the vector-based programming logic of Matlab so that you can use it in the most efficient manner during our labs. Try all the exercises below in Matlab and answer the related questions. Write down your answers in your report. You do not need to provide any code for the exercises in this part.

- a) First type **a=[3.2 34/7 -6 24]**, then type **a=[3.2; 34/7; -6; 24]**. What is the difference?
- b) Type **a=[3.2 34/7 -6 24];** and **b=[3.2; 34/7; -6; 24];**. What is the difference from part a?
- c) In order to do this item, you need to learn how to measure the computation time of a Matlab code. In order to do that, learn how *tic* and *toc* commands work. You can use $help$ command in Matlab for this purpose. Then measure the time difference

between generation of the variables **a=[3.2 34/7 -6 24]** and **a=[3.2 34/7 -6 24];**. When is it useful to put ";" at the end of a command line?

- d)Type **a=[3.2 34/7 -6 24];** and **b=[5.8 24/5 5 -102];**. Then type **c=a*b**. What message do you receive? Why do you receive this message?

- e) Type **a=[3.2 34/7 -6 24];** and **b=[5.8 24/5 5 -102];**. Then type **c=a.*b**. What is the result? What is the effect of adding the dot in front of the multiplication symbol *? If you type **c=b.*a**, does the result change?

- f) Type **a=[3.2 34/7 -6 24];** and **b=[5.8; 24/5; 5; -102];**. Then type **c=a*b**. What is the result? What has Matlab done now?

- g) Type **a=[3.2; 34/7; -6; 24];** and **b=[5.8 24/5 5 -102];**. Then type **c=a*b**. What is the result? What has Matlab done now?

- h) Type **a=[1:0.01:2]**. What does such a command do?

- i)Then type **a=[1:0.01:2];**. Measure the spent time while generating this variable.

- j) Now generate the variable **a** in part i by using a *for* loop. Again, measure the spent time.

- k) Again generate the variable **a** in part i by using a *for* loop, but, in this case, allocate memory for **a** before the loop. You can allocate memory by using *zeros* or *ones* commands. (You can do a web search to learn how to use these commands for memory allocation). Measure the spent time to generate the variable after memory allocation. Now compare the measured time in parts i, j and k. Which method is the most efficient one?

Now you have learned how to generate basic Matlab variables, and also you have compared the efficiency of different variable generation methods. From now on, you are expected to write your Matlab codes not only correctly but also efficiently. Besides these methods, there might be some other cases that make your Matlab code more efficient in this lab and further labs. You can always do a web search or get help from the TAs on how to make your code more efficient.

- l) Type **a=[0:pi/8:2*pi]**;. Then type **b=sin(a);**. Examine the resulting **b**. Notice that as the input argument for the **sin** function, we used the vector **a**. This does not make sense mathematically because we have to insert a single real-valued number into the $\sin(x)$ function. For instance, $\sin(\frac{\pi}{5})$ has a meaning but $\sin([\frac{\pi}{3}\,\frac{\pi}{5}])$ does not in the ordinary sense. But Matlab still returns a result when we insert **a** into the **sin** function. What does Matlab do?

- m) Type **t=[1:0.02:4];**, and type **x=cos(3/4*pi*t+pi/6);**. Now first type **plot(x)**, then type **plot(x,t)** and **plot(t,x)**. What is the difference? You do not need to include the plots in the report; just provide your answer to this question.

- n) For the above part, type **plot(t,x,'-+')**. What do you observe? Type **plot(t,x,'+')**. What happens?

Type **help plot** in the Matlab command window, and see what else you can do with the **plot** command, which is one of the vital commands of Matlab. Also study the Matlab commands **xlabel**, **ylabel**, **title**, **xlim**, **ylim** and **grid**. These commands are essential for producing professional looking plots in Matlab. Make sure that you can confidently use these commands. As a final exercise to further understand the **plot** command, we will graph the function $x(t) = \sin(2\pi t + \pi/3)$. For this part, provide your answers to the questions that are asked below. You will also provide a graph. Again, no codes are necessary.

- o) Let **t=[0 0.04 0.08 ... 0.92 0.96 1]**. By now, you know that we can prepare the array **t** with the single-line command **t=[0:0.04:1];**. How many time points are included in **t**?

