```matlab
%Cardy Wei
%DSP Proj 2
%Prof Keene

clear all; close all;
load ('projIB.mat');

Rp = 3;
Rs = 95;
Wp=2500;
Ws = 4000;
Fs=44100;

%Ellip Filter
[n,Wn] = ellipord(Wp/(Fs/2),Ws/(Fs/2),Rp,Rs);
%Df1
[b1, a1]=ellip(n, Rp, Rs, Wn);
Ellip1 = dfilt.df1(b1,a1);

figure;
[h,w]=freqz(Ellip1, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Ellip1, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')
%Df2
[b2, a2]=ellip(n, Rp, Rs, Wn);
Ellip2 = dfilt.df2(b2,a2);

figure;
[h,w]=freqz(Ellip2, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Ellip2, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

%Df2sos
[z1,p1,k1] = ellip(n, Rp, Rs, Wn);
```

```
[SOS1,G] = zp2sos(z1,p1,k1);
Ellip3 = dfilt.df2sos(SOS1,G);

figure;
[h,w]=freqz(Ellip2, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Ellip3, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')
%Df2tsos
[z2,p2,k2] = ellip(n, Rp, Rs, Wn);
[SOS2,G] = zp2sos(z2,p2,k2);
Ellip4 = dfilt.df2tsos(SOS2,G);

figure;
[h,w]=freqz(Ellip2, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Ellip4, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')
figure;
subplot(2,2,1)
zplane(Ellip1)
title('DB1')

subplot(2,2,2)
zplane(Ellip2)
title('DB2')

subplot(2,2,3)
zplane(Ellip3)
title('SOS')

subplot(2,2,4)
zplane(Ellip4)
title('SOS Transposed')

imp=[1 zeros(1,99)];

figure;
```

```
subplot(2,2,1)
resp =filter(Ellip1,imp);
stem(resp)
xlim([0,100])
title('DB1')

subplot(2,2,2)
resp =filter(Ellip2,imp);
stem(resp)
xlim([0,100])
title('DB2')

subplot(2,2,3)
resp =filter(Ellip3,imp);
stem(resp)
xlim([0,100])
title('SOS')

subplot(2,2,4)
resp =filter(Ellip4,imp);
stem(resp)
xlim([0,100])
title('SOS Transposed')

x = cost(Ellip1)
y = cost(Ellip2)
z = cost(Ellip3)
g = cost(Ellip4)

% Butterworth Filter
[n,Wn] = buttord(Wp/(Fs/2),Ws/(Fs/2),Rp,Rs);

%Df2sos
[z1,p1,k1] = butter(n-1, Wn);
[SOS1,G] = zp2sos(z1,p1,k1);
Hd3 = dfilt.df2sos(SOS1,G);

figure;
[h,w]=freqz(Hd3, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Hd3, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

%Df1
[z4,p4,k4]=butter(n-1, Wn);
Hd = Hd3.convert('df1');
```

```
figure;
[h,w]=freqz(Hd, 100);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Hd, 100);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

%Df2tsos
[z2,p2,k2] = butter(n-1, Wn);
[SOS2,G] = zp2sos(z2,p2,k2);
Hd4 = dfilt.df2tsos(SOS2,G);

figure;
[h,w]=freqz(Hd4, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Hd4, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

%Df2
[z3,p3,k3]=butter(n-1, Wn);
Hd2 = Hd4.convert('df2');

figure;
[h,w]=freqz(Hd2, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(Hd2, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

figure;
subplot(2,2,1)
```

```
zplane(Hd)
title('DB1')

subplot(2,2,2)
zplane(Hd2)
title('DB2')

subplot(2,2,3)
zplane(Hd3)
title('SOS')

subplot(2,2,4)
zplane(Hd4)
title('SOS Transposed')

figure;
subplot(2,2,1)
resp =filter(Hd,imp);
stem(resp)
xlim([0,100])
title('DB1')

subplot(2,2,2)
resp =filter(Hd2,imp);
stem(resp)
xlim([0,100])
title('DB2')

subplot(2,2,3)
resp =filter(Hd3,imp);
stem(resp)
xlim([0,100])
title('SOS')

subplot(2,2,4)
resp =filter(Hd4,imp);
stem(resp)
xlim([0,100])
title('SOS Transposed')

x = cost(Hd)
y = cost(Hd2)
z = cost(Hd3)
g = cost(Hd4)

%Cheby1
[n,Wn] = cheb1ord(Wp/(Fs/2),Ws/(Fs/2),Rp,Rs);
[b,a] = cheby1(n,Rp,Wn);

figure;
[h,w]=freqz(b,a, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
```

```
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(b,a, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

