Introduction to Matlab (Lab Section AD)

Hankoo Lee

Jeremy Liem

EE235 Continuous-Time Linear Systems

DEPARTMENT OF ELECTRICAL ENGINEERING, UNIVERSITY OF WASHINGTON, SEATTLE, WA, 98195

***Abstract* — This lab is used as an introduction to Matlab. This includes learning about variables, basic matrix operations, data visualizations, setting up, learning and plotting about sound files, and learning about how to write simple one-argument function with no return value. We look at plotting figures and loading sound files at different cases. All of this is done mathematically and graphically using Matlab.**

I. INTRODUCTION

In this 2-week lab section, we learn many important things, mainly, learning about the basic operations of Matlab, learning about using Matlab to do operations such as accessing matrix operations, getting its length/size of operations, doing mathematical operations of matrix, learnt about timing period and on graphical representation of equations over a time period. We also learnt about sound files, this involves loading sound files, playing sound files and plotting sound files. Lastly, we learnt about writing a simple one-argument function with no return value.

II. EXERCISE #1 – MATLAB VARIABLES AND BASIC OPERATIONS

In this exercise, we learnt about creating Matlab variables, suppression of output, Accessing elements from a vector/matrix, getting the size and the number of rows/columns from a vector/matrix and getting the length of a vector.

A. Results

Here is the screenshot of the Matlab code

Ex1.m

```
% FILE: Ex1.m
% NAME: Jeremy Liem
% DESCRIPTION: Matlab Variables and Basic Operatic
% Clear all variables and close all windows
clearvars;
close all;
% PART A
y1=[4;6;2];
z=[1 1 1;3 6 9;0 0 0];
% PART B
c = z(2,2)
c1= y1(2:3)
% PART C
d = z(2,3)
z(3,2:3)= -1
% PART D
e = size(y1,2)
```

Figure 1: Here is the code of Lab Exercise #1

In the figure above, the code creates the following Matlab variables

- y1 = Creates a 3x1 vector with value 4,6,2
- z = Creates a 3x3 matrix with the first row containing all 1, the second row containing values 3,6 and 9, the third row has all zeros.

There are 4 parts in this Exercise and here is the screenshot of the command output displayed in Figure 2.

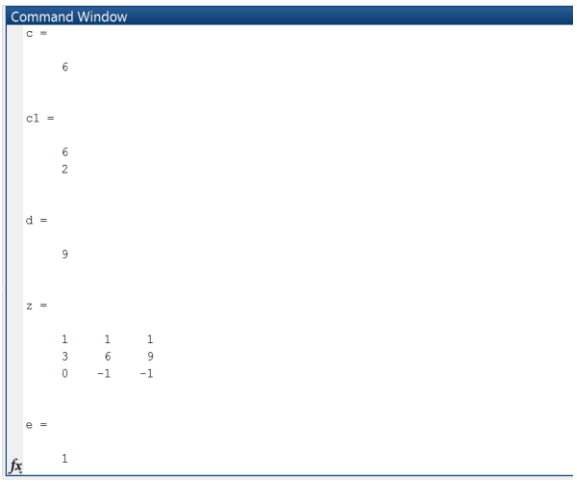

Figure 2 : Here is the output of Lab Exercise #1 in the Command Window when the Matlab Code in Figure 1 is run.

In the figure above, the command window presents

- c = which is the value 6 from vector y1.
- c1 = which stores the last 2 elements from vector y1.
- d = which stores the value 9 from matrix z.
- z = this is the matrix that was updated so that the last two columns in row 3 is changed to -1
- e = number of columns in vector y1

B. Discussion

"Not applicable for this exercise."

III. Exercise #2 – More Matlab Variables and Basic Operation

In this exercise, we learnt about the addition/subtraction of matrix, transposing a matrix, scalar multiplication of matrix, squaring elements of a matrix, creating time samples vector and accessing signals at particular time(s). All of this is done using Matlab.

A. Results

Here is the screenshot of the Matlab code

```
4   % Clear all variables and close all windows
5   clearvars;
6   close all;
7   % PART A
8   Fs=5;
9   t=-2:1/Fs:4;
10  x=2-t;
11  figure;
12  subplot(1,2,1);
13  plot(t,x);
14  xlabel('t');
15  ylabel('x(t)');
16  title('x(t)= 2-t');
17  grid on;
18  xlim([-5 5]);
19  ylim([-5 5]);
20  % PART B
21  Fs=5;
22  t=-2:1/Fs:4;
23  y=-0.5*t.^2;
24  subplot(1,2,2);
25  plot(t,y);
26  xlabel('t');
27  ylabel('y(t)');
28  title('y(t)=0.5t^2');
29  grid on;
30  xlim([-5 5]);
31  ylim([-5 5]);
32
```

Figure 3 – Here is the code for the Matlab Exercise #3

In the figure above, the code creates the following Matlab variables

- x(t) = 0.5t (Part A)
- y(t) = t2 (Part A)
- z(t) = x(t) - 2y(t) (Part B)
- Both x(t) and y(t) is defined over a range of t = 0 to t = 3 with sampling frequency of 2. (Part A)

There are 4 parts in this exercise but only 2 parts requires output to be displayed in the command window. Here is the output in Figure 4.

```
Command Window

>> Ex2

w1 =

    -7


w2 =

         0   -0.2500   -1.5000   -3.7500
```

Figure 4 – Here is the output of Lab Exercise #2 in the Command window when the Matlab code in Figure 3 is run.

