Noah Choe

179008726

LAB 2

Problem 1

The purpose of problem 1 is to create a finite-impulse response filter. In part 1.1, the filter was designed so that only the middle component will pass through the filter, and the output had a slight delay. We were also told to find the three unknown filter coefficients for a system of equations. This was done by using matrix division to find the values. In part 1.2, we then created a plot for the magnitude response and added the frequency points w1, w2, and w3 to the graph. In part 1.3, we created a plot of the input and output, and we also plotted s(n) which was similar to the output signal. We also created a table that compared n with s(n), s(n-2), y(n), v(n), and y_v(n). Lastly, in part 1.4, we compared the FIR filter to the IIR filter. We measured for the noise ratio and saw that the IIR filter had a big decrease compared to the FIR filter.

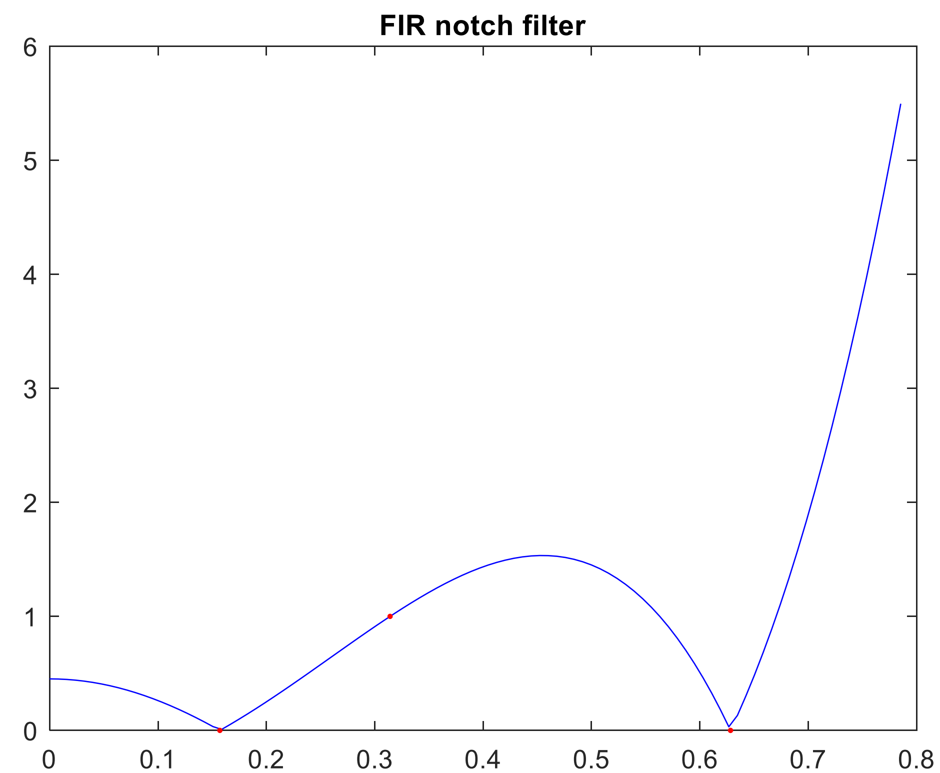
Problem 1.1

```
w1 = 0.05*pi;

w2 = 0.1*pi;

w3 = 0.2*pi;

B = [2*cos(2*w1) 2*cos(w1) 1; 2*cos(2*w2) 2*cos(w2) 1; 2*cos(2*w3) 2*cos(w3) 1];

Y = [0; 1; 0];

b = B\Y;


disp(b);


%output

-48.0477

172.6550

-249.6665
```

Problem 1.2

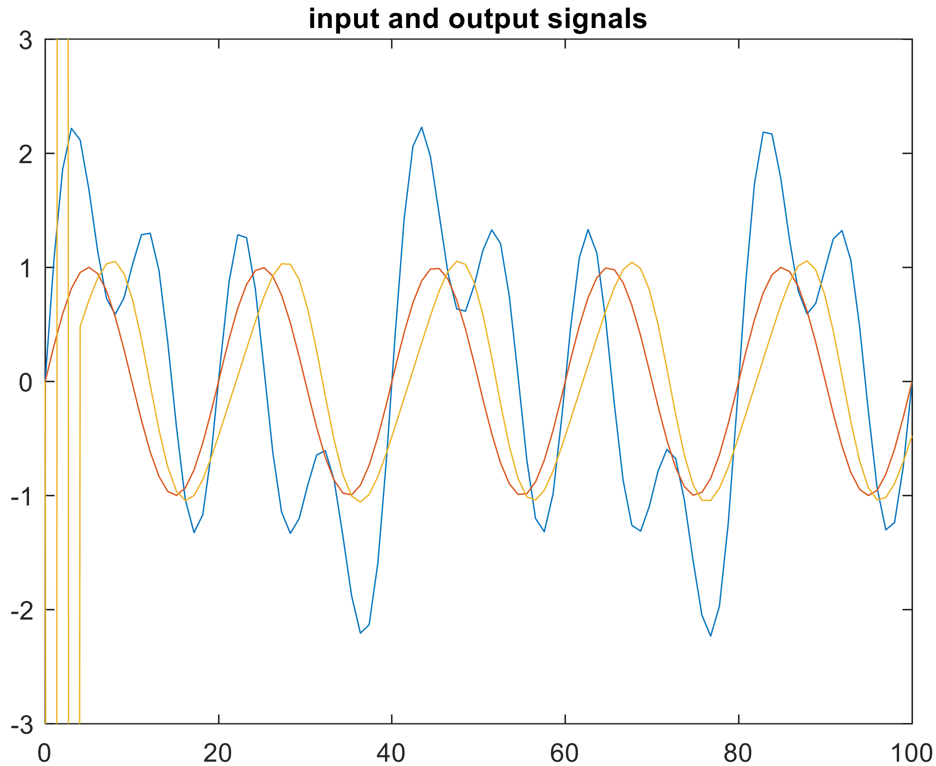
```
B = @(w) 2*b(1)*cos(2*w)+2*b(2)*cos(w)+b(3);

H = @(w) exp(-2*1i*w).*B(w);
```

```
n = linspace(0,0.25*pi);

figure;
plot(n,abs(H(n)),'-b');
hold on;
plot(w1,abs(H(w1)),'.r');
plot(w2,abs(H(w2)),'.r');
plot(w3,abs(H(w3)),'.r');
title("FIR notch filter");
hold off;
```



