Noah Choe

179008726

Lab 4

Problem 1

The purpose of this problem is to find the direct and transposed realization of a second – order IIR filter. We then compare the two realizations with the filter function. We also compare the vector vout for the transposed realization and the filter function. At the end we created a transposed realization table comparing different samples.

Problem 1.1:

direct.m

```
function y = direct(b, a, x)
   y(1) = b(1)*x(1);
   y(2) = b(1)*x(2) + b(2)*x(1) - a(2)*y(1);


   for i = 3:length(x)
      y(i) = b(1)*x(i) + b(2)*x(i-1) + b(3)*x(i-2)-a(2)*y(i-1)-a(3)*y(i-2);
   end
end
```

tran.m

```
function [y, vout] = tran(b, a, x, vin)

    if (nargin < 4)

        vin = [0;0];

    end



    v1(1) = vin(1);

    v2(1) = vin(2);



    for i = 1:length(x)

        y(i) = b(1)*x(i) + v1(i);

        v1(i+1) = b(2)*x(i) - a(2)*y(i) + v2(i);

        v2(i+1) = b(3)*x(i) - a(3)*y(i);

    end



    vout = [v1(length(x)+1); v2(length(x)+1)];

end
```

Problem 1.2:

```
x = [1, 1, 2, 1, 2, 2, 1, 1]';

b = [4, 2.4, -1.6];

a = [1, -0.5, 0.6];

vin = [0;0];


ydirect = direct(b, a, x);

ytran = tran(b, a, x);


[yfilter, voutfilter] = filter(b, a, x);

yfilter = yfilter';


display(ydirect);

display(ytran);

display(yfilter);


[yfilter, voutfilter] = filter(b, a, x);

[ytran, vouttran] = tran(b, a, x);

yfilter = yfilter';


display(vouttran);

display(voutfilter);
```

```
ydirect =

   4.0000   8.4000  10.6000   7.4600   4.5700   9.0090   7.3625   1.4759

ytran =

   4.0000   8.4000  10.6000   7.4600   4.5700   9.0090   7.3625   1.4759

yfilter =

   4.0000   8.4000  10.6000   7.4600   4.5700   9.0090   7.3625   1.4758

vouttran =

  -2.8796

  -2.4855

voutfilter =

  -2.8796

  -2.4855
```

We see that all outputs are similar if not the same.

<u>Problem 1.3:</u>

```
fprintf(' n    x       y       v1        v2\n')

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


for i = 1:length(x)

   [ytran, vouttran] = tran(b, a, x(i), vin);

   fprintf('%2i %4i %9.4f %9.4f %9.4f \n',  i-1, x(i), ytran(1), vin(1), vin(2));

   vin = vouttran;

end
```

```
fprintf('%2i %4s %7s   %9.4f %9.4f \n', 8, '-', '-', vin(1), vin(2));
```

```
 n   x      y      v1      v2
--------------------------------------
 0   1    4.0000   0.0000   0.0000
 1   1    8.4000   4.4000  -4.0000
 2   2   10.6000   2.6000  -6.6400
 3   1    7.4600   3.4600  -9.5600
 4   2    4.5700  -3.4300  -6.0760
 5   2    9.0090   1.0090  -5.9420
 6   1    7.3625   3.3625  -8.6054
 7   1    1.4759  -2.5241  -6.0175
 8   -        -  -2.8796  -2.4855
```

Problem 2

The purpose of this problem is to design two different notch filters and see how the R values changed the results. We compared the values between the two filters and also the graphs between the two filters. Through the values we saw little difference between the two, however, the graphs seemed to vary more so.

Problem 2.1 and 2.2:

```
f = linspace(0,10,1001);

f0 = 4;

fs = 200;

w0 = (2*pi*f0)/fs;

w = (2*pi*f)/fs;

R1 = 0.980;

R2 = 0.995;


G1 = (1 - 2*R1*cos(w0) + R1^2)/(1 - 2*cos(w0) + 1);

G2 = (1 - 2*R2*cos(w0) + R2^2)/(1 - 2*cos(w0) + 1);


b1 = G1*[1, -2*cos(w0), 1];

a1 = [1, -2*R1*cos(w0), R1^2];

b2 = G2*[1, -2*cos(w0), 1];

a2 = [1, -2*R2*cos(w0), R2^2];
```