In the figure above, the command window presents

- w1 = the value of z(t) at t = 2. (Part C)
- w2 = extracting values of z(t) from t = 0 to t = 1.5 . (Part D)

B. Discussion

"Not applicable for this exercise."

IV. EXERCISE #3 – MATLAB PLOTTING BASICS

In this exercise, we learnt about loading a new figure window, plotting a function, Labeling plots, Turning on the figure grid, adjusting axes limits and Plotting multiple plots using subplots. The Plotting basics learnt in this part is also used in Exercise #4.

A. Results

Here is the screenshot of the code.

```
% Clear all variables and close all windows
clearvars;
close all;
% PART A
Fs=5;
t=-2:1/Fs:4;
x=2-t;
figure;
subplot(1,2,1);
plot(t,x);
xlabel('t');
ylabel('x(t)');
title('x(t)= 2-t');
grid on;
xlim([-5 5]);
ylim([-5 5]);
% PART B
Fs=5;
t=-2:1/Fs:4;
y=-0.5*t.^2;
subplot(1,2,2);
plot(t,y);
xlabel('t');
ylabel('y(t)');
title('y(t)=0.5t^2');
grid on;
xlim([-5 5]);
ylim([-5 5]);
```

Figure 5 – This is the Matlab Code for Lab Exercise #3.

There are 2 parts of this exercise and it is displayed in a new window.

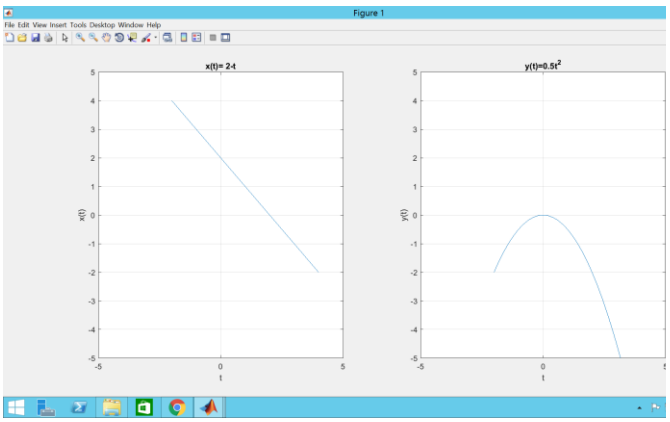

Figure 6 – Here is the output for Lab Exercise #3 when the code in Figure 5 is run.

In this part of the code, the exercise requires to plot 2 figures.

- x(t) = 2-t over range of t = -2 to t = 4 with sampling frequency of 5. This is to be plotted in the 1st subplot on 1x2 Figure Window. The x-axis range is between -5 and 5. The same range is used in the y-axis. The grid is turned on and both the plot and the axes are labeled.(Part A)

- $y(t) = -0.5t^2$ over the same range as x(t). This is to be plotted in the 2nd subplot on 1x2 Figure Window. The x-axis range is between -5 and 5. The same range is used in the y-axis. The grid is turned on and both the plot and the axes are labeled. (Part B)

B. Discussion

"Not applicable for this exercise."

V. EXERCISE #4 – SOUND FILES

In this exercise, we learnt about sound files. It involves loading a Matlab data file, Playing a sound file, plotting a sound file and computing time samples vector. It also involves learning about y and Fs. y is the audio samples vector and Fs is the sampling frequency of vector y.

A. Results

Here is the screenshot of the code.

```
% Clear all variables and close all windows
clearvars;
close all;
%PART A
load chirp.mat;
chirpSound = y ;
chirpFs = Fs ;
load gong.mat;
gongSound = y;
gongFs = Fs;
%PART B
t_chirp = (0:length(chirpSound)-1)*(1/chirpFs);
t_gong = (0:length(gongSound)-1)*(1/gongFs);
%PART C
figure;
subplot(2,1,1);
plot(t_chirp,chirpSound);
xlabel('t');
ylabel('y(t)');
title('chirp sound');
subplot(2,1,2);
plot(t_gong, gongSound);
xlabel('t');
ylabel('y(t)');
title('gong sound');
%PART D
sound(chirpSound,chirpFs);
pause(4);
sound(gongSound, gongFs);
```

Figure 7 – This is the Matlab code for Exercise #4.

