Lab 2

ELECENG 3TP3

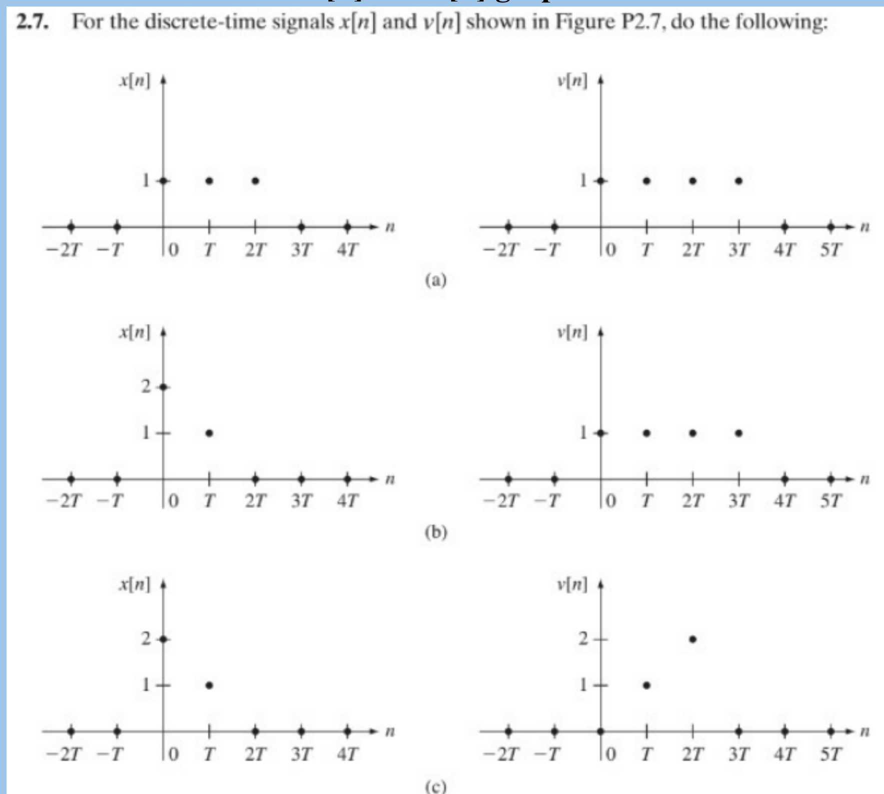
Gurleen Dhillon 400301955

Yuvraj Sandhu 400319134

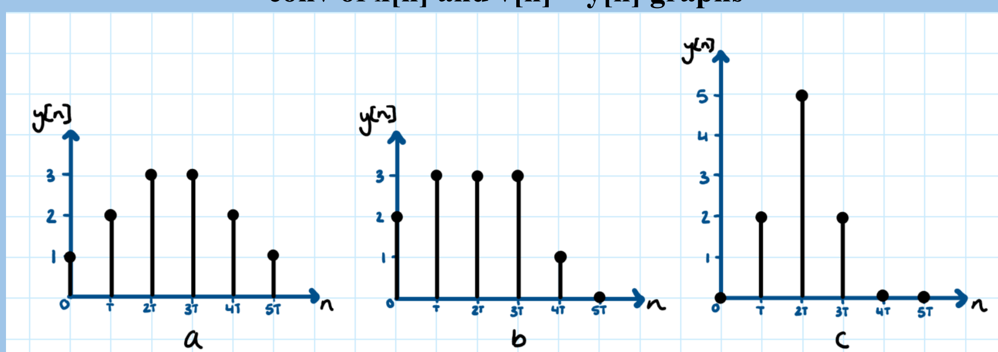
2022-10-21

Part 1

x[n] and v[n] graphs



conv of x[n] and v[n] = y[n] graphs



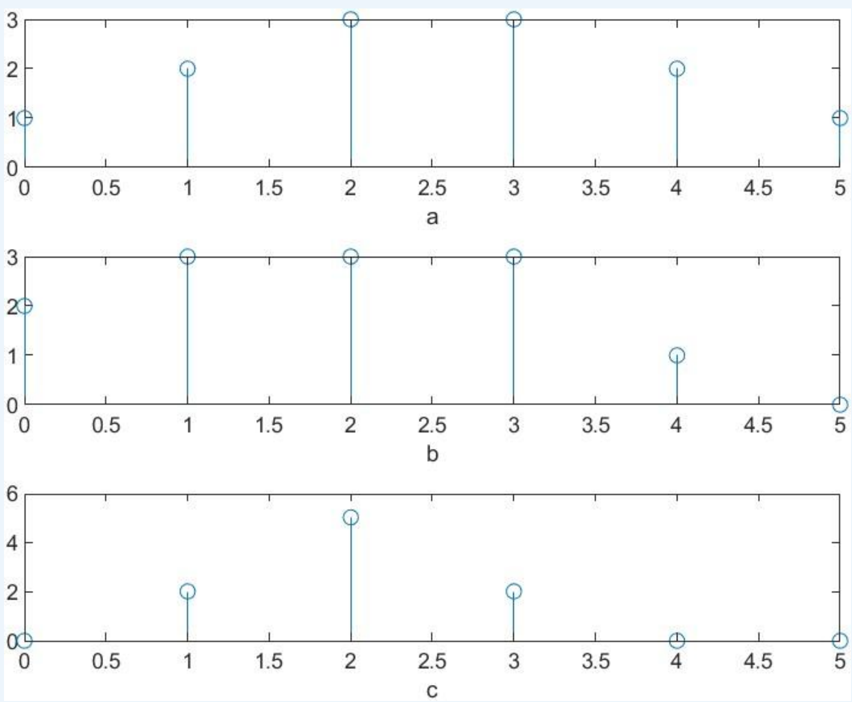

```
clear
f = SimpleFunctions;
t = 0:5;

%x[n] and v[n] functions in a,b,c
x1 = f.unitstep(t) - f.unitstep(t-3);
x2 = 2.*f.unitstep(t) - f.unitstep(t-1) - f.unitstep(t-2);
x3 = 2.*f.unitstep(t) - f.unitstep(t-1) - f.unitstep(t-2);
```

```
v1 = f.unitstep(t) - f.unitstep(t-4);
v2 = f.unitstep(t) - f.unitstep(t-4);
v3 = f.unitstep(t-1) + f.unitstep(t-2) - 2.*f.unitstep(t-3);

%convs of x[n] and v[n]
y1 = conv(x1, v1);
y2 = conv(x2, v2);
y3 = conv(x3, v3);

%plot of convs
subplot(3, 1, 1)
stem(t, y1(1:length(t)))
xlabel("a")
subplot(3, 1, 2)
stem(t, y2(1:length(t)))
xlabel("b")
subplot(3, 1, 3)
stem(t, y3(1:length(t)))
xlabel("c")
exportgraphics(gcf, '3tp3_lab2_1b.jpg');
```



