3TP3: Signals and Systems
Lab #2
2nd November 2020

Instructor: Dr. Jun Chen

Vishavjeet Singh - singhv30 - 400219273
Humza Ahmad - ahmadh36 - 400203417

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by:

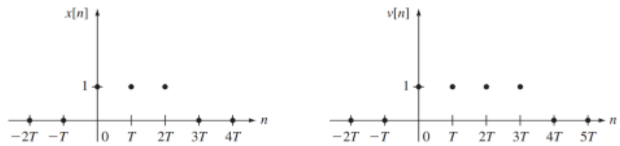
[Vishavjeet Singh, singhv30, 400219273]

[Humza Ahmad - ahmadh36 – 400203417]

<u>PART 2</u>: Experiments

Question 1: (Q2.7 from textbook)

a)



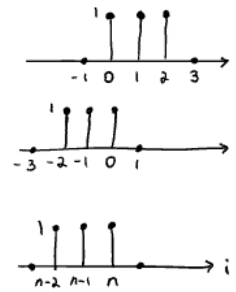
(a)

$x[n] = \begin{cases} 1 & n = 0,1,2 \\ 0 & \text{otherwise} \end{cases}$ $\qquad v[n] = \begin{cases} 1 & n = 0,1,2,3 \\ 0 & \text{otherwise} \end{cases}$

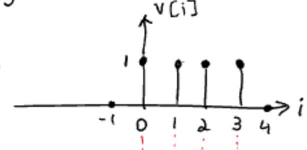
→ Shorter duration signal

$y[n] = x[n] * v[n] = v[n] * x[n] = \sum_{i=0}^{n} v[i]\, x[n-i]$

$x[i]$ ↓ $x[-i]$ ↓ $x[n-i]$



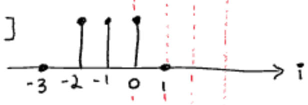
V[i]



✫ $V[n] * x[n]$ when $n < 0$, is zero (no overlap)

n = 0

x[0-i]



→ $y[0] = v[0] \times x[0] = 1 \times 1 = 1$

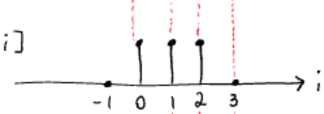
n = 1

x[1-i]

→ $y[1] = 1 \times 1 + 1 \times 1 = 2$

n = 2

x[2-i]



→ $y[2] = 1 \times 1 + 1 \times 1 + 1 \times 1 = 3$

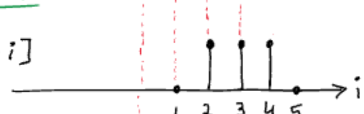
n = 3

x[3-i]

→ $y[3] = 1 \times 1 + 1 \times 1 + 1 \times 1 = 3$

n = 4

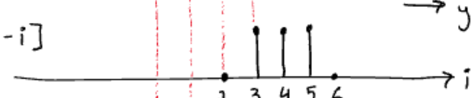
x[4-i]



→ $y[4] = 1 \times 1 + 1 \times 1 = 2$

n = 5

x[5-i]

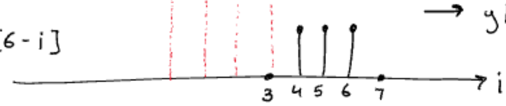


→ $y[5] = 1 \times 1 = 1$

n	$y[n]$
$n \le -1$	0
0	1
1	2
2	3
3	3
4	2
5	1
$n \geq 6$	0

n = 6

x[6-i]

