LAB-2

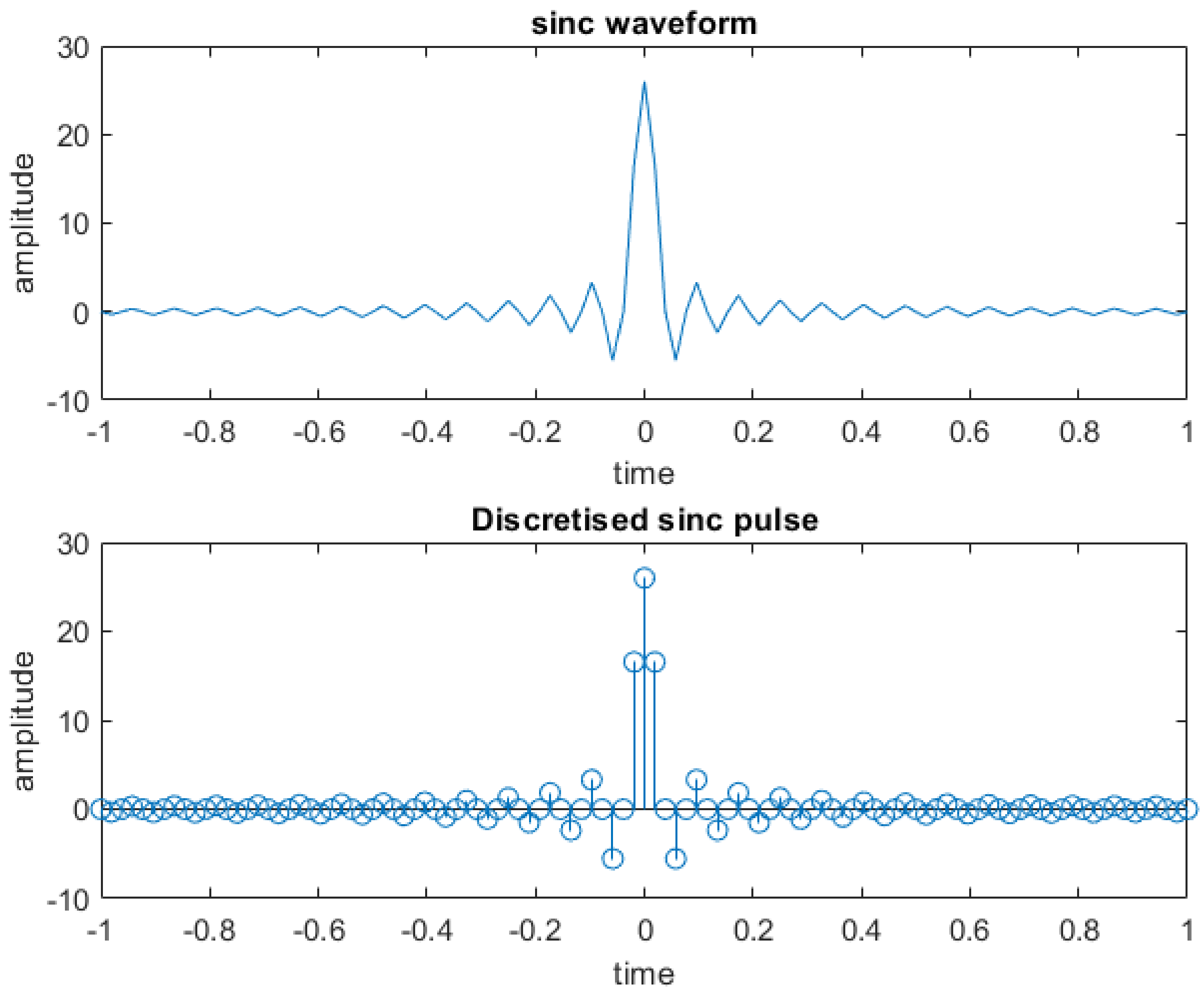
Name: **Sidharth S Nair**

ID: **2019A3PS0178P**

Task-1: Sinc waveform

Code

```
clc;
close all;
B=13;
f=13;%B=N+5
fs=4*f;
tmin=-1;
tmax=1;
t=tmin:(1/fs):tmax;
m = 2*B*sinc(2*B*t);
subplot(2,1,1);
plot(t,m);ylabel('amplitude');xlabel('time');title('sinc waveform');
subplot(2,1,2);
stem(t,m);ylabel('amplitude');xlabel('time');title('Discretised sinc pulse');
```



