

---

# Niruyan Rakulan 214343438

## EECS 4214: Lab 3

### Table of Contents

Problem 1 .....	1
Problem 2 .....	10
Problem 3 .....	11
Problem 4 .....	16

### Problem 1

```
%10MHz
close all;
clear all;
clc;

%1.1 and 1.2
f1=1e6;
R1=1;
C1=1.591549431e-8;
syms y1(t);
ode1=diff(y1)+1/(R1*C1)*y1==sin(2*pi*f1*t)/(R1*C1);
cond1=y1(0)==0;
y1Sol(t)=dsolve(ode1,cond1);
figure;
subplot(2,1,1);
t1=0:1e-9:5e-6-1e-9;
plot(t1,sin(2*pi*f1*t1));
axis([0 5e-6 -1 1]);
xlabel('Time(s)');
ylabel('Voltage(V)');
title('Input Voltage for 10MHz LP Filter');
grid;
subplot(2,1,2);
fplot(y1Sol(t));
axis([0 5e-6 -1 1]);
grid;
xlabel('Time(s)');
ylabel('Voltage(V)');
title('Output Voltage for 10MHz LP Filter');

%%1MHz
f1=1e6;
R2=1;
C2=1.591549431e-7;
syms y2(t);
```

```
ode2=diff(y2)+1/(R2*C2)*y2==sin(2*pi*f1*t)/(R2*C2);
cond2=y2(0)==0;
y2Sol(t)=dsolve(ode2,cond2);
figure;
subplot(2,1,1);
plot(t1,sin(2*pi*f1*t1));
axis([0 5e-6 -1 1]);
grid;
xlabel('Time(s)');
ylabel('Voltage(V)');
title('Input Voltage for 1MHz LP Filter');
subplot(2,1,2);
fplot(y2Sol(t));
axis([0 5e-6 -1 1]);
grid;
xlabel('Time(s)');
ylabel('Voltage(V)');
title('Output Voltage for 1MHz LP Filter');

%100 Khz
f1=1e6;
R3=1;
C3=1.591549431e-6;
syms y3(t);
ode3=diff(y3)+1/(R3*C3)*y3==sin(2*pi*f1*t)/(R3*C3);
cond3=y3(0)==0;
y3Sol(t)=dsolve(ode3,cond3);
figure;
subplot(2,1,1);
plot(t1,sin(2*pi*f1*t1));
axis([0 5e-6 -1 1]);
grid;
xlabel('Time(s)');
ylabel('Voltage(V)');
title('Input Voltage for 100 kHz LP Filter');
subplot(2,1,2);
fplot(y3Sol(t));
axis([0 5e-6 -1 1]);
grid;
xlabel('Time(s)');
ylabel('Voltage(V)');
title('Output Voltage for 100 kHz LP Filter');

%1.3 Bode
%10 MHz LP Filter
H1=tf([1/(R1*C1)],[1 1/(R1*C1)]);
figure();
bode(H1);
title('10 MHz LP Filter');

%1 MHz LP Filter
H2=tf([1/(R2*C2)],[1 1/(R2*C2)]);
figure();
bode(H2);
```

```
title('1 MHz LP Filter');

%100 kHz LP Filter
H3=tf([1/(R3*C3)],[1 1/(R3*C3)]);
figure();
bode(H3);
title('100 kHz LP Filter');

%1.4 FFT

%input
figure;
t1=0:1e-9:5e-2-1e-9;
x=sin(2*pi*f1*t1);
X=fft(x);
X=X/length(x);
X = X(1:length(X)/2+1);
X(2:end-1) = 2*X(2:end-1);
f = linspace(0,1e9/2,length(X));
subplot(2,1,1);
plot(t1,x);
title('Input x(t)');
axis([0 5e-6 -1 1]);
subplot(2,1,2);
plot(f,abs(X));
axis([0 2e6 0 2]);
title('X(jw)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
grid;

%1 MHz LP Filter
figure;
t1=0:1e-9:5e-6-1e-9;
y1a=double(y1Sol(t1));
Y1A=fft(y1a);
Y1A=Y1A/length(Y1A);
Y1A = Y1A(1:length(Y1A)/2+1);
Y1A(2:end-1) = 2*Y1A(2:end-1);
f = linspace(0,1e9/2,length(Y1A));
subplot(2,1,1);
plot(t1,y1a);
title('Output 10 MHz Filter y1(t)');
grid;
axis([0 5e-6 -1 1]);
subplot(2,1,2);
plot(f,abs(Y1A));
axis([0 2e6 0 2]);
title('Output 10 MHz Filter Y1(jw)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
grid;

%1 kHz LP Filter
```

