UNIVERSITY OF VICTORIA

Department of Electrical and Computer Engineering

ECE 534 Application of Digital Signal Processing Techniques

PROJECT REPORT

Project Option: C

Title: Design Method based on Balanced Realization in State Space

Date: April 22nd, 2020

Name: Jude Onyia

Student ID: V00947095

1. Objective

Design three IIR filters (high-pass, band-pass and band-stop) by first using a least-squared method to design FIR filters of a specific order, cut-off frequency and group delay. Then, utilize the balanced truncation technique to design IIR filters with a specified reduced order and the same frequency response as the FIR filters.

2. Introduction

The design of a high-order FIR filter to accommodate a required frequency response using an appropriate design method produces a filter that is always stable and has good quality. After realizing the FIR filter in the state space and obtaining a balanced realization of the system, it is evident that the components of the system's state vector are independent. Also, their importance in terms of obtaining the desired frequency response is in a descending order. This allows the reduction of the order of the system by keeping only the first few components of the state variable in order to simplify the system with minimum distortion. This method is known as the balanced truncation technique. This section will discuss balanced realization properties and the balanced truncation technique.

2.1. Balanced Realization of a Transfer Function

After obtaining a high-order FIR filter for a required frequency response, its transfer function can be realized in the N-dimensional state space with state vector $\boldsymbol{x}(k)$ as shown in (1). The input and output of the system is denoted as $u(k)$ and $y(k)$ respectively.

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u(k) \tag{1}$$

$$y(k) = \boldsymbol{c}\boldsymbol{x}(k) + du(k)$$

$$where\ \boldsymbol{x}(k) \in \boldsymbol{R}^{N\times 1}, \boldsymbol{A} \in \boldsymbol{R}^{N\times N}, \boldsymbol{b} \in \boldsymbol{R}^{N\times 1}, \boldsymbol{c} \in \boldsymbol{R}^{1\times N}, and\ d \in \boldsymbol{R}^{1\times 1}$$

A state-space realization {**A**, **b**, **c**, d} of the system relates to its transfer function H(z) according to equation (2). To achieve the objective of this project, the state-space realization must be balanced. Equation (3) shows a balanced realization of H(z).

$$H(z) = \boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} + d \tag{2}$$

$$where\ \boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_{N-1} \\ 0 & \boldsymbol{0} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} h_1 \\ \vdots \\ h_N \end{bmatrix}, \boldsymbol{c} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, d = h_0$$

The balanced state-space realization in equation (3) satisfies the relationship to H(z) as shown in equation (4) and has a controllability Gramian and an observability Gramian that are diagonal and equal in equation (5). The matrix $\boldsymbol{T}$ is an arbitrary non-singular matrix that, in this case of a balanced state-space realization, must be computed in regard to the condition of the Gramians in equation (5).

$$\boldsymbol{x}(k+1) = \boldsymbol{A}_b\boldsymbol{x}(k) + \boldsymbol{b}_b u(k) \tag{3}$$

$$y(k) = \boldsymbol{c}_b\boldsymbol{x}(k) + d_{\boldsymbol{b}}$$

$$where\ \boldsymbol{A_b} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}, \boldsymbol{b_b} = \boldsymbol{T}^{-1}\boldsymbol{b}, \boldsymbol{c_b} = \boldsymbol{c}\boldsymbol{T}, d_{\boldsymbol{b}} = d$$

$$H(z) = \boldsymbol{c}_b(z\boldsymbol{I} - \boldsymbol{A}_b)^{-1}\boldsymbol{b}_b + d_{\boldsymbol{b}} \tag{4}$$

$$\boldsymbol{K} = \boldsymbol{W} = diag\{\sigma_1, \sigma_2, \dots, \sigma_N\} \tag{5}$$

Equation (6) and (7) define the controllability Gramian $\boldsymbol{K}$ and the observability Gramian $\boldsymbol{W}$ for any stable state-space realization of H(z), respectively. Since the transfer function is FIR, the state-space realization produces a finite number N of nonzero terms for the Gramians.

$$\boldsymbol{K} = \sum_{i=0}^{N-1} \boldsymbol{A}^i\boldsymbol{b}\boldsymbol{b}^T(\boldsymbol{A}^i)^T \tag{6}$$

$$\boldsymbol{W} = \sum_{i=0}^{N-1} (\boldsymbol{A}^i)^T\boldsymbol{c}^T\boldsymbol{c}\boldsymbol{A}^i = \boldsymbol{I}_N \tag{7}$$

The Lyapunov equations in (8) and (9) can be used to calculate the Gramians more accurately and efficiently compared to the direct approach in equations (6) and (7).

$$\boldsymbol{K} - \boldsymbol{A}\boldsymbol{K}\boldsymbol{A}^T = \boldsymbol{b}\boldsymbol{b}^T \tag{8}$$

$$\boldsymbol{W} - \boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A} = \boldsymbol{c}^T\boldsymbol{c} \tag{9}$$

To simplify, one can obtain the balanced realization of a transfer function H(z) in three simple steps:

- Compute the controllability Gramian $\boldsymbol{K}$ using the Lyapunov equation in (8) or the direct approach in (6).
- Compute the matrix **T** by finding an orthogonal matrix $\boldsymbol{U}$ that meets the condition in (10), and construct matrix $\boldsymbol{V}$ as shown in (11). Then, compute $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{V}$.

$$\boldsymbol{U}^T\boldsymbol{K}\boldsymbol{U} = diag\{\xi_1, \xi_2, \dots, \xi_N\} \tag{10}$$

$$where\ \xi_1 \geq \xi_1 \geq \cdots \geq \xi_1 > 0\ are\ the\ eigenvalues\ of\ \boldsymbol{K}$$

$$\boldsymbol{V} = diag\{\sqrt{\sigma_1}, \sqrt{\sigma_2}, \ldots, \sqrt{\sigma_N}\}\ \ where\ \sigma_i^2 = \xi_i\ \ for\ i = 1,2,\ldots,N \tag{11}$$

- Finally, construct the balanced realization $\{\boldsymbol{A_b}, \boldsymbol{b_b}, \boldsymbol{c_b}, d_b\} = \{\boldsymbol{T^{-1}AT},\ \boldsymbol{T^{-1}b},\ \boldsymbol{cT}, d\}$.