Observations

- Sampling frequency is a critical component of plotting waveforms since it decides the shape and how much information of the actual waveform is trasferred to the sampled one

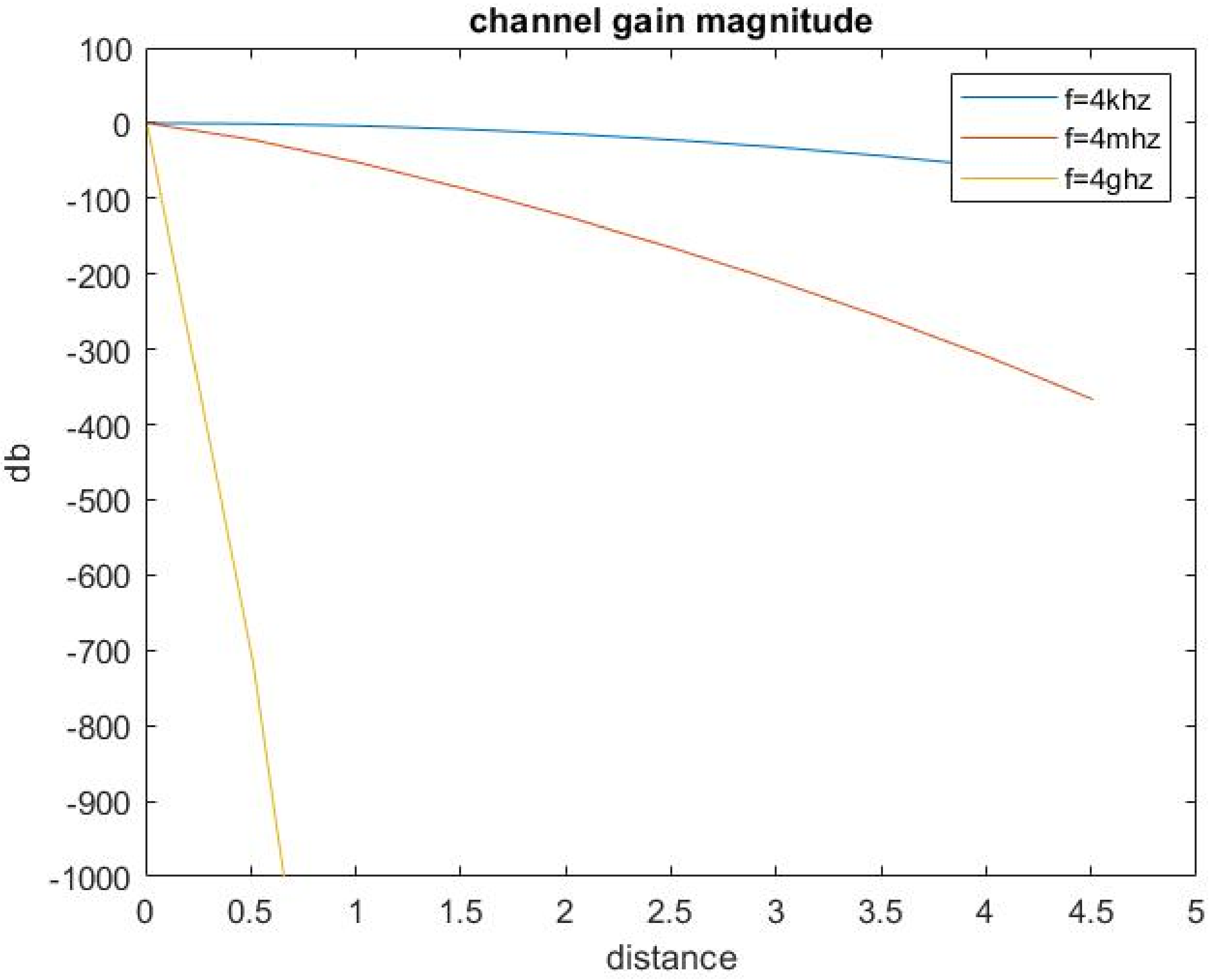
Task-2 : Channel Gain Model

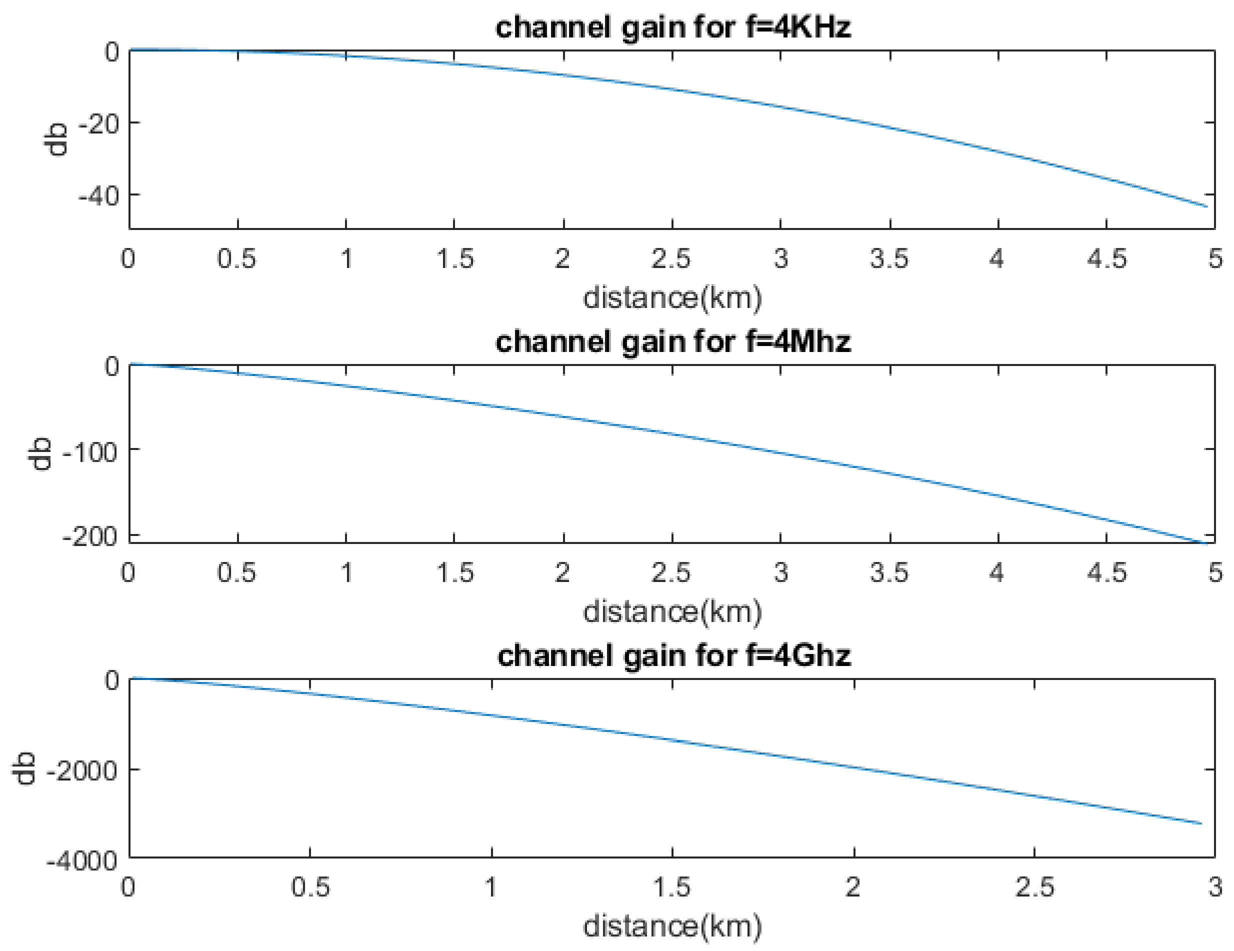
Code:

```
clc;
close all;
d_min=10e-3;
d_max=5000e-3;
```

```
d_step=500e-3;
f1=4e3;  %Hz
f2=4e6;
f3=4e9;
% for wire type=0.4mm
data=struct;
data.r_oc = 280;
data.a_c= 0.0969;
data.L_o = 587.3e-6;
data.L_inf = 426e-6;
data.b = 1.385;
data.fm = 745900;
data.c_inf = 50e-9;
data.c_0 = 0;
data.c_e = 1;
data.g_0 = 0;
data.g_e = 1;
t=d_min:d_step:d_max;
H1=zeros(length(t),1);
H2=zeros(length(t),1);
H3=zeros(length(t),1);
for i=1:length(t)
    H1(i)=10*log10(channel_gain(f1,t(i),data));
    H2(i)=10*log10(channel_gain(f2,t(i),data));
    H3(i)=10*log10(channel_gain(f3,t(i),data));
end
figure(1);
plot(t,H1);hold off;title('channel gain for f=4KHz');xlabel('distance(km)');ylabel('db');
figure(2);
plot(t,H2);title('channel gain for f=4Mhz');xlabel('distance(km)');ylabel('db');