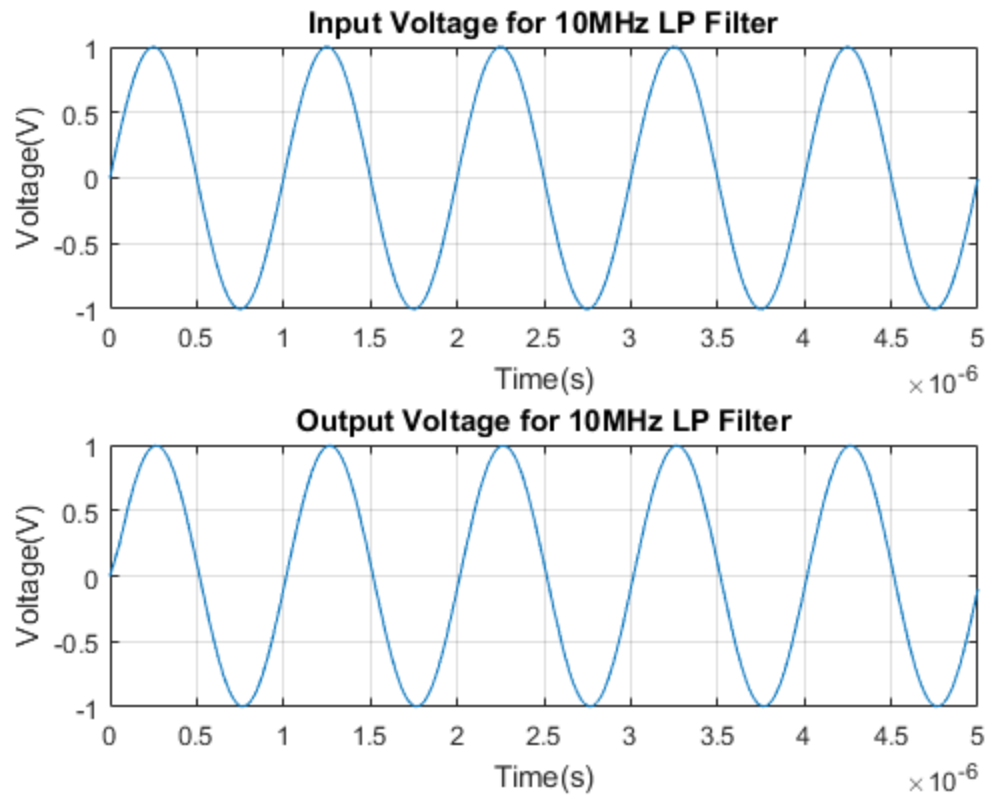
```
figure;
t1=0:1e-9:5e-6-1e-9;
y2a=double(y2Sol(t1));
Y2A=fft(y2a);
Y2A=Y2A/length(Y2A);
Y2A = Y2A(1:length(Y2A)/2+1);
Y2A(2:end-1) = 2*Y2A(2:end-1);
f = linspace(0,1e9/2,length(Y2A));
subplot(2,1,1);
plot(t1,y2a);
axis([0 5e-6 -1 1]);
title('Output 1 MHz Filter y2(t)');
grid;
subplot(2,1,2);
plot(f,abs(Y2A));
axis([0 2e6 0 2]);
title('Output 1 MHz Filter Y2(jw)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
grid;

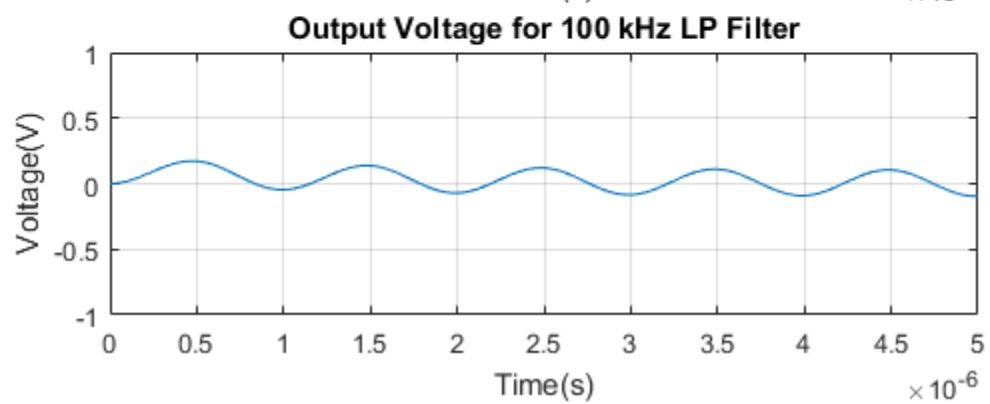
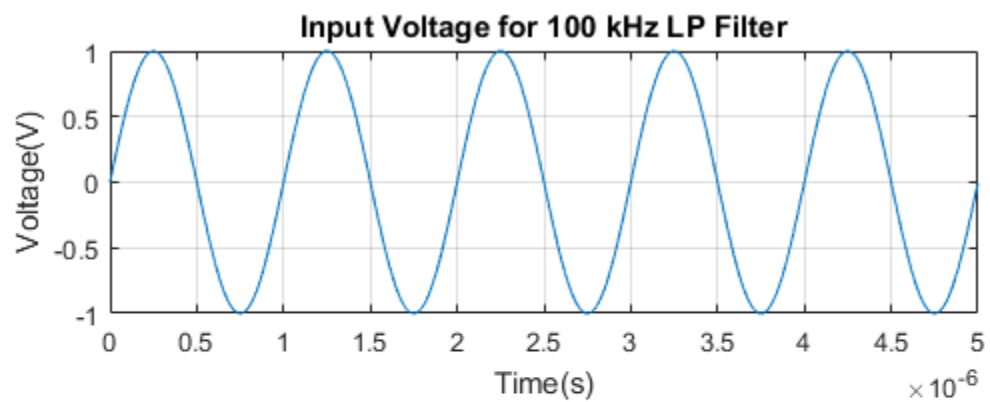
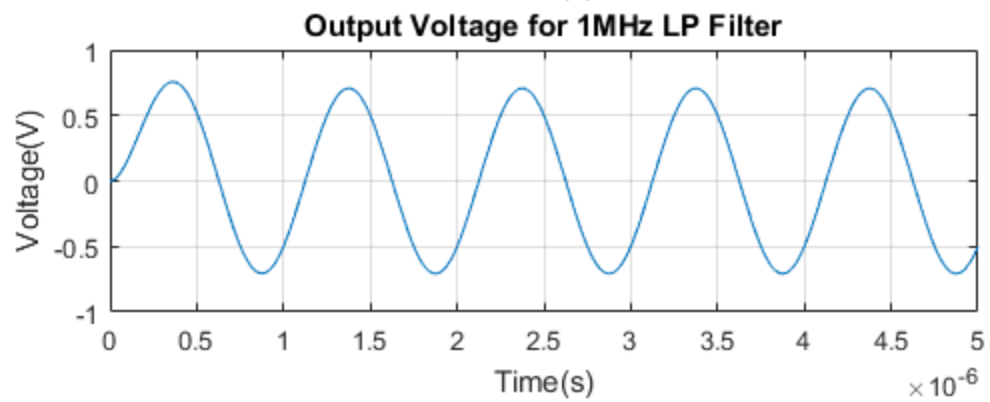
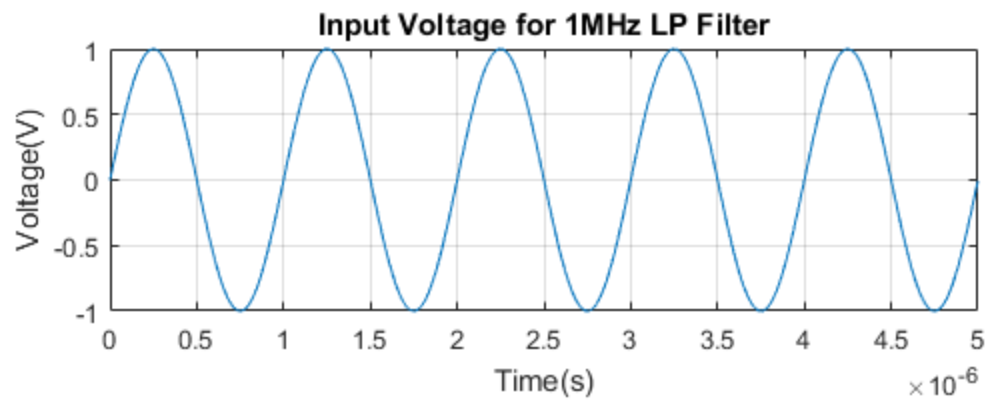
figure;
t1=0:1e-9:5e-6-1e-9;
y3a=double(y3Sol(t1));
Y3A=fft(y3a);
Y3A=Y3A/length(Y3A);
Y3A = Y3A(1:length(Y3A)/2+1);
Y3A(2:end-1) = 2*Y3A(2:end-1);
f = linspace(0,1e9/2,length(Y3A));
subplot(2,1,1);
plot(t1,y3a);
axis([0 5e-6 -1 1]);
title('Output 100 kHz Filter y3(t)');
grid;
subplot(2,1,2);
plot(f,abs(Y3A));
