SIMULATION TASK

Task 1

Here, we intend to simulate a realtime communication system in MATLAB. Consider the general block diagram of Fig. 1.

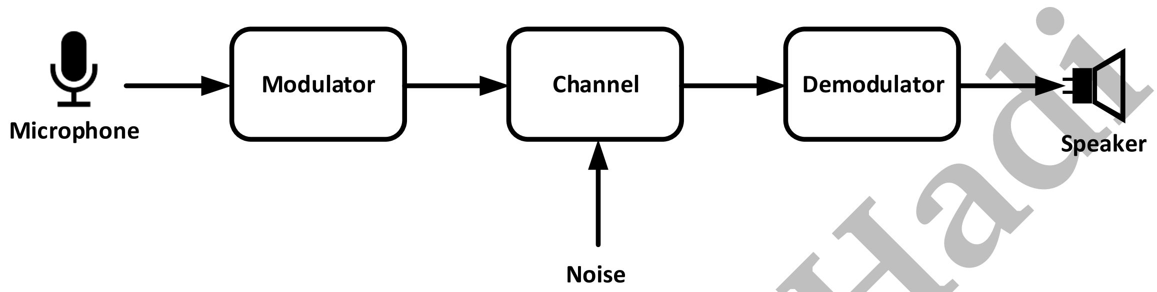


Figure 1: Block diagram of an analog communication system.

(a) Assume that the modulator and demodulator are PM. Write a MATLAB code to simulate the PM communication system. Create separate MATLAB functions for the modulator, demodulator, and channel. Then, connect them in a main mfile. Name the functions pm_modulator, channel, and pm_demodulator.

(b) Repeat the previous part for a DSB communication system. Note that the channel function does not change. You only need to code two new functions for the DSB modulator and demodulator. Name these new functions dsb_modulator and dsb_demodulator.

The picture of mfile codes for PM and DSB modulation and Channel and also PM and DSB demodulation are show below:

```
function u = PM_Modulation(m,dt)
fc=10^5; ac=1; kp=2;
t1=0:dt:((length(m)-1)*dt)
u=ac*cos(2*pi*fc*t1+kp*m);
end
```

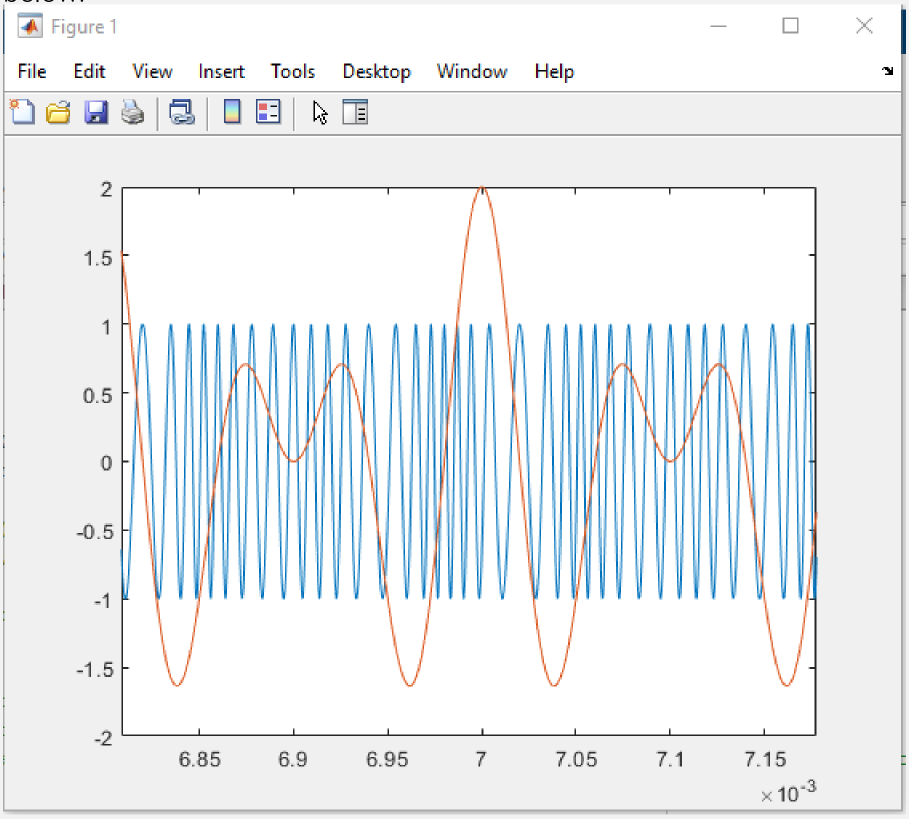
```
function m = PM_Demodulation(u,dt)
fc=10^5; ac=1; kp=2;
t=0:dt:(length(u)-1)*dt;
yq = hilbert(u).*exp(-1i*(2*pi*fc*t));
m = (1/kp)*angle(yq);
end
```


Task 1 continued on next page. . .