- p) How would you generate the variable **t** in part m using **linspace** command?

- q) Now compute $x(t)$ over the time grid specified by **t**, and denote the resulting array with **x**. By now, you know that we can obtain **x** with the single-line command **x=sin(2*pi*t+pi/3);**.

- r) Type **figure;**. You will see that an empty figure window will be opened. Then type **plot(t,x,'b');**. You will see that your function is plotted, where the color of the curve is blue. Then type **hold on;**. This command will enable you to make further plots within the same figure window while preserving all the old plots.

- s) Now let **t=[0 0.01 0.02 0.03 ... 0.98 0.99 1]** and compute **x** once again. How many time points do we take this time? Type **plot(t,x,'r');** this time. You will notice that a red curve is added to the old figure window.

- t) Add the same sine signal plot with the time index **t=[0 0.1 ... 0.9 1]** with a different color.

- u) Add the same sine signal plot with the time index **t=[0 0.2 0.4 0.6 0.8 1]** with a different color.

- v) Closely examine the figure that you obtained, perhaps zooming in or out. Which choice of **t** produces the plot that is most likely for the continuous $x(t)$? Why?

- w) How does the **plot** command "fill" the space between data points?

- x) After closing the figure window you used during the previous items, repeat the exercise in item s by using the **stem** command instead of the **plot** command. Do not provide any graphs, but just answer the following question: What is the difference between the **plot** command and **stem** command?

Part 2

In this part, we will experience how different signal waveforms sound. Take **t=[0:1/8192:1]** for all the exercises in this part. First, consider a signal of the form

$$x_1(t) = \cos(2\pi f_0 t). \tag{1}$$

- a) First examine **sound** and **soundsc** commands. Are both appropriate to listen the discrete version of above signal in Matlab?
- b) Take $f_0 = 440$. Compute $x_1(t)$ and store it in an array named **x1**. Plot **x1** versus **t**. Turn on the speakers of your computer. Then type **sound(x1)** or **soundsc(x1)**. Listen to the sound.
- c) Repeat a for $f_0 = 687$, but do not produce any plot.
- d) Repeat a for $f_0 = 883$, but do not produce any plot.

What happens to the pitch of the sound as the frequency increases?
Now consider a second signal defined as

$$x_2(t) = e^{-at}\cos(2\pi f_0 t). \tag{2}$$

Take $f_0 = 330$ and $a = 7$. Compute $x_2(t)$ and store it in an array named **x2**. Write a single line code for computing **x2**. In this code, make use of the element-wise multiplication facility of Matlab while computing the product of e^{-at} and $\cos(2\pi f_0 t)$. (Recall that in Matlab, elementwise multiplication of arrays is achieved by placing a dot in front of the multiplication symbol.) Provide this code to your report. Make a plot of **x2** versus **t**, and listen to **x2** by the **sound** command. Compare your plot and what you hear to the results you obtained for $x_1(t)$ when f_0 is 440. What is the effect of adding the e^{-at} term to the sound that you hear? Which one of $x_1(t)$ and $x_2(t)$ resembles the sound produced by a piano more, which one resembles that of a flute more? Now take $a = 2$, and recompute **x2** (do not change **t** and f_0). Compare the sound you hear with that of the $a = 4$ case. Repeat for $a = 11$. How does the duration of the sound that you hear change as a increases?
Next, consider the signal

$$x_3(t) = \cos(2\pi f_1 t)\cos(2\pi f_0 t), \tag{3}$$

where $f_1 \ll f_0$. Take $f_0 = 510$ and $f_1 = 4$. Again, using a single line command, compute **x3** (provide this code to your report) and plot and listen to it. Compare your results with that of $x_1(t)$. How does the low-frequency cosine term $\cos(2\pi f_1 t)$ affect the sound you hear? Recompute **x3** for $f_1 = 2$ and $f_1 = 6$. What is the change in the sound that you hear? By using a trigonometric identity, write $x_3(t)$ as a sum of two different frequency cosine signals and also interpret what you hear in this part using this identity.