axis([0 2e6 0 2]);
title('Output 100 kHz Filter Y3(jw)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
grid;

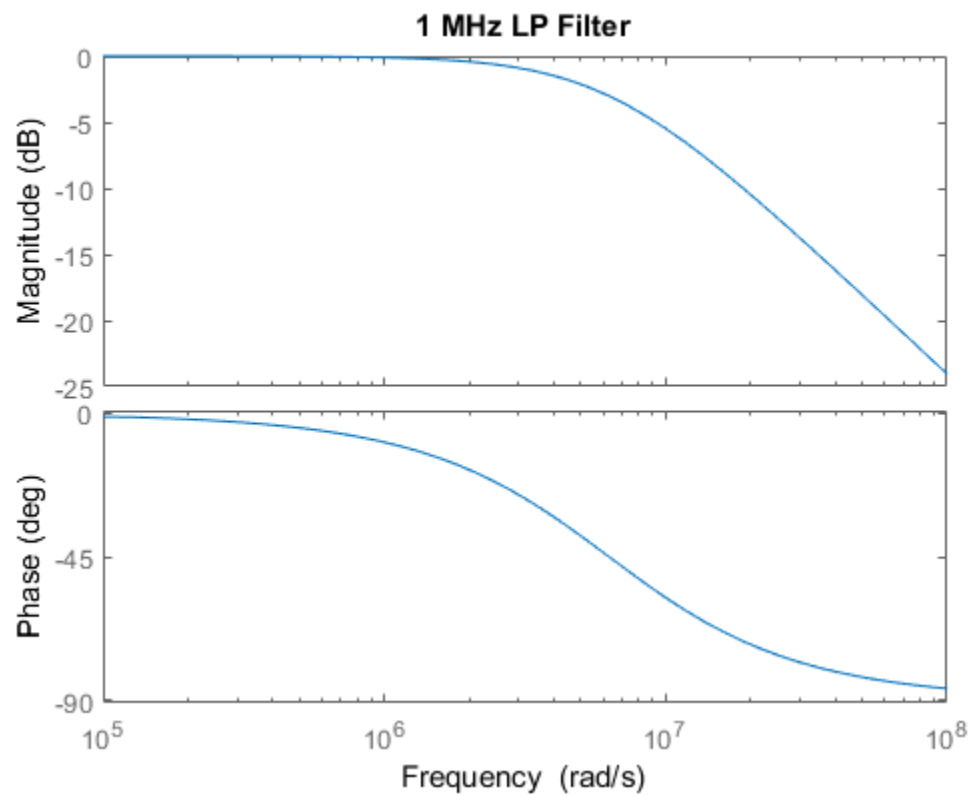
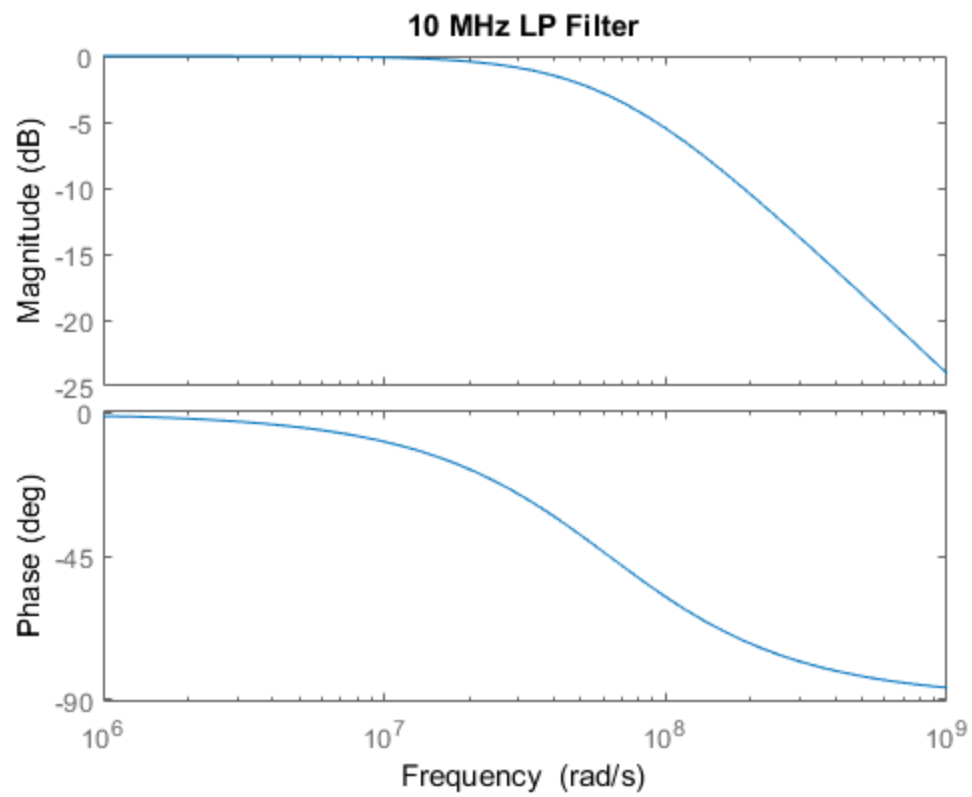
%1.5
%The 10 MHz gain allows the 1 MHz signal to pass through without
%attenuation(Good-fidelity output). The 1 MHz filter attenuates the
%signal to 1/sqrt(2); this is the cutoff frequency of the filter
%(Good-recognition output. The 100kHz filter severely attenuates the
%signal(Poor-Recognition output).

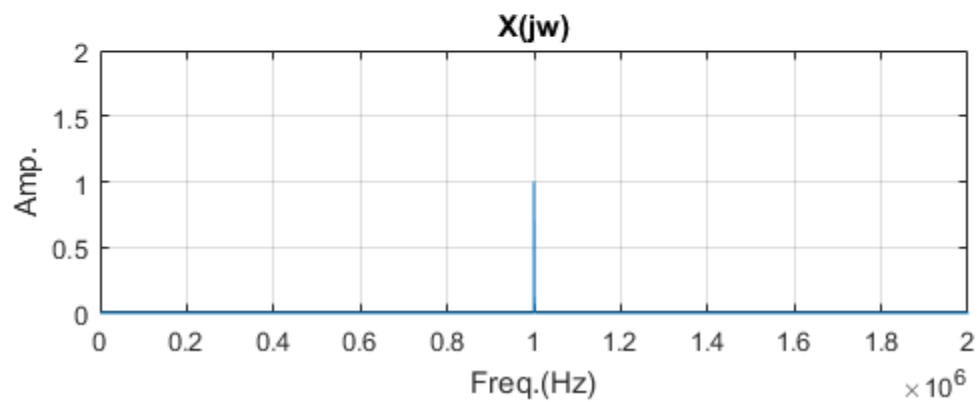
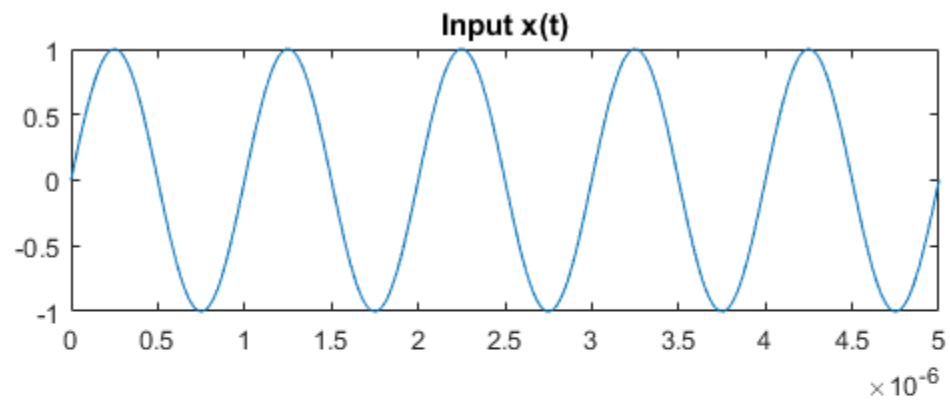
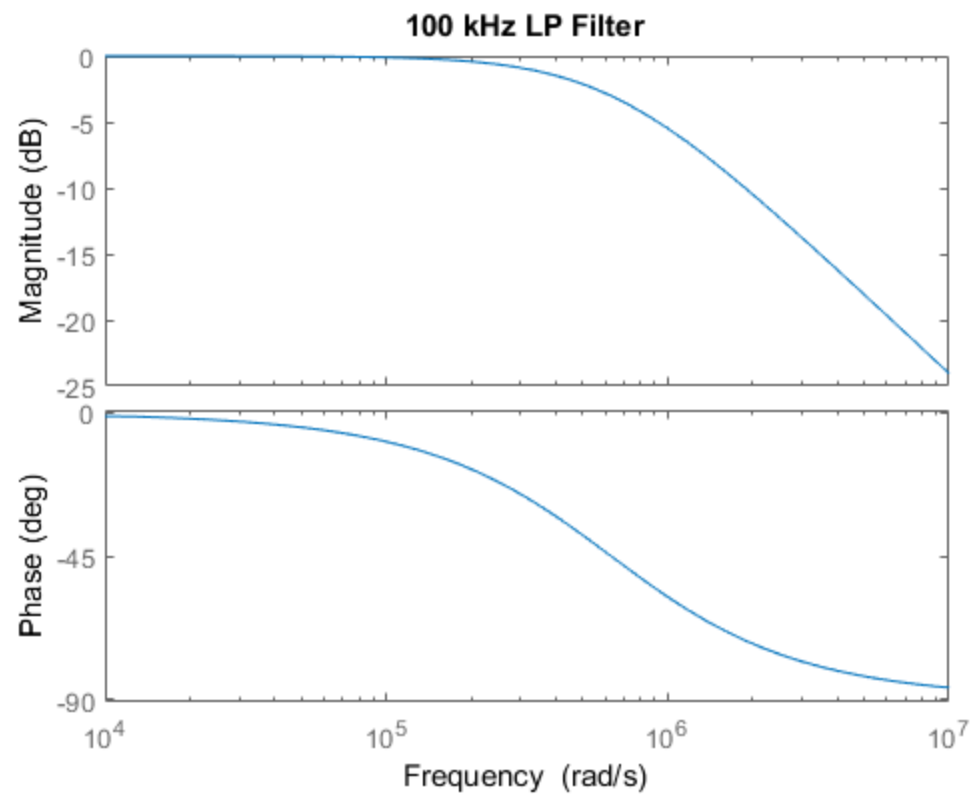
disp('1.5: The 10 MHz gain allows the 1 MHz signal to pass through
without attenuation(Good-fidelity output). The 1 MHz filter
attenuates the signal to 1/sqrt(2); this is the cutoff frequency
of the filter (Good-recognition output. The 100kHz filter severely
attenuates the signal(Poor-Recognition output)');
```