figure(3)
plot(t,H3);%legend('f=4khz','f=4mhz','f=4ghz');
title('channel gain for f=4Ghz');xlabel('distance(km)');ylabel('db');
fig1=figure;
plot(t,H1);hold on;ylim([-1000 100]);plot(t,H2);plot(t,H3);title('channel gain magnitude');xlabel('distance(km)');ylabel('db');legend('f
=4khz','f=4mhz','f=4ghz');

function gain = channel_gain(x,d,data)
    gain=abs(exp(-1*gamma(x,d,data)*d));
end

function gamma = gamma(x,d,data)
    gamma = sqrt((resistance(x,d,data)+1i*omega(x)*inductance(x,d,data))*(conductance(x,d,data)+1i*omega(x)*capacitance(x,d,data)));
end

function omega = omega(x)
    omega = 2*pi*x;
end

function cond = conductance(x,d,data)
    cond = data.g_0*d*x^(data.g_e);
end
function res = resistance(x,d,data)
    res = (((data.r_oc)^4 +(data.a_c)*x^2)*d)^(1/4);
end
function ind = inductance(x,d,data)
    ind = (data.L_o*d+(data.L_inf*d*(x/data.fm)^data.b))/(1+(x/data.fm)^data.b);
end
function cap = capacitance(x,d,data)
    cap = (data.c_inf*d+data.c_0*d*x^(-data.c_e));
end
```