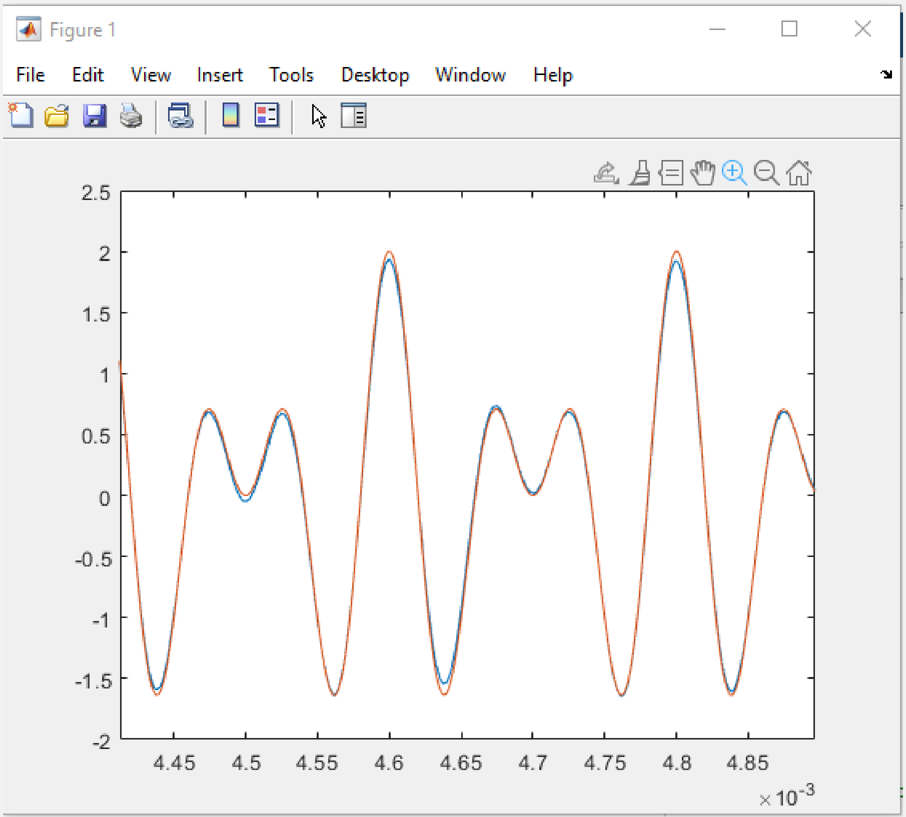
```
function u=DSB_Modulation(m,dt)
fc=10^5; ac=2;
t=0:dt:(length(m)-1)*dt;
u=ac*m.*cos(2*pi*fc*t);
end
```

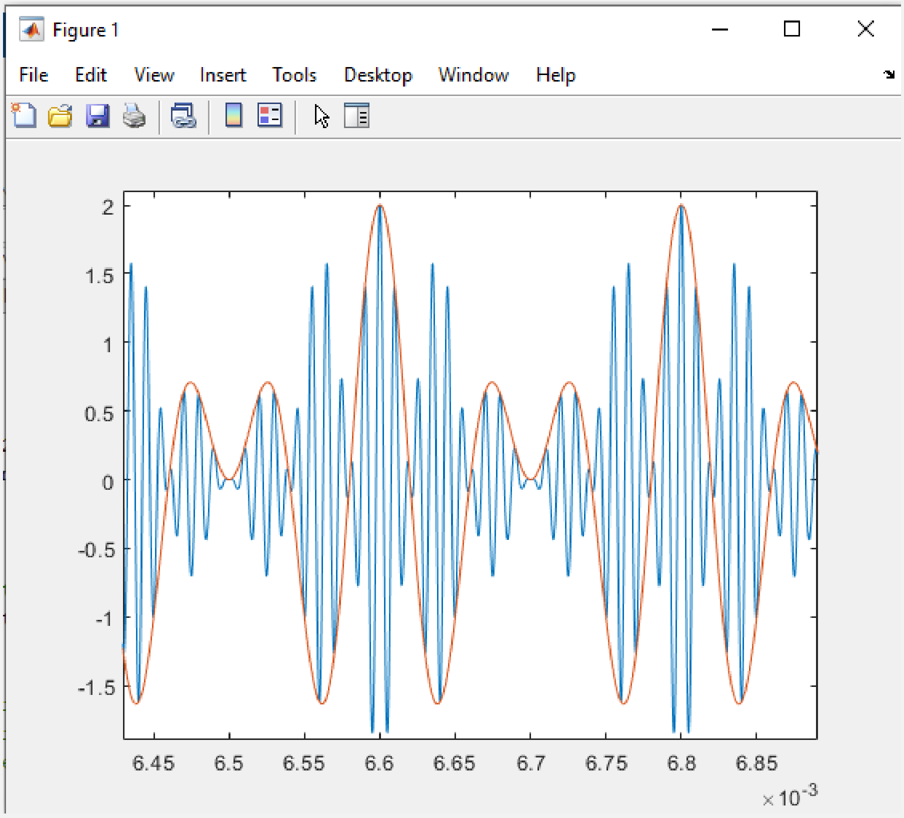
```
function m=DSB_Demodulation(u,dt,bw)
fc=10^5; ac=2;
t=0:dt:(length(u)-1)*dt;
m=2*lowpass(u.*cos(2*pi*fc*t),bw/2,1/dt)/ac;
end
```