1.5: The 10 MHz gain allows the 1 MHz signal to pass through without attenuation (Good-fidelity output). The 1 MHz filter attenuates the signal to  $1/\sqrt{2}$ ; this is the cutoff frequency of the filter (Good-recognition output). The 100 kHz filter severely attenuates the signal (Poor-Recognition output)

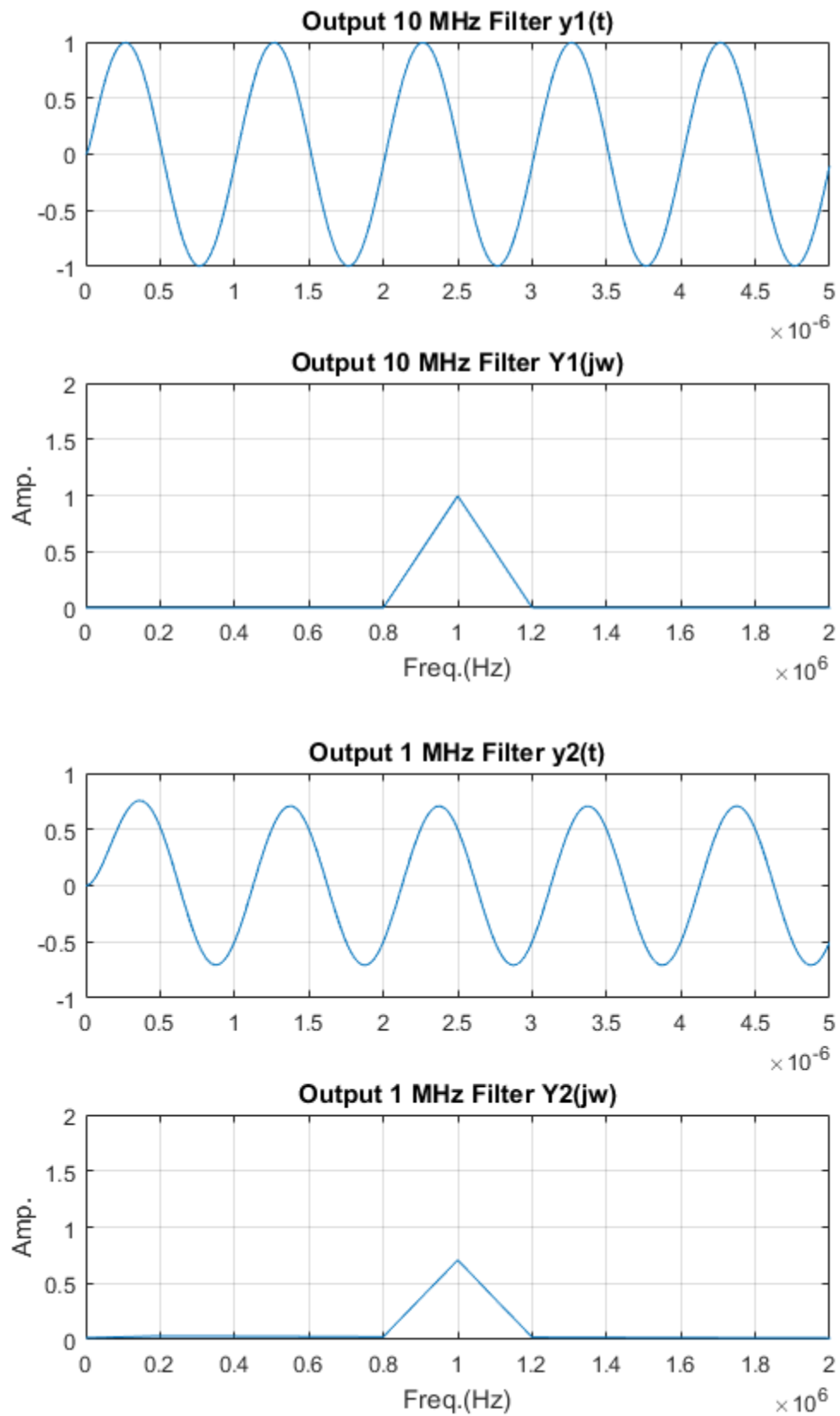


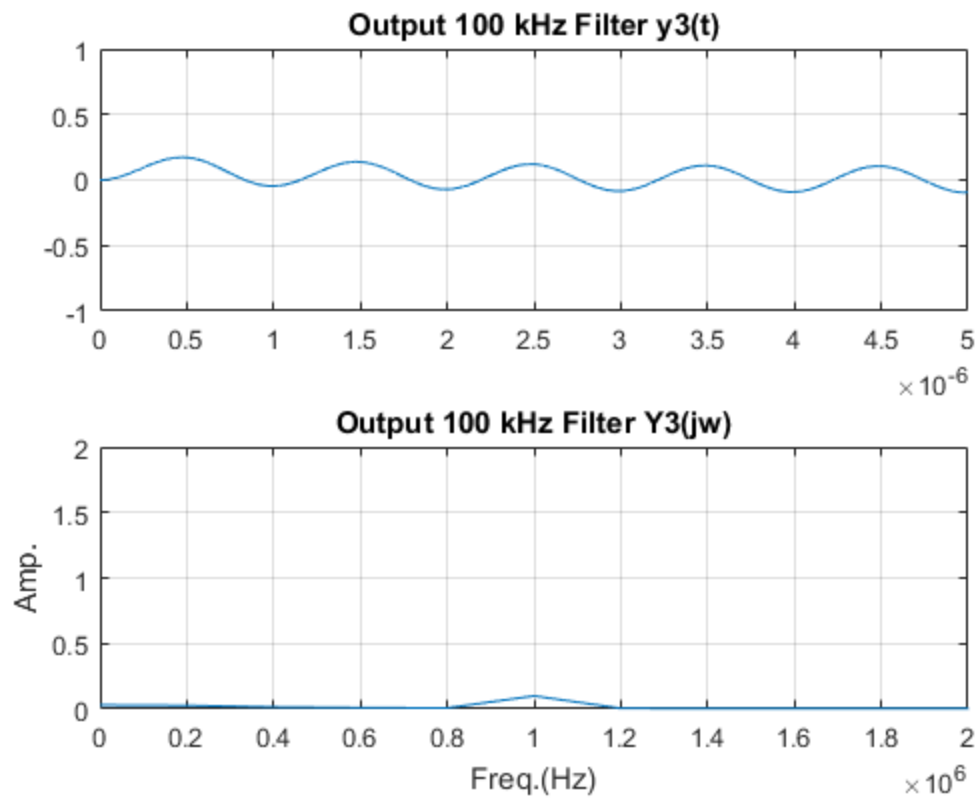












## Problem 2

```
clc;
close all;
clear all;
% 30% to 70%

R2=1;
C2=1.591549431e-7;
Tau=R2*C2;

%30%
hold on;
time_to_30=5.676658041e-8;
thirty_per=-1:1:5;
plot(time_to_30*ones(size(thirty_per)),thirty_per,'c');

thirty_per_volt=-0.4;
thirty_per_volt_time=0:1:2;
plot(thirty_per_volt_time,thirty_per_volt*ones(size(thirty_per_volt_time)),'g');

%70%
hold on;
time_to_70=1.916182232e-7;
seventy_per=-1:1:5;
```