2.2. Balanced Truncation Technique

After obtaining the balanced realization $\{\boldsymbol{A_b}, \boldsymbol{b_b}, \boldsymbol{c_b}, d_b\}$ of a stable high-order FIR filter H(z), one can use the balanced truncation technique to reduce the order with minimum distortion relative to the original system. The transfer function of the desired reduced order IIR filter $H_r(z)$ can be obtained by partitioning the balanced realization as shown in (12). The transfer function $H_r(z)$ is then computed as shown in equation (13).

$$\boldsymbol{A_b} = \begin{bmatrix} \boldsymbol{A_r} & * \\ * & * \end{bmatrix}, \boldsymbol{b_b} = \begin{bmatrix} \boldsymbol{b_r} \\ * \end{bmatrix}, \boldsymbol{c_b} = \begin{bmatrix} \boldsymbol{c_r} & * \end{bmatrix} \tag{12}$$

$$where\ \{*\}\ denotes\ the\ discarded\ N - r\ portion\ of\ the\ balanced\ realization$$

$$H_r(z) = \boldsymbol{c_r}(z\boldsymbol{I} - \boldsymbol{A_r})^{-1}\boldsymbol{b_r} + \boldsymbol{d_b} \tag{13}$$

3. Implementation

The specification of the project was to design three IIR filters (high-pass, band-pass and band-stop) using the balanced truncation technique. The order of each IIR filter was set to 12 and the group delay was set to 7 samples. For high-pass, the normalized cut-off frequency was 0.7. For band-pass and band-stop, the normalized upper and lower cut-off frequencies were 0.2 and 0.6, respectively. The order of the starting FIR filter was 28. The implementation of the project was achieved using the following steps:

1. Write a function that implements a least-squared method for a band-pass FIR filter. This function takes as inputs the following:
    a. The order of the filter
    b. The group delay
    c. The weight of the stopbands (always set to 1 in this project)
    d. The normalized lower stopband edge (fa1)
    e. The normalized lower passband edge (fp1)
    f. The normalized higher passband edge (fp2)
    g. The normalized higher stopband edge (fa2)

2. Call the function is step 1 with inputs that result in the function producing a high-pass filter. The inputs involved for the high-pass filter are as follows (where fc is the cut-off frequency):
    a. fa1 = fc - 0.09
    b. fp1 = fc + 0.09
    c. fp2 = 0.9999
    d. fa2 = 0.9999
3. Extract the state-space realization {**A**, **b**, **c**, d} from the filter.
4. Compute the balance realization $\{\boldsymbol{A}_b, \boldsymbol{b}_b, \boldsymbol{c}_b, d_b\}$.
5. Obtain the reduced order filter properties as shown in equation (12).
6. Compute the IIR transfer function using the ***ss2tf()*** function on MATLAB.
7. Repeat from step 2 to step 6 for a band-pass filter and a band-stop filter (the subtraction of an all-pass filter and a band-pass filter). The input parameters are as follows (where fc1 and fc2 are the lower and upper cut-off frequencies, respectively):
    a. fa1 = fc1 - 0.09
    b. fp1 = fc1 + 0.09
    c. fp2 = fc2 - 0.09
    d. fa2 = fc2 + 0.09

4. Results

The results of the implementation are shown below. The FIR filters obtained from the least-squared method are compared with their corresponding IIR versions obtained from the balanced truncation technique.

4.1. High-pass filter Design Results

The least-squared implementation successfully produced a high-pass FIR filter with the normalized cut-off frequency of 0.7 as shown in Fig. 1.

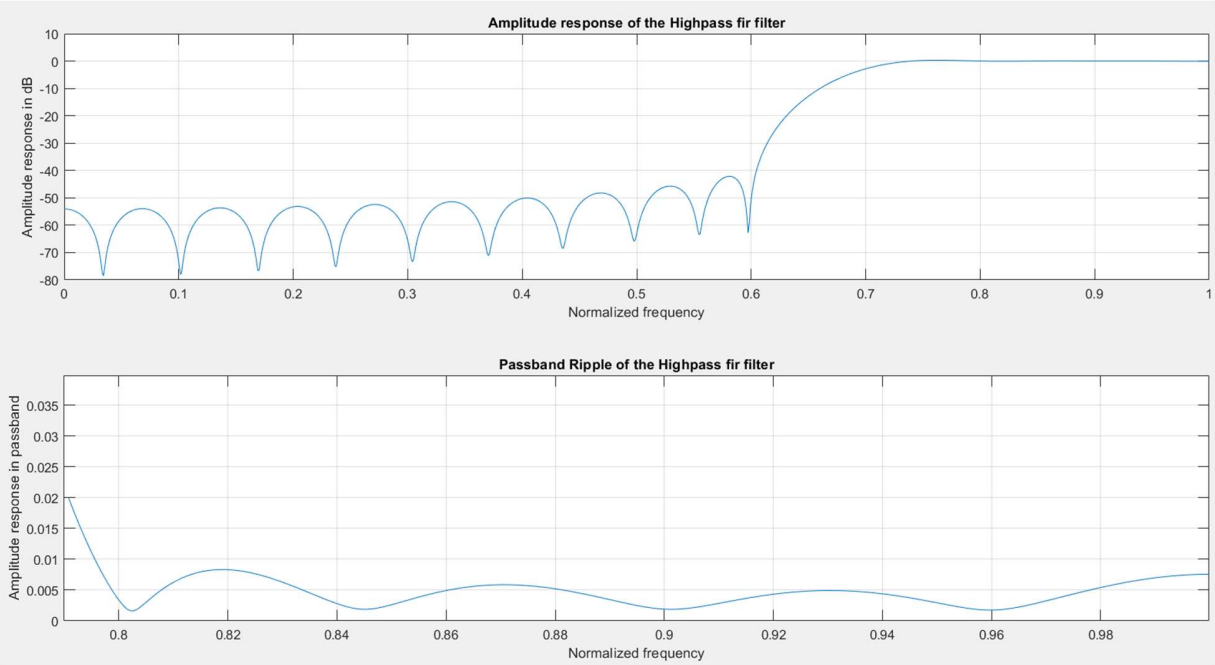

Fig. 1. Amplitude Response of the High-pass FIR Filter

The passband ripple between the normalized passband frequency range of 0.7 to 1 in Fig.1 is less than 0.02.