Problem 1.3

```
n = linspace(0,100);


H = [b(1), b(2), b(3), b(2), b(1)];

s = @(n) sin(w2.*n);

v = @(n) sin(w1.*n) + sin(w3.*n);

x = @(n) s(n) + v(n);

y =@(n) filter(H, 1, x(n));

yv = @(n) filter(H, 1, v(n));

s2 = @(n) sin(w2*(n-2)).*(n>=2);


figure;

plot(n, x(n));

hold on;

plot(n, s(n));

plot(n, y(n));

xlim([0, 100]);

ylim([-3, 3]);

title('input and output signals');

hold off;


n_table = 0:9;

fprintf('n     s(n)     s(n-2)     y(n)      v(n)      y_v(n)\n');

fprintf('-------------------------------------------------------\n');

fprintf('%d %9.4f %9.4f %9.4f %9.4f %9.4f\n',[n_table; s(0:9); s2(0:9); y(0:9); v(0:9); yv(0:9)]);
```



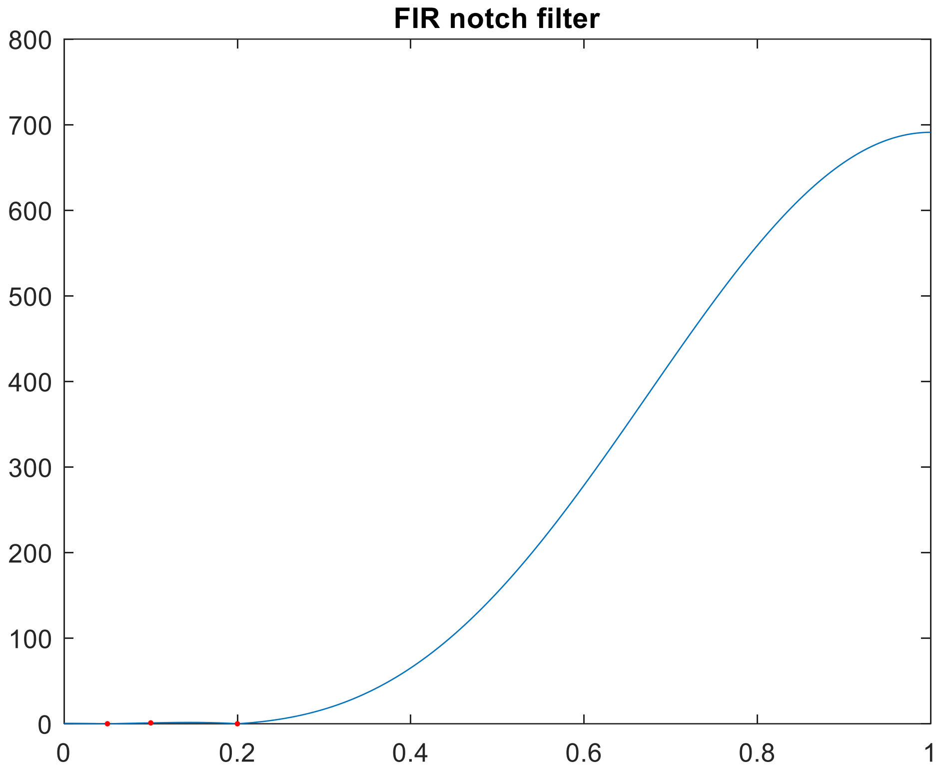
n	s(n)	s(n-2)	y(n)	v(n)	y_v(n)
0	0.0000	-0.0000	0.0000	0.0000	0.0000
1	0.3090	-0.0000	-50.6056	0.7442	-35.7580
2	0.5878	0.0000	93.0613	1.2601	67.9497
3	0.8090	0.3090	-50.2965	1.4050	-35.7580
4	0.9511	0.5878	0.5878	1.1756	-0.0000
5	1.0000	0.8090	0.8090	0.7071	-0.0000
6	0.9511	0.9511	0.9511	0.2212	-0.0000
7	0.8090	1.0000	1.0000	-0.0600	-0.0000
8	0.5878	0.9511	0.9511	-0.0000	-0.0000
9	0.3090	0.8090	0.8090	0.3999	0.0000

