3TP3 LAB TWO

McMaster University

2021.10.19

Cheng Fei_400228518

Yichen Lu_400247938

Question 1

Part A

(a)

2.7 Yichen Lu 400247938 Cheng Fei 400218518

(a)

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

$x[i] = [1,1,1]$, $v[i] = [1,1,1,1]$

$\therefore y[0] = x[0]\,v[0] = 1 \cdot 1 = 1$

$y[1] = x[0]\,v[1] + x[1]\,v[0] = 1+1 = 2$

$y[2] = x[0]\,v[2] + x[1]\,v[1] + x[2]\,v[0] = 3$

$y[3] = x[0]\,v[3] + x[1]\,v[2] + x[2]\,v[1] = 3$

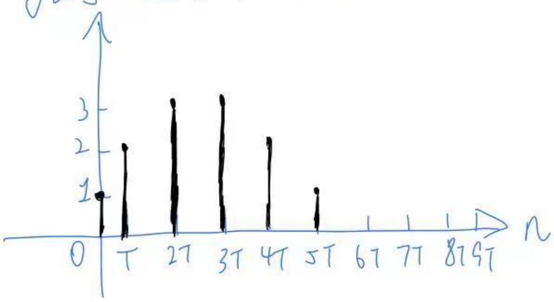
$y[4] = x[1]\,v[3] + x[2]\,v[2] = 2$

$y[5] = x[2]\,v[3] = 1$

$y=[n] = 0$, for $n \geq 6$

$$\therefore y[n] = \begin{cases} 1, & n = 0,5 \\ 2, & n = 1,4 \\ 3, & n = 2,3 \\ 0, & \text{otherwise} \end{cases}$$

$y[n] = x[n] * v[n]$



(b)

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

$x[i] = [2, 1]$, $v[i] = [1, 1, 1, 1]$ $\quad \because x[0] = 2 \quad x[1] = 1$

$\therefore y[0] = x[0]\, v[0] = 2$

$y[1] = x[0]\, v[1] + x[1]\, v[0] = 2 + 1 = 3$

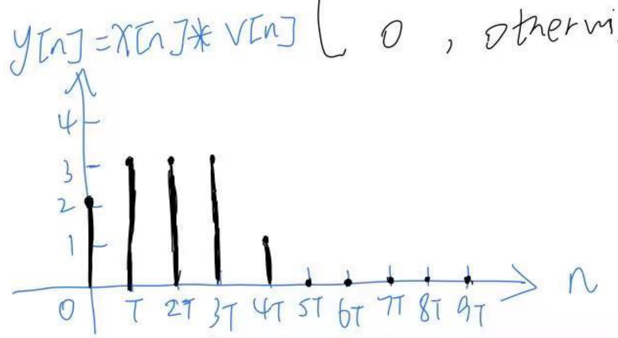
$y[2] = x[0]\, v[2] + x[1]\, v[1] = 2 + 1 = 3$

$y[3] = x[0]\, v[3] + x[1]\, v[2] = 2 + 1 = 3$

$y[4] = x[1]\, v[3] = 1$

$y[n] = 0$, for $n \geq 5$

$$\therefore y[n] = \begin{cases} 1, & n = 4 \\ 2, & n = 0 \\ 3, & n = 1, 2, 3 \\ 0, & \text{otherwise} \end{cases}$$



(c)

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

$x[i] = [2, 1]$, $v[i] = [0, 1, 2]$

$\therefore y[0] = x[0]v[0] = 2 \cdot 0 = 0$

$y[1] = x[0]v[1] + x[1]v[0] = 2$

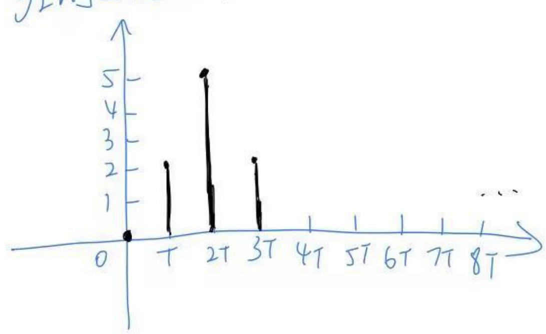
$y[2] = x[0]v[2] + x[1]v[1] = 5$

$y[3] = x[1]v[2] = 2$

$y[n] = 0$, for $n \geq 4$

$$\therefore y[n] = \begin{cases} 2, & n = 1, 3 \\ 5, & n = 2 \\ 0, & \text{otherwise} \end{cases}$$

