RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	ELE532
Course Title	Signals and Systems I
Semester/Year	Fall 2021

Instructor	Dr. Javad Alirezaie

ASSIGNMENT No.	**1**

Assignment Title	Working with MATLAB, Visualization of Signals

Submission Date	October 1, 2021
Due Date	October 3, 2021

Student Name	Reza Aablue
Student ID	500966944
Signature*	R.A.

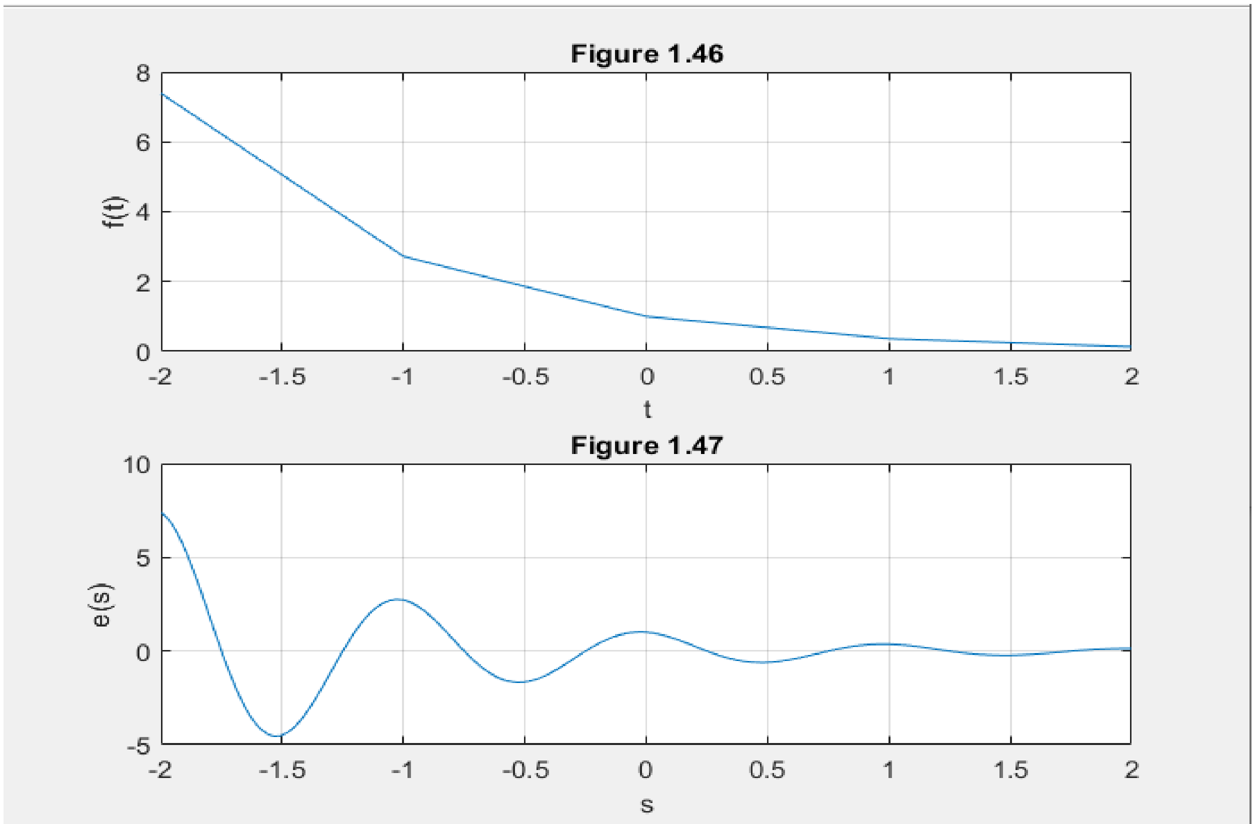
Student Name	Rendel Abrasia
Student ID	500942743
Signature*	R.A.