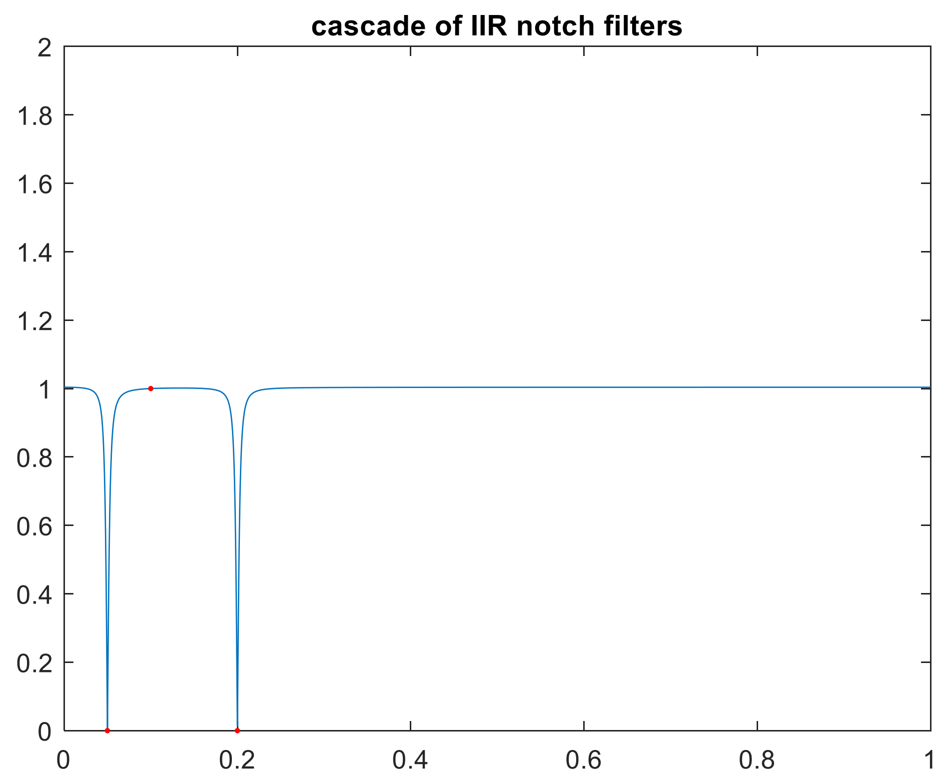
Problem 1.4

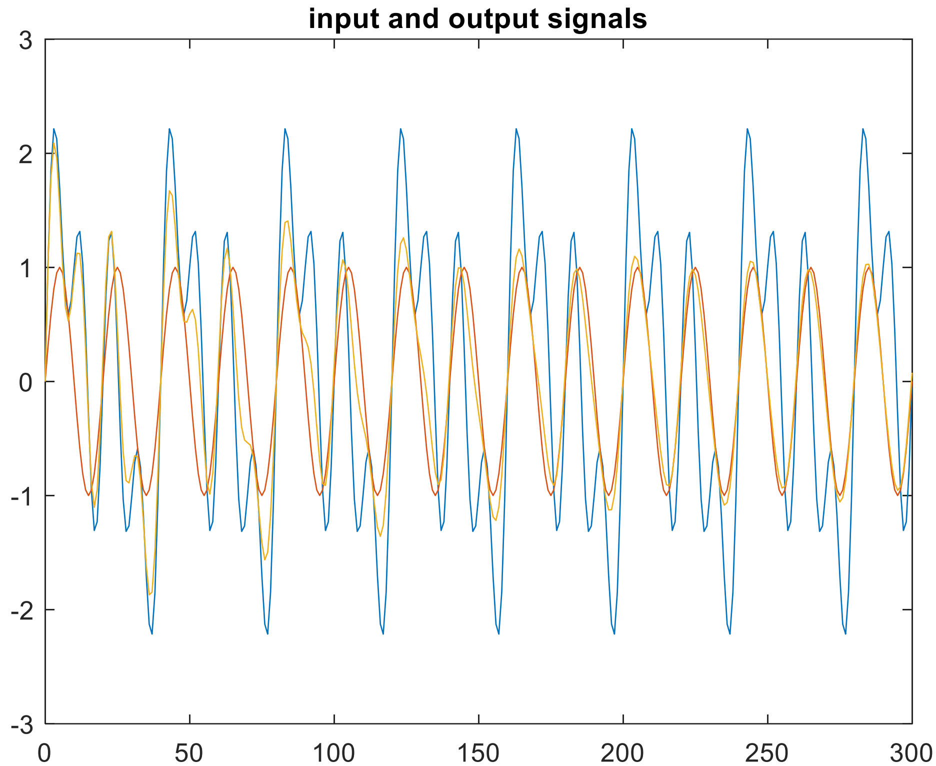
```
H = @(w) exp(-2*1i*w).*B(w);
n = linspace(0, pi, 5000);

figure;
plot(n/pi, abs(H(n)));
hold on;
plot(w1/pi, abs(H(w1)), '.r');
plot(w2/pi, abs(H(w2)), '.r');
plot(w3/pi, abs(H(w3)), '.r');
xlim([0, 1]);
ylim([0, 800]);
title('FIR notch filter');
hold off;

for i=1:length(H)
   sum = sum - H(i).^2;
end
noise = sqrt(sum);
display(noise);

b = [.984011, -3.535954, 5.113142, -3.535954, 0.984011];
a = [1, -3.557832, 5.093644, -3.487380, 0.960788];
Hmag = abs(freqz(b, a, n));

figure;
plot(n/pi, Hmag);
hold on;
```

```
plot(w1/pi, abs(H(w1)), '.r');

plot(w2/pi, abs(H(w2)), '.r');

plot(w3/pi, abs(H(w3)), '.r');

xlim([0, 1]);

ylim([0, 2]);

title('cascade of IIR notch filters');

hold off;


n = 0:300;

H = impz(b, a, 301);

y =@(n) filter(H, 1, x(n));


figure;

plot(n, x(n));

hold on;

plot(n, s(n));

plot(n, y(n));

xlim([0, 300]);

ylim([-3, 3]);

title('input and output signals');

hold off;


n = 0:600;

H = impz(b, a, 601);

for i=1:length(H)

   sum = sum - H(i).^2;

end

noise = sqrt(sum);
```

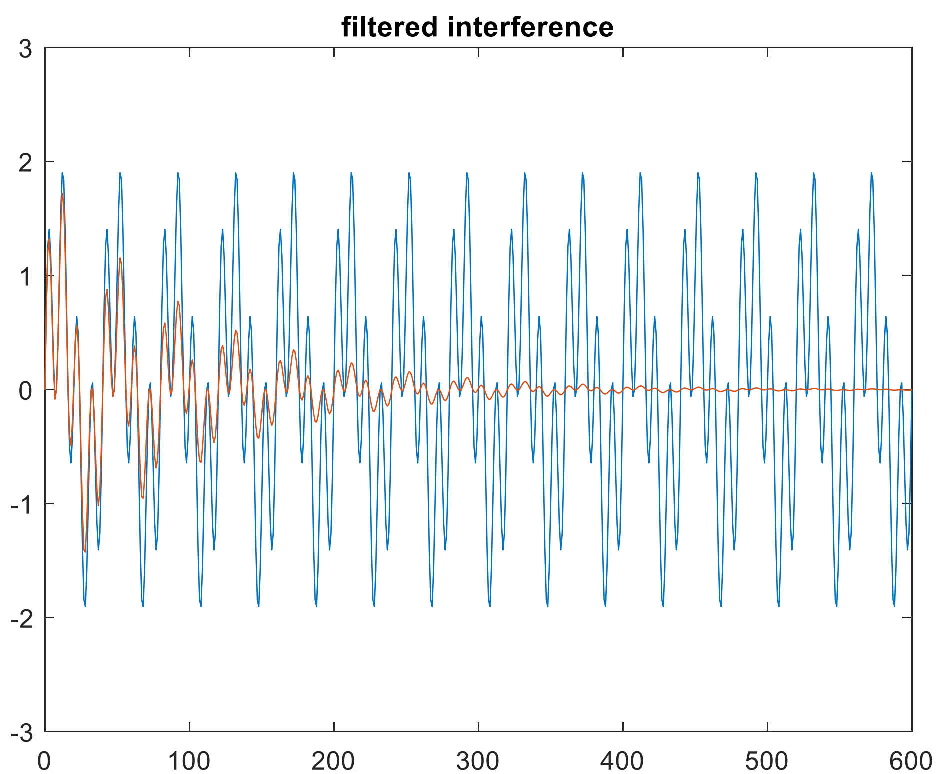
```
display(noise);

yv = @(n) filter(H, 1, v(n));

figure;
plot(n, v(n));
hold on;
plot(n, yv(n));
xlim([0, 600]);
ylim([-3, 3]);
title('filtered interference');
hold off;

n40 = log(0.01)/log(max(abs(roots(a))));
display(n40);
%output
noise =
  21.0093 - 9.6151i
noise =
  20.9899 - 9.6240i
n40 =
  460.5267
```


