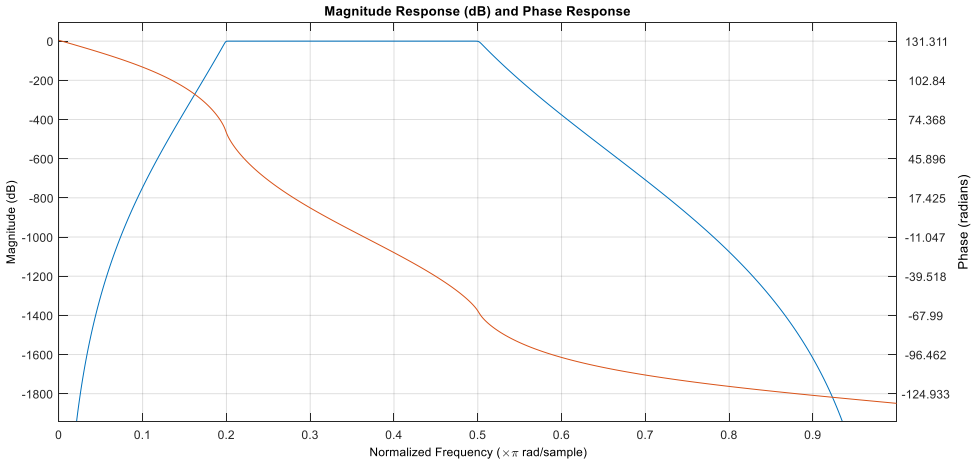
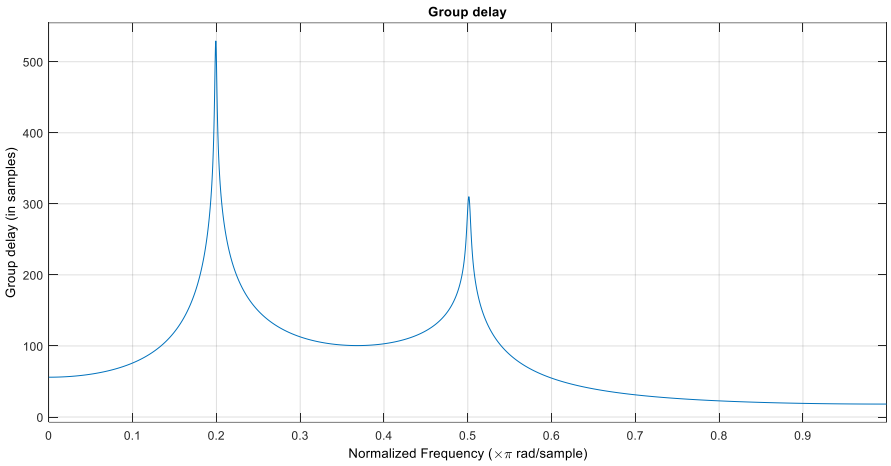
Άσκηση 2.3

- IIR με αναλογικό Butterworth

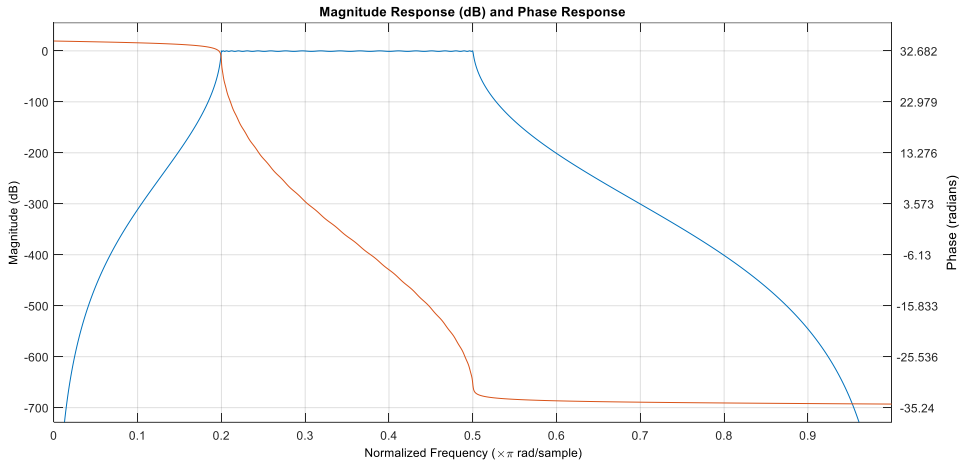
```
function Hd = butterworth
%BUTTERWORTH Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.4 and Signal Processing Toolbox 8.0.
% Generated on: 09-May-2019 23:20:31

% Butterworth Bandpass filter designed using FDESIGN.BANDPASS.

% All frequency values are normalized to 1.

Fstop1 = 0.18;         % First Stopband Frequency
Fpass1 = 0.2;          % First Passband Frequency
Fpass2 = 0.5;          % Second Passband Frequency
Fstop2 = 0.52;         % Second Stopband Frequency
Astop1 = 80;           % First Stopband Attenuation (dB)
Apass  = 1;            % Passband Ripple (dB)
Astop2 = 80;           % Second Stopband Attenuation (dB)
match  = 'passband';   % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h  = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass,
...
                      Astop2);
Hd = design(h, 'butter', 'MatchExactly', match);

% [EOF]
```

