Emre Can Şen- 21902516

Section-2

Lab-03

Analytical Part

- Before this lab assignment, I didn't know the phone signals were generated using this procedure. So I opened up my phone's dialing app & started dialing. I was able to get similar sounds using the app and the code.
- It is not, because some of the signals share the same frequencies. So they seem intertwined.
- When we isolate the signal, we known that only one signal is being received. So the peaks must belong to that frequency.
- First thing I noticed was there was some white noise. When the data went through it was introduced some white noise. Also at the start of the recording there is some echo leaking into the sound. Then I'm able to hear unechoed sound with white noise. Also since my microphones sampling rate is higher then that said in the lab, I slow down the sound in Matlab to get 10 secs.

Figure 1- Written Answers

Analytical Part with Calculations

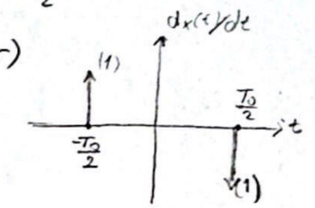
a-) $2\pi f_0 = w_0$, $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(w)\, e^{jwt}\, dw \Rightarrow \left(\begin{matrix}\text{Assume}\\ X(w)=\delta(w-w_0)\end{matrix}\right)$ $x'(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\delta(w-w_0)\, e^{jwt}\, dw$

$= \frac{e^{jw_0 t}}{2\pi}\int_{-\infty}^{\infty}\delta(w-w_0)\,dw = \frac{e^{jw_0 t}}{2\pi} = x'(t)$. So $\frac{e^{jw_0 t}}{2\pi} \xrightarrow{\mathcal{F}} \delta(w-w_0)$, Scale it $e^{jw_0 t} \xrightarrow{\mathcal{F}} 2\pi\delta(w-w_0)$ (linearity property)

b-) $X(w) = \int_{-\infty}^{\infty} \cos(w_0 t)\, e^{jwt}\, dt = \int_{-\infty}^{\infty}\left(\frac{e^{jw_0 t}+\overset{\to e^{j(-w_0)t}}{e^{-jw_0 t}}}{2}\right)e^{-jwt}\,dt$. from linearity and part a

$\Rightarrow \frac{1}{2}\left[2\pi\delta(w-w_0) + 2\pi\delta(w+w_0)\right] = \delta(w-w_0)+\delta(w+w_0)$

c-)



$\frac{dx(t)}{dt} = -\delta\left(t-\frac{T_0}{2}\right) + \delta\left(t+\frac{T_0}{2}\right) \xrightarrow[\text{diff. prop.}]{\mathcal{F}} jwX(w) = e^{jw\frac{T_0}{2}} - e^{-jw\frac{T_0}{2}}$

$\Rightarrow X(w) = \frac{2}{w}\sin\left(w\frac{T_0}{2}\right) = \frac{1}{\cancel{w}}\,\overset{\to \text{sinc}(w\frac{T_0}{2})}{\frac{2\sin(w\frac{T_0}{2})}{w\frac{T_0}{2}}}\,\cancel{w}\cdot\frac{T_0}{2}$

$= T_0\,\text{sinc}\left(w\frac{T_0}{2}\right)$

d-) multiplication in time domain is equal to convolution in fourier dom. $\Rightarrow \underset{\text{part a}}{X_a(w)} * \underset{\text{part c}}{X_c(w)}$

$2\pi\delta(w-w_0) * T_0\,\text{sinc}\left(w\frac{T_0}{2}\right) \xrightarrow{\text{scale \& shift}} \frac{2\pi}{2\pi} T_0\,\text{sinc}\left([w-w_0]\frac{T_0}{2}\right)$

e-) $(\delta(w-w_0)+\delta(w+w_0)) * T_0\,\text{sinc}\left(w\frac{T_0}{2}\right) \xrightarrow{\text{distributive prop.}} \frac{T_0\,\text{sinc}([w-w_0]\frac{T_0}{2})}{2\pi} + \frac{T_0\,\text{sinc}([w+w_0]\frac{T_0}{2})}{2\pi}$

f-) Time shift property $\to e^{-jwt_0}\, T_0\,\text{sinc}\left(w\frac{T_0}{2}\right)$

g-) $2\pi\delta(w-w_0) * \left[e^{-jwt_0}\, T_0\,\text{sinc}\left(w\frac{T_0}{2}\right)\right] = \frac{2\pi}{2\pi} e^{-j(w-w_0)t_0}\, T_0\,\text{sinc}\left([w-w_0]\frac{T_0}{2}\right)$

h-) $\frac{e^{-j(w-w_0)t_0}}{2\pi} T_0\,\text{sinc}\left([w-w_0]\frac{T_0}{2}\right) + \frac{e^{-j(w+w_0)t_0}}{2\pi} T_0\,\text{sinc}\left([w+w_0]\frac{T_0}{2}\right)$