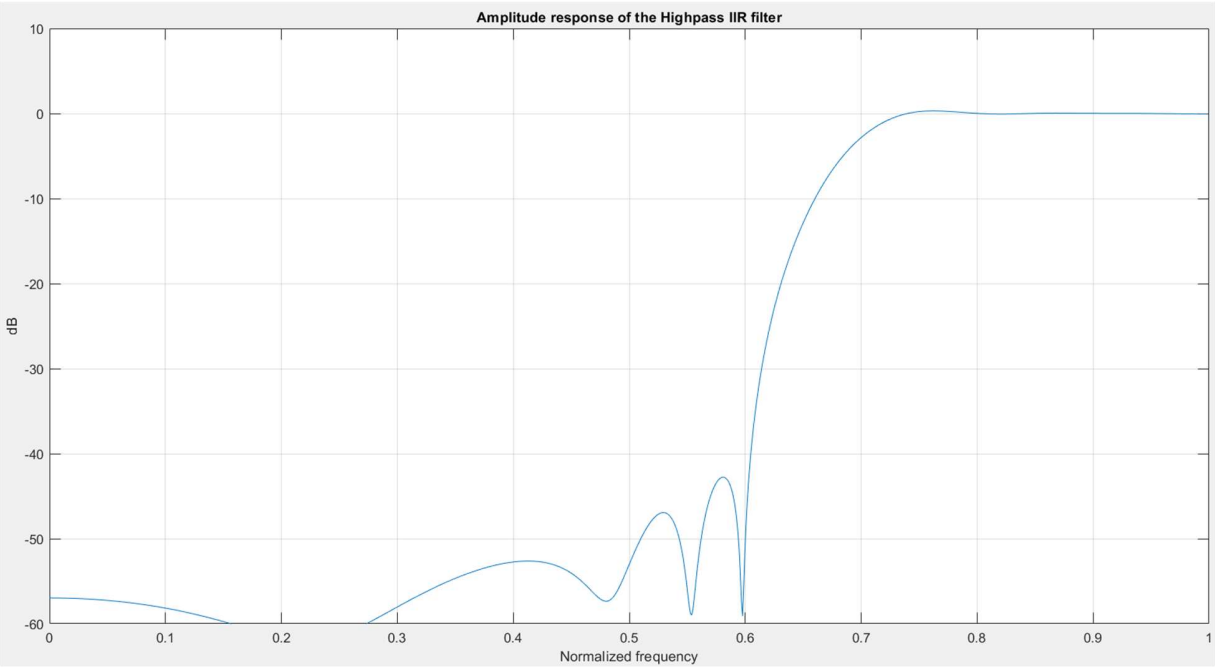

Fig. 2. Amplitude Response of the High-pass IIR Filter

Fig. 2 shows the high-pass IIR filter obtained through the balanced truncation technique, this is very similar the result of the FIR design shown in Fig. 1.

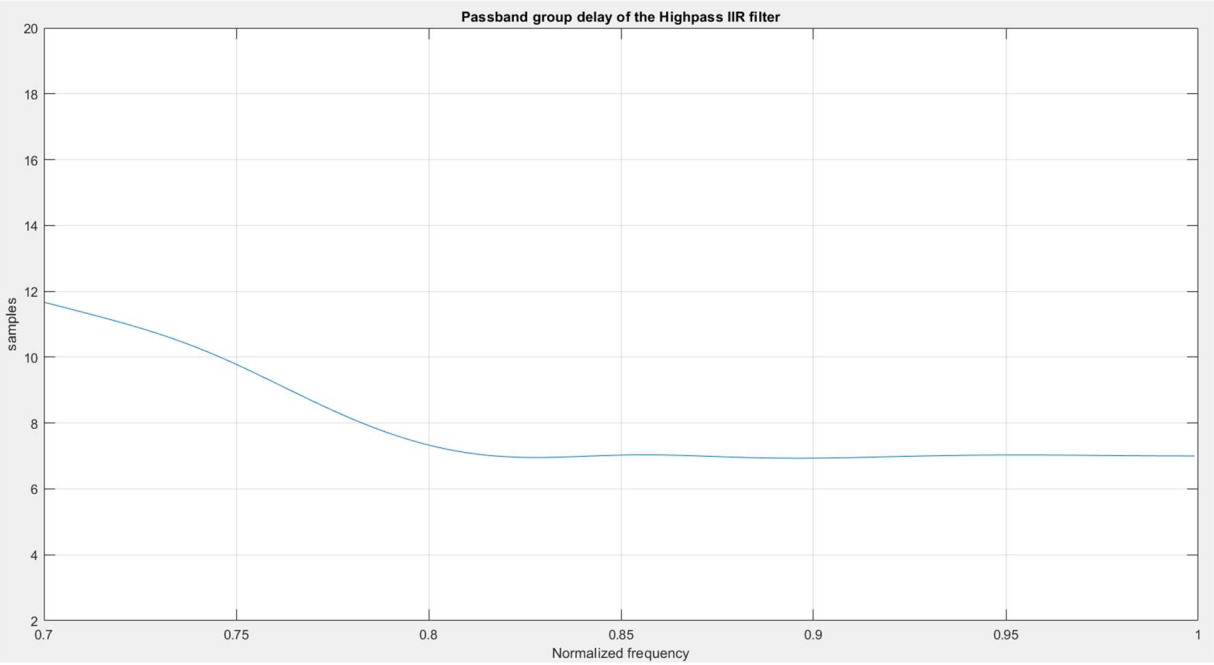

Fig. 3. Passband Group Delay of the High-pass IIR Filter

Fig. 3 shows that the group delay between the normalized passband frequency range is 7 sample as required.

4.2. Band-pass filter Design Results

The result of the band-pass FIR filter is shown in Fig. 4, its passband ripple is below 0.02 in the desired frequency range.