**By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.*

Problem A:

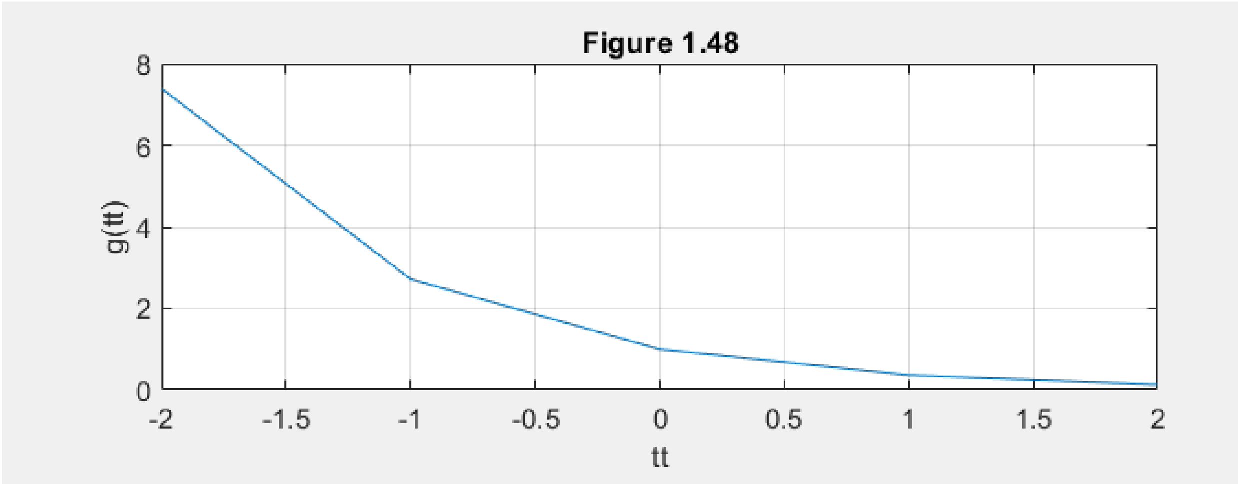
A.1

```
%Rendel Abrasia, Reza Aablue
%500942743, 500966944
%Section 4

%Problem A.1

%Setting up the values for t and s
t = (-2:2);
s = (-2:0.01:2);

%Function of f(t) and e(s)
f = @(t) exp(-t).*cos(2*pi*t);
e = @(s) exp(-s).*cos(2*pi*s);

%Allows to separate graphs into multiple windows if more than 1 figure.
figure(1);

%Setting up the graph
subplot(211);
plot(t,f(t));
xlabel('t');
ylabel('f(t)');
title('Figure 1.46');
grid;

subplot(212);
plot(s,e(s));
xlabel('s');
ylabel('e(s)');
title('Figure 1.47');
grid;
```



A.2

```
%Problem A.2
%Setting up the values for tt
tt = (-2:2);

%Function for g(tt)
g = @(tt) exp(-t);

%New Window for Problem A.2
figure(2);

subplot(211);
plot(tt,g(tt));
xlabel('tt');
ylabel('g(tt)');
title('Figure 1.48');
grid;
```



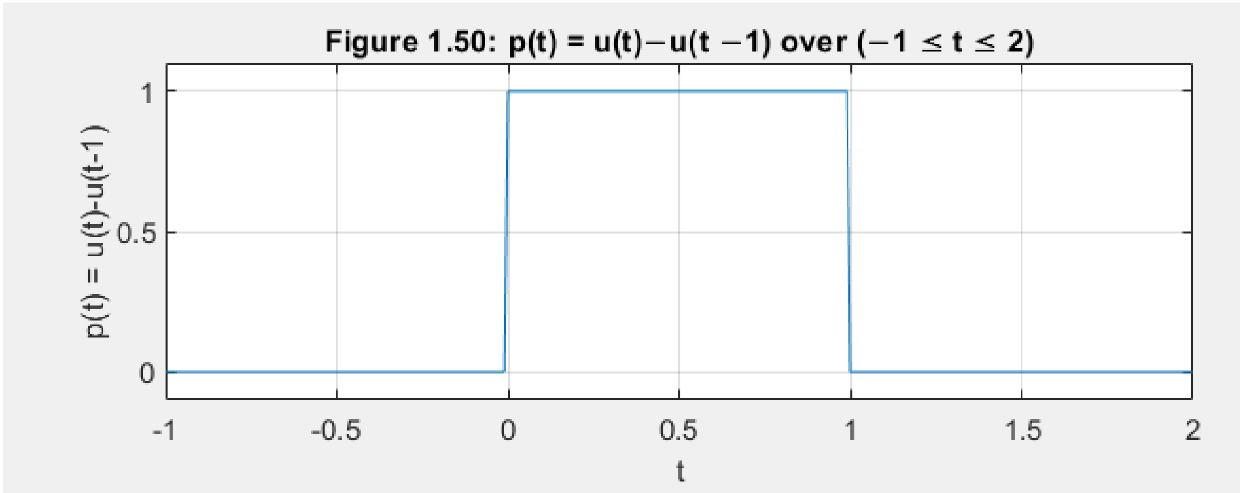
A.3

When comparing the graphs of problem A.2 with figure 1.46 of problem A.1. It is observed that the graphs are very identical.