FIR notch filter
800
700
600
500
400
300
200
100
0
0
0.2
0.4
0.6
0.8
1

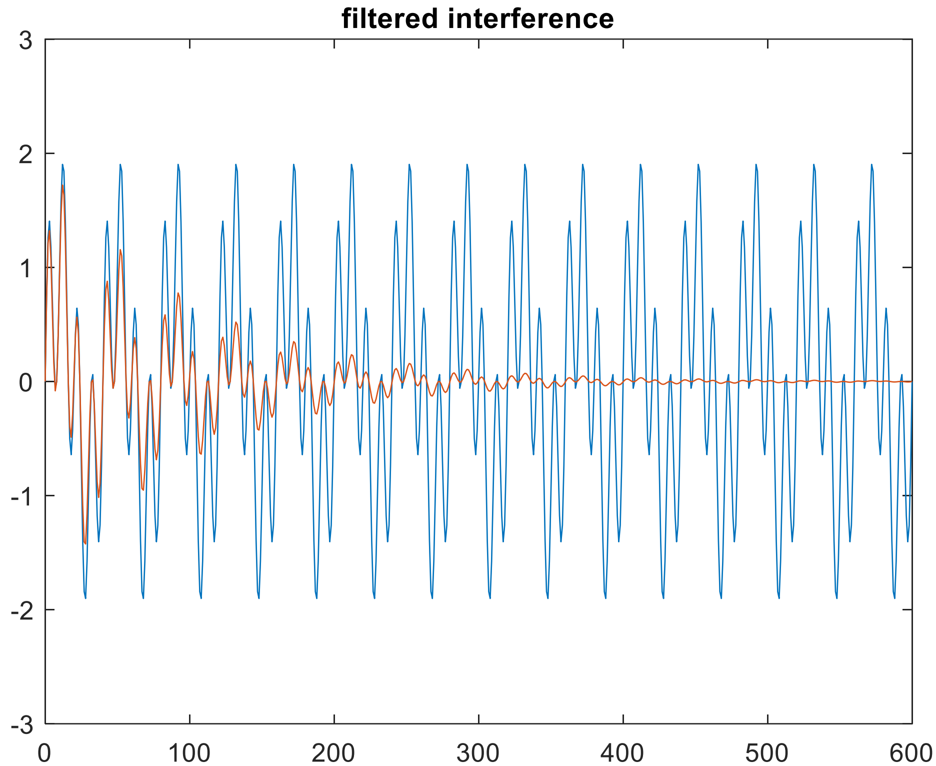

cascade of IIR notch filters
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1



input and output signals
3
2
1
0
-1
-2
-3
0
50
100
150
200
250
300



filtered interference
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600


filtered interference
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600

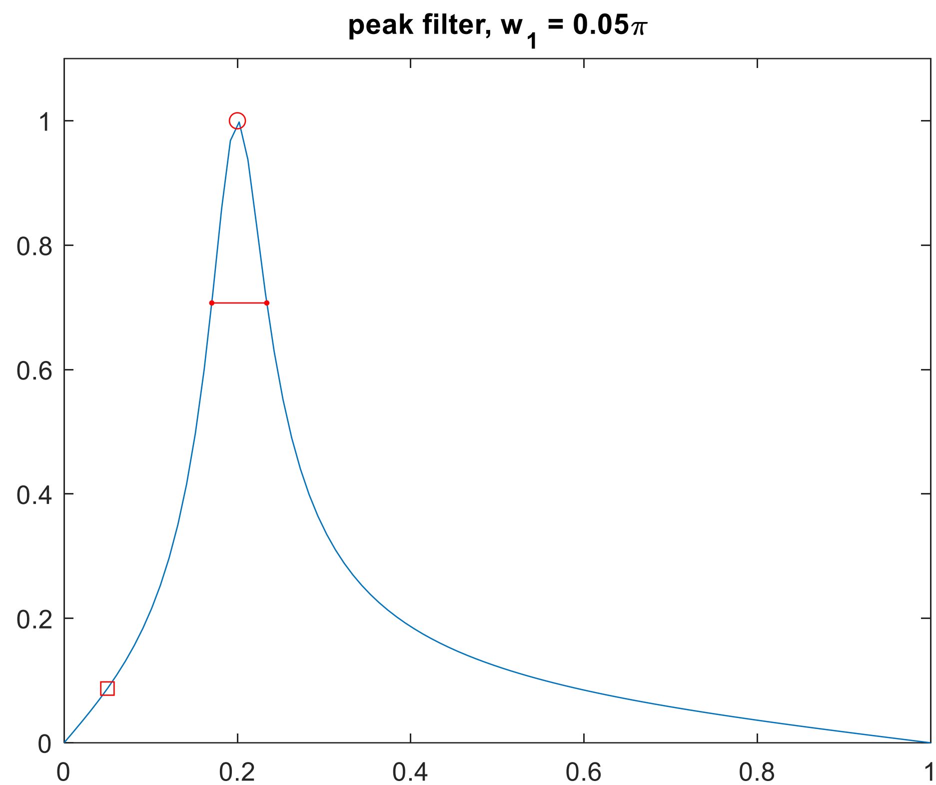
Problem 2

In part 2.1, we had to plot the magnitude response of the filter with peak and side frequency points w1 and w2. On the graph, we see a peak at .2pi with an output of 1 telling us that the frequency fully passes through the filter at that point. For this part, we also had to calculate the left and right 3-db frequencies and place them on the graph too. In part 2.2, we had to plot the phase delay of the filter. There is another peak at %.2pi which tells us that at that point, frequencies fully passes through the filter with no changes. In part 2.3, we see how the phase affects the input, there is a delay in the output signal compared to the input signal. In part 2.4, we changed the value of w1 to .3pi, and repeated parts 2.1-2.3 for this part. We saw a time delay and a shift due to this compared to before.