$y[n] = x[n] * v[n]$



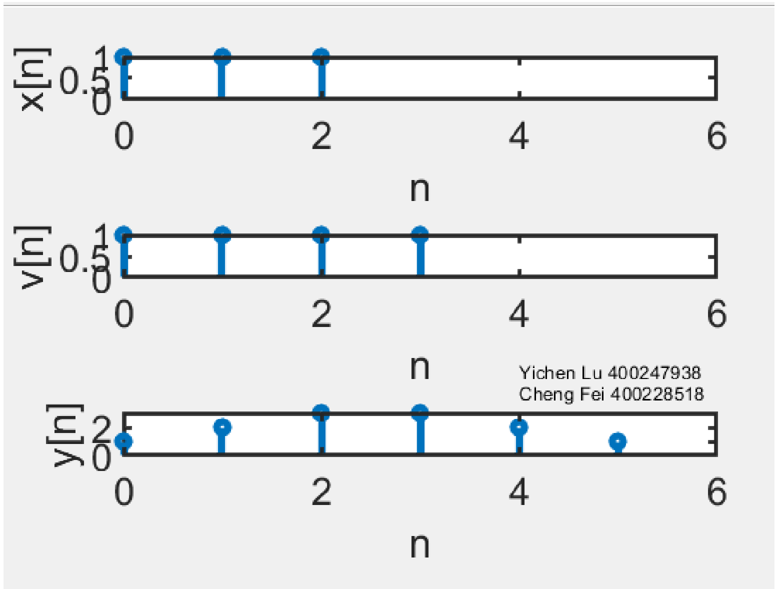
Part B

The code for (a)

The first part of MATLAB code given below are used to verify the handwritten result. Basically, the code just simply assigns the data for each signal into an array and calculate the convolution of x[n] and y[n]. In addition, the other part of MATLAB code is used to plot and export the stem plot by setting some configurations.

```
%Cheng Fei_400228518
%Yichen Lu_400247938
clc;
clear;

x=[1,1,1];
v=[1,1,1,1];

%calculate the convolution
convolution=conv(x,v);
conv_length=length(x)+length(v)-1;
nx=0:length(x)-1;
nv=0:length(v)-1;
nout=0:length(convolution)-1;

%plot
subplot(3,1,1) %plot at the first place
stem(nx,x,'LineWidth',4);
axis([0 conv_length 0 max(x)]);
xlabel('n'); %name the x lable
ylabel('x[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

subplot(3,1,2) %plot at the second place
stem(nv,v,'LineWidth',4);
axis([0 conv_length 0 max(v)]);
xlabel('n'); %name the x lable
ylabel('v[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

subplot(3,1,3) %plot at the third place
stem(nout,convolution,'LineWidth',4);
axis([0 conv_length 0 max(convolution)]);
xlabel('n'); %name the x lable
ylabel('y[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

text(4,4.5,'Cheng Fei 400228518');
text(4,6,'Yichen Lu 400247938');
```

The output for (a)

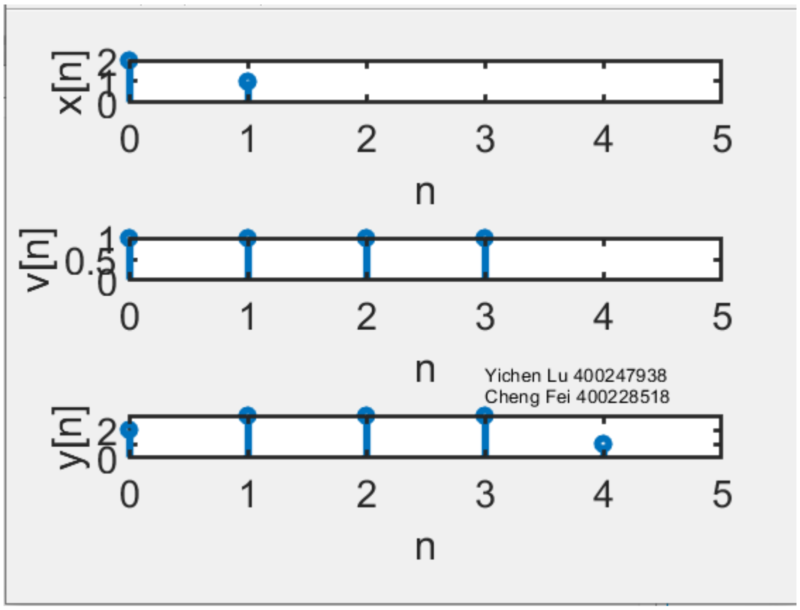


The first part of MATLAB code given below are used to verify the handwritten result. Basically, the code just simply assigns the data for each signal into an array and calculate the convolution of x[n] and y[n]. In addition, the other part of MATLAB code is used to plot and export the stem plot by setting some configurations.