Problem B

B.1

```
%Rendel Abrasia, Reza Aablue
%500942743,  500966944
%Section 4

%Problem B.1

%Setting up the values for t and p
t = (-1:0.01:2);
p = @(t) 1.0.*((t>=0)&(t<1));

%Allows to separate graphs into multiple windows if more than 1 figure.
figure(1);

%Setting up the graph
subplot(211);
plot(t,p(t));
xlabel('t');
ylabel('p(t) = u(t)-u(t-1)');
axis([-1 2 -.1 1.1]);
title('Figure 1.50: p(t) = u(t)-u(t -1) over (-1 ≤ t ≤ 2)');
grid;
```



B.2

```
%Problem B.2

%Functions of r(t) and n(t)
r = @(t)t.*p(t);
n = @(t)r(t) + r(-t+2);

%Second window for Problem B.2
figure(2);

%Setting up the graphs
subplot(211);
plot(t,r(t));
xlabel('t');
ylabel('r(t) = tp(t)');
title('Figure 1.51: r(t) = tp(t) over (-1 ≤ t ≤ 2) ');
axis([-1 2 -.1 1.1]);
grid;

subplot(212);
plot(t,n(t));
xlabel('t');
ylabel('n(t)= r(t) + r(-t+2)');
title('Figure 1.52: n(t) = r(t) + r(-t+2) over (-1 ≤ t ≤ 2)');
axis([-1 2 -.1 1.1]);
grid;
```