Problem 2.1

```
w0 = 0.2*pi;

B = 0.1;

w1 = 0.05*pi;

n = linspace(0, pi);


H = @(w) 1i*B.*sin(w)./(cos(w)-cos(w0)+1i*B.*sin(w));


left = acos((cos(w0)+B*sqrt(B^2+(sin(w0))^2))/(1+B^2));

right = acos((cos(w0)-B*sqrt(B^2+(sin(w0)^2)))/(1+B^2));

w3dB = [left right];


figure;

plot(n/pi,abs(H(n)));

hold on;

plot(w0/pi,abs(H(w0)), 'ro');

plot(w1/pi,abs(H(w1)), 'rs');

plot(w3dB/pi,abs(H(w3dB)), 'r.-');

xlim([0, 1]);

ylim([0, 1.1]);

title('peak filter, w_1 = 0.05\pi');
```

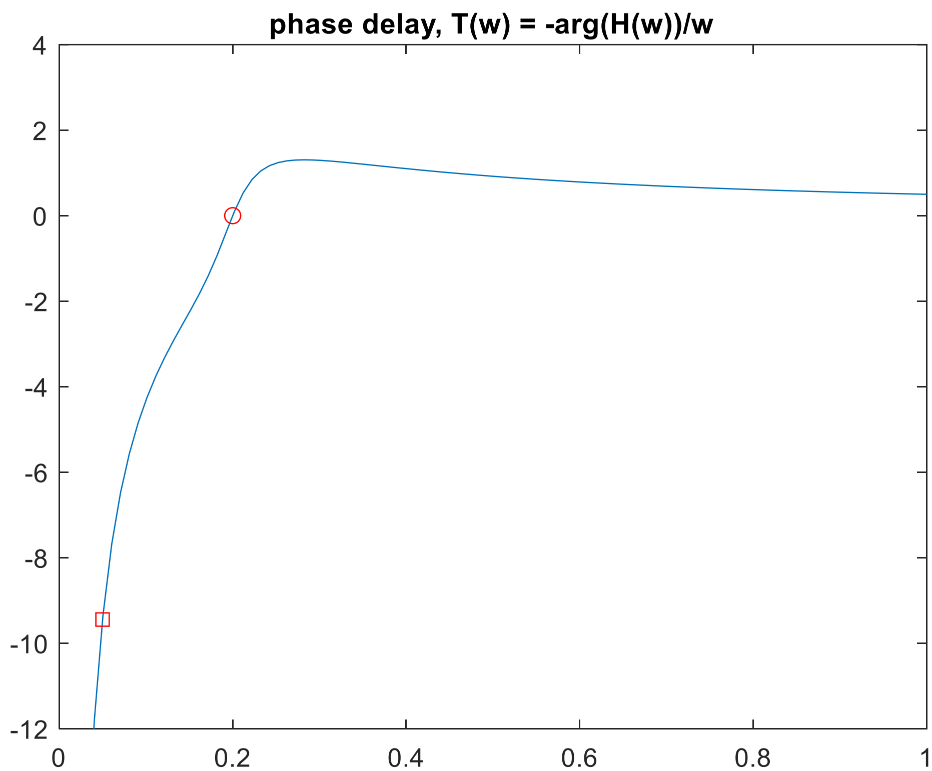
```
hold off;
```



Problem 2.2

```
w0 = 0.2*pi;

B = 0.1;

w1 = 0.05*pi;

n = linspace(0, pi);


T = @(w) -(1./w).*atan((cos(w)-cos(w0))./(B.*sin(w)));


figure;

plot(n/pi,T(n));

hold on;

plot(w0/pi,T(w0), 'ro');

plot(w1/pi,T(w1), 'rs');
```

```
xlim([0, 1]);

ylim([-12, 4]);

title('phase delay, T(w) = -arg(H(w))/w');

hold off;
```

