3TP3: Signals & Systems
Lab #2

Instructor: Dr. Kirubarajan

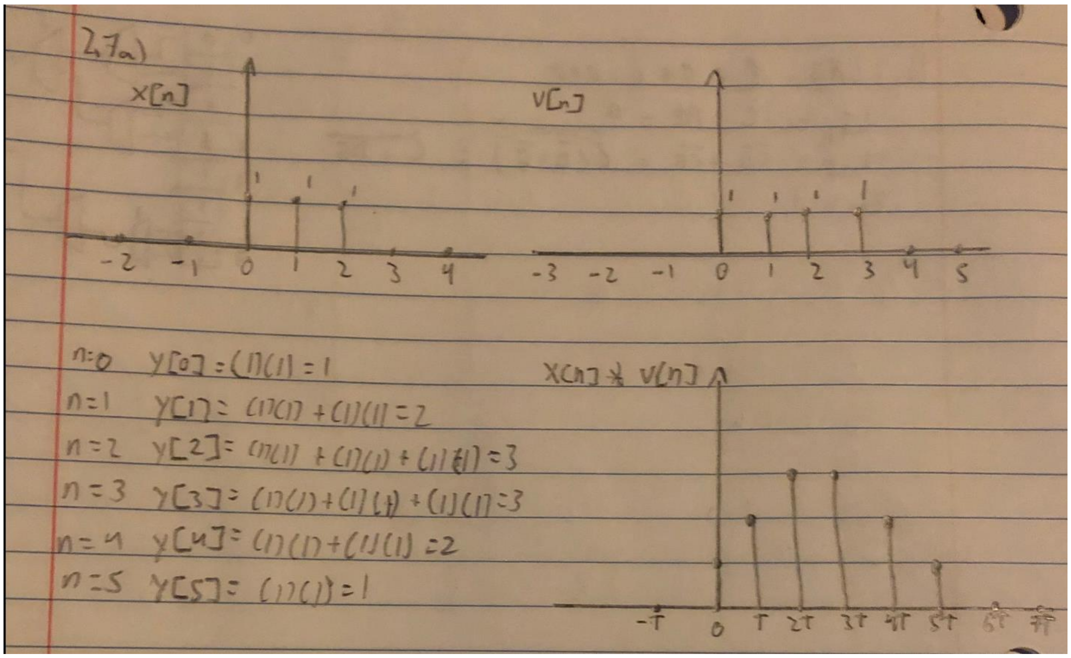
Tamer Rafidi – L02 – rafidit- 400333527
Samer Rafidi – L02 – rafidis- 400333524

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 **[Tamer Rafidi, rafidit, 400333527]**

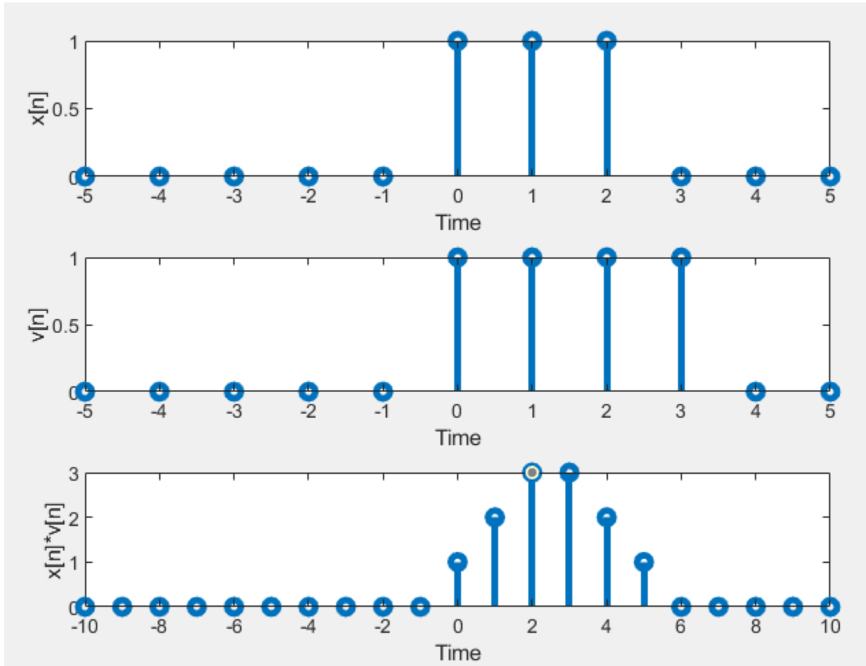
As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Samer Rafidi, rafidis, 400333524]**