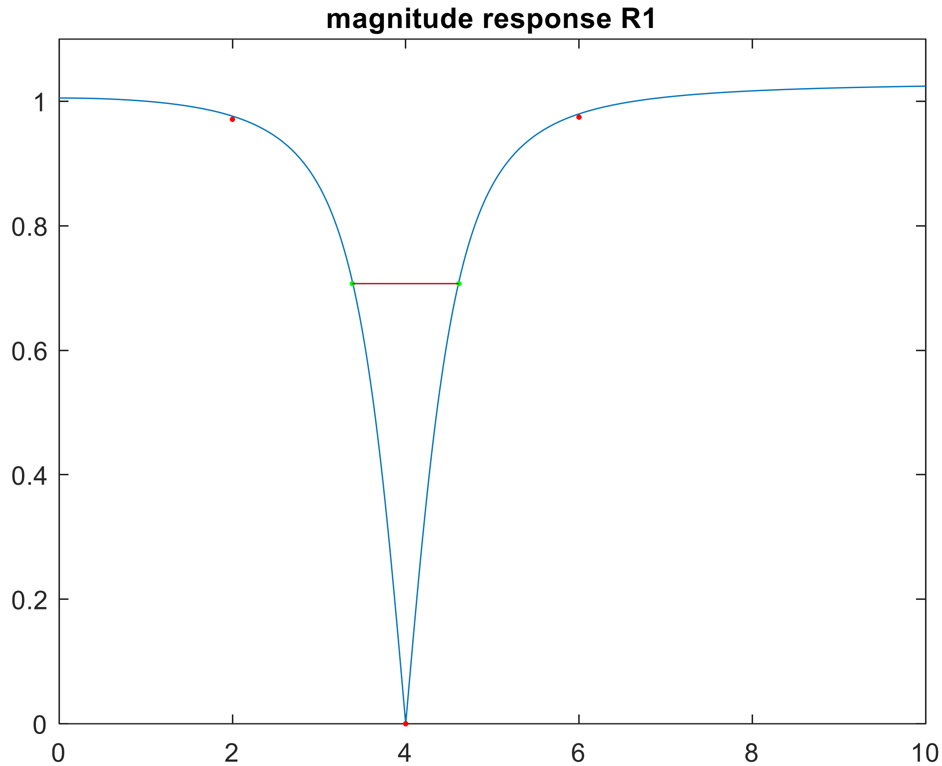
```
fresp = @(b,a,w) polyval(flip(b),exp(-j*w))./polyval(flip(a),exp(-j*w));

mag = @(b,a,w) abs(fresp(b,a,w));


fprintf('filter 1\n');

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

fprintf('b = [ %1.6f %4.6f %4.6f] \n', b1(1), b1(2), b1(3));

fprintf('a = [ %1.6f %4.6f %4.6f] \n\n', a1(1), a1(2), a1(3));


fprintf('filter 2\n');

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

fprintf('b = [ %1.6f %4.6f %4.6f] \n', b2(1), b2(2), b2(3));

fprintf('a = [ %1.6f %4.6f %4.6f] \n', a2(1), a2(2), a2(3));


%plot of filter 1

f1 = 2;

f2 = 6;

fa = fs/pi*(1-R1);

fLa = f0 - 0.5*fa;

fRa = f0 + 0.5*fa;

fL =[fLa+1 fLa-1];

fR =[fRa+1 fRa-1];

F = @(f) mag(b1,a1,2*pi*f/fs) - 1/sqrt(2);
```

```
fLexact = fzero(F, fL);

fRexact = fzero(F, fR);

wleft = fLexact*2*pi/fs;

wright = fRexact*2*pi/fs;


figure;

plot(f, G1*mag(b1, a1, w));

hold on;

plot(fLexact, mag(b1, a1, wleft), 'g.', fRexact, mag(b1, a1, wright), 'g.', f1,mag(b1, a1,
f1*2*pi/fs), 'r.', f2, mag(b1, a1, f2*2*pi/fs),'r.', f0, mag(b1, a1, f0*2*pi/fs), 'r.');

plot([fLexact, fRexact], [mag(b1, a1, wleft), mag(b1, a1, wright)]);

title('magnitude response R1');

xlim([0 10]);

ylim([0 1.1]);

hold off;


error = 100*(abs(fLexact - fLa) + abs(fRexact - fRa))/(fLexact + fRexact);

fprintf('\n\nF1: exact approx\n');

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

fprintf('fL = %.4f %.4f \n', fLexact, fLa);

fprintf('fR = %.4f %.4f \n', fRexact, fRa);

fprintf('------------------\n')
```