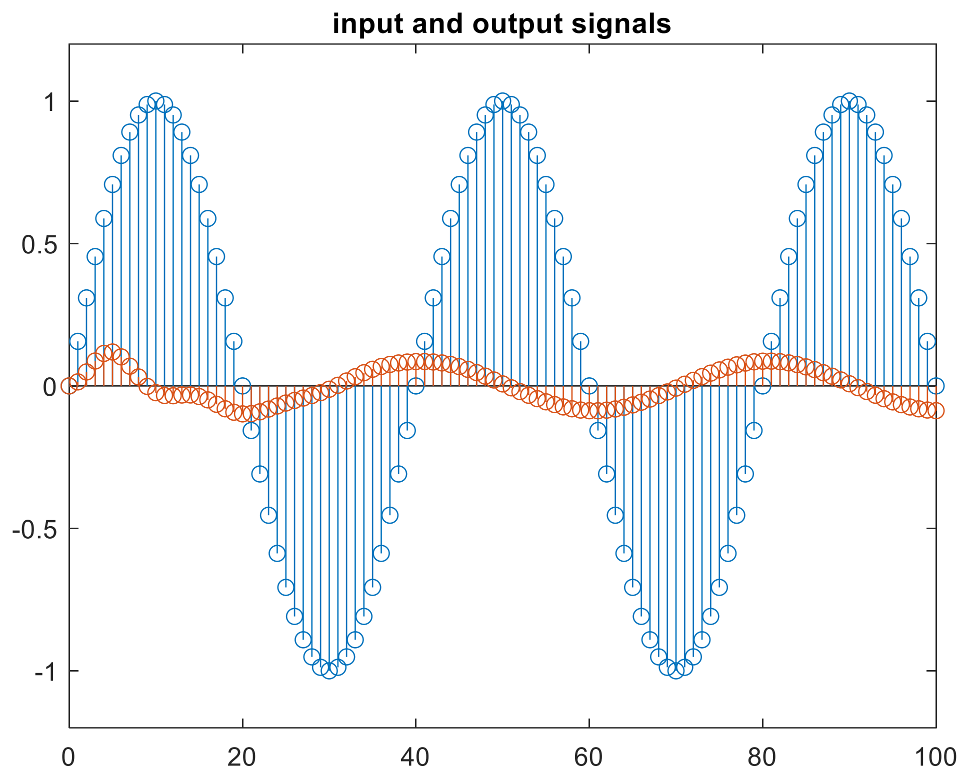


Problem 2.3

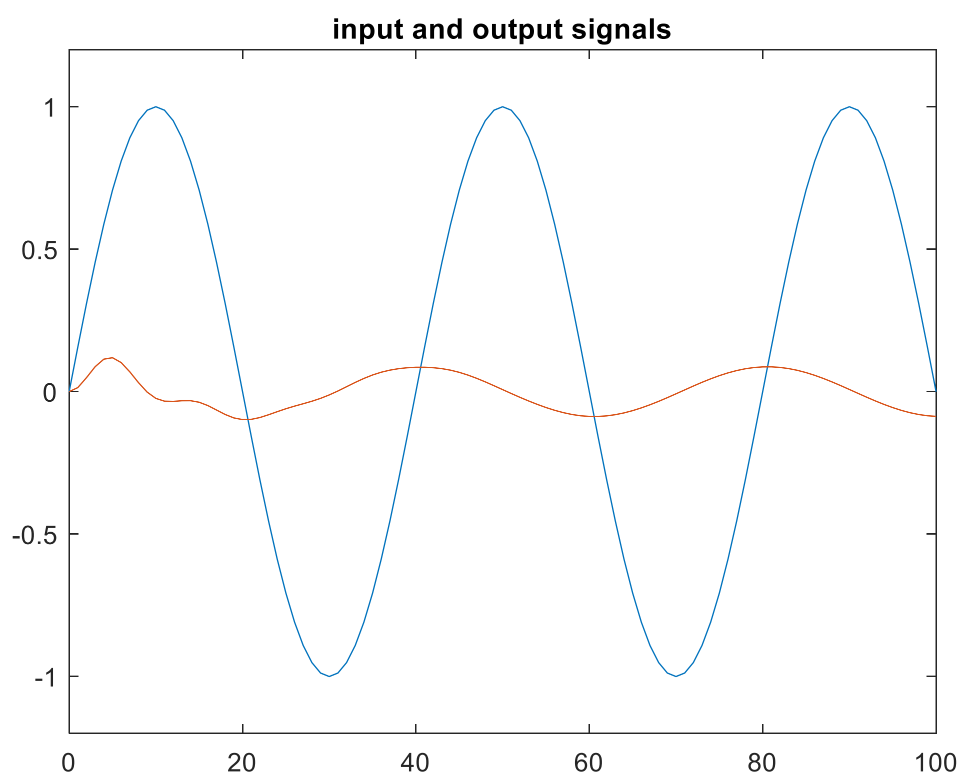
```
w0 = 0.2*pi;

B = 0.1;

w1 = 0.05*pi;

n = 0:100;


x = @(n) sin(w1.*n);

b = (B/(1+B)).*[1,0,-1];

a = [1,-2*cos(w0)/(1+B),(1-B)/(1+B)];

y = filter(b, a, x(n));
```

```
figure;
stem(n, x(n));
hold on;
stem(n, y);
xlim([0, 100]);
ylim([-1.2, 1.2]);
title('input and output signals');
hold off;

figure;
plot(n,x(n));
hold on;
plot(n, y);
xlim([0, 100]);
ylim([-1.2, 1.2]);
title('input and output signals');
hold off;

display(T(w1));
display(abs(T(w1)));
%output
-9.4440
 9.4440
```

This seems correct as we see on the graph that it seems there is around a 10 unit shift.



