SUMMARY

## Code for Part 1- Converting Input Characters to Sound:

function character\_to\_sound

CharacterSet = {'1','2','3','4','5','6','7','8','9','\*','0','#','A','B','C','D'} %only to be displayed in command window

CharMatrix = ['1' '2' '3' '4' '5' '6' '7' '8' '9' '\*' '0' '#' 'A' 'B' 'C' 'D'];

F1 = [697 770 852 941]; % Low frequency group

F2 = [1209 1336 1477 1633]; % High frequency group

%////////////////////Generating Frequency Table//////////////////////

f = [];

for i=1:4,

for j=1:4,

f = [ f [F1(i);F2(j)] ];

end

end

f'; % Frequency table

%///////////////////Generating Signal of each character/////////////////////////////

Fs = 8000; % Sampling rate = 8 kHz

N = 1600; % Number of samples for 200 ms (0.2 sec) sound signal

N1 = 400; % Number of samples for 50 ms (0.05 sec) silence period

t = (0:N-1)/Fs; % Time duration of sound signal

%t1 = (0:N1-1)/Fs;

sig = zeros(N,size(f,2)); %signal for each character

sil = zeros(N1,size(f,2)); %0.05 sec silence signal

%////////////////////Playing sound//////////////////

x=input('Please enter 10 Characters: '); % Enter matrix of 10 rendom characters from given Character set e.g. ['1' '6' 'D' '8' '\*'.....upto 10 chars

if length(x)~=10

x=input('Enter 10 Characters from above Character Set:');

end

y=ismember(x, CharMatrix); %checks if the entered characters belongs to character matrix

if y(:)==1 %true for all characters

T=[];

for i=1:10

posX=find(CharMatrix==x(i)); %It individualy finds the position of entered character in the CharMatrix

sig(:,posX) = 1/2\*sum(sin(f(:,posX)\*pi\*t)); % 1/2 is multiplied to limit the amplitude to one. f(:,posX) selects two frequencies from freq table corresponding to that character

sil(:,posX) = sin(0); % zero signal for silence period

S(:,:,i)=cat(1, sig(:,posX), sil(:,posX)); %concatenation of sound signal and silence period

T = [T; S(:,:,i)]; %Adds the character signal to previous one

subplot(2,5,i);

plot(t\*1e3,sig(:,posX));

title(['Character "', x(i),'": [',num2str(f(1,posX)),',',num2str(f(2,posX)),']'])

ylabel('Amplitude'); xlabel('Time (ms)');

end

soundsc(T,Fs);

else

disp('Input Characters are not valid. Please Try Again');

end

%////////////////////Saving the sound signal as file////////////////////

filename = 'character\_to\_sound\_data';

auwrite(T,filename) % Use audiowrite(filename,T) if auwrite() doesn't work for your version of matlab

## Outputs

**In Command Window:**

%for wrong inputs

CharacterSet =

'1' '2' '3' '4' '5' '6' '7' '8' '9' '\*' '0' '#' 'A' 'B' 'C' 'D'

Please enter 10 Characters: ['A' 'B' 'C' 'D' '#' '6' '2' '3']

Enter 10 Characters from above Character Set:['A' 'B' 'C' 'D' '\*' '0' 'X' '6' '12' '3']

Input Characters are not valid. Please Try Again

%for valid input

CharacterSet =

'1' '2' '3' '4' '5' '6' '7' '8' '9' '\*' '0' '#' 'A' 'B' 'C' 'D'

Please enter 10 Characters: ['A' 'B' 'C' 'D' '\*' '0' '#' '6' '2' '3']

## Plots:



## Audio File:

.au file will be saved in Matlab files. The audio output in this case is attached



## Code for Part 2-Converting Sound Signal To Character:

## For Part-2 (a & b):

function sound\_to\_character

filename = input('Please type the file name here: ');

[AudioSignal,Fs] = auread(filename); % Use audioread(filename) if auread() doesn't work for your version of matlab

[nS C] = auread('character\_to\_sound\_data','size'); % gives a matrix of [Total No. of Samples Total no.of Channels]

F1 = [697 770 852 941]; % Low frequency group

F2 = [1209 1336 1477 1633]; % High frequency group

FR = horzcat(F1,F2); % Total Range of frequencies

%////////////////Segmentation of Individual Character/////////////////

t = 0.2; %Time duration for each character i.e. 200 ms

N = t\*Fs; %it gives N=1600 that is the no. samples per character

Tsil=0.05; %Silence period t=0.05sec

N1 = Tsil\*Fs; % Number of samples for 0.05 sec silence period

dur = N+N1; %Total duration per character

ch=1;%:n/dur % Total no. of characters present in the file

for s=1:dur:nS

%It no. of total characters in the file is 10, this loop will run 10 times with the increment of total duration of one character i.e. here it is 250 ms

%(0.2 + 0.05 sec) or 2000 (1600 + 800) samples upto 20000 i.e. the total

%samples in this file i.e for i=1:2000:20000

x=auread(filename,[s s+N]); %auread('aufile',[N1 N2]) returns only samples N1 through N2 from each channel in the file.