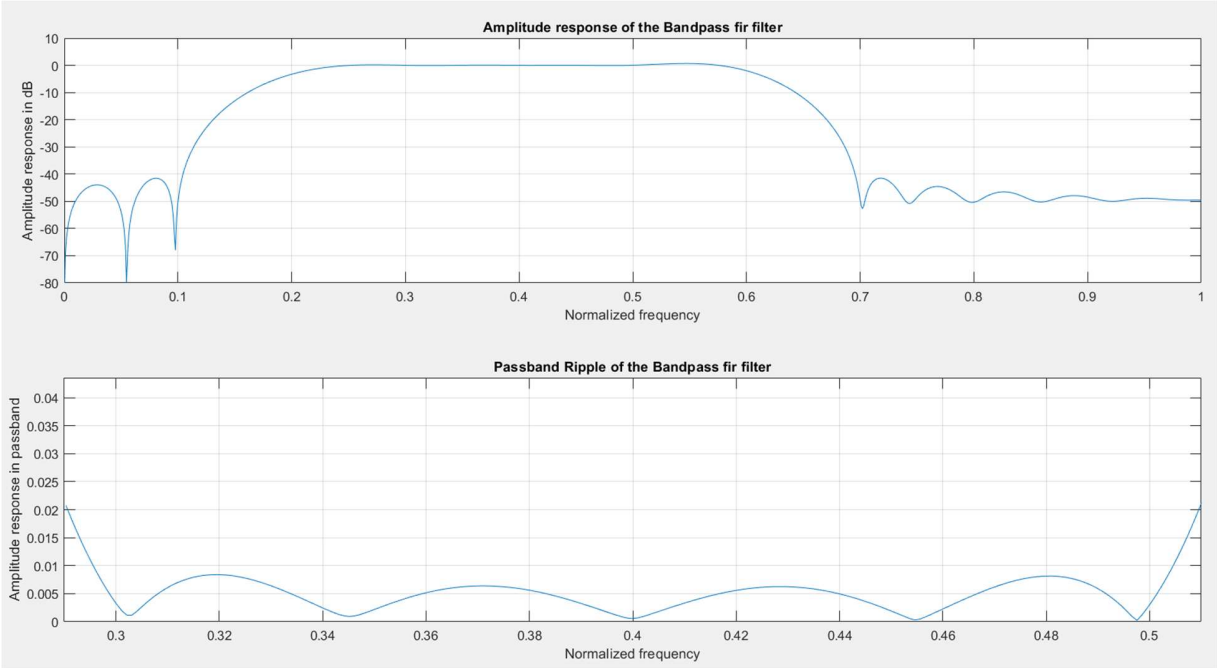

Fig. 4. Amplitude Response of the Band-pass FIR Filter

The desired frequency response is obtained, as shown in Fig. 4, despite the irregular shape of the response at the stopband frequencies.

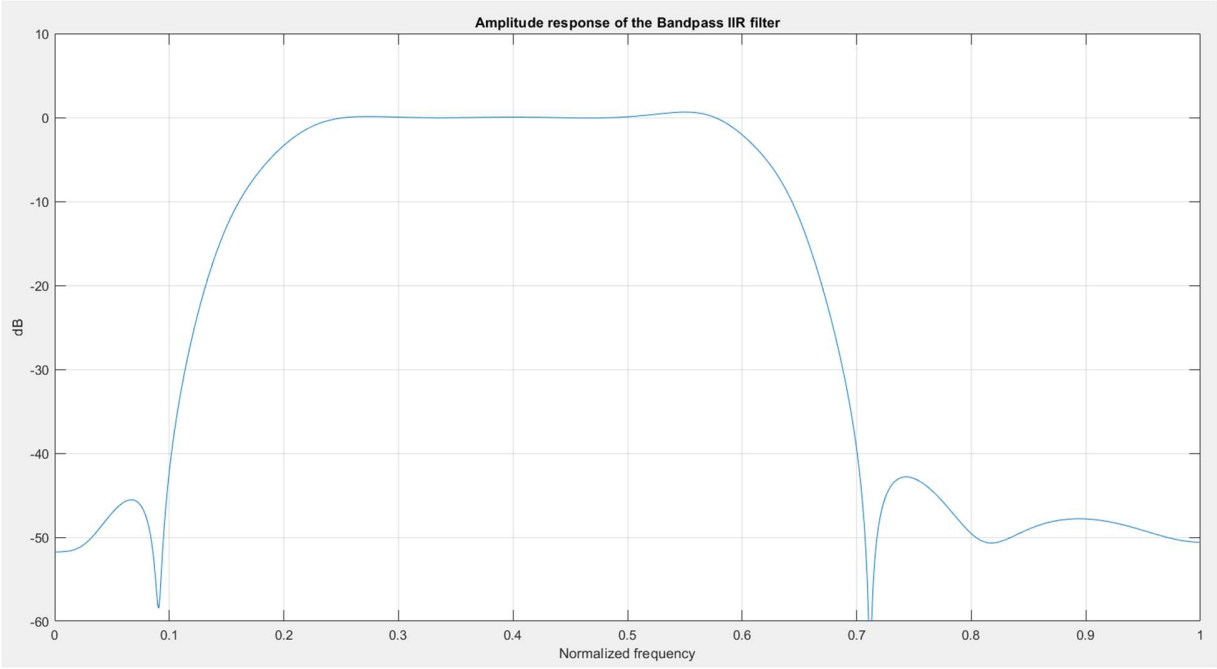

Fig. 5. Amplitude Response of the Band-pass IIR Filter

Fig. 5 shows the amplitude response of the IIR filter obtained. Like the results for the FIR filter, it allows the required frequency range and attenuates others.

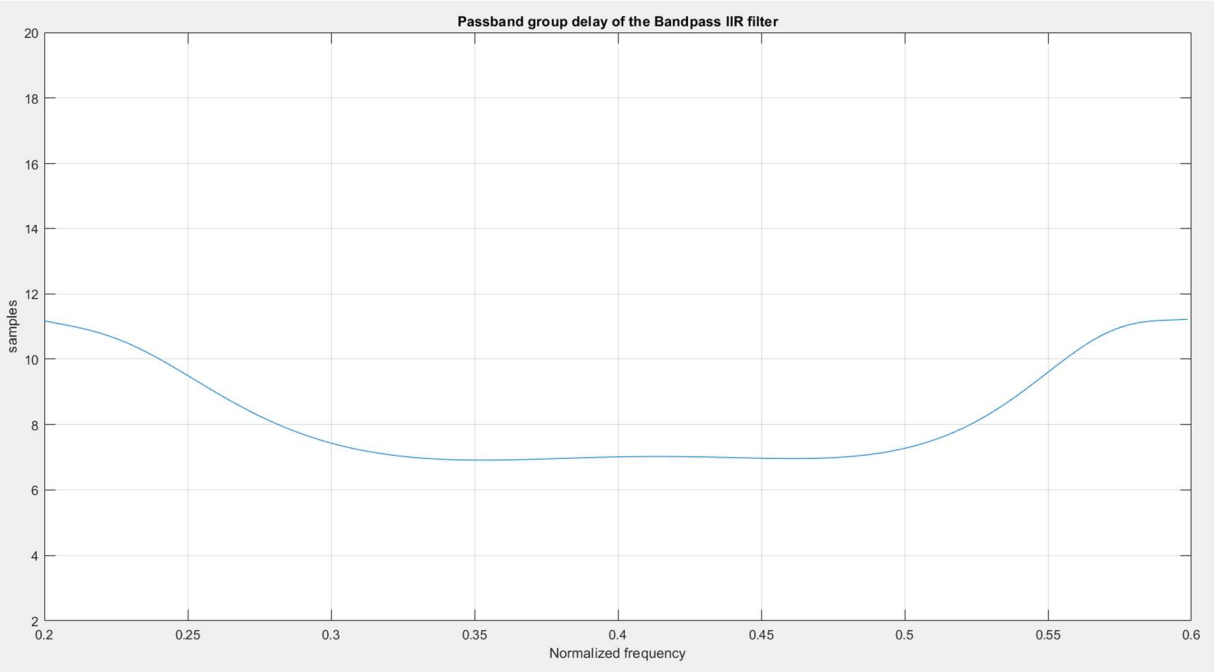

Fig. 6. Passband Group Delay of the Band-pass IIR Filter

Fig. 6 shows that the group delay is 7 samples within the passband range as expected.