Part-2

a-) Convolving $\delta(t-t_0)$ with $x(t)$ will give $x(t-t_0)$, using this fact:

$h(t) = \delta(t) + \sum_{i=1}^{M} A_i\,\delta(t-t_i)$

b-) $h(t) \xrightarrow{\mathcal{F}} 1 + \sum_{i=1}^{M} A_i\, e^{-jwt_i} = H(jw)$

c-) $X(w)\,H(w) = Y(w)$

d-) $\frac{Y(w)}{H(w)} = X(w)$

Figure 2- Calculated Answers

Plots

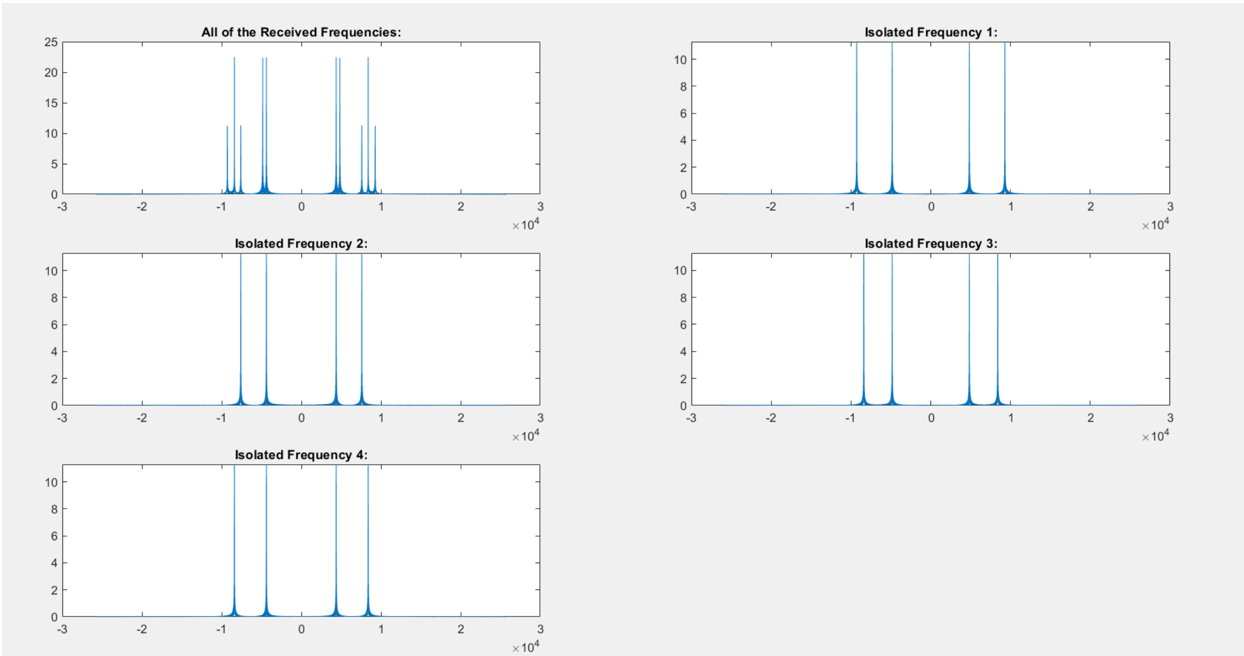


Figure 3- Frequency Plots

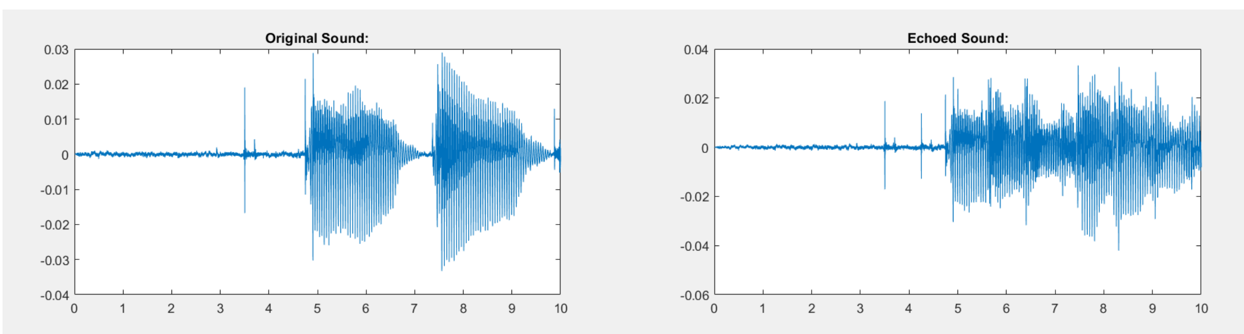


Figure 4- Sound Plots

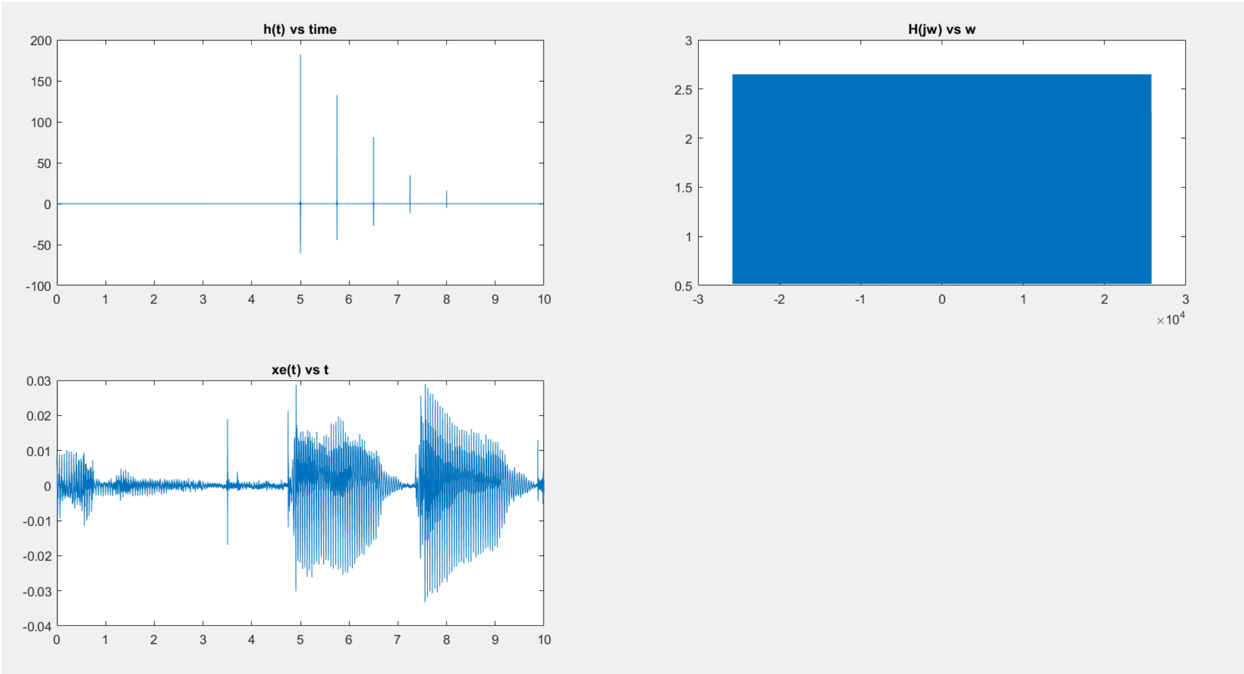


Figure 5- Transform Plots

MATLAB Code

```
Number=[6 1 5 2];
y=DTMFTRA(Number);
%soundsc(y);
figure(1)
X1=FT(y);
omega=linspace(-8192*pi,8192*pi,8193);
omega=omega(1:8193);
subplot(3,2,1)
plot(omega,abs(X1))
title('All of the Received Frequencies:')

kek1=ones(1,2049);
kek2=zeros(1,6144);
rect1=horzcat(kek1,kek2);
x1a=y.*rect1;
X1A=FT(x1a);
subplot(3,2,2)
plot(omega,abs(X1A))
title('Isolated Frequency 1:')
```

```
kek1=ones(1,2049);
kek3=zeros(1,2048);
rect2=horzcat(kek3,kek1,kek3,kek3);
x2a=y.*rect2;
X2A=FT(x2a);
subplot(3,2,3)
plot(omega,abs(X2A))
title('Isolated Frequency 2:')

rect3=horzcat(kek3,kek3,kek1,kek3);
x3a=y.*rect3;
X3A=FT(x3a);
subplot(3,2,4)
plot(omega,abs(X3A))
title('Isolated Frequency 3:')