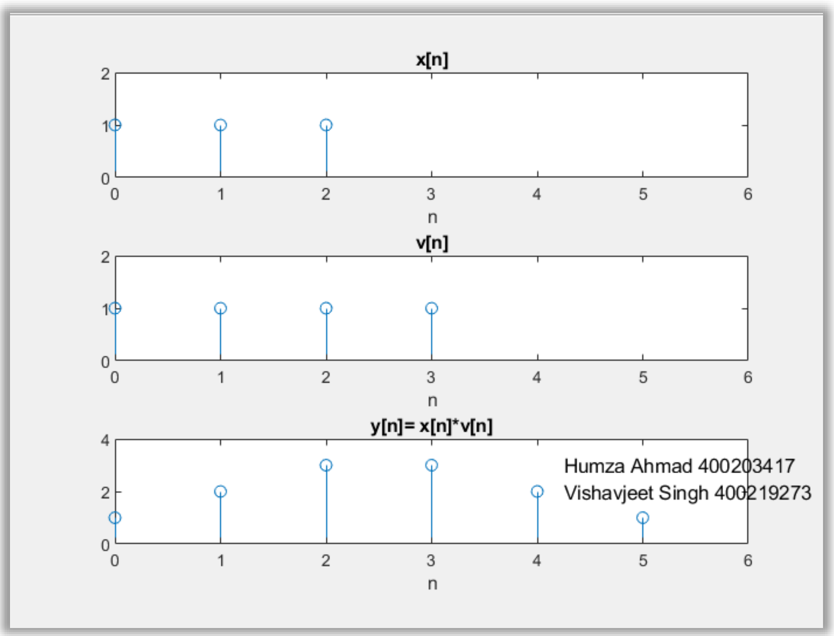


→ $y[6] = 1 \times 0 = 0$

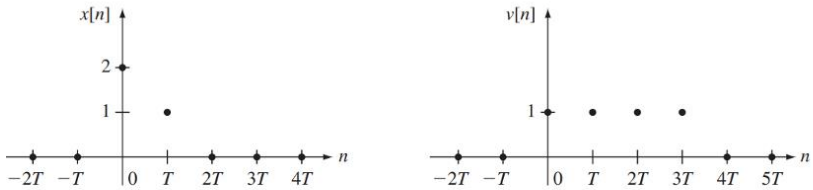
✫ $V[n] * x[n]$ when $n > 6$, is zero (no overlap)

```
>> x=[1 1 1]; %input x[n]
n_x=0:length(x)-1; %indecies for x[n]
v=[1 1 1 1]; %input v[n]
n_v= 0:length(v)-1; %indecies for v[n]

y = conv(x,v); %output y[n] = x[n]*v[n]
n_o= 0:length(y)-1; %indecies for y[n]

subplot(3,1,1);
stem(n_x,x); %plotting x[n]
title("x[n]");
xlabel('n');
xlim([0,6]); %adjust axis for better view
ylim([0,2]);

subplot(3,1,2);
stem(n_v,v); %plotting v[n]
title("v[n]");
xlabel('n');
xlim([0,6]);  %adjust axis for better view
ylim([0,2]);

subplot(3,1,3);
stem(n_o,y); %plotting y[n]
title("y[n]= x[n]*v[n]");
xlabel('n');
xlim([0,6]);  %adjust axis for better view
ylim([0,4]);
text(4.25,3,'Humza Ahmad 400203417');
text(4.25,2,'Vishavjeet Singh 400219273');
```

Each input was first initialized as a vector. Then the ‘conv’ function was called to convolute the two input signals. The ‘subplot’ command was called to break the output window into three separate graphs, for which the two input signals and the resultant convoluted signal, y[n], were then plotted using the ‘stem’ function on the respective graph.



b)



(b)

$$x[n] = \begin{cases} 2 & n=0 \\ 1 & n=1 \\ 0 & \text{otherwise} \end{cases} \qquad v[n] = \begin{cases} 1 & n=0,1,2,3 \\ 0 & \text{otherwise} \end{cases}$$

$$y[n] = x[n] * v[n] = \sum_{i=0}^{\infty} x[i]\, v[n-i] = x[0]\, v[n] + x[1]\, v[n-1]$$
$$= 2\, v[n] + v[n-1]$$