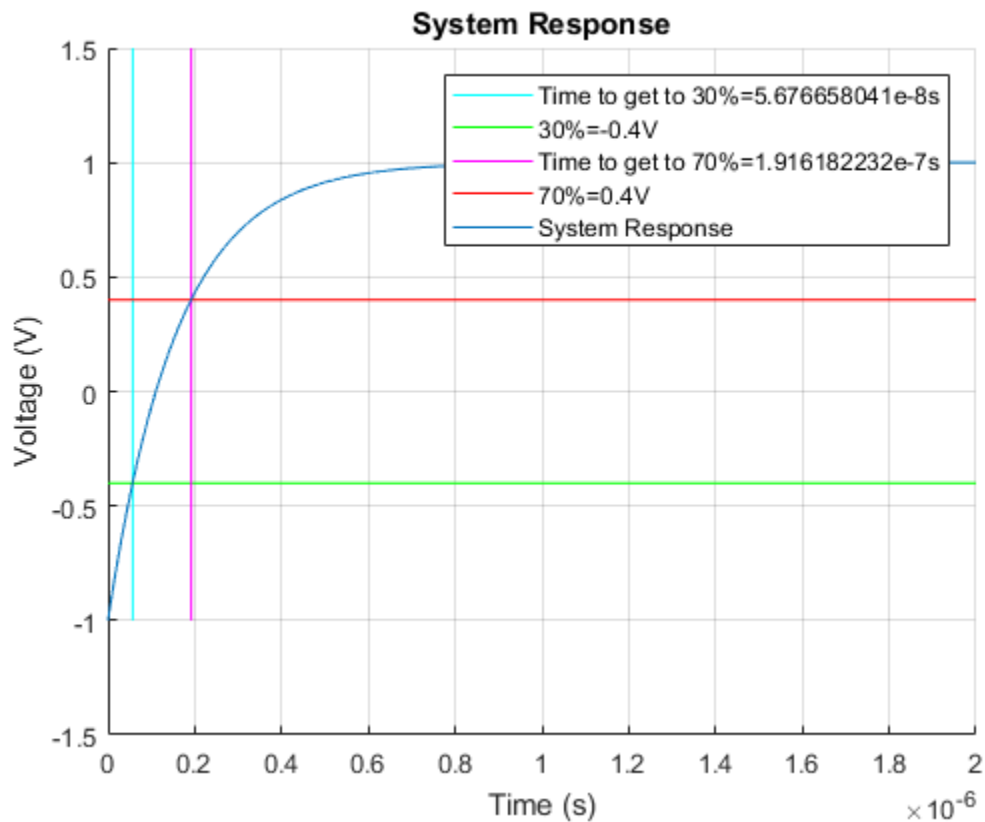
```
plot(time_to_70*ones(size(seventy_per)),seventy_per,'m');

seventy_per_volt=0.4;
seventy_per_volt_time=0:1:2;
plot(seventy_per_volt_time,seventy_per_volt*ones(size(seventy_per_volt_time)),'r')

t=0:1e-10:5e-3;
y=2*(1-exp(-t./(R2*C2)))-1;
plot(t,y);
axis([0 2e-6 -1.5 1.5]);
legend('Time to get to 30%=5.676658041e-8s','30%=-0.4V','Time to get  
to 70%=1.916182232e-7s','70%=0.4V','System Response');
ylabel('Voltage (V)');
xlabel('Time (s)');
title('System Response');
grid;

% Max bit rate = 5.2 MHz
disp('2.1: Time to get to 70%=1.348516428e-7s. Max bit rate = 7.4  
MHz');
```

2.1: Time to get to 70%=1.348516428e-7s. Max bit rate = 7.4 MHz



## Problem 3

```
clc;
```

```
close all;
clear all;

%noise in time
fs=30e6;
t=0:1/fs:1;
noise=randn(1,length(t));
figure;
subplot(2,1,1);
plot(t,noise);
title('Noise in Time.(Unfiltered)');
xlabel('Time(s)');
ylabel('Amp. ');
axis([0 1 -5 5]);
subplot(2,1,2);

%noise in frequency.
NOISE=fft(noise);
NOISE=NOISE/length(noise);
NOISE = NOISE(1:length(NOISE)/2+1);
NOISE(2:end-1) = 2*NOISE(2:end-1);
f = linspace(0,fs/2,length(NOISE));
plot(f,abs(NOISE));
axis([0 1.5e7 0 0.2e-2]);
title('Noise in Freq.(Unfiltered)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');

%10 MHz
figure;
R1=1;
C1=1.591549431e-8;
H1=1./sqrt(1+(2*pi.*f*R1*C1).^2);
subplot(3,1,1);
plot(f,H1);
axis([0 1.5e7 0 1]);
title('10 MHz Filter');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
Y1=H1.*abs(NOISE);
subplot(3,1,2);
plot(f,Y1);
title('Noise in Freq.(Filtered by 10 MHz)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
axis([0 1.5e7 0 0.2e-2]);
y1=ifft(Y1);
subplot(3,1,3);
plot(t(1:length(y1)),y1);
title('Noise in Time.(Filtered by 10 MHz)');
xlabel('Time(s)');
ylabel('Amp. ');
axis([0 1 -1e-6 1e-6]);
```

```
%1 MHz
figure;
R2=1;
C2=1.591549431e-7;
H2=1./sqrt(1+(2*pi.*f*R2*C2).^2);
subplot(3,1,1);
plot(f,H2);
title('1 MHz Filter');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
axis([0 1.5e7 0 1]);
Y2=H2.*abs(NOISE);
subplot(3,1,2);
plot(f,Y2);
title('Noise in Freq.(Filtered by 10 MHz)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
axis([0 1.5e7 0 0.2e-2]);
y2=ifft(Y2);
subplot(3,1,3);
plot(t(1:length(y2)),y2);
title('Noise in Time.(Filtered by 1 MHz)');
xlabel('Time(s)');
ylabel('Amp. ');
axis([0 1 -1e-6 1e-6]);

%100 kHz
figure;
R3=1;
C3=1.591549431e-6;
H3=1./sqrt(1+(2*pi.*f*R3*C3).^2);
subplot(3,1,1);
plot(f,H3);
title('100 kHz Filter');
xlabel('Freq.(Hz)');

ylabel('Amp. ');
axis([0 1.5e7 0 1]);
Y3=H3.*abs(NOISE);
subplot(3,1,2);
plot(f,Y3);
title('Noise in Freq.(Filtered by 10 MHz)');
xlabel('Freq.(Hz)');
ylabel('Amp. ');
axis([0 1.5e7 0 0.2e-2]);
y3=ifft(Y3);
subplot(3,1,3);
plot(t(1:length(y3)),y3);
title('Noise in Time.(Filtered by 1 kHz)');
xlabel('Time(s)');
ylabel('Amp. ');
axis([0 1 -1e-6 1e-6]);
```

```
% The low pass filter with the lower cutoff frequency (100 kHz) was  
better in  
% removing the noise. It is assumed the noise is existent in all  
frequencies  
% so the 100 kHz filter was able to get rid of more noise.  
  
disp('The low pass filter with the lower cutoff frequency (100 kHz)  
was better in removing the noise. It is assumed the noise is existent  
in all frequencies so the 100 kHz filter was able to get rid of more  
noise.');
```

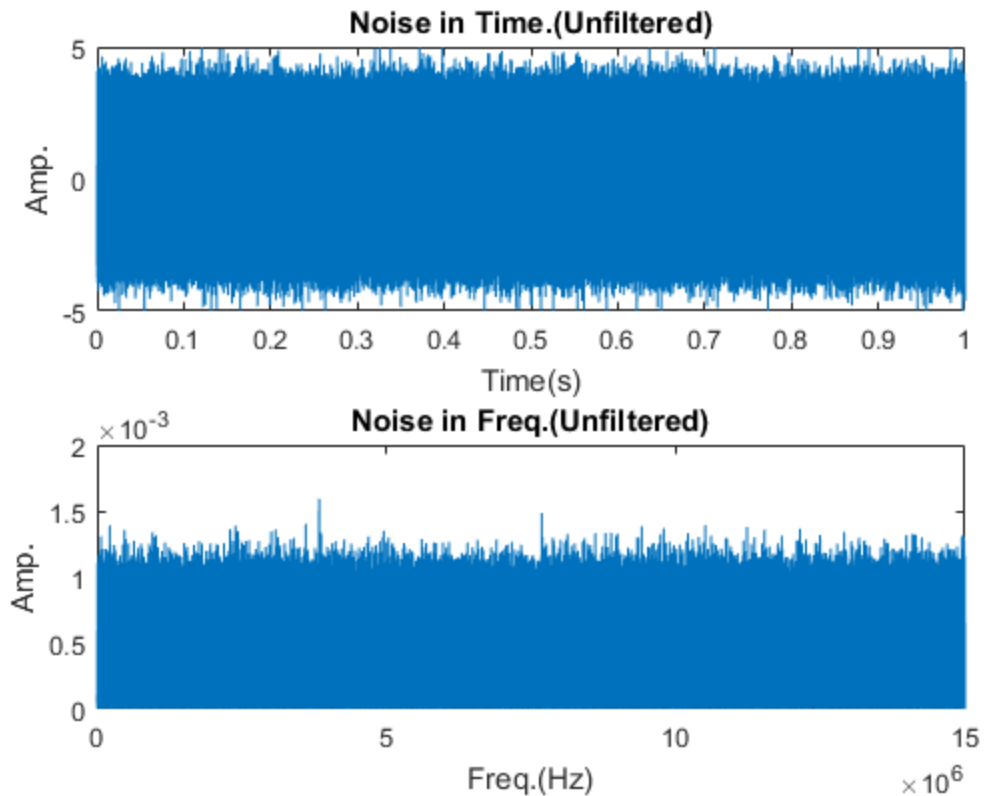
*Warning: Integer operands are required for colon operator when used as index*