Plots





Observations

- Gain falloff is much heavier for Higher bandwidth signal than for lower bandwidth one
- This means that for faster sampling we need more bandwidth, but will have a heavy attenuation factor with distance
- Bit speed v/s Transmission tradeoff arises

Task-3 : Huffman Encoding

Code:

```
clc;
close;
clear;
%decomposed my name into ascii coded symbols
symbols=double(['s' 'a' 'h' 'i' 'r' ' ' 'd' 't' 'n']);
%adjusted for sum of probability to be 1
probs=[.133 .133 .133 .133 .133 .133 .066 .066 .07];
[dict,avglen] = huffmandict(symbols,probs);
inputsig=['sidharth s nair'];
code=huffmanenco(double(inputsig),dict);
disp(char(huffmandeco(code,dict)));
```

Outputs:

Binary Code for my name using the `huffmanenco` function

```
>> code

code =

  Columns 1 through 18

     0     0     1     0     1     0     1     1     0     1     0     1     1     0     0     0     1     0

  Columns 19 through 36

     1     1     1     0     0     0     1     1     1     0     0     0     0     1     1     0     0     1

  Columns 37 through 47

     1     1     0     0     0     0     1     0     1     0     1
```

Dictionary for the ascii symbols in my name generated using the command

`[dict,avglen] = huffmandict(symbols,probs);`

```
>> dict

dict =

  9×2 cell array

    {[115]}    {1×3 double}
    {[ 97]}    {1×3 double}
    {[104]}    {1×3 double}
    {[105]}    {1×3 double}
    {[114]}    {1×3 double}
    {[ 32]}    {1×3 double}
    {[100]}    {1×4 double}
    {[116]}    {1×4 double}
    {[110]}    {1×3 double}
```

Decoded version of the code using the `huffmandeco` command

```
>> char(huffmandeco(code,dict))

ans =

    'sidharth s nair'
```

Observations

- Average Length of huffman encoder for my name is

```
>> avglen

avglen =

    3.1320
```

Task-4 : Ringing tone

Code:

```
clc;
clear;
f1=440;
f2=480;
fs=10000;
t=0:(1/fs):2;
note1=0.1*sin(2*pi*f1.*t);
note2=0.1*sin(2*pi*f2.*t);
while(1)
    sound(note1+note2,fs);
    pause(6);  %continues the execution after waiting till 6sec
end
```