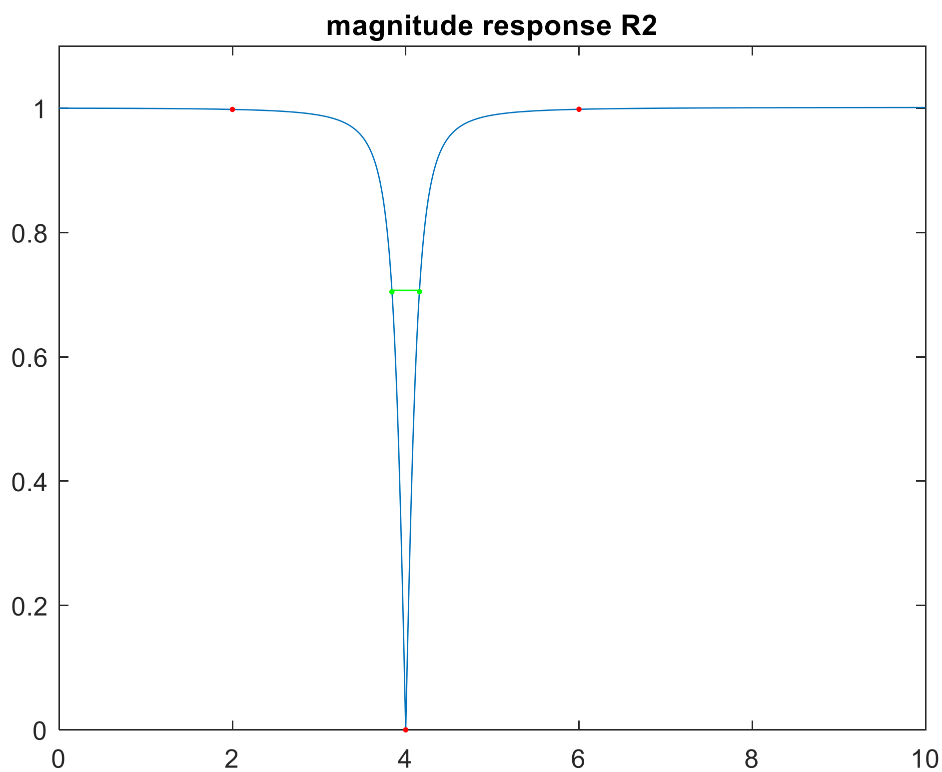
```
fprintf('percent error = %.4f%%\n', error);


fa = (fs/pi)*(1-R2);

fLa = f0 - 0.5*fa;

fRa = f0 + 0.5*fa;

fL = [fLa-0.1 fLa+0.1];

fR = [fRa-0.1 fRa+0.1];

F = @(f) mag(b2,a2,2*pi*f/fs) - 1/sqrt(2);

fLexact = fzero(F,fL);

fRexact = fzero(F,fR);

wleft = fLexact*2*pi/fs;

wright = fRexact*2*pi/fs;


figure;

plot(f, mag(b2,a2,w));

hold on;

plot(fLexact, G2*mag(b2, a2, wleft), 'g.', fRexact, G2*mag(b2, a2, wright), 'g.', f1, mag(b2, a2,
f1*2*pi/fs), 'r.', f2, mag(b2, a2, f2*2*pi/fs), 'r.', f0, mag(b2, a2, f0*2*pi/fs), 'r.');

plot([fLexact, fRexact], [mag(b2,a2,wleft),mag(b2,a2,wright)], 'g-')

title('magnitude response R2');

xlim([0 10]);

ylim([0 1.1]);
```

```
hold off;

error = 100*(abs(fLexact - fLa) + abs(fRexact - fRa))/(fLexact + fRexact);
fprintf('\n\nF2: exact approx\n');
fprintf('------------------\n');
fprintf('fL = %.4f %.4f \n', fLexact, fLa);
fprintf('fR = %.4f %.4f \n', fRexact, fRa);
fprintf('------------------\n');
fprintf('percent error = %.4f%%\n', error);
```

```
filter 1
-----------------------------------
b = [ 1.005364 -1.994872 1.005364]
a = [ 1.000000 -1.944545 0.960400]

filter 2
-----------------------------------
b = [ 0.996585 -1.977454 0.996585]
a = [ 1.000000 -1.974308 0.990025]
```