$\underline{n=0}$ $\quad y[0] = 2\, v[0] + v[0-1]$
$= 2(1) + 0$
$= 2$

$\underline{n=1}$ $\quad y[1] = 2\, v[1] + v[0]$
$= 2(1) + 1$
$= 3$

$\underline{n=2}$ $\quad y[2] = 2\, v[2] + v[1]$
$= 2(1) + 1$
$= 3$

$\underline{n=3}$ $\quad y[3] = 2\, v[3] + v[2]$
$= 2(1) + 1$
$= 3$

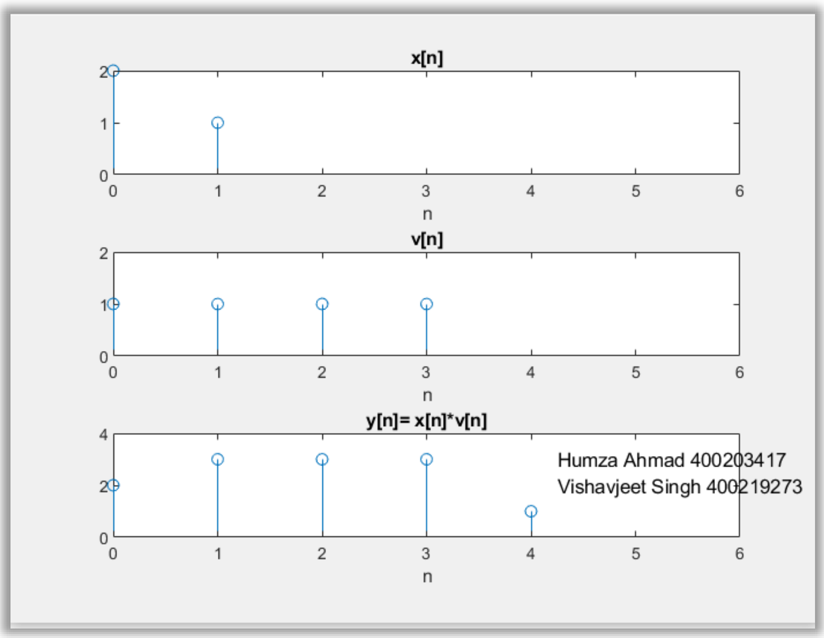
$\underline{n=4}$ $\quad y[4] = 2\, v[4] + v[3]$
$= 2(0) + 1$
$= 1$

$\underline{n=5}$ $\quad y[5] = 2\, v[5] + v[4]$
$= 2(0) + 0$
$= 0$

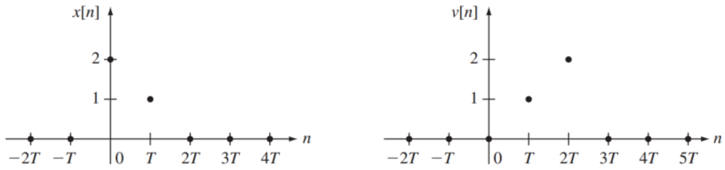
n	$y[n]$
$n \le -1$	0
0	2
1	3
2	3
3	3
4	1
$n \ge 5$	0

```
x=[2 1]; %input x[n]
n_x=0:length(x)-1; %indecies for x[n]
v=[1 1 1 1]; %input v[n]
n_v= 0:length(v)-1; %indecies for v[n]

y = conv(x,v); %output y[n] = x[n]*v[n]
n_o= 0:length(y)-1; %indecies for y[n]

subplot(3,1,1);
stem(n_x,x); %plotting x[n]
title("x[n]");
xlabel('n');
xlim([0,6]); %adjust axis for better view
ylim([0,2]);

subplot(3,1,2);
stem(n_v,v); %plotting v[n]
title("v[n]");
xlabel('n');
xlim([0,6]);  %adjust axis for better view
ylim([0,2]);

subplot(3,1,3);
stem(n_o,y); %plotting y[n]
title("y[n]= x[n]*v[n]");
xlabel('n');
xlim([0,6]);  %adjust axis for better view
ylim([0,4]);
text(4.25,3,'Humza Ahmad 400203417');
text(4.25,2,'Vishavjeet Singh 400219273');
```

In this question, the exact same process was implemented as in part (a). Only the values of the input signals, x[n] and v[n], were altered.



c)



(c)

$$x[n] = \begin{cases} 2 & n=0 \\ 1 & n=1 \\ 0 & \text{otherwise} \end{cases} \qquad v[n] = \begin{cases} 1 & n=1 \\ 2 & n=2 \\ 0 & \text{otherwise} \end{cases}$$

$$y[n] = x[n] * v[n] = \sum_{i=0}^{n} x[n]\,v[n-i] = x[0]\,v[n] + x[1]\,v[n-1]$$
$$= 2v[n] + v[n-1]$$