Question 1

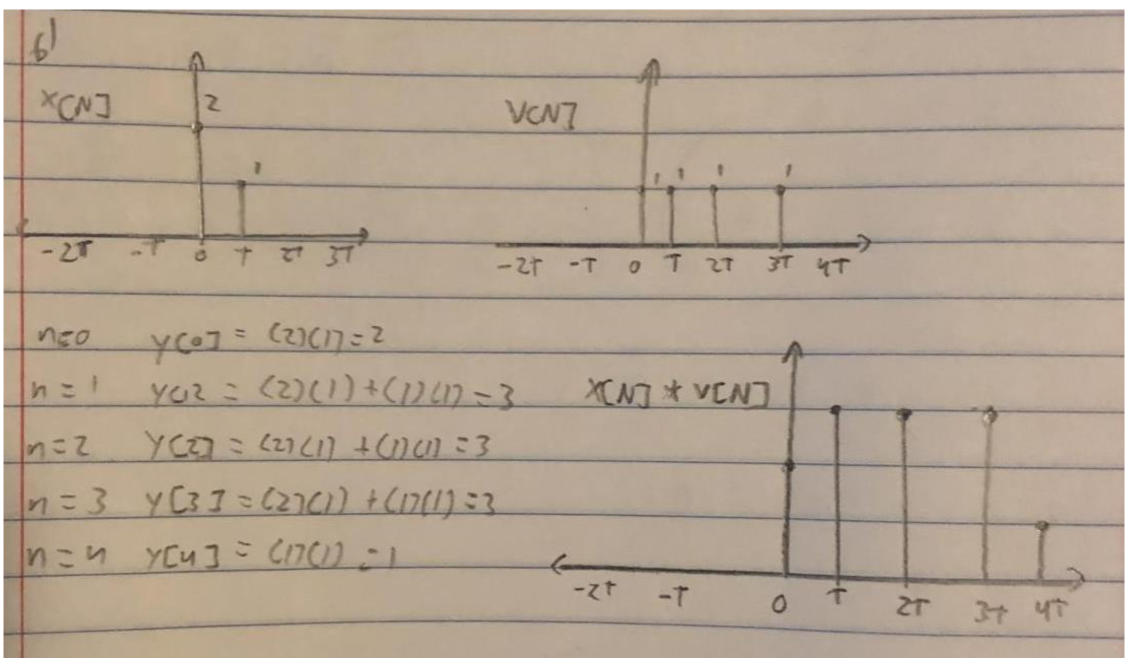
1a)



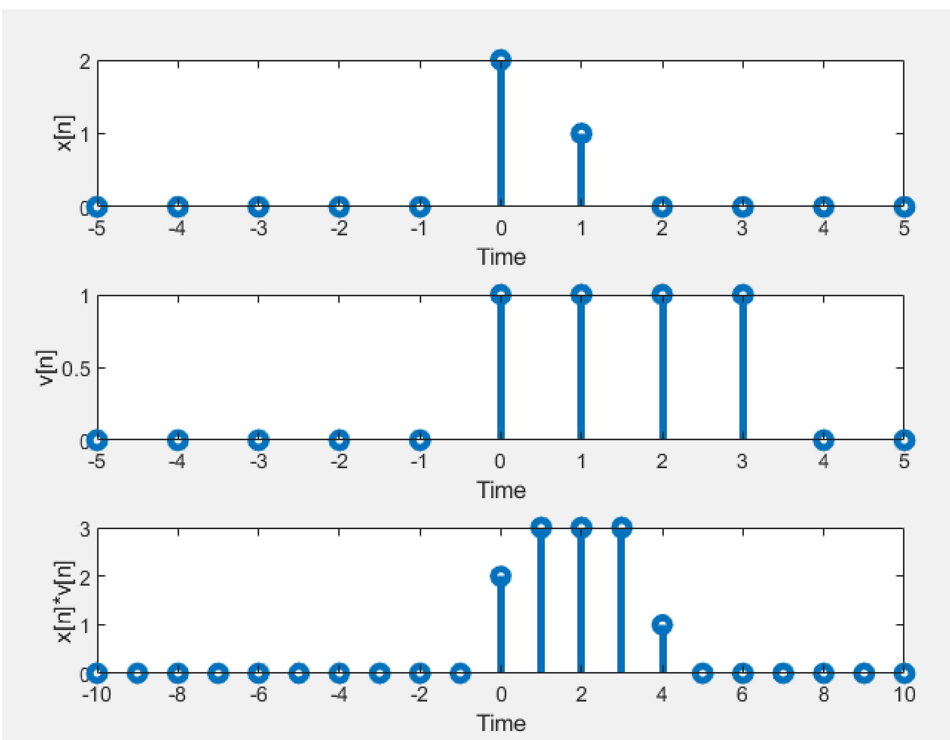

```
%Set time ranges
t = -5:5;
t_2 = -10:10;

%Set two variables to SimpleFunctions() object
x = SimpleFunctions();
v = SimpleFunctions();

%Created x[n] and v[n]
x = x.unitstep(t)-x.unitstep(t-3);
v = v.unitstep(t)-v.unitstep(t-4);

%Computing convolution of x[n] and v[n]
con = conv(x, v);

%Creating three subplots x[n], v[n], and x[n]*v[n]
subplot(3,1,1);
stem(t, x, "LineWidth",3);
xlabel('Time')
ylabel('x[n]')

subplot(3,1,2);
stem(t, v, "LineWidth",3);
xlabel('Time')
ylabel('v[n]')

subplot(3,1,3);
stem(t_2, con, "LineWidth",3);
xlabel('Time')
ylabel('x[n]*v[n]')
```