figure;
subplot(2,1,1)
zplane(b,a)
title('Cheby1')
subplot(2,1,2)
impz(b,a)

%Cheby2
[n,Wn] = cheb2ord(Wp/(Fs/2),Ws/(Fs/2),Rp,Rs);
[b,a] = cheby2(n,Rp,Wn);

figure;
[h,w]=freqz(b,a, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(b,a, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

figure;
subplot(2,1,1)
zplane(b,a)
title('Cheby2')
subplot(2,1,2)
impz(b,a)

%Parks
ripple = [(10^(Rp/20)-1)/(10^(Rp/20)+1)  10^(-Rs/20)];
f = [2500 4000];
a = [100 0];

[n,fo,ao,w] = firpmord(f,a,ripple,Fs);
x = firpm(n,fo,ao,w);
filt = dfilt.dffir(x);
soundsc(filter(Ellip1,noisy), Fs);
x = cost(filt)

figure;
[h,w]=freqz(filt, 1000);
```

```
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(filt, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

figure;
subplot(2,1,1)
[z,p,k]=zpk(filt);
zplane(z,p)
title('Parks')
subplot(2,1,2)
resp =filter(filt,imp);
stem(resp)
xlim([0,100])

%Kaiser
f = [2500 4000];
mags = [1 0];
devs = [0.5 0.001];

[n,Wn,beta,ftype] = kaiserord(f,mags,devs,Fs);
x = fir1(n,Wn,ftype);
x=x*100;
filt = dfilt.dffir(x);


z = cost(filt)

figure;
[h,w]=freqz(filt, 1000);
subplot(3,1,1)
plot(w/pi,20*log10(abs(h)))
title('Magnitute Response')
subplot(3,1,2)
plot(w/pi,20*log10(abs(h)))
xlim([0 Wp/(Fs/2)])
title('Magnitute Response Passband Ripple')
[gd,w] = grpdelay(filt, 1000);
subplot(3,1,3)
plot(w/pi, gd)
title('Group Delay')

figure;
subplot(2,1,1)
[z,p,k]=zpk(filt);
zplane(z,p)
title('Kaiser')
```

```
subplot(2,1,2)
resp =filter(filt,imp);
stem(resp)
xlim([0,100])
```

```
x =

Number of Multipliers            : 17
Number of Adders                 : 16
Number of States                 : 16
Multiplications per Input Sample : 17
Additions per Input Sample       : 16

y =

Number of Multipliers            : 17
Number of Adders                 : 16
Number of States                 : 8
Multiplications per Input Sample : 17
Additions per Input Sample       : 16

z =

Number of Multipliers            : 16
Number of Adders                 : 16
Number of States                 : 8
Multiplications per Input Sample : 16
Additions per Input Sample       : 16

g =

Number of Multipliers            : 16
Number of Adders                 : 16
Number of States                 : 8
Multiplications per Input Sample : 16
Additions per Input Sample       : 16

x =

Number of Multipliers            : 45
Number of Adders                 : 44
Number of States                 : 44
Multiplications per Input Sample : 45
Additions per Input Sample       : 44

y =

Number of Multipliers            : 45
Number of Adders                 : 44
Number of States                 : 22
Multiplications per Input Sample : 45
Additions per Input Sample       : 44
```

```
z =

Number of Multipliers            : 34
Number of Adders                 : 44
Number of States                 : 22
Multiplications per Input Sample : 34
Additions per Input Sample       : 44

g =

Number of Multipliers            : 34
Number of Adders                 : 44
Number of States                 : 22
Multiplications per Input Sample : 34
Additions per Input Sample       : 44

x =

Number of Multipliers            : 125
Number of Adders                 : 124
Number of States                 : 124
Multiplications per Input Sample : 125
Additions per Input Sample       : 124

z =

Number of Multipliers            : 108
Number of Adders                 : 107
Number of States                 : 107
Multiplications per Input Sample : 108
Additions per Input Sample       : 107
```

Published with MATLAB® R2016b