Part 3

The instantaneous frequency of a signal of the form

$$x(t) = \cos(2\pi\phi(t)) \tag{4}$$

is defined as

$$f_{ins}(t) = \frac{d\phi(t)}{dt}. \tag{5}$$

Show that the instantaneous frequency of the signal $x_1(t)$ given in Eq. 1 is given as $f_{ins}(t) = f_0$ for all t. Next, consider a signal of the following form

$$x_4(t) = \cos(\pi\alpha t^2). \tag{6}$$

Show that the instantaneous frequency of the signal $x_4(t)$ is given as $f_{ins}(t) = \alpha t$ for all t. Thus, instantaneous frequency changes linearly with time. What is the frequency at $t = 0$? What is the frequency at some $t = t_0$? To get a feeling about the physical implication of the linearly changing instantaneous frequency, let us compute $x_4(t)$ and listen to it. Take **t=[0:1/8192:1]**. Then go to the website **www.random.org** and generate a random integer between 1600 and 2048. You will use this generated number as α. With these selections, what are the values between which the frequency will change? Now compute **x4** again with a single line command and provide this command in your report. Then, listen to **x4**. Now, comment on the physical implication of the linearly changing instantaneous frequency. Keep **t** the same, and repeat the experiment with $\alpha_1 = \alpha/2$ and $\alpha_2 = 2\alpha$. Comment on the changes. In these examples, we only increased the frequency of the sound signal from 0 Hz to α Hz, α_1 Hz and α_2 Hz. Now consider the following signal:

$$x_5(t) = \cos(2\pi(-500t^2 + 1600t)). \tag{7}$$

Take **t=[0:1/8192:2]**. Prepare **x5** with a single line command and provide your code. Then listen to **x5**. How does the frequency of the signal change as time goes on? Find the instantaneous frequency for $x_5(t)$. What is the frequency at $t = 0$? What is the frequency at $t = 1$? What is the frequency at $t = 2$? As you see, you can make so much fantasy about sounds using the concepts you learn in the signals and systems course!

Part 4

Let $x(t) = \cos(2\pi\alpha t + \phi)$, where α is the same value as in Part 3. Take **t=[0:1/8192:1];**.

- a) Let $\phi = 0$. Compute and listen to **x**.
- b) Repeat taking $\phi = \frac{\pi}{4}$.
- c) Repeat taking $\phi = \frac{\pi}{2}$.

- d) Repeat taking $\phi = \frac{3\pi}{4}$.
- e) Repeat taking $\phi = \pi$.

How does the volume of the sound that you hear change? How does the pitch of the sound that you hear change?

Part 5

Let $x_1(t) = A_1 \cos(2\pi f_0 t + \phi_1)$ and $x_2(t) = A_2 \cos(2\pi f_0 t + \phi_2)$ where $A_1 \geq A_2 \geq 0$. Let $x_3(t) = x_1(t) + x_2(t)$. I claim that $x_3(t)$ has the form $x_3(t) = A_3 \cos(2\pi f_3 t + \phi_3)$ where $A_3 \geq 0$. Find A_3, f_3 and ϕ_3 in terms of A_1, A_2, ϕ_1, ϕ_2 and f_0 by providing full derivation. Given A_1, A_2 and f_0, find a condition on ϕ_1 and ϕ_2 such that

- a) A_3 is minimum.
- b) A_3 is maximum.

What are the maximum and minimum possible values for A_3?

Part 6 (optional, not grading, do not put your results to your report)

Write the following function:

```
function [note] = notecreate(frq_no, dur)
    note = sin(2*pi* [1:dur]/8192 * (440*2.^((frq_no-1)/12)));
end
```

Then, write the following script:

```
notename = {'A' 'A#' 'B' 'C' 'C#' 'D' 'D#' 'E' 'F' 'F#' 'G' 'G#'};
song = {'A' 'A' 'E' 'E' 'F#' 'F#' 'E' 'E' 'D' 'D' 'C#' 'C#' 'B' 'B' 'A' 'A'};

for k1 = 1:length(song)
    idx = strcmp(song(k1), notename);
    songidx(k1) = find(idx);
end

dur = 0.3*8192;
songnote = [ ];

for k1 = 1:length(songidx)
    songnote = [songnote; [notecreate(songidx(k1),dur)  zeros(1,75)]'];
end
soundsc(songnote, 8192)
```

Try to understand the code, and try to compose different songs. Try to add some different sound effects on it. Have fun!