input and output signals
1
0.5
0
-0.5
-1
0
20
40
60
80
100



input and output signals
1
0.5
0
-0.5
-1
0
20
40
60
80
100

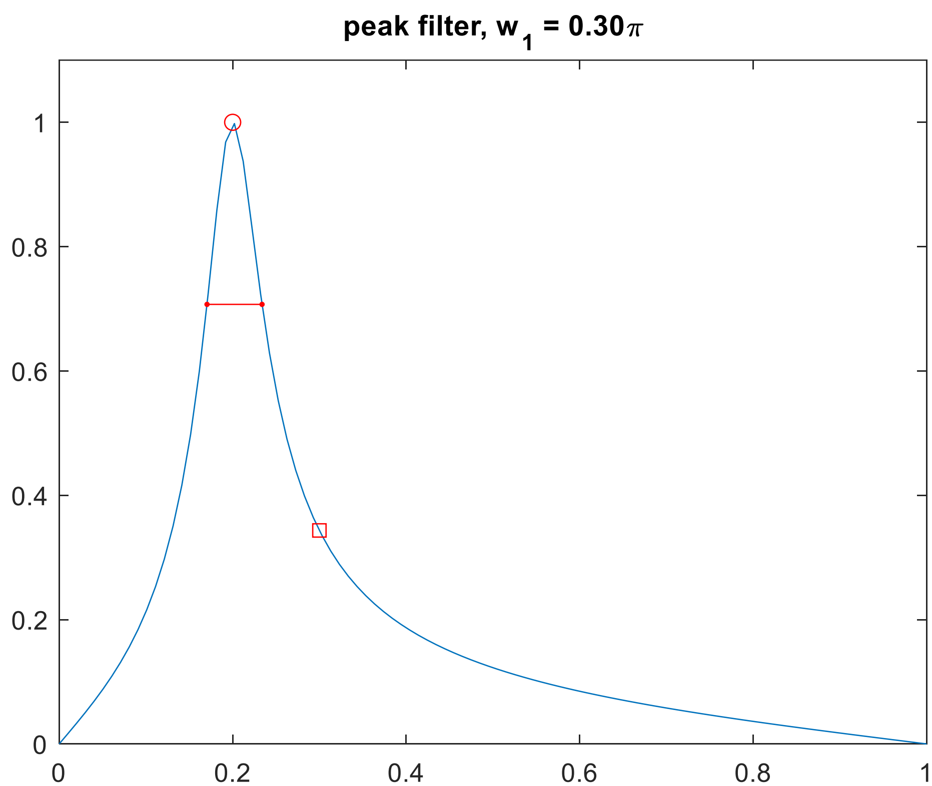
Problem 2.4

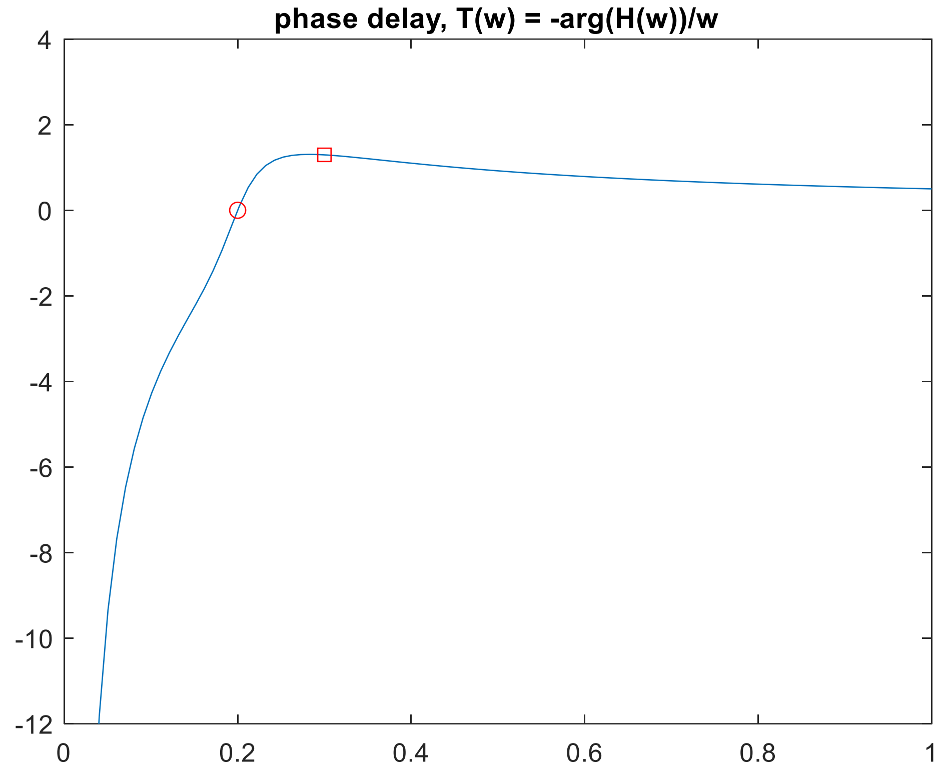
```
w0 = 0.2*pi;

B = 0.1;

w1 = 0.3*pi;

n = linspace(0, pi);


H = @(w) 1i*B.*sin(w)./(cos(w)-cos(w0)+1i*B.*sin(w));


left = acos((cos(w0)+B*sqrt(B^2+(sin(w0))^2))/(1+B^2));

right = acos((cos(w0)-B*sqrt(B^2+(sin(w0)^2)))/(1+B^2));

w3dB = [left right];


figure;

plot(n/pi, abs(H(n)));

hold on;

plot(w0/pi, abs(H(w0)), 'ro');

plot(w1/pi, abs(H(w1)), 'rs');

plot(w3dB/pi, abs(H(w3dB)), 'r.-');

xlim([0, 1]);

ylim([0, 1.1]);

title('peak filter, w_1 = 0.30\pi');

hold off;


T = @(w) -(1./w).*atan((cos(w)-cos(w0))./(B.*sin(w)));


figure;

plot(n/pi, T(n));

hold on;
```

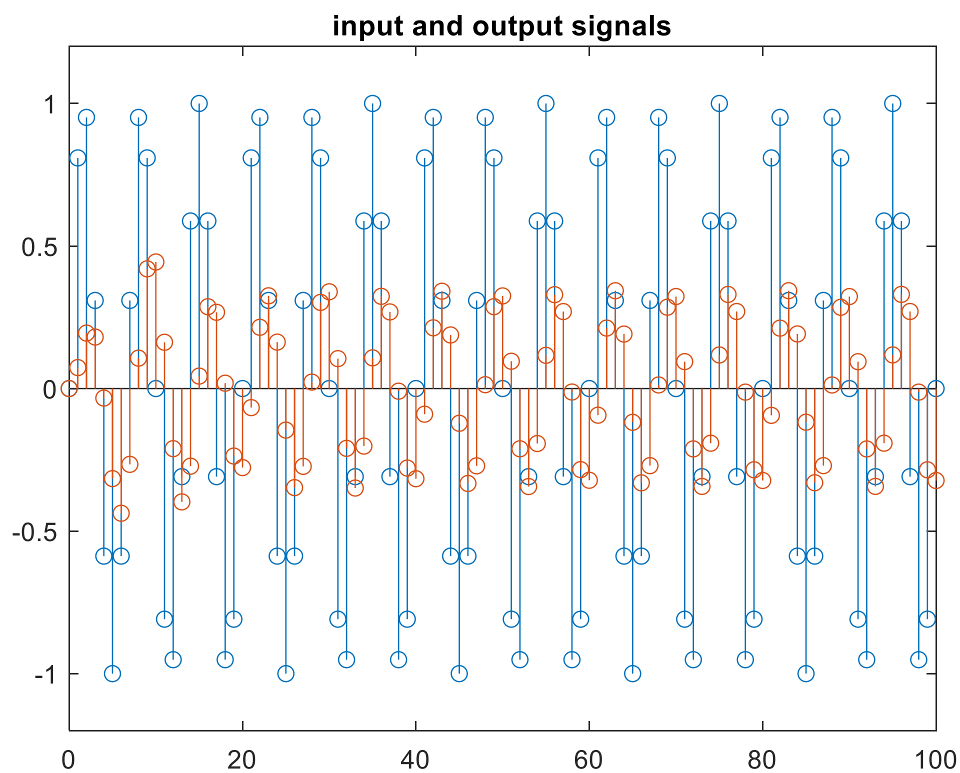
```
plot(w0/pi, T(w0), 'ro');

plot(w1/pi, T(w1), 'rs');

xlim([0, 1]);

ylim([-12, 4]);

title('phase delay, T(w) = -arg(H(w))/w');

hold off;


n = 0:100;


x = @(n) sin(w1.*n);

b = (B/(1+B)).*[1,0,-1];

a = [1,-2*cos(w0)/(1+B),(1-B)/(1+B)];

y = filter(b,a,x(n));


figure;

stem(n, x(n));

hold on;

stem(n, y);

xlim([0, 100]);

ylim([-1.2, 1.2]);

title('input and output signals');

hold off;


figure;

plot(n, x(n));

hold on;

plot(n, y);

xlim([0, 100]);
```