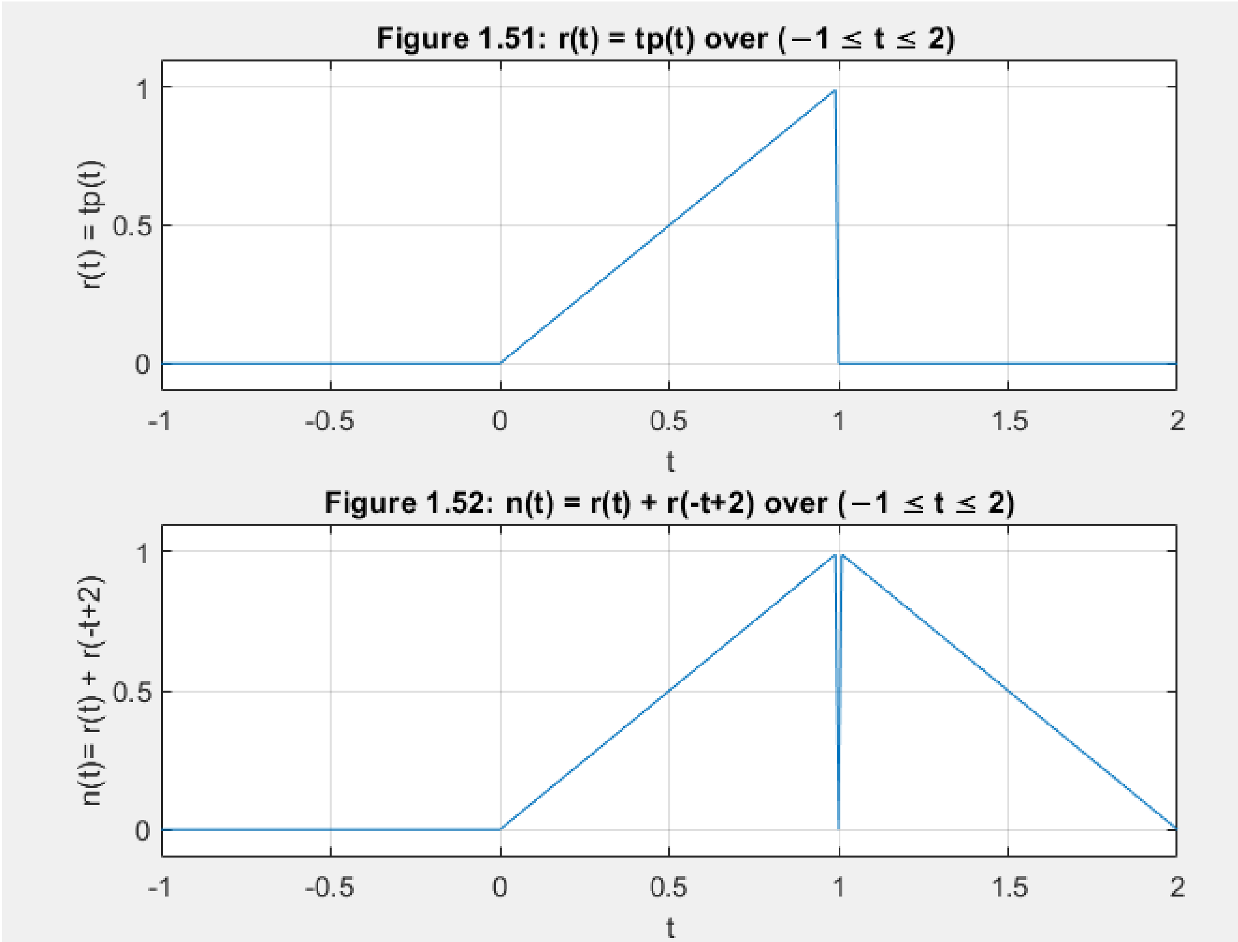
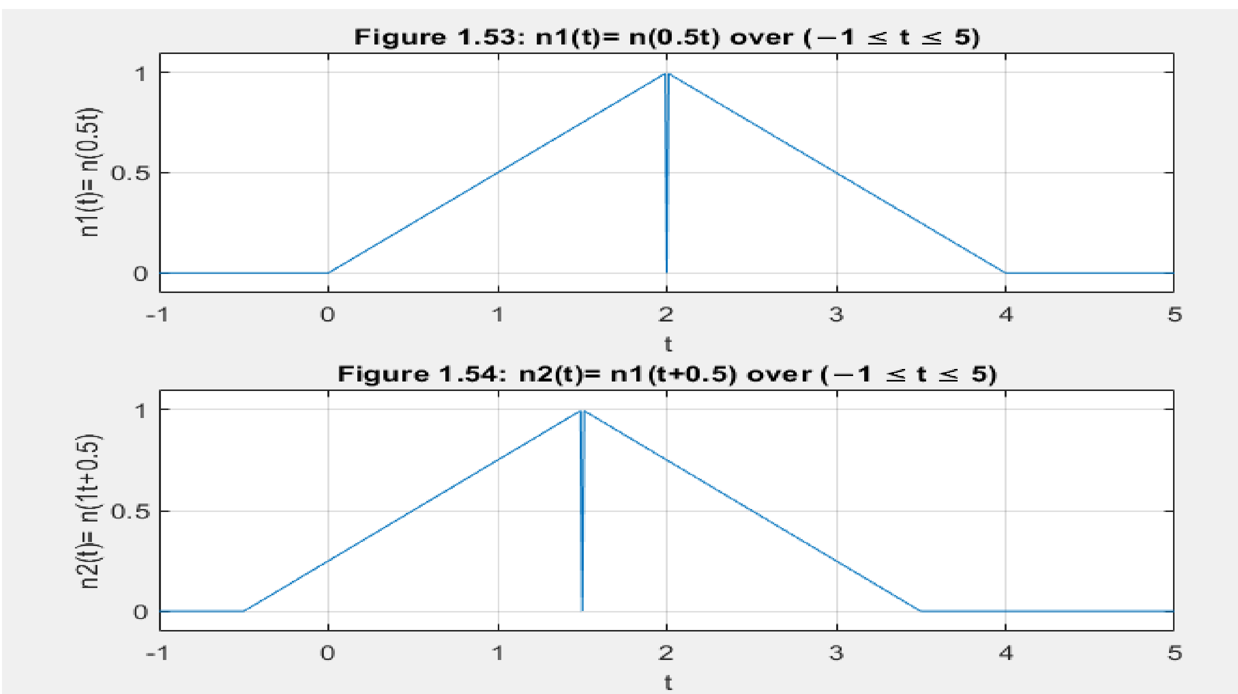


Figure 1.51: r(t) = tp(t) over (−1 ≤ t ≤ 2)
r(t) = tp(t)
1
0.5
0
-1
-0.5
0
0.5
1
1.5
2
t
Figure 1.52: n(t) = r(t) + r(-t+2) over (−1 ≤ t ≤ 2)
n(t)= r(t) + r(-t+2)
1
0.5
0
-1
-0.5
0
0.5
1
1.5
2
t