4.3. Band-stop filter Design Results

The band-stop filter was obtained by subtracting a band-pass filter of the required cut-off frequencies from an all-pass filter that allowed all normalized frequencies from 0 to 1.

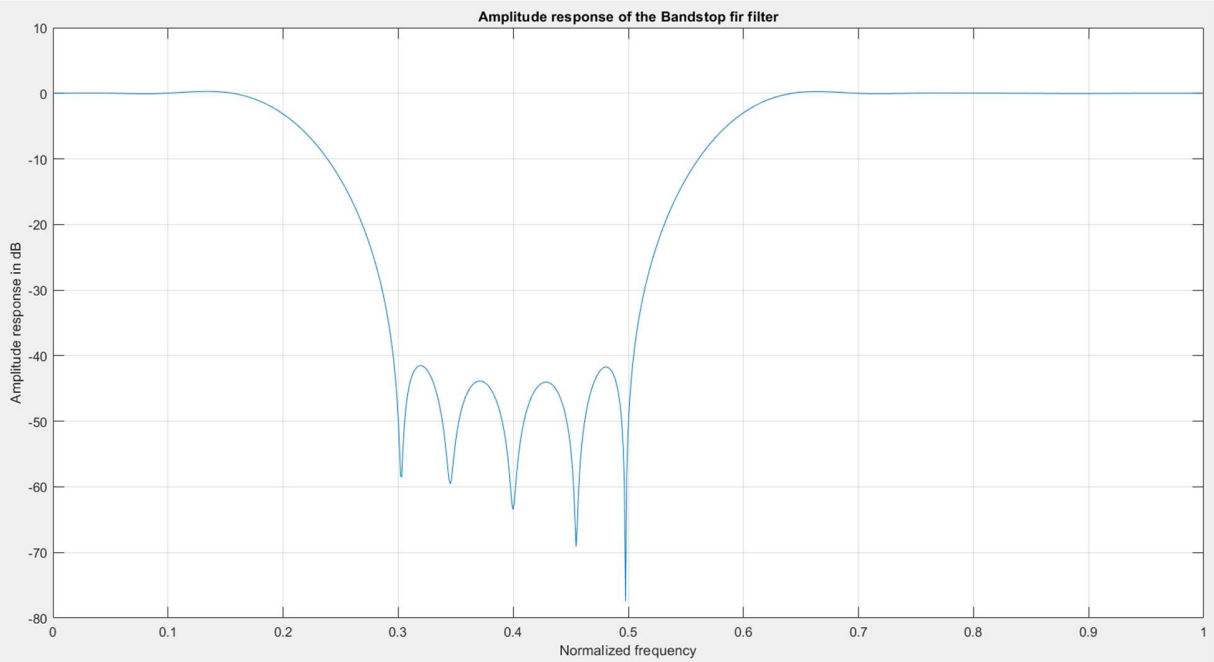

Fig. 7. Amplitude Response of the Band-stop FIR Filter

The amplitude response of the band-stop FIR filter obtained from the least-squared implementation is shown in Fig. 7.

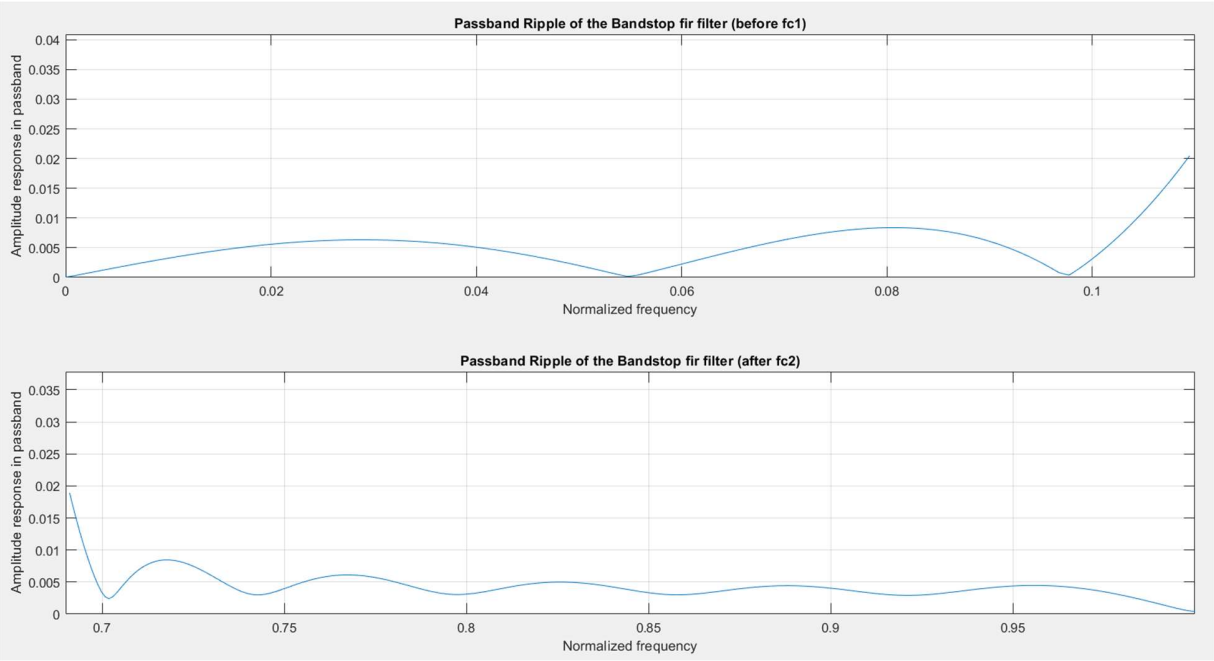

Fig. 8. Passband Ripple of the Band-stop FIR Filter before fc1 and after fc2

The passband ripple before the lower cut-off frequency and after the higher cut-off frequency is shown in Fig. 8. Both do not go beyond a magnitude of 0.02.