The modulated signals and demodulated signals which passed from channel are shown below:

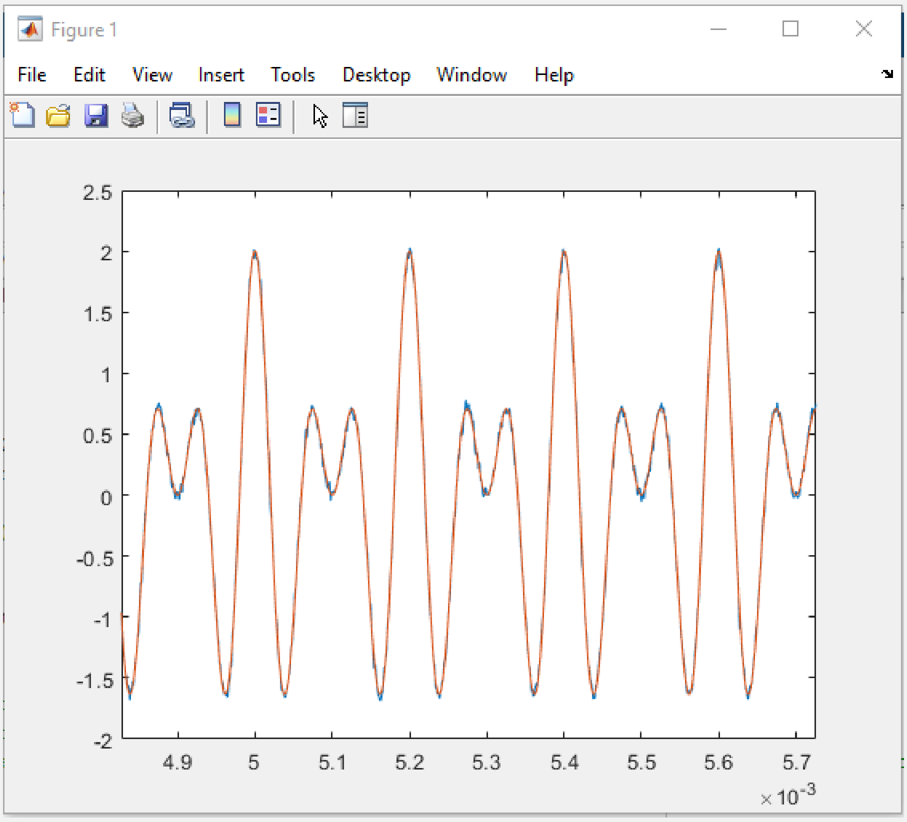


Task 1 continued on next page. . .





Task 1 continued on next page. . .



(c) Feed your simulation codes with a recorded audio file and play the demodulated signal and hear it for different noise levels in the channel. How do you feel when you hear the demodulated signal? Note that you can record your voice from your laptop microphone and feed it to the modulator. You can also play the demodulated signal and hear it from your laptop speaker. MATLAB has useful internal commands for working with microphones and speakers!

the mp3 file name is Gangnam.mp3 which will be the feed. The mfiles are shown below:

```
[signal,FS]=audioread('Gangnam.mp3');
m=signal';
dt=1/length(m(1,:));
y11=PM_Modulation(m(1,:),dt);
y21=Channel(y11);
y12=PM_Modulation(m(2,:),dt);
y22=Channel(y12);
z1=PM_Demodulation(y21,dt);
z2=PM_Demodulation(y22,dt);
out=transpose([z1;z2]);
audiowrite('GangnamPM.wav',out,FS);
```