rect4=horzcat(kek3,kek3,kek3,kek1);
x4a=y.*rect4;
X4A=FT(x4a);
subplot(3,2,5)
plot(omega,abs(X4A))
title('Isolated Frequency 4:')

%part2
sample=[1,(8192/4800)*Fs];
[y_sound,Fs] = audioread('alphabet.wav',sample);
y_sound_original=y_sound;
%soundsc(y_sound,Fs/6)

syms k
t=0:1/8192:10-1/8192;
M=4;
A_i=[0.75 0.5 0.25 0.15];
t_i=[0.75 1.5 2.25 3];
t_i=t_i.*8192;
z1=zeros(6144,1);
y1_sound=vertcat(z1,y_sound);
y1_sound=(0.75).*y1_sound(1:81920);
z2=zeros(12288,1);
y2_sound=vertcat(z2,y_sound);
y2_sound=(0.5).*y2_sound(1:81920);
z3=zeros(18432,1);
y3_sound=vertcat(z3,y_sound);
y3_sound=(0.25).*y3_sound(1:81920);
```

```
z4=zeros(24576,1);
y4_sound=vertcat(z4,y_sound);
y4_sound=(0.15).*y4_sound(1:81920);

y_sound=y_sound+y1_sound+y2_sound+y3_sound+y4_sound;
%soundsc(y_sound,Fs)
figure(2)
subplot(2,2,1)
plot(t,y_sound_original)
title('Original Sound:')
subplot(2,2,2)
plot(t,y_sound)
title('Echoed Sound:')

sound_fourier=FT(y_sound);
omega=linspace(-8192*pi,8192*pi,81921);
omega=omega(1:81920);
H_jw=1+0.75*exp(-1i*omega*0.75)+0.5*exp(-
1i*omega*1.5)+0.25*exp(-1i*omega*2.25)+0.15*exp(-
1i*omega*3);
h_time=IFT(H_jw);
figure(3)
subplot(2,2,1)
plot(t,h_time)
title('h(t) vs time')
subplot(2,2,2)