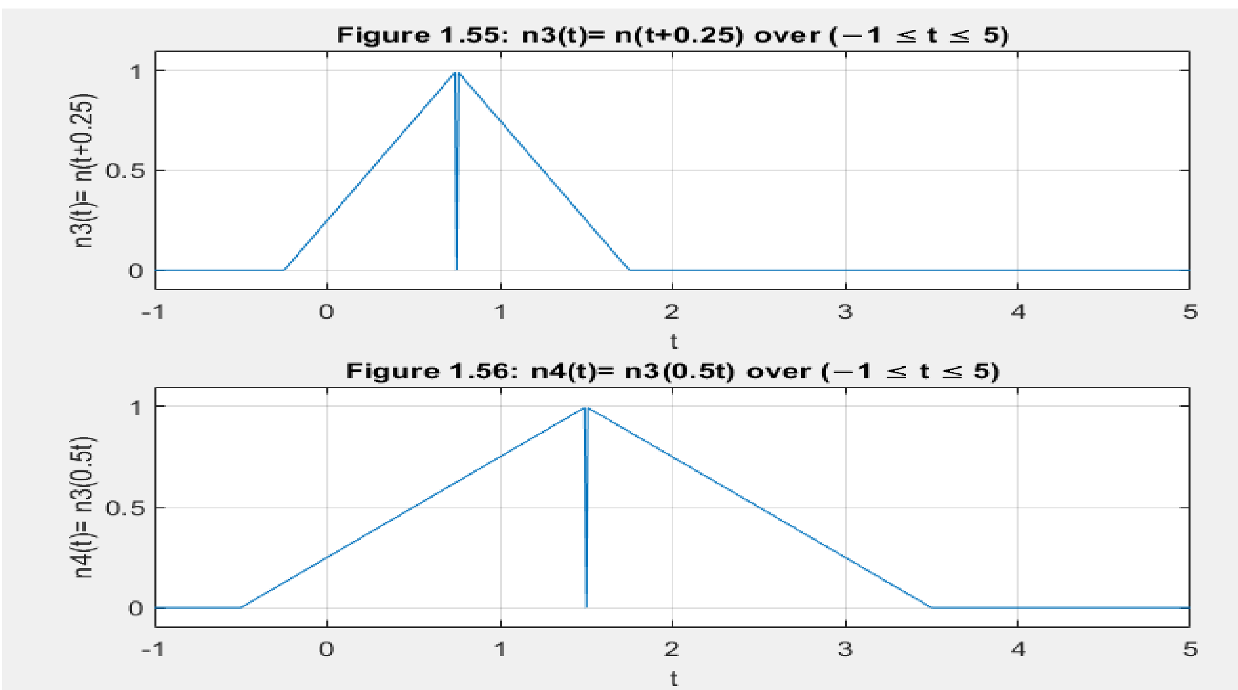
B.3

```
%Problem B.3

%A new range of values for t to satisfy the graph
t = (-1:0.01:5);

%Functions for n1(t) and n2(t)
n1 = @(t)n(0.5*t);
n2 = @(t)n1(t+0.5);

%New window for Problem B.3
figure(3)

%Setting up the graphs
subplot(211);
plot(t,n1(t));
xlabel('t');
ylabel('n1(t)= n(0.5t)');
title('Figure 1.53: n1(t)= n(0.5t) over (-1 ≤ t ≤ 5)');
axis([-1 5 -.1 1.1]);
grid;

subplot(212);
plot(t,n2(t));
xlabel('t');
ylabel('n2(t)= n(1t+0.5)');
title('Figure 1.54: n2(t)= n1(t+0.5) over (-1 ≤ t ≤ 5)');
axis([-1 5 -.1 1.1]);
grid;
```



B.4

```
%Problem B.4

%Functions for n3(t) and n4(t)
n3 = @(t)n(t+0.25);
n4 = @(t)n3(0.5*t);

%New window for Problem B.4
figure(4)

%Setting up the graphs
subplot(211);
plot(t,n3(t));
xlabel('t');
ylabel('n3(t)= n(t+0.25)');
title('Figure 1.55: n3(t)= n(t+0.25) over (-1 ≤ t ≤ 5)');
axis([-1 5 -.1 1.1]);
grid;

subplot(212);
plot(t,n4(t));
xlabel('t');
ylabel('n4(t)= n3(0.5t)');
title('Figure 1.56: n4(t)= n3(0.5t) over (-1 ≤ t ≤ 5)');
axis([-1 5 -.1 1.1]);
grid;
```