x[n]
v[n]
x[n]*v[n]
Time
Time
Time

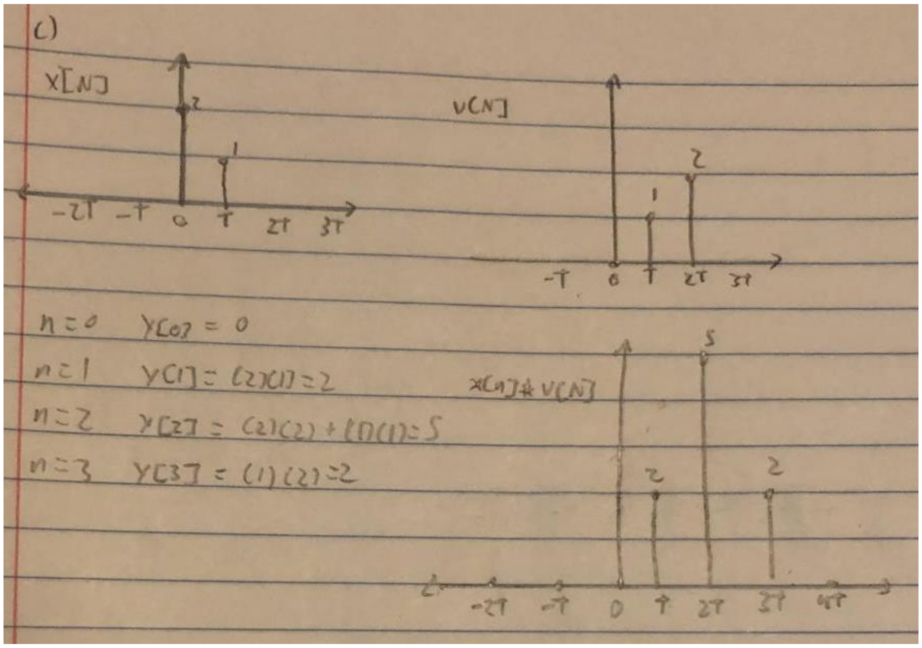
1b)


b)
X[N]
V[N]
n=0 Y[0] = (2)(1) = 2
n=1 Y[1] = (2)(1) + (1)(1) = 3
n=2 Y[2] = (2)(1) + (1)(1) = 3
n=3 Y[3] = (2)(1) + (1)(1) = 3
n=4 Y[4] = (1)(1) = 1
X[N] * V[N]