The code for (b)

```
%Cheng Fei_400228518
%Yichen Lu_400247938
clc;
clear;

x=[2,1];
v=[1,1,1,1];

%calculate the convolution
convolution=conv(x,v);
conv_length=length(x)+length(v)-1;
nx=0:length(x)-1;
nv=0:length(v)-1;
nout=0:length(convolution)-1;

%plot
subplot(3,1,1) %plot at the first place
stem(nx,x,'LineWidth',4);
axis([0 conv_length 0 max(x)]);
xlabel('n'); %name the x lable
ylabel('x[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

subplot(3,1,2) %plot at the second place
stem(nv,v,'LineWidth',4);
axis([0 conv_length 0 max(v)]);
xlabel('n'); %name the x lable
ylabel('v[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

subplot(3,1,3) %plot at the third place
stem(nout,convolution,'LineWidth',4);
axis([0 conv_length 0 max(convolution)]);
xlabel('n'); %name the x lable
ylabel('y[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

text(3,4.5,'Cheng Fei 400228518');
text(3,6,'Yichen Lu 400247938');
```

The output for (b)

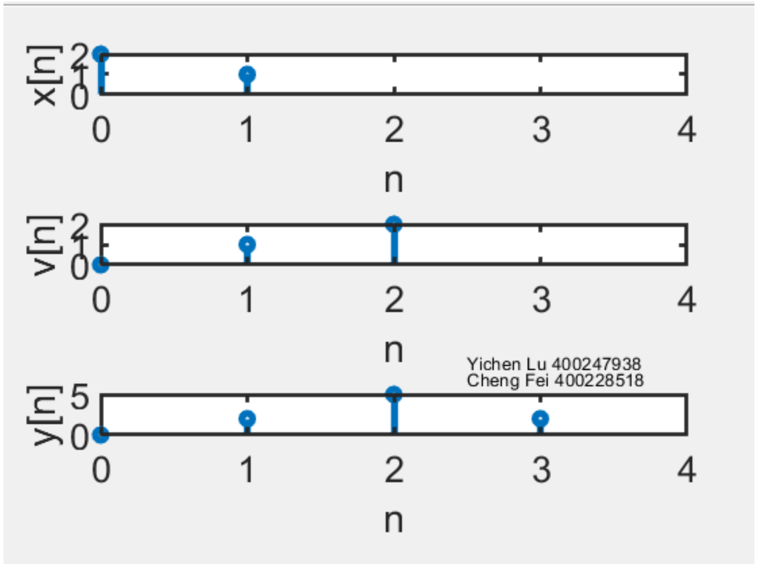


The first part of MATLAB code given below are used to verify the handwritten result. Basically, the code just simply assigns the data for each signal into an array and calculate the convolution of x[n] and y[n]. In addition, the other part of MATLAB code is used to plot and export the stem plot by setting some configurations.

The code for (c)

```
%Cheng Fei_400228518
%Yichen Lu_400247938
clc;
clear;

x=[2,1];
v=[0,1,2];

%calculate the convolution
convolution=conv(x,v);
conv_length=length(x)+length(v)-1;
nx=0:length(x)-1;
nv=0:length(v)-1;
nout=0:length(convolution)-1;

%plot
subplot(3,1,1) %plot at the first place
stem(nx,x,'LineWidth',4);
axis([0 conv_length 0 max(x)]);
xlabel('n'); %name the x lable
ylabel('x[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

subplot(3,1,2) %plot at the second place
stem(nv,v,'LineWidth',4);
axis([0 conv_length 0 max(v)]);
xlabel('n'); %name the x lable
ylabel('v[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

subplot(3,1,3) %plot at the third place
stem(nout,convolution,'LineWidth',4);
axis([0 conv_length 0 max(convolution)]);
xlabel('n'); %name the x lable
ylabel('y[n]'); %name the y lable
set(gca, 'LineWidth', 2);
set(gca, 'fontsize', 20);

text(2.5,7,'Cheng Fei 400228518');
text(2.5,9,'Yichen Lu 400247938');
```

The output for (c)



Question 2-4

The code for Q2-Q4

The MATLAB code below are used to generate a distorted signal. First, the code read the my_speech_clip.wav file and set some variables , such as number of samples in the signal, sampling period in seconds and time vector in seconds. Afterwards, the script calculates the echo delay, Te, in seconds and stores this value in Te_sec. And then L_delay is stored the value that is dividing the echo delay in seconds, Te_sec, by the original signal T. The value of it is rounded to make sure the sample delay is always an integer.