magnitude response R1
0
0.2
0.4
0.6
0.8
1
0
2
4
6
8
10



magnitude response R2
0
0.2
0.4
0.6
0.8
1
0
2
4
6
8
10

F1: exact approx

----

fL = 3.3837   3.3634

fR = 4.6151   4.6366

----

percent error = 0.5236%

F2: exact   approx

----

fL = 3.8409   3.8408

fR = 4.1591   4.1592

----

percent error = 0.0011%

<u>Problem 2.3:</u>

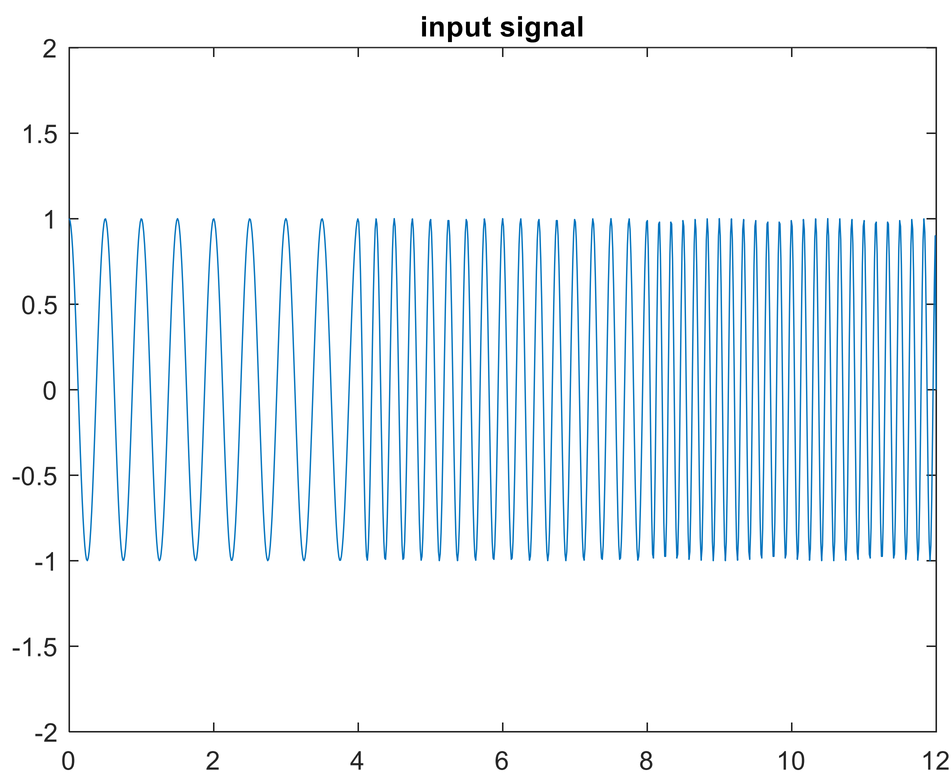
```
f0 = 4;

f1 = 2;

f2 = 6;

fs = 200;

Ts = 1/fs;


x=@(t) cos(2*pi*f1.*t).*(t >= 0 & t < 4) + cos(2*pi*f0.*t).*(t >= 4 & t < 8) + cos(2*pi*f2.*t).*(t >=
8 & t < 12);
```

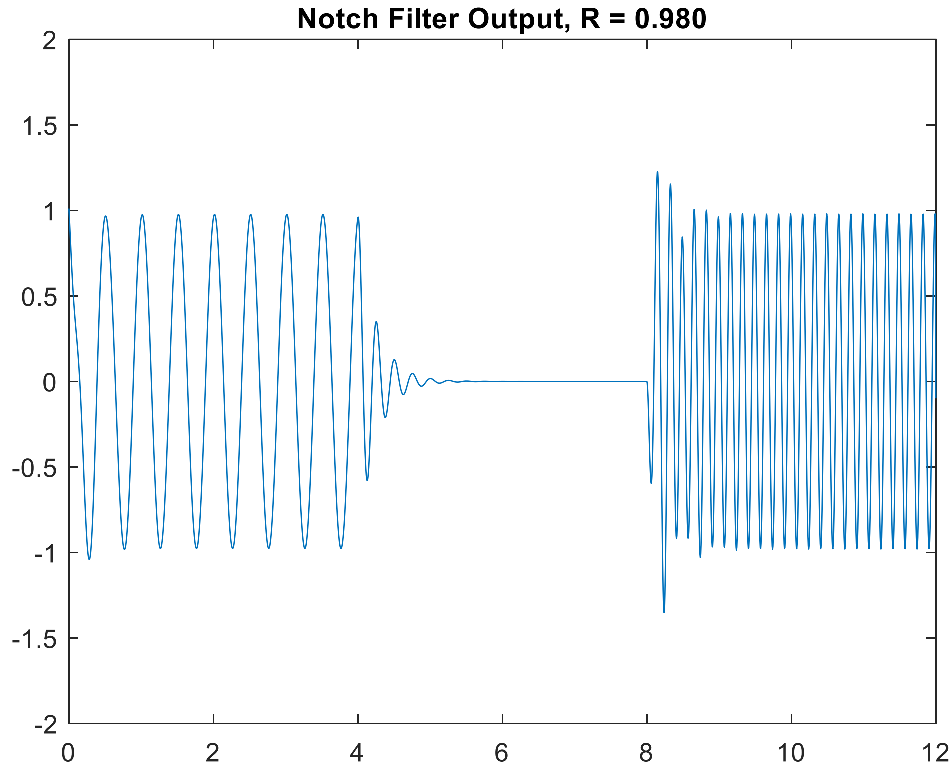
```
t = linspace(0,12,1001);

figure;
plot(t, x(t));
hold on;
title('input signal');
xlim([0 12]);
ylim([-2 2]);
hold off;

G1 = (1 - 2*R1*cos(w0) + R1^2)/(1 - 2*cos(w0) + 1);
G2 = (1 - 2*R2*cos(w0) + R2^2)/(1 - 2*cos(w0) + 1);

b1 = G1*[1, -2*cos(w0), 1];
a1 = [1, -2*R1*cos(w0), R1^2];
b2 = G2*[1, -2*cos(w0), 1];
a2 = [1, -2*R2*cos(w0), R2^2];

t = 0:Ts:12;

figure;
```