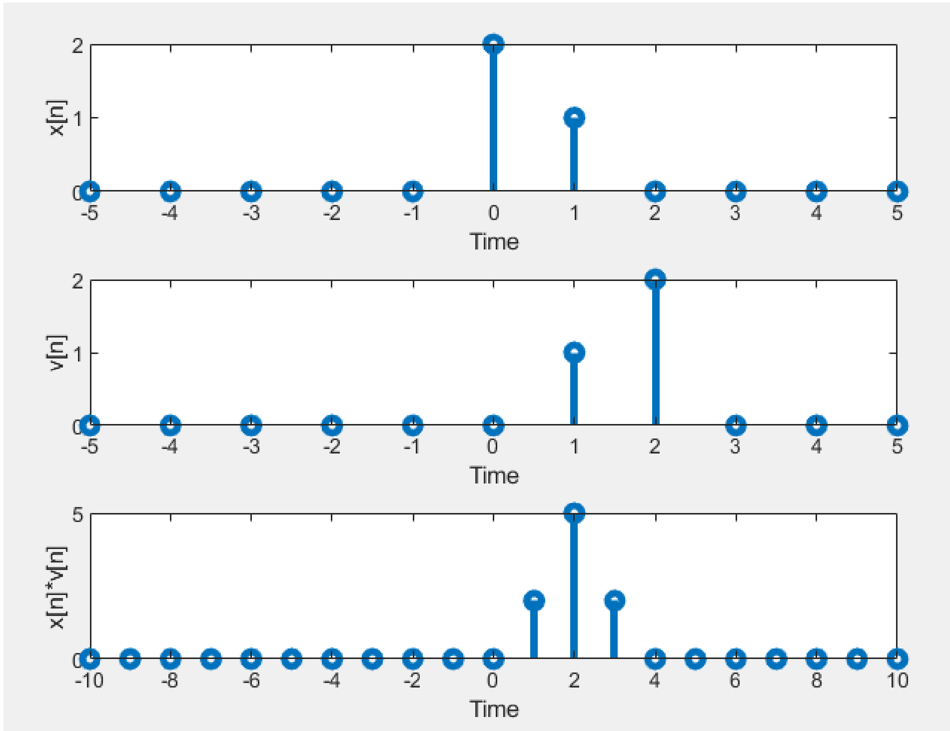
```matlab
%Set time ranges
t = -5:5;
t_2 = -10:10;

%Set two variables to SimpleFunctions() object
x = SimpleFunctions();
v = SimpleFunctions();

%Created x[n] and v[n]
x = 2*x.unitstep(t) - x.unitstep(t-1) - x.unitstep(t-2);
v = v.unitstep(t)-v.unitstep(t-4);

%Computing convolution of x[n] and v[n]
con = conv(x, v);

%Creating three subplots x[n], v[n], and x[n]*v[n]
subplot(3,1,1);
stem(t, x, "LineWidth",3);
xlabel('Time')
ylabel('x[n]')

subplot(3,1,2);
stem(t, v, "LineWidth",3);
xlabel('Time')
ylabel('v[n]')

subplot(3,1,3);
stem(t_2, con, "LineWidth",3);
xlabel('Time')
ylabel('x[n]*v[n]')
```



1c)




```
%Set time ranges
t = -5:5;
t_2 = -10:10;

%Set two variables to SimpleFunctions() object
x = SimpleFunctions();
v = SimpleFunctions();

%Created x[n] and v[n]
x = 2*x.unitstep(t) - x.unitstep(t-1) - x.unitstep(t-2);
v = v.unitstep(t-1) + v.unitstep(t-2) - 2*v.unitstep(t-3);

%Computing convolution of x[n] and v[n]
con = conv(x, v);

%Creating three subplots x[n], v[n], and x[n]*v[n]
subplot(3,1,1);
stem(t, x, "LineWidth",3);
xlabel('Time')
ylabel('x[n]')

subplot(3,1,2);
stem(t, v, "LineWidth",3);
xlabel('Time')
ylabel('v[n]')

subplot(3,1,3);
stem(t_2, con, "LineWidth",3);
xlabel('Time')
ylabel('x[n]*v[n]')
```



Question 3

```matlab
[signal, Fs] = audioread('my_speech_clip.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.
```

Question 4

```
[signal, Fs] = audioread('my_speech_clip.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.

% Creates variable for alpha and echo delay(msec).
Te = 1000;
alpha = 0.3;

%Creating a variable that keeps track of the number of samples delayed.
S_delay = round((Te/1000) / T);

%Vector created to store echos
signalplusecho = zeros(L+S_delay, 1);
signalplusecho(1:L) = signal;

%Echo is added
signalplusecho(S_delay+1:end) = signalplusecho(S_delay+1:end) + alpha*signal;
%Rescales to ensure that the magntiude doesn't exceed 1 (avoids clipping)
signalplusecho = signalplusecho/max(abs(signalplusecho));

% Used to check the magntiude to make sure that no sample exceeds 1
magnitude = sqrt(signalplusecho);
magnitude;

audiowrite('speechwithecho.wav', signalplusecho, Fs);
```