The signal signalpulsecho is created as a column vector of size L+L_delay filled with zeros which is like a container for the signal. And then, we import the original signal into it from one to L, which means the beginning of the new signal is the same as the original signal. Moreover, according to $f_r(t)=f_s(t)+\alpha f_s(t-t_e)$, we add the new signal with the amplified echo signal together. The new signal is re-scaled to make sure the signal not to clip. Lastly, we can write a new .wav file.

```
%Yichen Lu 400247938
%Cheng Fei_400228518
clc;
clear;
[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

Te=579; %the delay is 579ms
Te_sec=Te/1000; %the delay in second
L_delay=round(Te_sec/T); %the number of delay samples
alpha=0.05;

signalpulsecho=zeros(L+L_delay,1); %creat empty output
signalpulsecho(1:L)=signal; %import original signal
%change the signal
signalpulsecho(L_delay+1:end)=signalpulsecho(L_delay+1:end)+signal*alpha;
signalpulsecho=signalpulsecho/max(abs(signalpulsecho));
audiowrite('speechwithecho.wav',signalpulsecho,Fs); %output the file
```

Question 5

The code for Q5

The MATLAB code below are used to generate a distorted signal. First, the code read the my_speech_clip.wav file and set some variables, such as number of samples in the signal, sampling period in seconds and time vector in seconds. Afterwards, the script calculates the echo delay, Te, in seconds and stores this value in Te_sec. And then L_delay is stored the number of samples that created during the delay time. The value of it is rounded to make sure the sample delay is always an integer.

IR is created as a column vector of size 1+L_delay filled with zeros which work with the size of signal to decide the size of outcome audio. The first element of IR is set to 1 because this would transfer the original signal to the output signal, and the alpha at the place of L_delay+1 would create the noise signal due to the delay, since during the convolution,

signalpulsecho_convolution[n] = IR[1]*signal[n-1]+IR[L_delay+1]*signal[n-L_delay-1] = signal[n-1]+alpha*signal[n-L_delay-1], which just satisfies the requirement of $f_r(t)=f_s(t)+\alpha f_s(t-t_e)$ referred in Q4. After the convolution we need to rescale it and finally outport the file with name of ‘speechwithcho_convolution.wav’.

```
%Cheng Fei 400228518
%Yichen Lu 400247938
clc;
clear;

[signal, Fs] = audioread('my_speech_clip.wav'); %import the audio file
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=579; %the delay is 579ms
Te_sec=Te/1000; %the delay in second
L_delay=round(Te_sec/T); %the number of delay samples
alpha=0.05;

IR=zeros(L_delay+1,1); %creat empty IR
%set IR number
IR(1)=1;
IR(L_delay+1)=alpha;
%change the original audio using convolution
signalpulsecho_convolution=conv(signal,IR);
signalpulsecho_convolution=signalpulsecho_convolution/max(abs(signalpulsecho_convolution));
audiowrite('speechwithecho_convolution.wav',signalpulsecho_convolution,Fs); %output the audio file
```

Question 6

After we set the alpha to 1 and test many values of Te, we find that when the delay is smaller than 15ms the quality of speech is acceptable. The Te can be larger to get an acceptable audio quality with the decreasing of alpha.

Question 7

The MATLAB code below are used to generate a distorted signal. First, the code read the my_speech_clip.wav file and set some variables, such as number of samples in the signal, sampling period in seconds and time vector in seconds. Afterwards, the script calculates the echo delay, Te, in seconds and stores this value in Te_sec. And then L_delay is stored the number of samples that created during the delay time. The value of it is rounded to make sure the sample delay is always an integer.

IR is created as a column vector of size 1+L_delay filled with zeros which work with the size of signal to decide the size of outcome audio. 'Ne' decides the times of echo and 'x' represents the which time we are calculating with. The first element of IR is set to 1 because this would transfer the original signal to the output signal, and the alpha^x at the place of x*L_delay+1 would create the noise signal due to the delay. For example if Ne=2, during the convolution, signalpulsecho_convolution[n] = IR[1]*signal[n-1]+IR[L_delay+1]*signal[n-L_delay-1]+ IR[2*L_delay+1]*signal[n-2*L_delay-1] = signal[n-1]+alpha*signal[n-L_delay-1]+ alpha^2*signal[n-2*L_delay-1], here we can find the final signal has two echoes and the strength of second echo is alpha times of the first one. After the convolution we need to rescale it and finally outport the file with name of 'speechwithcho_convolution.wav'.

```
%Cheng Fei 400228518
%Yichen Lu 400247938
clc;
clear;

[signal, Fs] = audioread('my_speech_clip.wav'); %import the audio file
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=579; %the delay is 579ms
Te_sec=Te/1000; %the delay in second
L_delay=round(Te_sec/T); %the number of delay samples
alpha=0.05;
Ne=1; %number of echos

IR=zeros(Ne*L_delay+1,1); %creat empty IR
%set IR number
IR(1)=1;
for x=1:Ne
    IR(x*L_delay+1)=IR(x*L_delay+1)+alpha^x;
end
%change the original sudio
signalpulsecho_reverberation=conv(signal,IR);
signalpulsecho_reverberation=signalpulsecho_reverberation/max(abs(signalpulsecho_reverberation));
audiowrite('speechwithecho_reverberation.wav', signalpulsecho_reverberation,Fs);
```