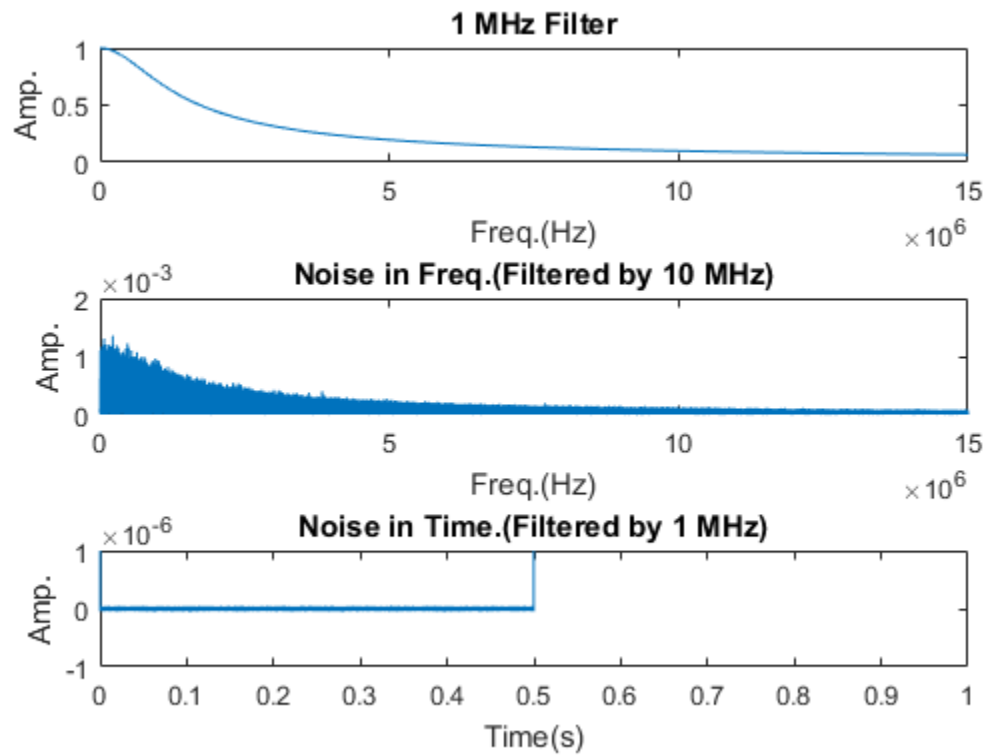
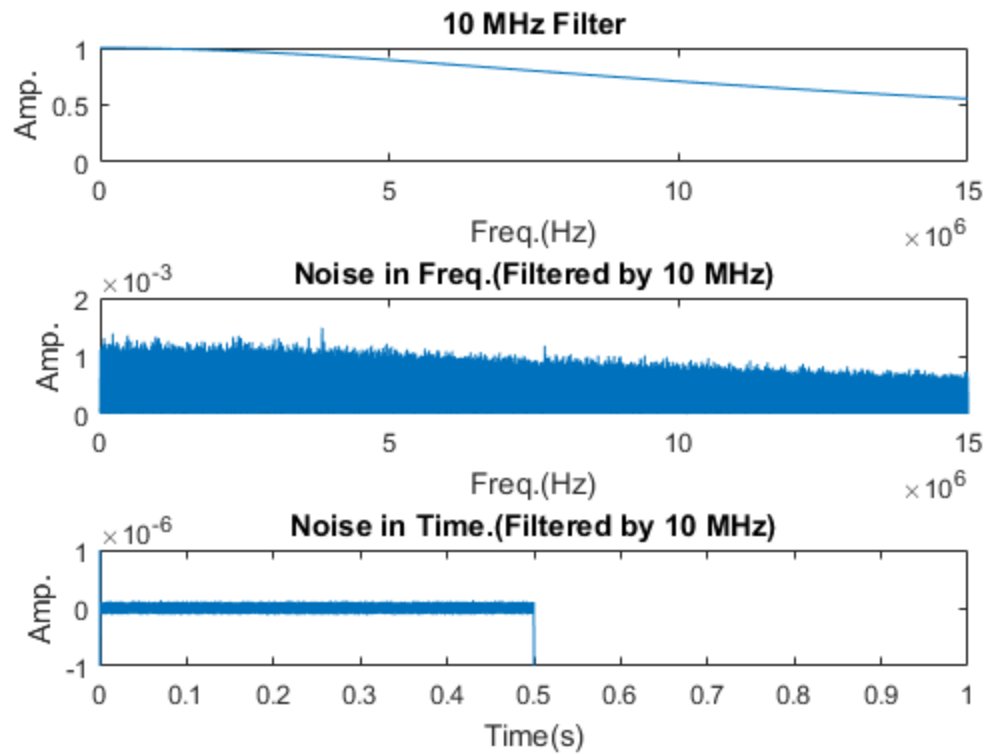
*Warning: Imaginary parts of complex X and/or Y arguments ignored*

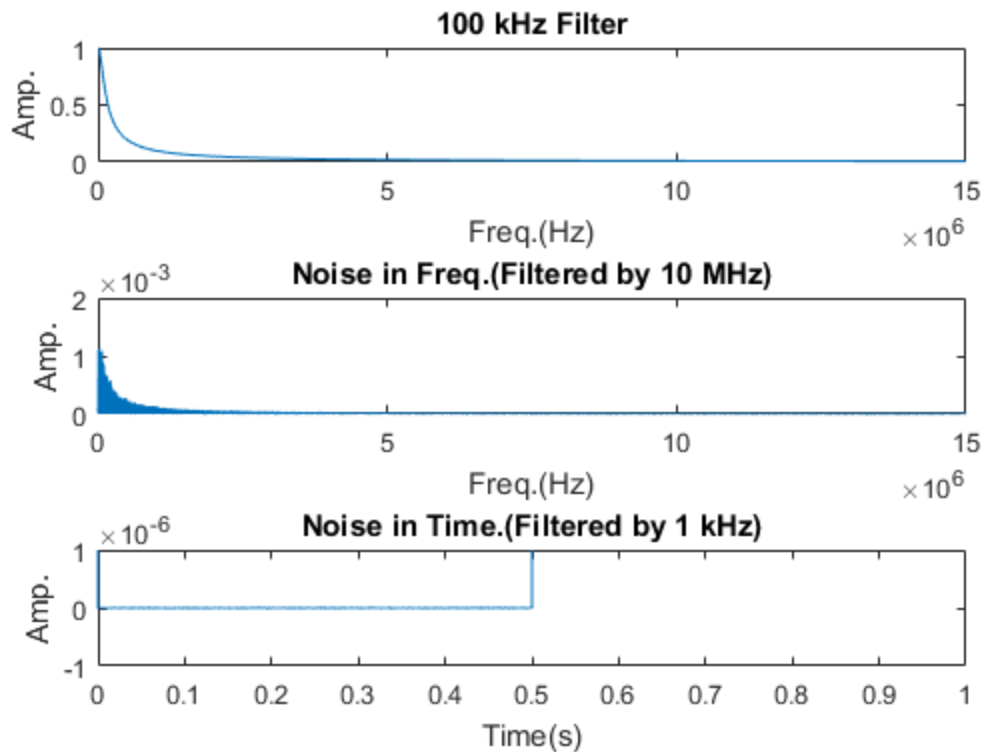
*Warning: Imaginary parts of complex X and/or Y arguments ignored*

*Warning: Imaginary parts of complex X and/or Y arguments ignored*

*The low pass filter with the lower cutoff frequency (100 kHz) was better in removing the noise. It is assumed the noise is existent in all frequencies so the 100 kHz filter was able to get rid of more noise.*