Question 5

```
[signal, Fs] = audioread('my_speech_clip.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.

% Creates variable for alpha and echo delay(msec).
Te = 250;
alpha = 0.3;

%Creating a variable that keeps track of the number of samples delayed.
S_delay = round((Te/1000) / T);

%Vector created to store impulse response
IR = zeros(S_delay+1,1);
%Value is set to 1 so we don't lose original signal
IR(1) = 1;
%The impulse response at the delayed time is set to alpha
IR(S_delay) = alpha;


%Echo is added after convolution
signalplusecho = conv(signal, IR);
%Rescales to ensure that the magntiude doesn't exceed 1 (avoids clipping)
signalplusecho = signalplusecho/max(abs(signalplusecho));

% Used to check the magntiude to make sure that no sample exceeds 1
magnitude = sqrt(signalplusecho);
magnitude;

audiowrite('speechwithconvo.wav', signalplusecho, Fs);
```

Question 7

```
[signal, Fs] = audioread('my_speech_clip.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.

% Creates variable for alpha and echo delay(msec).
Te = 250;
Ne = 2; % Number of echos
alpha = 0.3;

%Creating a variable that keeps track of the number of samples delayed.
S_delay = round((Te/1000) / T);

%Vector created to store impulse response
IR = zeros(Ne*S_delay+1,1);
%Value is set to 1 so we don't lose original signal
IR(1) = 1;
%The impulse response at the delayed time is set to alpha
IR(S_delay) = alpha*Ne;

%For loop that runs from 0 to Ne
for j = 0: Ne
    IR(j * S_delay+1) = alpha^(j); %IR is set to alpha ^ j
end

%Echo is added after convolution
signalplusecho = conv(signal, IR);
%Rescales to ensure that the magntiude doesn't exceed 1 (avoids clipping)
signalplusecho = signalplusecho/max(abs(signalplusecho));

% Used to check the magntiude to make sure that no sample exceeds 1
magnitude = sqrt(signalplusecho);
magnitude;

audiowrite('speechwithreverb.wav', signalplusecho, Fs);
```

4) Report

Question #1 asks us to manually compute the convolutions of x[n] and v[n] in question 2.7 from the textbook, then verify our results using MATLAB. On MATLAB, we use the SimpleFunctions() function from lab 1 to create x[n] and v[n] from the questions. Then we use the built-in function *conv* to compute the convolution for x[n] and v[n]. We then set the length of the x-axis to range from -5 to 5 for x[n] and v[n], as well as set another time variable from -10 to 10 for x[n]*v[n]. These variables are used in the stem function when creating the subplots for x[n], v[n], and the convolution of x[n]*v[n]. After plotting the graph, we use xlabel and ylabel to complete our plot.

Question #4 asks us to create an echo and put that echo into our voice recording. First, I start by creating Te which is the time when the echo will start. Alpha is created and this is used to scale the amplitude of the signal. I use S_delay to keep track of how many signals are going to be delayed. Signalplusecho is the audio that will be distorted.

Question #5 asks us to create an echo using convolution. First, we create a vector (IR) which stores the impulse response. We set it to 1 so we can store the original value of signal and not lose that. Then, we set alpha to the delayed time. Lastly, we use the convolution function to create the new distorted sound and then we scale it to ensure that no sound is lost in the process.

Question #6 asks us to make alpha = 1 and see at which value of Te we get an acceptable sound. I tested values from 500 msec down to 10 msec. In my testing, I heard the echo throughout all the values, however at Te = 20 is the maximum I can go before I hear the echo again. Te = 30 was close however the echo was still noticeable. When decreasing the alpha, I noticed that I could have a higher value of Te without the echo being noticeable.

Question #7 asks us to create an echo using reverberation. This was done by using a for loop which created Ne amount of impulse responses. Repeating #6, I observed that the same pattern was very close in this code. I heard the echo throughout all the values, however it was very low at Te = 30. At Te = 40, the echo was noticeable. When I decreased the alpha value, I did not hear a change.