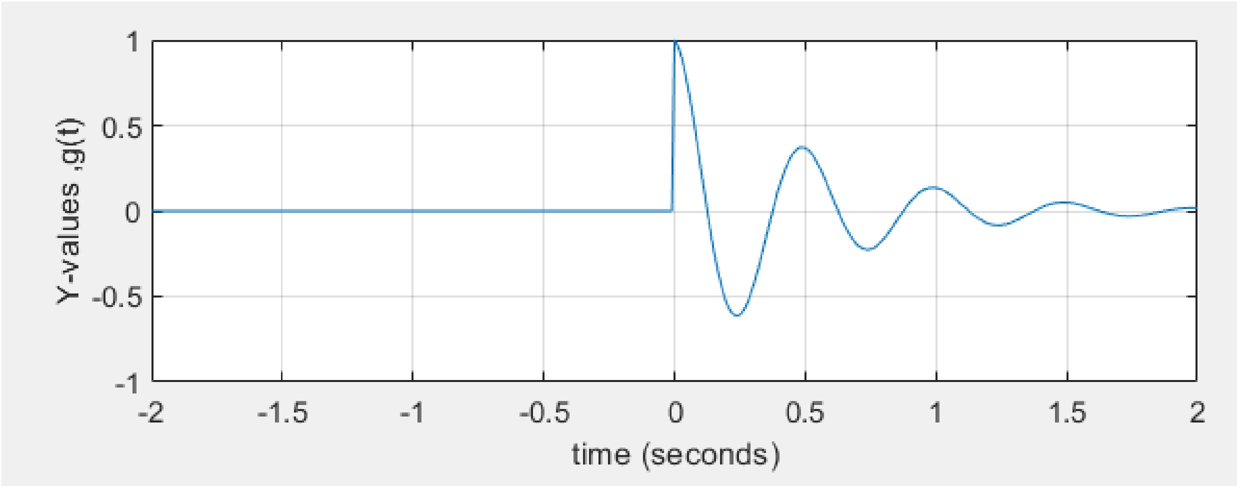
B.5

When comparing the graphs of n2(t) and n4(t) from Figures 1.54 and 1.56, It is observed that the graphs are the same. A reason why it is the same is due to the methods used in class. Two methods are learned which are doing the time shift then the time scale which is method 1 and the time scale then time shift which is method 2. For n2, we used method 1 to do the timescale first then timeshift. It is being expanded by 2 due to the timescale of 0.5. Then it is being shifted to 0.5 units to the left thus n2 starting at x-axis of [-0.5,3.5]. For n4, we used method 2 to show the graph function moving to the left by 0.25 then being expanded by 2 due to its timescale of 0.5. Therefore, both functions are the same.

Problem C

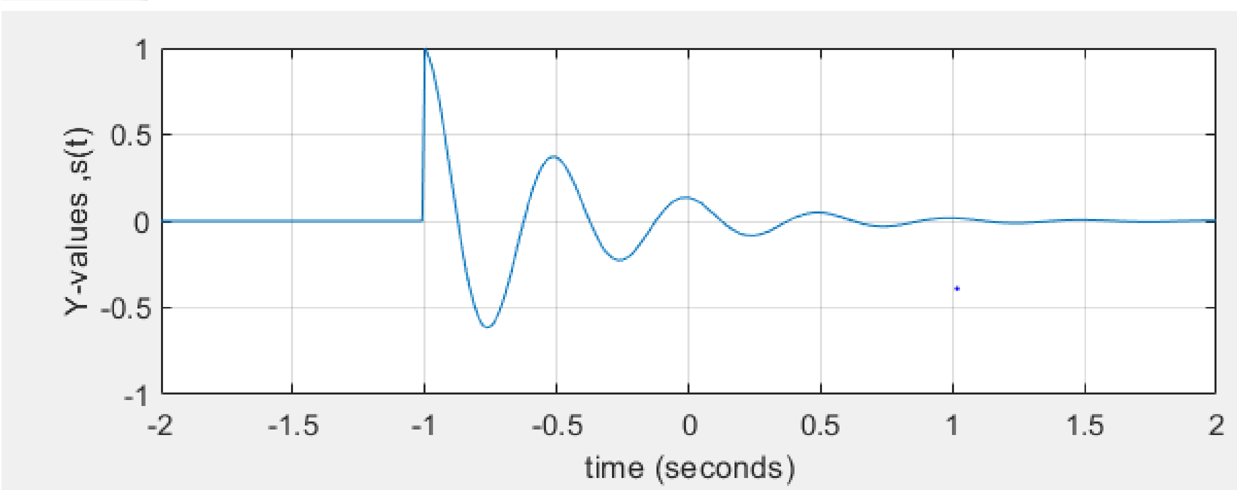
C.1

```
%Rendel Abrasia, Reza Aablue
%500942743, 500966944
%Section 4

%Problem C.1
t = (-2:0.01:2); % Time intervals and step sizes.

f = @(t) exp(-2.*t) .* cos(4*pi*t); % Create function f(t).

u = @(t) 1.0 .* (t>=0); % Create unit step function, u(t).

axis([-2 2 -1 1]); % Creating axis boundaries.



g = @(t) f(t).*u(t); % Create g(t) = f(t) * u(t).

subplot (211);

plot (t, g(t)); % Plotting g(t).

xlabel ("time (seconds)"); % Label for x-axis.
ylabel ("Y-values ,g(t)"); % Label for y-axis.

grid(); % Set up grid lines.
```