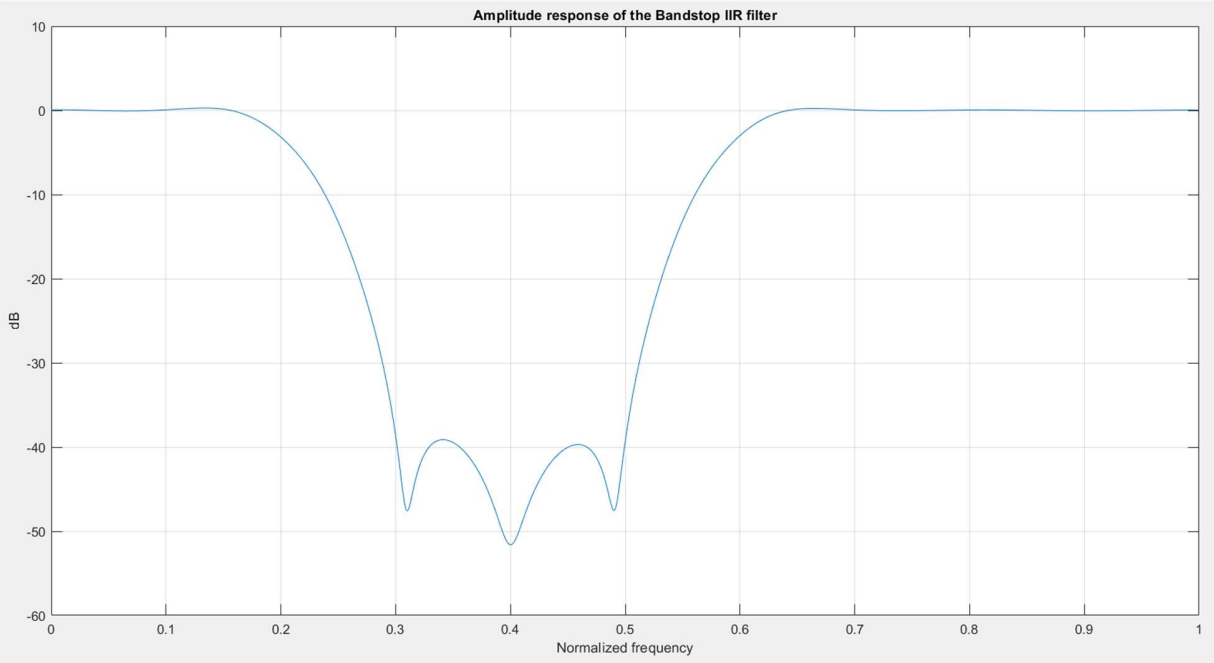

Fig. 9. Amplitude Response of the Band-stop IIR Filter

Fig. 9 shows the amplitude response of the band-stop IIR filter obtained from the balanced truncation technique. As seen, its response is like its FIR version.

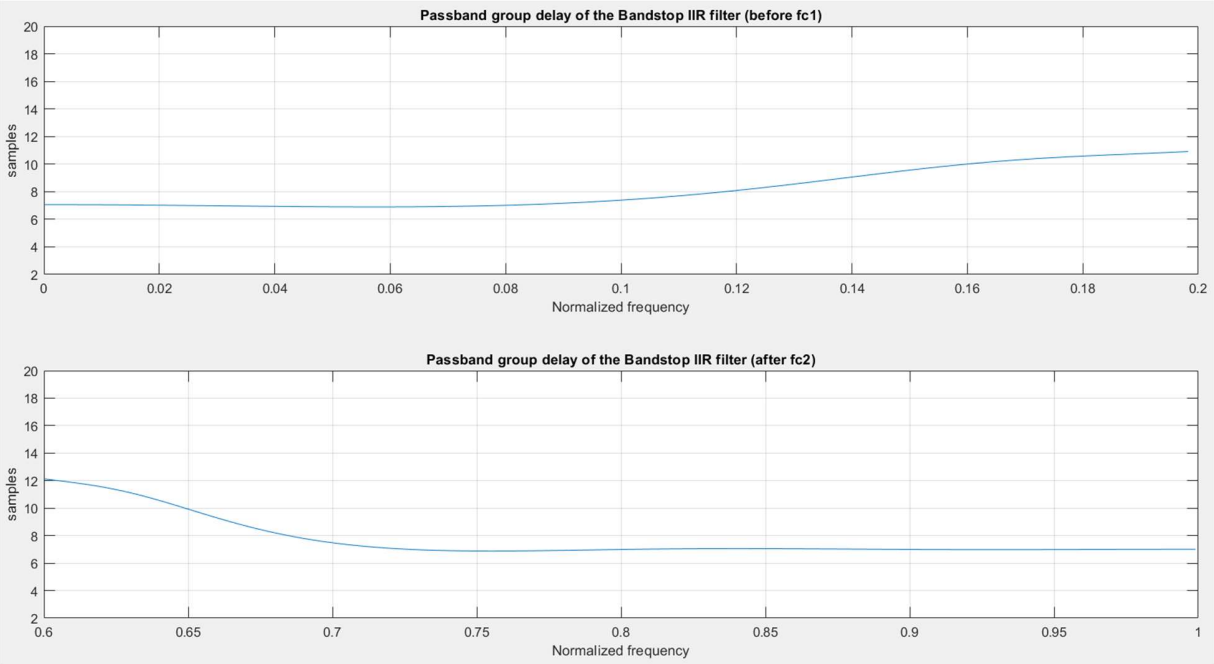

Fig. 10. Passband Group Delay of the Band-stop IIR Filter before fc1 and after fc2

Fig. 10 shows the group delay before the lower cut-off frequency and after the higher cut-off frequency. As expected, they are both 7 samples.

5. Conclusion

With a stable high-order FIR filter designed to approximate a desired frequency response, one can obtain a stable IIR filter of lower order using the balanced truncation technique. This project shows that this technique provides an efficient way to approximate high-order FIR filters with IIR filters of much lower order and ensured stability. Achieving this can be quite useful in reducing cost by using a lower order filter and provides a safe system with a stable filter. This technique is also desirable because the approximation error due to the reduction of the filter size is nearly optimal in terms of amplitude response.

6. Reference

[1] Lecture Notes for ECE 459/534 Applications of Digital Signal Processing Techniques. W.-S. Lu, University of Victoria Department of Electrical and Computer Engineering.

7. Appendix

Below are the MATLAB scripts associated to this project, five scripts in total. The first is the main script to run on MATLAB, it calls three functions that use the balance truncation technique to design high-pass, band-pass and band-stop IIR filters. These three functions obtain their corresponding FIR filters needed for the balance truncation technique by calling another function that implements a least-squared method for FIR filter design.