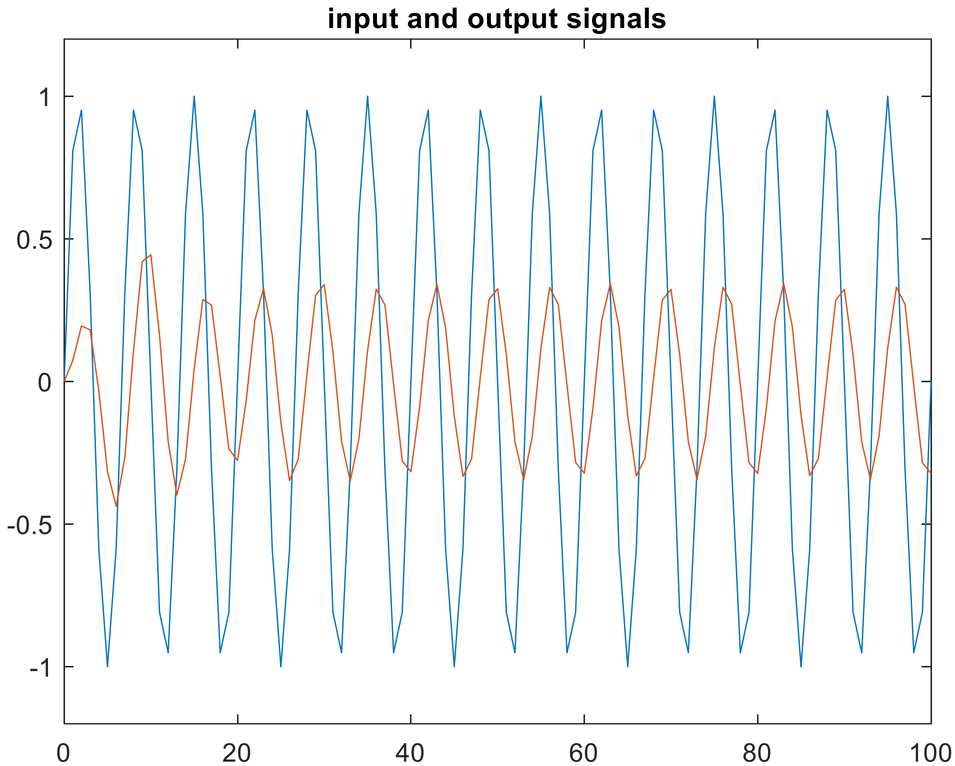
```
ylim([-1.2, 1.2]);

title('input and output signals');

hold off;


display(T(w1));

display(abs(T(w1)));

%output

1.2947

1.2947
```

This seems correct as we see on the graph that it seems there is around a 1 unit shift.




phase delay, T(w) = -arg(H(w))/w
4
2
0
-2
-4
-6
-8
-10
-12
0
0.2
0.4
0.6
0.8
1


input and output signals
1
0.5
0
-0.5
-1
0
20
40
60
80
100



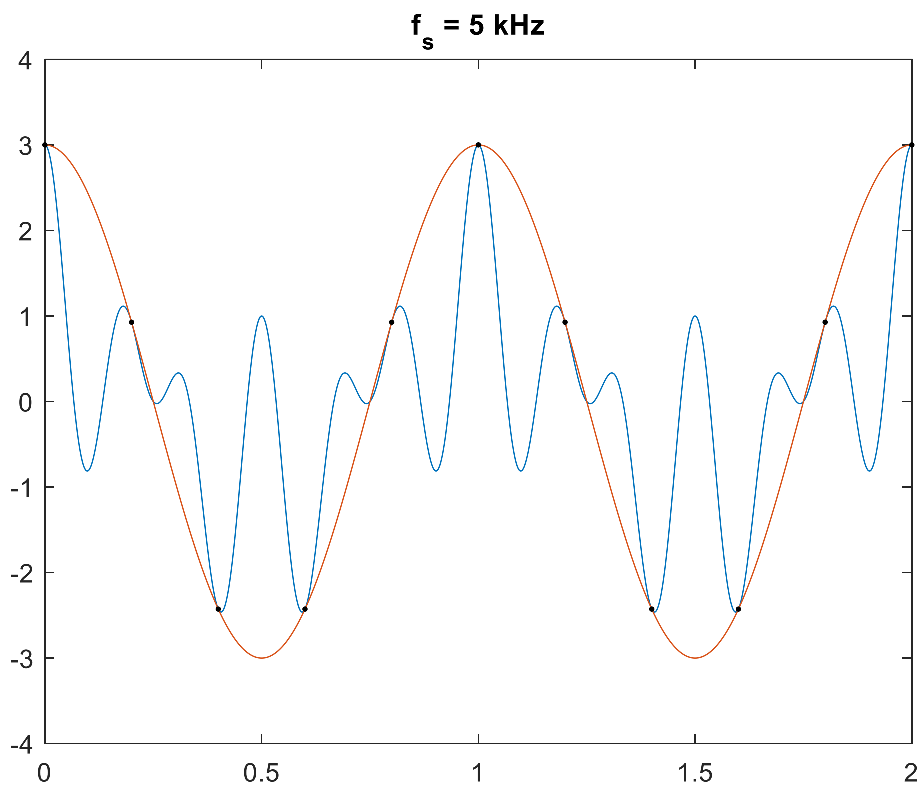
Problem 3

In problem3, we demonstrated the effects of aliasing from improper sampling. We did this with 2 different frequencies of 5KHz and 10KHz. We were able to see that the 10KHz had less aliasing than the 5KHz sampled signal.

Problem 3.1

```
T = 0:.2:2;

n = linspace(0, 2, 1000);


x = @(t) cos(2*pi*t) + cos(8*pi*t) + cos(12*pi*t);

xa = @ (t) 3 * cos(2*pi*t);


figure;

plot(n, x(n));

hold on;
```

```
plot(n, xa(n));

plot(T,x(T),'k.');

xlim([0, 2]);

ylim([-4, 4]);

title('f_s = 5 kHz')

hold off;
```



Problem 3.2

```
T = 0:.1:2;

n = linspace(0, 2, 1000);


x = @(t) cos(2*pi*t) + cos(8*pi*t) + cos(12*pi*t);

x_a = @ (t) cos(2*pi*t) + 2 * cos(8*pi*t);


figure;
```

```
plot(n,x(n));
hold on;
plot(n,x_a(n));
plot(T,x(T),'k.');
xlim([0, 2]);
ylim([-4, 4]);
title('f_s = 10 kHz')
```