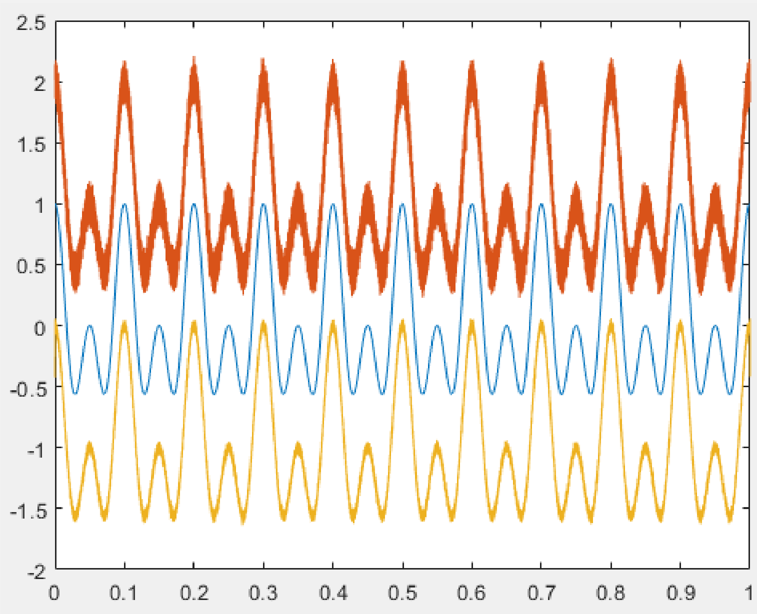
```
[signal,FS]=audioread('Gangnam.mp3');
m=signal';
dt=1/length(m(1,:));
bw=meanfreq(m(1,:),1/length(m(1,:)))+obw(m(1,:),1/length(m(1,:)));
y11=DSB_Modulation(m(1,:),dt);
y21=Channel(y11);
y12=DSB_Modulation(m(2,:),dt);
y22=Channel(y12);
z1=DSB_Demodulation(y21,dt,bw);
z2=DSB_Demodulation(y22,dt,bw);
out=transpose([z1;z2]);
audiowrite('GangnamDSB.wav',out,FS);
```

and the output file will be created.

(d) *Compare the SNR performance of the simulated PM and DSB communication systems. To do this, you can plot the output SNR of both systems in terms of the channel noise level, message bandwidth, and so on.*

The Red signal is the PM output(with offset to be visible easily) and yellow signal is DSB output and he blue signal is the Message.

```
dt=1e-6;
fc=1e5;
t=0:dt:1;
m=0.5*cos(2*pi*10*t)+0.5*cos(2*pi*20*t);
bw=obw(m,1/length(m))+meanfreq(m,1/length(m));
y11=PM_Modulation(m,dt);
y12=Channel(y11);
y13=PM_Demodulation(y12,dt);
pmANDdc=y13+1;
y21=DSB_Modulation(m,dt);
y22=Channel(y21);
y23=DSB_Demodulation(y22,dt,bw);
dsbANDdc=y23-1;
plot(t,m,t,pmANDdc,t,dsbANDdc);
```



Task 1 continued on next page...

Note that the PM signal is for kp=2.

(e) Make your simulation setup realtime. In this way, you talk to the microphone and hear the demodulated signal from the speaker simultaneously without any delay and lag.

```
%PM REAL TIME
deviceReader = audioDeviceReader(1/dt,1/dt);
deviceWriter = audioDeviceWriter('SampleRate',deviceReader.SampleRate);
process = @(m,dt) PM_Demodulation(Channel(PM_Modulation(m,dt)),dt);
tic
while toc>0
    m=transpose(deviceReader());
    deviceWriter(transpose(PM_Demodulation(Channel(PM_Modulation(m,dt)),dt)));
    pause(dt)
end
release(deviceReader);
release(deviceWriter);
```

DSB:

```
deviceReader = audioDeviceReader(fs,fs);
deviceWriter = audioDeviceWriter('SampleRate',deviceReader.SampleRate);
tic
while toc>0
    m=transpose(deviceReader());
    deviceWriter(transpose(DSB_Demodulation(Channel(DSB_Modulation(m,dt)),dt,2*(meanfreq(m,length(m))+obw(m,length(m))/2)));
    pause(dt)
end
release(deviceReader);
release(deviceWriter);
%}
```

(f) Prepare a short report and describe your work concisely. Use suitable figures to better describe the developed codes and to make your report more readable and understandable. Attach a sample of the recorded audios as well as the developed codes to your sent report.

(g) ***Bonus!*** *Write your report in LaTeX.*