$\underline{n=0}$
$$y[0] = 2v[0] + v[-1] = 2(0) + 0 = 0$$

$\underline{n=1}$
$$y[1] = 2v[1] + v[0] = 2(1) + 0 = 2$$

$\underline{n=2}$
$$y[2] = 2v[2] + v[1] = 2(2) + 1 = 5$$

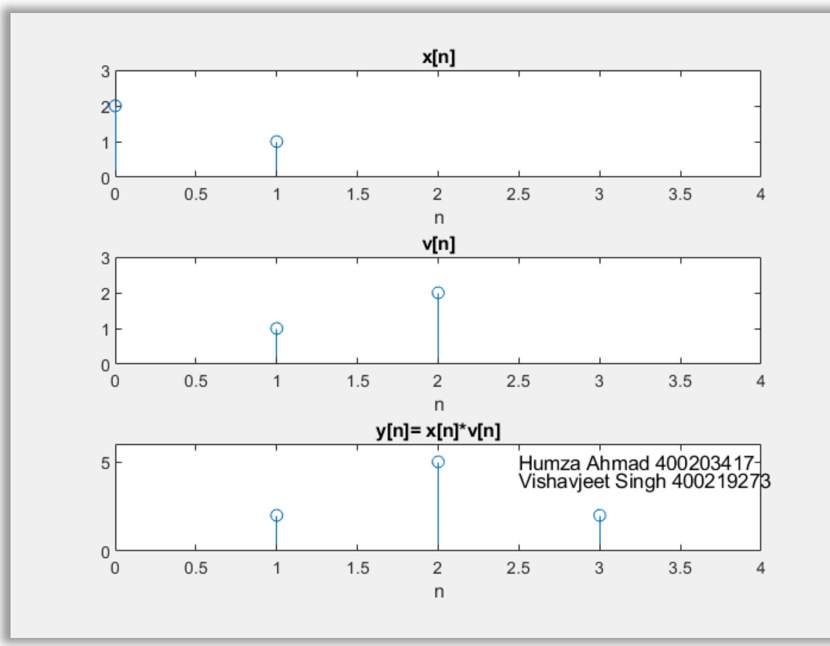
$\underline{n=3}$
$$y[3] = 2v[3] + v[2] = 2(0) + 2 = 2$$

$\underline{n=4}$
$$y[4] = 2v[4] + v[3] = 2(0) + 0 = 0$$

n	$y[n]$
$n \leq 0$	0
1	2
2	5
3	2
$n \geq 4$	0

```
x=[2 1]; %input x[n]
n_x=0:length(x)-1; %indecies for x[n]
v=[1 2]; %input v[n]

y = conv(x,v); %output y[n] = x[n]*v[n]

subplot(3,1,1);
stem(n_x,x); %plotting x[n]
title("x[n]");
xlabel('n');
xlim([0,4]); %adjust axis for better view
ylim([0,3]);

subplot(3,1,2);
stem(v); %plotting v[n], since the first value where v[n]>0 occurs at n=1, no need for unique index
title("v[n]");
xlabel('n');
xlim([0,4]);  %adjust axis for better view
ylim([0,3]);

subplot(3,1,3);
stem(y); %plotting y[n], since the first value where y[n]>0 occurs at n=1, no need for unique index
title("y[n]= x[n]*v[n]");
xlabel('n');
xlim([0,4]);  %adjust axis for better view
ylim([0,6]);
text(2.5,5,'Humza Ahmad 400203417');
text(2.5,4,'Vishavjeet Singh 400219273');
```

In this question, the exact same process was implemented as in part (a). Only the values of the input signals and axis limits were altered. Additionally, since the MATLAB ‘stem’ function starts plotting from 1 on the horizontal axis by default, there was no need to include distinct indices for v[n] and y[n].



Question 4:

In this part, the signal with echo is created by inserting zeroes equal to the desired delay in the beginning of the signal vector. This signal is then truncated for vector addition with the original signal. Alpha value picked close to 1 so as to easily hear the echo. This final signal with echo is then saved to the disk.