%subplot(2,5,ch)

%plot(x)

ch = ch + 1;

end

%Frequensy Domain

nfft=length(x); %length of signal in time domain

nfft2=2.^nextpow2(nfft); %next nearest 2.^n

fx=fft(x,nfft2); %x in freq domain with signal length nfft2. gives a complex n i.e mag+phase of signal also its a mirror signal

fx=fx(1:nfft2/2); %toget only +ve real values by deviding by 2

Xn=fx/max(fx); %to normalize the magnitude to one

%size(Xn)

xfft=Fs.\*(0:nfft2/2-1)/nfft2; %define x axis in freq domain

%mul=[];

c=1; %for no. of figures

for len=40:10:80 %for optimal lengths i.e N=40,50,60,80

n=0:len-1;

figure(c)

for i=1:8

h=cos(2\*pi\*FR(i)/Fs\*n); % Impulse response

fh=fft(h,nfft2); %impulse response in freq domain

fh=fh(1:nfft2/2); %deviding x axis to half to get +ve real values only

B=fh/max(fh); %to normalize the magnitude to one

mul=B'.\*Xn; %filtered signal in freq domain (complex signal)

subplot(2,4,i)

%plot(abs(mul));

plot(xfft,abs(Xn))

hold on

plot(xfft,abs(B),'r')

title(FR(i)); % shows the filter freq as title

legend('Signal','Filter');

end

c=c+1;

end

## Outputs:

**In Command window:**

Please type the file name here: 'character\_to\_sound\_data'

## Plots:

**(a). Plot of Segmented Signal for each character:**

In each segment below, Initial 1600 samples (0.2 sec long) represents the sound signal of each character, while next 400 samples is silence period of 0.05 sec.



**(b). Plot of filters and character signal in frequency domain:**

It will give frequency response of all 8 filters for one character, varying the length of the filter N=40:10:80 in separate figure. Frequency response of both the signal (only one character) and impulse response is plotted on same plot.

**If N=40:**

****

**If N=50:**

****

**If N=60:**

****

**If N=70:**

****

**If N=80:**

****

## Code For Part-2 (merged all parts):

function sound\_to\_character

filename = input('Please type the file name here: ');

[AudioSignal,Fs] = auread(filename); % Use audioread(filename) if auread() doesn't work for your version of matlab

[nS C] = auread('character\_to\_sound\_data','size'); % gives a matrix of [Total No. of Samples Total no.of Channels]

F1 = [697 770 852 941]; % Low frequency group

F2 = [1209 1336 1477 1633]; % High frequency group

CharMatrix = ['1' '2' '3' '4' '5' '6' '7' '8' '9' '\*' '0' '#' 'A' 'B' 'C' 'D'];

FR = horzcat(F1,F2); % Total Range of frequencies

%////////////////////Generating Frequency Table//////////////////////

f = [];% Frequency table

for i=1:4,

for j=1:4,

f = [ f [F1(i);F2(j);F1(i)+F2(j)] ];

end

end

FS = f(3,:); FS' % Range of frequencies i.e. sum of both lower & higher frequency

len=40; %Optimal length is selected 40 here as determined by using N=40,50,60,80.

n=0:len-1; %0<n<N-1 for h(n)

%////////////////Segmentation of Individual Character/////////////////

t = 0.2; %Time duration for each character i.e. 200 ms

N = t\*Fs; %it gives N=1600 that is the no. samples per character

Tsil=0.05; %Silence period t=0.05sec

N1 = Tsil\*Fs; % Number of samples for 0.05 sec silence period

dur = N+N1; %Total duration per character

%ch=1:n/dur; % Total no. of characters present in the file

c=1; % to define the no. of character in AudioSignal

for s=1:dur:nS

%It no. of total characters in the file is 10, this loop will run 10 times with the increment of total duration of one character i.e. here it is 250 ms

%(0.2 + 0.05 sec) or 2000 (1600 + 800) samples upto 20000 i.e. the total

%samples in this file i.e for i=1:2000:20000

x=auread(filename,[s s+N]); %auread('aufile',[N1 N2]) returns only samples N1 through N2 from each channel in the file.

%subplot(2,5,ch)

%plot(x)

%////converting to Frequensy Domain//////

nfft=length(x); %length of signal in time domain

nfft2=2.^nextpow2(nfft); %next nearest 2.^n

fx=fft(x,nfft2); %x in freq domain with signal length nfft2. gives a complex n i.e mag+phase of signal also its a mirror signal

fx=fx(1:nfft2/2); %toget only +ve real values by deviding by 2

Xn=fx/max(fx); %to normalize the magnitude to one

xfft=Fs.\*(0:nfft2/2-1)/nfft2; %define x axis in freq domain

freq=0;

figure(c)

for i=1:8

h=cos(2\*pi\*FR(i)/Fs\*n); % Impulse response

fh=fft(h,nfft2); %impulse response in freq domain

fh=fh(1:nfft2/2); %deviding x axis to half to get +ve real values only

B=fh/max(fh); %to normalize the magnitude to one

mul=B'.\*Xn; %filtered signal in freq domain (complex signal)

subplot(2,4,i)

plot(abs(mul)) %shows the output of all 8 filters for each character in seperate figure

title(FR(i)); % shows the filter freq as title

if max(mul)>=0.59

Char\_No=c;

freq=FR(i); %Adds both lower & Higher freq of each character

end

% plot(xfft,abs(Xn))

% hold on

% plot(xfft,abs(B),'r')

% legend('Signal','Filter');

end

y=ismember(freq, FS); %checks if the freq belongs to Range of (sum of) frequencies

if y==1

posC=find(FS==freq);

Character=CharMatrix(:,posC); %Checks the position of Character in CharMatrix

end

c=c+1;

end

**Comments:**

You can change filter length by increasing or decreasing the parameter ‘len’ in the code (on line 17), to get optimized filter results for all characters. Also you can check results of all N using previous code containing 'a' & 'b' part only

## Output:

Plots of all characters from all 8 filters.

**Character 1:**

****

**Character 2:**

****

**Character 3:**

****

**Character 4:**

****

**Character 5:**

****

**Character 6:**

****

**Character 7:**

****

**Character 8:**

****

**Character 9:**

****

**Character 10:**

****