7.1. MATLAB Code of the Main Script

```
close all
clear
clc

% Filter specifications
N = 28; % High-order of stable FIR filter
r = 12; % Low-order of obtained IIR filter
D = 7; % Group Delay
fc = 0.7; % cut-off frequency for High-pass filter
% upper and lower cut-off frequencies for band-pass and
band-stop
fc1 = 0.2;
fc2 = 0.6;

% Display filter coefficients of High-pass IIR filter
[a,b] = High_Pass_bt_iir(N,r,D,fc);
format long
fprintf('Denominator of High Pass Filter (a):\n');
display(a);
fprintf('Numerator of High Pass Filter (b):\n');
display(b);

% Display filter coefficients of Band-pass IIR filter
[a,b] = Band_Pass_bt_iir(N,r,D,fc1,fc2);
format long
fprintf('Denominator of Band Pass Filter (a):\n');
display(a);
fprintf('Numerator of Band Pass Filter (b):\n');
display(b);

% Display filter coefficients of Band-stop IIR filter
[a,b] = Band_Stop_bt_iir(N,r,D,fc1,fc2);
```

```
format long
fprintf('Denominator of Band Stop Filter (a):\n');
display(a);
fprintf('Numerator of Band Stop Filter (b):\n');
display(b);
```

7.2. MATLAB Code of the Least-squares FIR design function

```
% This function implements the least-squares design of
nonlinear-phase
% lowpass FIR filters.
% Inputs:
% N: Order of the FIR filter, N must be an even integer.
% d: group-dealy in passband
% w: weight for the stopbands
% (assuming the weight in passband is 1)
% fa1: normalized lower stopband edge between 0 and pi
% fp1: normalized lower passband edge between 0 and pi
% with omi_p1 > omi_a1
% fp2: normalized higher passband edge between 0 and pi
% with omi_p2 > omi_p1
% fa2: normalized higher cutoff frequency between 0 and pi
% with omi_c2 > omi_p2.
% Output:
% h: impulse response of the bandpass FIR filter.
% Written by W.-S. Lu, University of Victoria
function h = bandpass_fir(N,d,w,fa1,fp1,fp2,fa2)
a1 = pi*fa1;
p1 = pi*fp1;
p2 = pi*fp2;
a2 = pi*fa2;
q = zeros(N+1,N+1);
b = zeros(N+1,1);
c = zeros(N+1,1);
c(1) = w*(a1-a2+pi)+p2-p1;
b(1) = (sin(d*p2)-sin(d*p1))/d;
for i = 1:N,
 z1 = sin(i*a1) - sin(i*a2);
 z2 = sin(i*p2 )- sin(i*p1);
 c(i+1) = (w*z1 + z2)/i;
```

```
 if i == d,
 b(i+1) = p2 - p1;
 else
 b(i+1) = (sin((i-d)*p2) -sin((i-d)*p1))/(i-d);
 end
end
Q = toeplitz(c);
h = inv(Q)*b;
```

7.3. MATLAB Code of the High-pass IIR filter function

```
% Design highpass IIR filter of order r with constant
% passband group delay using the balanced truncation
technique.
% Inputs:
% N: order of the high-order FIR filter
% r: order of the IIR filter.
% D: desired group delay.
% fc: normalized cutoff frequency between 0 and 1.
% Outputs:
% a,b: denominator and numerator of the IIR filter.
% Written by W.-S. Lu, University of Victoria.
% Example: [a,b] = bt_iir(28,12,7,0.7);
function [a,b] = High_Pass_bt_iir(N,r,D,fc)
ep = 0.09;
h = bandpass_fir(N,D,1,fc-ep,fc+ep,0.9999,0.9999);

% plot FIR frequeny response
figure(1)
fp1 = fc+ep;
fp2 = 0.9999;
j = sqrt(-1);
f = 0:1/1023:1;
ff = f*pi;
ff = ff(:);
F = freqz(h,1,ff);
subplot(211)
amp = abs(F);
plot(f,20*log10(amp))
axis([0 1 -80 10])
```

```
title('Amplitude response of the Highpass fir filter')
xlabel('Normalized frequency')
ylabel('Amplitude response in dB')
grid
subplot(212)
i1 = ceil(1024*fp1) + 1;
i2 = floor(1024*fp2)+ 1;
Hdp = exp(-j*D*ff(i1:i2));
er_p = abs(F(i1:i2)-Hdp);
plot(f(i1:i2),er_p)
axis([fp1 fp2 0 2*max(er_p)])
grid
title('Passband Ripple of the Highpass fir filter')
xlabel('Normalized frequency')
ylabel('Amplitude response in passband')


% State-space realization {A, b, c, d}
d = h(1);
b = h(2:end);
I = eye(N-1);
z1 = zeros(N-1,1);
A = [z1 I; 0 z1'];
c = [1 z1'];

% Finding Balanced realization {Ab, bb, cb, d}
Q = b*b';
K = dlyap(A,Q); % The controllability Gramian
[U,S,V] = svd(K); % Orthogonal matrix U
s = diag(S);
s = s.^0.25;
V = diag(s); % Diagonal matrix V
T = U*V;
Ti = inv(T);
Ab = Ti*A*T;
bb = Ti*b;
cb = c*T;

% Finding reduced order of filter properties {Ar, br, cr,
d} using
% the balanced truncation technique
Ar = Ab(1:r,1:r);
br = bb(1:r);
cr = cb(1:r);
```

```
[b,a] = ss2tf(Ar,br,cr,d,1);
a = a(:);
b = b(:);
[H,w] = freqz(b,a,1024);

% plot IIR frequeny response
figure(2)
plot(w/pi,20*log10(abs(H)))
grid
axis([0 1 -60 10])
xlabel('Normalized frequency')
ylabel('dB')
title("Amplitude response of the Highpass IIR filter")
figure(3)
[gd,w] = grpdelay(b,a,1024);
np = floor(1024*fc);
plot(w(np:1024)/pi,gd(np:1024));
axis([fc 1 2 20])
grid
xlabel('Normalized frequency')
ylabel('samples')
title("Passband group delay of the Highpass IIR filter")
```