## Problem 4

```
clc;
clear all;
close all;

fs=1*10^9;
t=0:1/fs:1e-3;

fc=1*10^6;

figure;
[b1,a1]=butter(1,fc/(fs/2));
title('Butterworth Filter');
freqz(b1,a1);
hold on;
[b2,a2]=butter(3,fc/(fs/2));
freqz(b2,a2);
[b3,a3]=butter(5,fc/(fs/2));
freqz(b3,a3);
lines = findall(gcf,'type','line');
set(lines(1),'color','red');
set(lines(2),'color','green');
set(lines(3),'color','blue');
legend('1st Order', '3rd Order', '5th Order');
```



```
title('Butterworth Filter');
hold off;

%%100 kHz
figure;
subplot(2,1,1);
x1=square(2*pi*100*10^3*t);
plot(t,x1);
title('100 kHz Square Wave');
axis([0 2/(100*10^3) -1.5 1.5]);
subplot(2,1,2);
y1=filter(b1,a1,x1);
y2=filter(b2,a2,x1);
y3=filter(b3,a3,x1);
plot(t,y1);
hold on;
plot(t,y2);
plot(t,y3);
hold off;
legend('1st Order', '3rd Order', '5th Order');
axis([0 2/(100*10^3) -1.5 1.5]);

%%900 kHz
figure;
subplot(2,1,1);
x1=square(2*pi*900*10^3*t);
plot(t,x1);
title('900 kHz Square Wave');
axis([0 2/(900*10^3) -1.5 1.5]);
subplot(2,1,2);
y1=filter(b1,a1,x1);
y2=filter(b2,a2,x1);
y3=filter(b3,a3,x1);
plot(t,y1);
hold on;
plot(t,y2);
plot(t,y3);
hold off;
legend('1st Order', '3rd Order', '5th Order');
axis([0 2/(900*10^3) -1.5 1.5]);

%%1 MHz
figure;
subplot(2,1,1);
x1=square(2*pi*1*10^6*t);
plot(t,x1);
title('1 MHz Square Wave');
axis([0 2/(1*10^6) -1.5 1.5]);
subplot(2,1,2);
y1=filter(b1,a1,x1);
y2=filter(b2,a2,x1);
y3=filter(b3,a3,x1);
plot(t,y1);
```

```
hold on;
plot(t,y2);
plot(t,y3);
hold off;