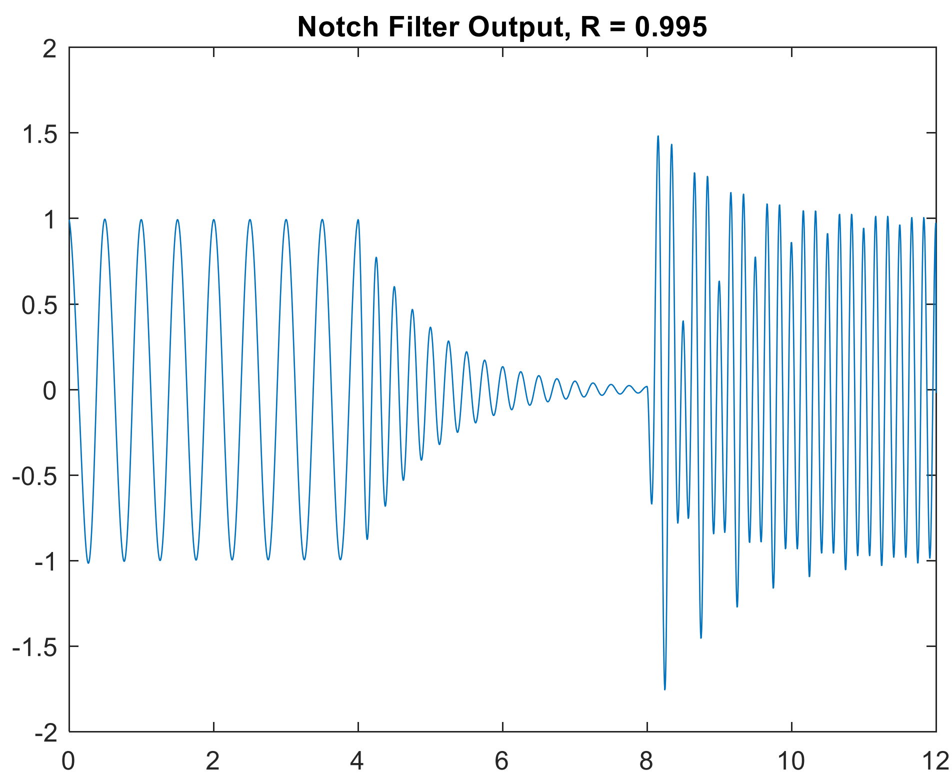
```
y = tran(G1*b1, a1, x(t));

plot(t, y);

hold on;

title('Notch Filter Output, R = 0.980');

xlim([0 12]);

ylim([-2 2]);

hold off;


figure;

y = tran(G2*b2, a2, x(t));

plot(t, y);

hold on;

title('Notch Filter Output, R = 0.995');

xlim([0 12]);

ylim([-2 2]);

hold off;


t_eff1 = Ts * log(0.01)/log(R1);

t_eff2 = Ts * log(0.01)/log(R2);


fprintf(' R       t_eff\n');

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

```
fprintf('%.4f  %2.4f (sec)\n', R1, t_eff1);

fprintf('%.4f  %2.4f (sec)\n\n', R1, t_eff2);
```

```
R       t_eff

--------------------

0.9800  1.1397 (sec)

0.9800  4.5936 (sec)
```




Notch Filter Output, R = 0.980
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
2
4
6
8
10
12


Notch Filter Output, R = 0.995
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
2
4
6
8
10
12