7.4. MATLAB Code of the Band-pass IIR filter function

```
% Design bandpass IIR filter of order r with constant
% passband group delay using the balanced truncation
technique.
% Inputs:
% N: order of the high-order FIR filter
% r: order of the IIR filter.
% D: desired group delay.
% fc: normalized cutoff frequency between 0 and 1.
% Outputs:
% a,b: denominator and numerator of the IIR filter.
% Written by W.-S. Lu, University of Victoria.
% Example: [a,b] = bt_iir(28,12,7,0.7);
function [a,b] = Band_Pass_bt_iir(N,r,D,fc1,fc2)
ep = 0.09;
h = bandpass_fir(N,D,1,fc1-ep,fc1+ep,fc2-ep,fc2+ep);
```

```
% plot FIR frequeny response
figure(4)
fp1 = fc1+ep;
fp2 = fc2-ep;
j = sqrt(-1);
f = 0:1/1023:1;
ff = f*pi;
ff = ff(:);
F = freqz(h,1,ff);
subplot(211)
amp = abs(F);
plot(f,20*log10(amp))
axis([0 1 -80 10])
title('Amplitude response of the Bandpass fir filter')
xlabel('Normalized frequency')
ylabel('Amplitude response in dB')
grid
subplot(212)
i1 = ceil(1024*fp1) + 1;
i2 = floor(1024*fp2)+ 1;
Hdp = exp(-j*D*ff(i1:i2));
er_p = abs(F(i1:i2)-Hdp);
plot(f(i1:i2),er_p)
axis([fp1 fp2 0 2*max(er_p)])
grid
title('Passband Ripple of the Bandpass fir filter')
xlabel('Normalized frequency')
ylabel('Amplitude response in passband')


% State-space realization {A, b, c, d}
d = h(1);
b = h(2:end);
I = eye(N-1);
z1 = zeros(N-1,1);
A = [z1 I; 0 z1'];
c = [1 z1'];

% Finding Balanced realization {Ab, bb, cb, d}
Q = b*b';
K = dlyap(A,Q); % The controllability Gramian
[U,S,V] = svd(K); % Orthogonal matrix U
s = diag(S);
```

```
s = s.^0.25;
V = diag(s); % Diagonal matrix V
T = U*V;
Ti = inv(T);
Ab = Ti*A*T;
bb = Ti*b;
cb = c*T;

% Finding reduced order of filter properties {Ar, br, cr,
d} using
% the balanced truncation technique
Ar = Ab(1:r,1:r);
br = bb(1:r);
cr = cb(1:r);
[b,a] = ss2tf(Ar,br,cr,d,1);
a = a(:);
b = b(:);
[H,w] = freqz(b,a,1024);


% plot IIR frequeny response
figure(5)
plot(w/pi,20*log10(abs(H)))
grid
axis([0 1 -60 10])
xlabel('Normalized frequency')
ylabel('dB')
title("Amplitude response of the Bandpass IIR filter")
figure(6)
[gd,w] = grpdelay(b,a,1024);
np1 = floor(1024*fc1);
np2 = floor(1024*fc2);
plot(w(np1:np2)/pi,gd(np1:np2));
axis([fc1 fc2 2 20])
grid
xlabel('Normalized frequency')
ylabel('samples')
title("Passband group delay of the Bandpass IIR filter")
```

7.5. MATLAB Code of the Band-stop IIR filter function

```
% Design bandstop IIR filter of order r with constant
% passband group delay using the balanced truncation
technique.
% Inputs:
% N: order of the high-order FIR filter
% r: order of the IIR filter.
% D: desired group delay.
% fc: normalized cutoff frequency between 0 and 1.
% Outputs:
% a,b: denominator and numerator of the IIR filter.
% Written by W.-S. Lu, University of Victoria.
% Example: [a,b] = bt_iir(28,12,7,0.7);
function [a,b] = Band_Stop_bt_iir(N,r,D,fc1,fc2)
ep = 0.09;
h1 = bandpass_fir(N,D,1,0,0,0.9999,0.9999); % an Allpass
filter
h2 = bandpass_fir(N,D,1,fc1-ep,fc1+ep,fc2-ep,fc2+ep);
h = h1 - h2; % A bandpass filter subtracted from an
allpass filter results in a bandstop filter!

% plot frequeny response
figure(7)
fp1 = 0;
fp2 = fc1-ep;
j = sqrt(-1);
f = 0:1/1023:1;
ff = f*pi;
ff = ff(:);
F = freqz(h,1,ff);
amp = abs(F);
plot(f,20*log10(amp))
axis([0 1 -80 10])
title('Amplitude response of the Bandstop fir filter')
xlabel('Normalized frequency')
ylabel('Amplitude response in dB')
grid
figure(8)
subplot(211)
i1 = ceil(1024*fp1) + 1;
i2 = floor(1024*fp2)+ 1;
Hdp = exp(-j*D*ff(i1:i2));
er_p = abs(F(i1:i2)-Hdp);
```

```
plot(f(i1:i2),er_p)
axis([fp1 fp2 0 2*max(er_p)])
grid
title('Passband Ripple of the Bandstop fir filter (before
fc1)')
xlabel('Normalized frequency')
ylabel('Amplitude response in passband')
subplot(212)
fp1 = fc2+ep;
fp2 = 0.9999;
i1 = ceil(1024*fp1) + 1;
i2 = floor(1024*fp2)+ 1;
Hdp = exp(-j*D*ff(i1:i2));
er_p = abs(F(i1:i2)-Hdp);
plot(f(i1:i2),er_p)
axis([fp1 fp2 0 2*max(er_p)])
grid
title('Passband Ripple of the Bandstop fir filter (after
fc2)')
xlabel('Normalized frequency')
ylabel('Amplitude response in passband')


% State-space realization {A, b, c, d}
d = h(1);
b = h(2:end);
I = eye(N-1);
z1 = zeros(N-1,1);
A = [z1 I; 0 z1'];
c = [1 z1'];

% Finding Balanced realization {Ab, bb, cb, d}
Q = b*b';
K = dlyap(A,Q); % The controllability Gramian
[U,S,V] = svd(K); % Orthogonal matrix U
s = diag(S);
s = s.^0.25;
V = diag(s); % Diagonal matrix V
T = U*V;
Ti = inv(T);
Ab = Ti*A*T;
bb = Ti*b;
cb = c*T;
```

```
% Finding reduced order of filter properties {Ar, br, cr,
d} using
% the balanced truncation technique
Ar = Ab(1:r,1:r);
br = bb(1:r);
cr = cb(1:r);
[b,a] = ss2tf(Ar,br,cr,d,1);
a = a(:);
b = b(:);
[H,w] = freqz(b,a,1024);


% plot IIR frequeny response
figure(9)
plot(w/pi,20*log10(abs(H)))
grid
axis([0 1 -60 10])
xlabel('Normalized frequency')
ylabel('dB')
title("Amplitude response of the Bandstop IIR filter")
figure(10)
[gd,w] = grpdelay(b,a,1024);
np1 = floor(1024*(fc1));
np2 = floor(1024*(fc2));
subplot(211)
plot(w(1:np1)/pi,gd(1:np1));
axis([0 fc1 2 20])
grid
xlabel('Normalized frequency')
ylabel('samples')
title("Passband group delay of the Bandstop IIR filter
(before fc1)")
subplot(212)
plot(w(np2:1024)/pi,gd(np2:1024));
axis([fc2 1 2 20])
grid
xlabel('Normalized frequency')
ylabel('samples')
title("Passband group delay of the Bandstop IIR filter
(after fc2)")
```