C.2

```
% Problem C.2
s = @(t) g(t+1);

subplot (212);

plot(t, s(t));

xlabel ("time (seconds)"); % Label for x-axis.

ylabel ("Y-values ,s(t)"); % Label for y-axis.

grid();
```

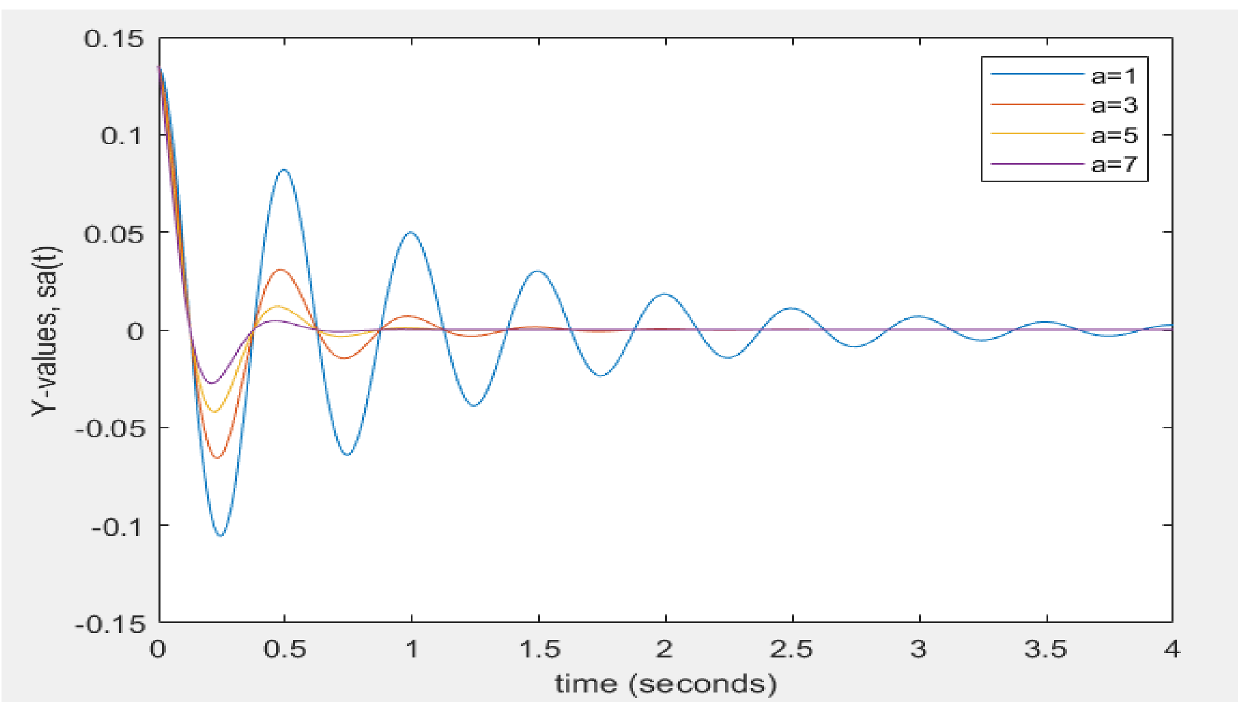


C.3

Another approach to graph sa(s) is by plotting the points for sa(s) for one alpha value at a time. It is less efficient but a more straightforward method. The value of alpha is substituted into the function 4 times separately and 4 different plot() commands are used.

```
% Problem C.3

matrix = zeros (401, 4); % Create matrix s(t).
s = (0:0.01:4); % Time intervals and step sizes.

figure (2);

for a = 1:2:7  % using alpha values 1, 3, 5 and 7.
    sa = @(t) exp (-2) .* exp(-a.*t) .* cos (4*pi*t) .* u(t); % Create function sa(t) for each alpha value.
    plot (s, sa(s));
    xlabel ("time (seconds)"); % Label for x-axis.
    ylabel ("Y-values, sa(t)"); % Label for y-axis.

    hold on; % Retains current plot and axis properties.

end

legend ('a=1', 'a=3', 'a=5', 'a=7'); % Legend for each value of alpha.

hold off; % Reset axis properties to default.
```

```
% Another approach to graph sa(s) is by plotting the points for sa(s) for
% one alpha value at a time. It is less efficient but a more
% straightforward method. The value of alpha is subsituted into the
% function 4 times separately and 4 different plot() commands are used.
```



C.4

The size of the matrix s(t) in problem C.3 is 401 x 4 => 1604. Matrix s(t) has 1604 elements.

Problem D

D.1

```
%Rendel Abrasia, Reza Aablue
%500942743, 500966944
%Section 4

% Problem D.1

% a) A(:) lists all elements vertically, moving from leftmost column to the
% last column. It moves down each column and prints out every value until
% it reaches the end of the column and moves to the top of the next column.

% b) A([2 4 7]) creates a 1x3 (rows x columns) cell array with the first
% elements in each column. Elements from (2,1), (4,1) and (2,2) are printed
% out as a result of this operation.

% c) [A >= 0.2] creates a 5x4 logical array that checks each elements to
% see if it's greater than or equal to 0.2. Digit 1 is printed for the
% element if condition is true, 0 otherwise. The result is shown below:
% 1   0   0   0
% 1   0   1   0
% 0   1   1   1
% 1   1   0   1
% 1   1   1   1

% d) A([A >= 0.2]) prints out all values that are greater that or equal to
% 0.2 in order of elements 1 to 25. There are 13 elements that are printed