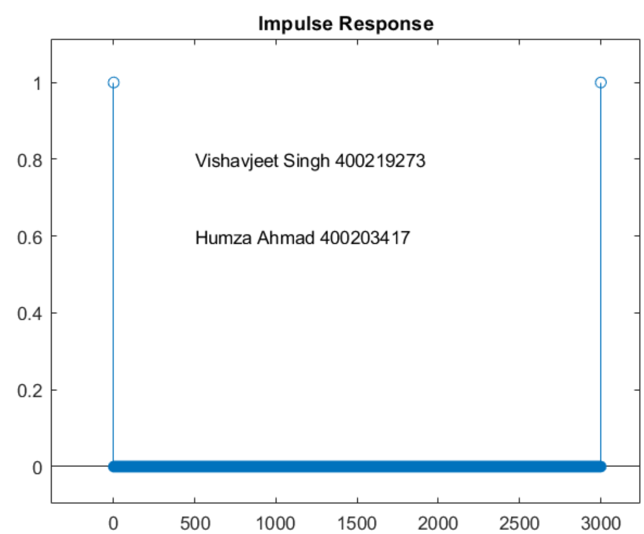
$$f_r(t) = f_s(t) + \alpha f_s(t - t_e)$$

$$t_e = 3000ms \;\&\; \alpha = 0.9$$

```
%%%%%% 3TP3 - Signals and Systems - LAB 2 - Signal Convolution %%%%%%
%%%%%% Vishavjeet Singh 400219273 %%%%%% Humza Ahmad 400203417 %%%%%%
clc; clear;

[signal, Fs] = audioread('speech.wav');
L = length(signal); % Number of samples in the signal.
T = 1/Fs; % Sampling period in seconds
t = [0:L-1]*T; % Time vector in seconds

Te = 3000; % Echo delay in msec
alpha = 0.9; % Amplitude factor by which the original signal is reduced
signal_delayed = [zeros(Te,1);signal]; % Add zeroes to the start of signal vector for the delay
signal_delayed = signal_delayed(1:L,1); % Truncate the array for vector addition
signalplusecho = signal + alpha.*signal_delayed; % Create signal with echo
signalplusecho = signalplusecho/max(abs(signalplusecho)); % Re-scaling
audiowrite('speechwithecho.wav', signalplusecho, Fs); % Saving to disk
```

Question 5:

Impulse function chosen has a maximum magnitude of the given signal at approximately t = 1ms which is almost zero as well as for the variable delay. Length of this impulse signal is equal to the length of the original signal. Thus, Impulse Response at t = 1ms and t = delay (ms) is equal to 1. When this Impulse Response is convoluted with the signal, the output is the signal with echo.



```
%%%%%% 3TP3 - Signals and Systems - LAB 2 - Signal Convolution %%%%%%
%%%%%% Vishavjeet Singh 400219273 %%%%%% Humza Ahmad 400203417 %%%%%%
clc; clear;

[signal, Fs] = audioread('speech.wav');
L = length(signal); % Number of samples in the signal.
T = 1/Fs; % Sampling period in seconds
t = [0:L-1]*T; % Time vector in seconds

delay = 3000; % Delay in msec
IR_base = zeros(L,1); % Base Matrix spanning the length of signal
IR(1) = 1; % Setting up magnitude to max at the beginning
IR(delay) = 1; % Setting up delay in the impulse signal
signalplusecho = conv(signal,IR); % Convulate signal with impulse
signalplusecho = signalplusecho/max(abs(signalplusecho)); % Re-scaling
audiowrite('speechwithecho_conv.wav', signalplusecho, Fs); % Saving to disk
```

Question 6:

The group experimented with multiple values for the time delay (Te). The signal starts to become acceptable beneath the 1000ms mark but 750ms was the chosen minimum delay for the signal to still be acceptable with a little to no echo. This chosen delay is subjective and can slightly deviate from this chosen value.

```
%%%%%% 3TP3 - Signals and Systems - LAB 2 - Signal Convolution %%%%%%
%%%%%% Vishavjeet Singh 400219273 %%%%%% Humza Ahmad 400203417 %%%%%%
clc; clear;

[signal, Fs] = audioread('speech.wav');
L = length(signal); % Number of samples in the signal.
T = 1/Fs; % Sampling period in seconds
t = [0:L-1]*T; % Time vector in seconds

Te = 750; % Echo delay in msec
alpha = 0.9; % Amplitude factor by which the original signal is reduced
signal_delayed = [zeros(Te,1);signal]; % Add zeroes to the start of signal vector for the delay
signal_delayed = signal_delayed(1:L,1); % Truncate the array for vector addition
signalplusecho = signal + alpha.*signal_delayed; % Create signal with echo
signalplusecho = signalplusecho/max(abs(signalplusecho)); % Re-scaling
audiowrite('speechwithecho_smallTe.wav', signalplusecho, Fs); % Saving to disk
```

As α is the amplitude factor so deceasing it allows us to further decrease the time delay. When $\alpha = 0.2$, the signal was still acceptable when Te = 2000ms. So decreasing the amplitude factor allows us to increase the time delay while still keeping the signal acceptable.

Question 7:

For the “reverberation” effect we used a for loop with a time delay of 2000ms and an alpha value of 0.98 so as the reverb effect is audible. We decided on 4 echoes and thus, the for loop iterates 4 times governed by the equation.

$$\sum_{i=1}^{N_e} \alpha^i f_s(t - iT_e)$$

```
%%%%%% 3TP3 - Signals and Systems - LAB 2 - Signal Convolution %%%%%%
%%%%%% Vishavjeet Singh 400219273 %%%%%% Humza Ahmad 400203417 %%%%%%
clc; clear;

[signal, Fs] = audioread('speech.wav');
L = length(signal); % Number of samples in the signal.
T = 1/Fs; % Sampling period in seconds
t = [0:L-1]*T; % Time vector in seconds

Te = 2000; % Echo delay in msec
alpha = 0.98; % Amplitude factor by which the original signal is reduced
Ne = 4; % Number of echoes
for i = 1 : Ne % For loop iterates over the number of echoes wanted
    signal_delayed = [zeros(i.*Te,1);signal]; % Add zeroes to the start of signal vector for the delay
    signal_delayed = signal_delayed(1:L,1); % Truncate the array for vector addition
    signalplusecho = signal + alpha^i.*signal_delayed; % Create signal with echo
end
signalplusecho = signalplusecho/max(abs(signalplusecho)); % Re-scaling
audiowrite('speechwithecho_reverberation.wav', signalplusecho, Fs); % Saving to disk
```

Same format is used as Q4 for creating the progressive time delay with adding zeroes and truncating the signal vector. The reverb effect can be heard in the output file.

Keeping alpha value at 1, when we set Te value so that the signal is acceptable. The signal starts to become acceptable beneath the 500ms mark but 450ms was the chosen minimum delay for the signal to still be acceptable with a little to no echo. This chosen delay is subjective and can slightly deviate from this chosen value. When decreasing the alpha value we can increase the time delay while still keeping the signal acceptable. For example for alpha = 0.7, we can set Te = 800ms with the signal being acceptable.

```
%%%%%% 3TP3 - Signals and Systems - LAB 2 - Signal Convolution %%%%%%
%%%%%% Vishavjeet Singh 400219273 %%%%%% Humza Ahmad 400203417 %%%%%%
clc; clear;

[signal, Fs] = audioread('speech.wav');
L = length(signal); % Number of samples in the signal.
T = 1/Fs; % Sampling period in seconds
t = [0:L-1]*T; % Time vector in seconds

Te = 800; % Echo delay in msec
alpha = 0.7; % Amplitude factor by which the original signal is reduced
Ne = 4; % Number of echoes
for i = 1 : Ne % For loop iterates over the number of echoes wanted
    signal_delayed = [zeros(i.*Te,1);signal]; % Add zeroes to the start of signal vector for the delay
    signal_delayed = signal_delayed(1:L,1); % Truncate the array for vector addition
    signalplusecho = signal + alpha^i.*signal_delayed; % Create signal with echo
end
signalplusecho = signalplusecho/max(abs(signalplusecho)); % Re-scaling
audiowrite('speechwithecho_reverberation_smallTe.wav', signalplusecho, Fs); % Saving to disk
```