There are 4 parts for this Exercise.

- In part A, we load 2 sound files, chirp. mat and gong.mat . The variable y is stored in chirpSound (for Chirp) and gongSound (for Gong). The variable Fs is store in chirpFs(for Chirp) and gongFs(for Gong).
- In part B, we compute the time samples for chip and gong and call the vector t_chirp(for chirp) and t_ gong.
- In part C, we plot the two sound signals using the sound and time samples from part a and b to 2x1 subplot window. The plot and axes are labeled properly.
- In part D, we played chirpSound and gongSound with a 4 second pause in between.

Here is the screenshot of the output when the Matlab code is run in Figure 8.

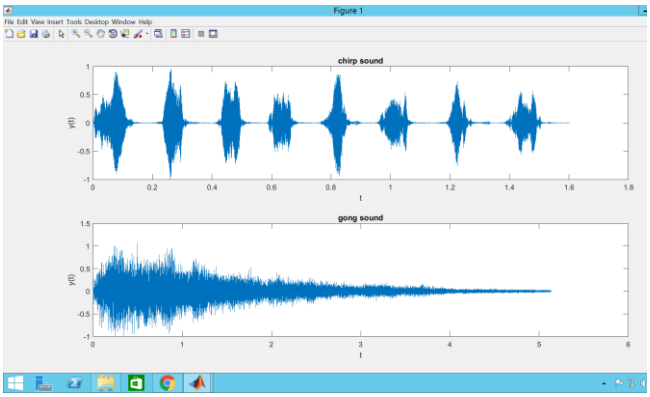

Figure 8 – This is the output when the Matlab code in Figure 7 is run.

B. Discussion

"Not applicable for this exercise."

VI. Exercise #5 – Functions

In this exercise, we learnt about Matlab functions. It can take in parameters and output return values. We learnt about writing a Matlab function and calling a Matlab function in this Exercise.

A. Results

The exercise requires us to load a train sound file and plot it with y-axes from -3 to 3.
Here is the Screenshot of the code.

```
% EXAMPSCALE Calling functions to call scale a certain sound file.
% USAGE: Play a unscaled train sound followed by scaled sound
% AUTHOR: Jeremy Liem
function ExAmpScale(A)
% PART A
load train.mat;
t_y = (0:length(y)-1)*(1/Fs);
% PART B
yScaled = A*y;
% PART C
close all;
figure;
subplot(1,2,1);
plot(t_y,y);
xlabel('t');
ylabel('y(t)');
title('Original Y sound');
ylim([-3 3]);
subplot(1,2,2);
plot(t_y, yScaled);
xlabel('t');
ylabel('y(t)');
title('Scaled Y sound');
ylim([-3 3]);
% PART D
sound(y, Fs);
pause(4);
sound(yScaled, Fs);
end
```

Figure 9 – Matlab Code for Exercise 5.

In the exercise, we are also asked to plot the 2 signal(unscaled and scaled) in a 1x2 subplot window. We are also asked to close the window as soon as another function is called. In this exercise, calling the function can be done by typing in the command window "ExAmpScale(5)".

In the exercise we are asked to run using a scale factor of 3 and 0.25. Here is the Screenshot.

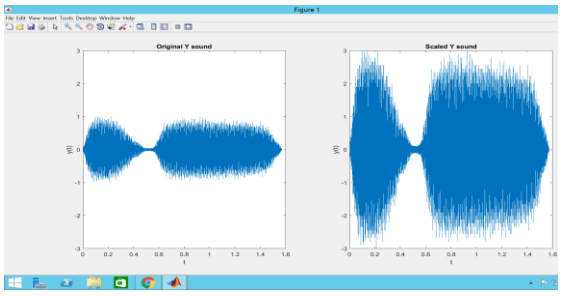

Figure 10 – ExAmpScale(3) produces this output.

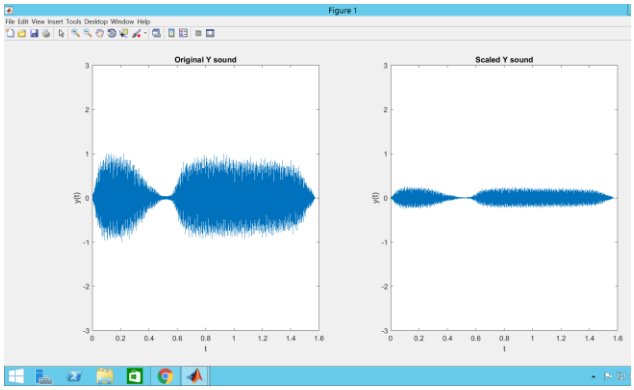

Figure 11- ExAmpScale(0.25) Produces this output.

B. *Discussion*

"Not applicable for this exercise."