legend('1st Order', '3rd Order', '5th Order');
axis([0 2/(1*10^6) -1.5 1.5]);

%%1.1 MHz
figure;
subplot(2,1,1);
x1=square(2*pi*1.1*10^6*t);
plot(t,x1);
title('1.1 MHz Square Wave');
axis([0 2/(1.1*10^6) -1.5 1.5]);
subplot(2,1,2);
y1=filter(b1,a1,x1);
y2=filter(b2,a2,x1);
y3=filter(b3,a3,x1);
plot(t,y1);
hold on;
plot(t,y2);
plot(t,y3);
hold off;
legend('1st Order', '3rd Order', '5th Order');
axis([0 2/(1.1*10^6) -1.5 1.5]);

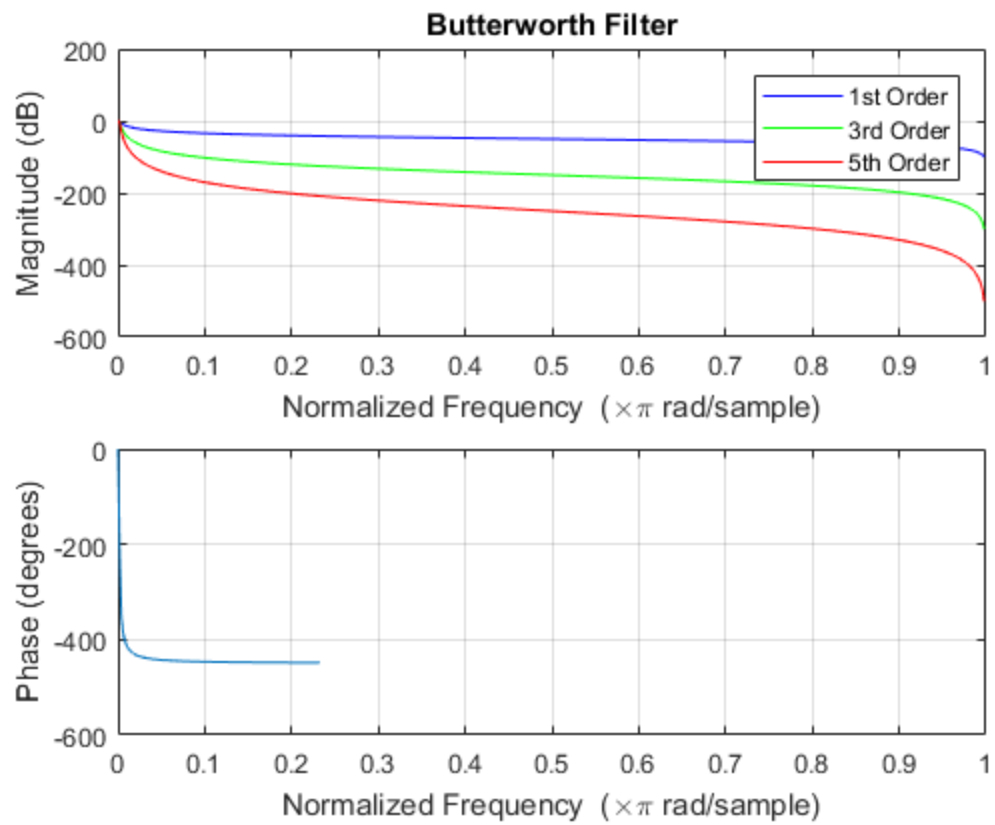
%%10 MHz

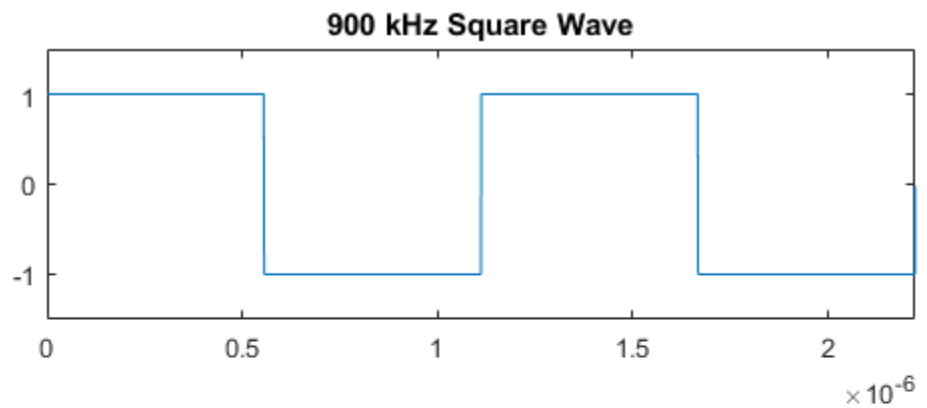
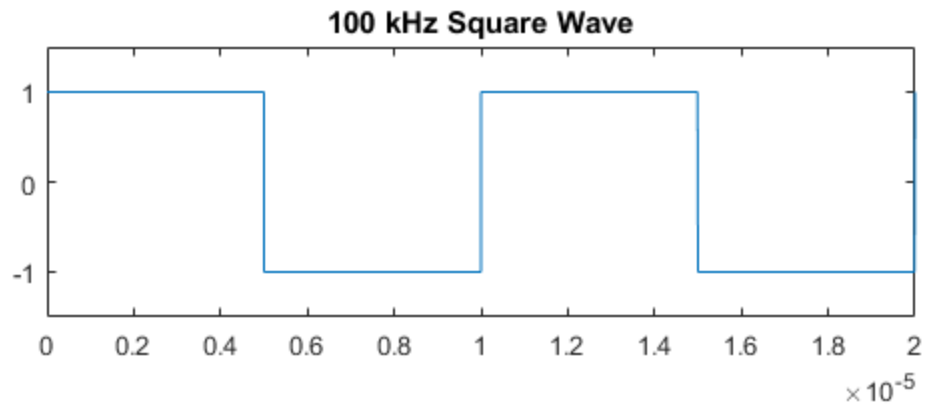
figure;
subplot(2,1,1);
x1=square(2*pi*10*10^6*t);
plot(t,x1);
title('10 MHz Square Wave');
axis([0 2/(10*10^6) -1.5 1.5]);
subplot(2,1,2);
y1=filter(b1,a1,x1);
y2=filter(b2,a2,x1);
y3=filter(b3,a3,x1);
plot(t,y1);
hold on;
plot(t,y2);
plot(t,y3);
hold off;
legend('1st Order', '3rd Order', '5th Order');
axis([0 2/(10*10^6) -1.5 1.5]);

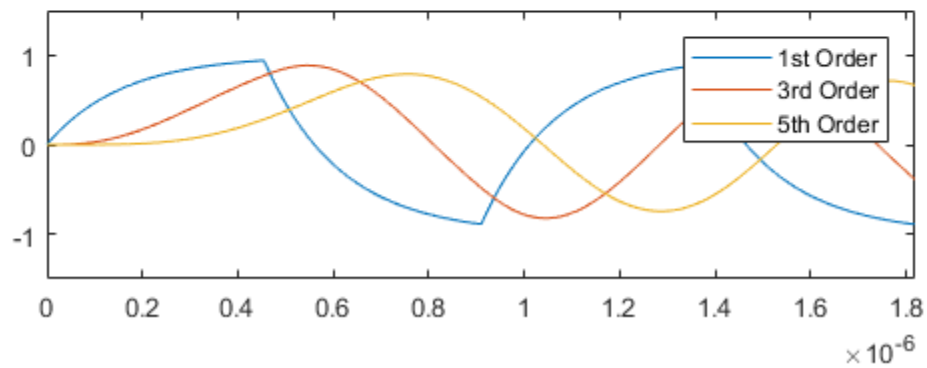
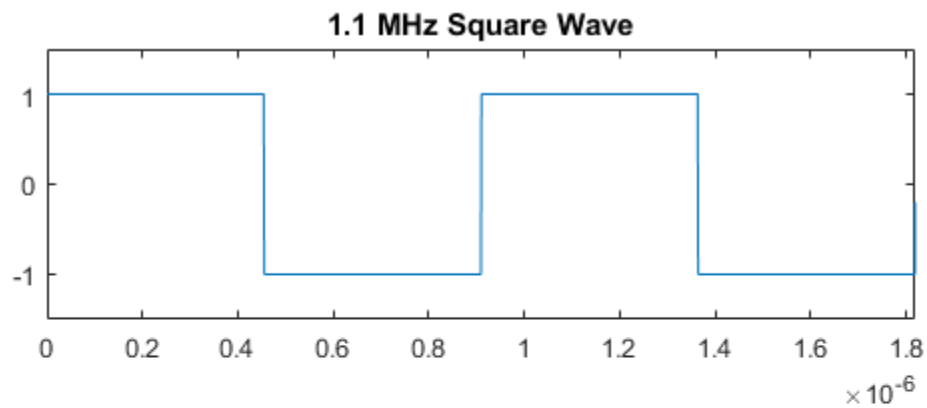
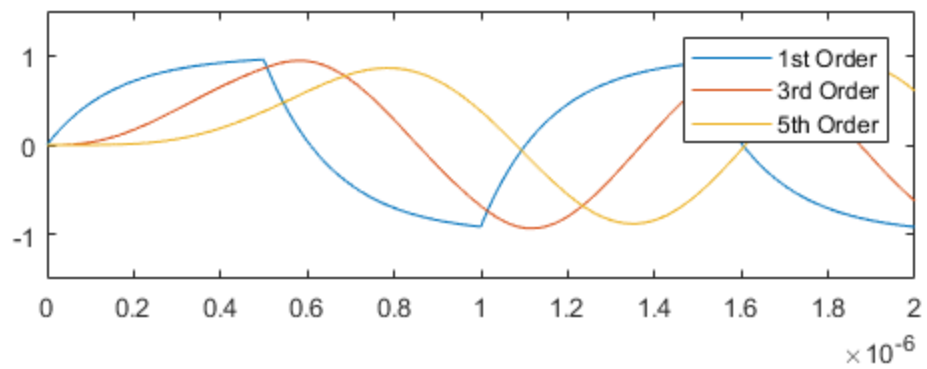
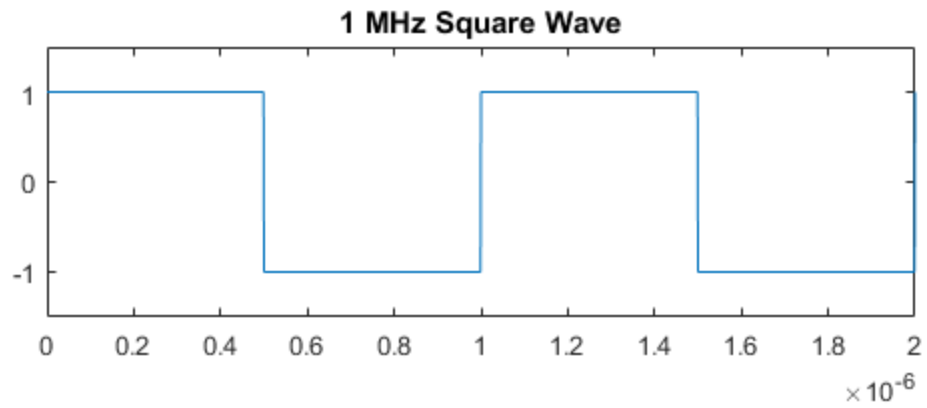
%The first order filter doesn not have overshoot compared to the other
two.
%The first order has less rise time.

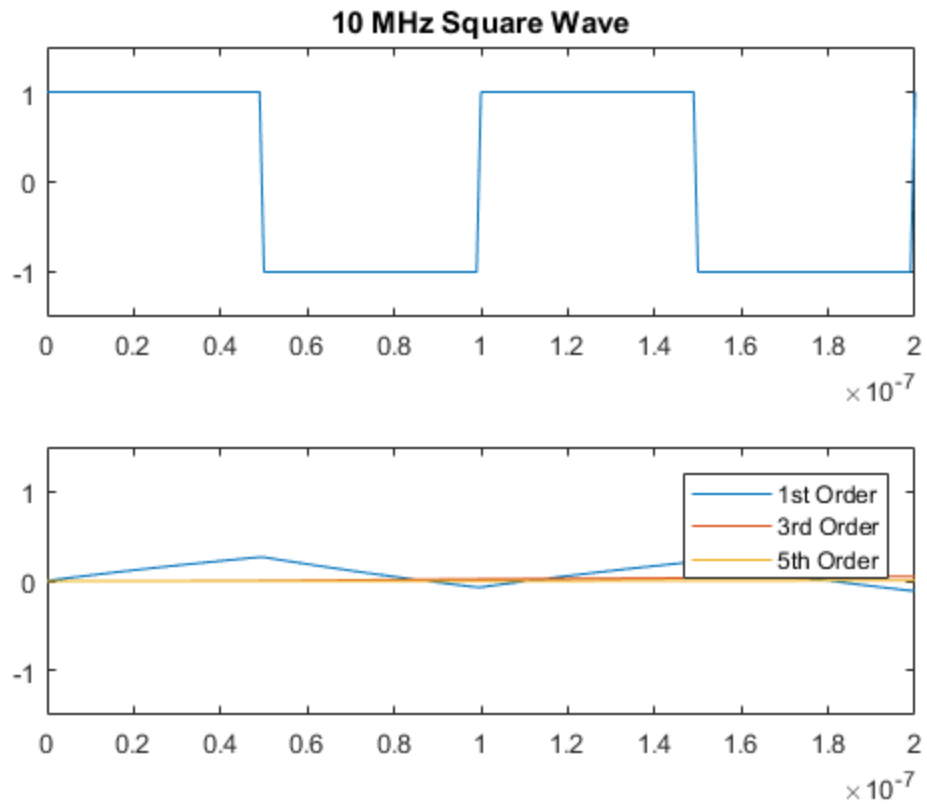
disp('%The first order filter doesn not have overshoot compared to the
other two. The first order has less rise time. ');
```

*%The first order filter doesn't have overshoot compared to the other two. The first order has less rise time.*









*Published with MATLAB® R2017a*