plot(omega,abs(H_jw))
title('H(jw) vs w')
H_jw=transpose(H_jw);

X_jw=rdivide(sound_fourier,H_jw);
x_e=IFT(X_jw);
soundsc(x_e,Fs/6)
subplot(2,2,3)
plot(t,x_e)
title('xe(t) vs t')

function x=DTMFTRA(Number)
N=(0.25).*[0:4/8192:length(Number)];
x=zeros(1,length(N));

for a= [1:length(Number)]
    T=((a-1)*0.25<=N & N<=(0.25)*a);
    if Number(a)==0
```

```
        x(T)=x(T)+cos(2*pi*941*N(T))+cos(2*pi*1336*N(T));
    elseif Number(a)== 1
        x(T)=x(T)+cos(2*pi*697*N(T))+cos(2*pi*1209*N(T));
    elseif Number(a)== 2
        x(T)=x(T)+cos(2*pi*697*N(T))+cos(2*pi*1336*N(T));
    elseif Number(a)== 3
        x(T)=x(T)+cos(2*pi*697*N(T))+cos(2*pi*1477*N(T));
    elseif Number(a)== 4
        x(T)=x(T)+cos(2*pi*770*N(T))+cos(2*pi*1209*N(T));
    elseif Number(a)== 5
        x(T)=x(T)+cos(2*pi*770*N(T))+cos(2*pi*1336*N(T));
    elseif Number(a)== 6
        x(T)=x(T)+cos(2*pi*770*N(T))+cos(2*pi*1477*N(T));
    elseif Number(a)== 7
        x(T)=x(T)+cos(2*pi*852*N(T))+cos(2*pi*1209*N(T));
    elseif Number(a)== 8
        x(T)=x(T)+cos(2*pi*852*N(T))+cos(2*pi*1336*N(T));
    elseif Number(a)== 9
        x(T)=x(T)+cos(2*pi*852*N(T))+cos(2*pi*1477*N(T));
    else
        x(T)=x(T)+0;
    end
end
end
```