% out.

% e) A([A >= 0.2]) = 0 changes all elements greater than or equal to 0.2 to a
% value of zero.
```

D.2

```
% Problem D.2a
load ('ELE532_Lab1_Data.mat'); % Load data from the provided file for this problem.

rows = size (B,1); % Row size of matrix B with respect to 1 dimension.
columns = size (B,2); % Column size of matrix B with respect to 2 dimensions.

for i=1:1:rows % 2 nested for-loops for setting all elements with magnitude
               % below 0.01 to zero.
    for j=1:1:columns
        if (abs(B(i,j)) < 0.01)
            B(i,j) = 0;
        end
    end
end

% Problem D.2b
load ('ELE532_Lab1_Data.mat'); % Load data from the provided file for this problem.
B([abs(B) < 0.01])=0 % Set all elements less than 0.01 to zero.
```

```
% Problem D.2c - part 1 (Elapsed time is 0.008202 seconds.)
tic
load ('ELE532_Lab1_Data.mat');

rows = size (B,1); % Row size of matrix B with respect to 1 dimension.
columns = size (B,2); % Column size of matrix B with respect to 2 dimensions.

for i=1:1:rows % 2 nested for-loops for setting all elements with magnitude
               % below 0.01 to zero.
    for j=1:1:columns
        if (abs(B(i,j)) < 0.01)
            B(i,j) = 0;
        end
    end
end

fprintf ("\nExecution time for code in Problem D.2a: ");
toc

% Problem D.2c - part 2 (Elapsed time is 0.165708 seconds.)
tic

load ('ELE532_Lab1_Data.mat'); % Load data from the provided file for this problem.
B([abs(B) < 0.01])=0 % Set all elements less than 0.01 to zero.

fprintf ("\nExecution time for code in Problem D.2b: ");
toc
```

D.3

An alternative is to use the nnz() function and then subtract the number

% of zeros from the total number of elements in the data array.

```
% Problem D.3
load ('ELE532_Lab1_Data.mat'); % Load data from the provided file for this problem.

audio_array = x_audio; % Copy x_audio array into a new array.

row = size (audio_array,1); % Row size of matrix B with respect to 1 dimension.
column = size (audio_array,2); % Column size of matrix B with respect to 2 dimensions.

num_of_zeros=0;

% An alternative is to use the nnz() function and then subtract the number
% of zeros from the total number of elements in the data array.

for i=1:row % 2 nested for-loops for finding the number of zeros in the data array.
    for j=1:column
        if (abs(audio_array(i,j))==0)
            num_of_zeros=num_of_zeros+1;
        end
    end
end
```