Using the graphs from the textbook under question 2.7, we are able to create 6 equations, 3 for x[n] and 3 for v[n] for each part of the question. Using these equations, we can use the `conv` function 3 times for each pair of x[n] and v[n] equations. We are then able to plot these `conv` functions as subplots.

Part 3

```
clear
[signal, Fs] = audioread('3TP3_lab2_original.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.
```

This code allows us to open the original audio file and create the variables signal (which contains the speech samples), Fs (which contains the sampling frequency), and L (which is the number of samples). From these values, we can obtain the sampling period and the time vector. These values will then be used in parts 4, 5, and 6 to create the distorted audio clips.

Part 4

```
Te = 0.75; %echo delay in sec
alpha = 0.3; %amplitude/strength of echo
Ldelay = Te*Fs; %delay

%creating echo
signalplusecho = zeros(size(signal));
signalplusecho(1:Ldelay) = signal(1:Ldelay);
for i=Ldelay+1:length(signal)
  signalplusecho(i) = signal(i) + alpha*signal(i-Ldelay);
end

%output
signalplusecho = signalplusecho/max(abs(signalplusecho));
audiowrite('3TP3_lab2_part4_echo.wav', signalplusecho, Fs);
```

We are able to set up the variables for the time delay (Te) and amplitude (alpha), and by using these variables and setting up a new empty array, we are able to use the equation $fr(t) = fs(t) + \alpha fs(t - te)$ to create the distorted audio clip.

Part 5

```
clear
[signal, Fs] = audioread('3TP3_lab2_original.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 = 0.75; %echo delay in sec
alpha = 0.3; %amplitude/strength of echo
Ldelay = Te*Fs; %delay
IR = zeros(Ldelay+1, 1); %create an empty impulse response
to generate numbers within it

%Setting the number for the impulse response
IR(1) = 1;
IR(Ldelay+1) = alpha;

signalpulse_conv = conv(signal, IR); %Convolution signal is
updated
signalpulse_conv =
signalpulse_conv/max(abs(signalpulse_conv));

%Output
audiowrite('3TP3_lab2_part5_echo.wav', signalpulse_conv,
Fs);
```

The question distorts the signal and first goes through the original audio file. We initialize the number of samples, sampling period, and time in seconds. The echo delay is calculated by dividing the echo delay by the original signals. After setting up the empty impulse response, we transfer over the original signal with noise. The convolution is then outputted to the wav file.

The impulse response is calculated using a column of a size of the delay +1 with zeroes. The first inputted value is 1 as it transfers the original audio signal to the output and the alpha would then generate the signal noise.

Part 6

How small does Te have to be before the quality of the speech is acceptable? Does your answer change when the value of alpha is decreased?

The value of Te must be at least 0.15 seconds (150 msec) or smaller so the quality of speech is acceptable. At this Te value, the audio does not seem to have the delay which resulted in the distorted audio from part 4, but instead makes it seem as though the volume of the original audio has been increased, like when you hear a voice over a speaker. Decreasing the value of alpha (the amplitude of the echo) does not change the fact that the value of Te must be 0.15 seconds or less to produce an acceptable audio, unless the value of alpha is zero. This is because no matter how small the amplitude of the echo is, it can still be heard in the audio clip. The only way to fix the quality of speech is to ensure that the value of Te is as small as possible.

Part 7

```
clear
[signal, Fs] = audioread('3TP3_lab2_original.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 = 0.75; %echo delay in sec
alpha = 0.3; %amplitude/strength of echo
Ldelay = Te*Fs; %delay
Echonum = 2; %Represent the number of echos

IR = zeros(Echonum*Ldelay+1,1);%create an empty impulse
response to generate numbers within it

IR(1) = 1;%IR Number set

%Multiply by number of echos added with the echo strength
for x = 1:Echonum
    IR(x*Ldelay +1) = IR(x*Ldelay +1) + alpha^x;
end

%Signal Updated
signalpulse_reverb = conv(signal,IR);
signalpulse_reverb =
signalpulse_reverb/max(abs(signalpulse_reverb));

%output
audiowrite('3TP3_lab2_part7_echo.wav', signalpulse_reverb,
Fs);
```

We are able to set up the variables for the time delay (Te), amplitude (alpha), and number of echos (Echonum) and by using these variables and setting up an empty impulse response, we are able to use a for loop to multiply the number of echos in the distorted audio.