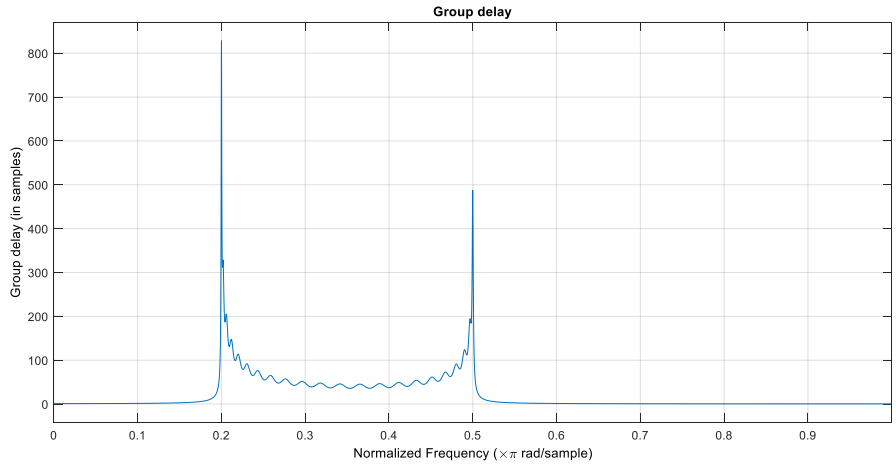




❖ IIR με αναλογικό Chebyshev I

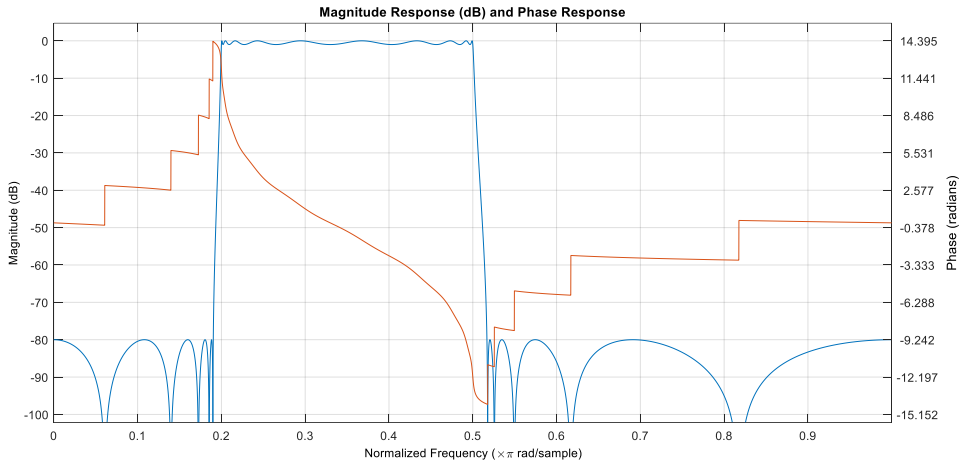
```
function Hd = chebyshevI
%CHEBYSHEVI Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.4 and Signal Processing Toolbox 8.0.
% Generated on: 09-May-2019 23:19:21

% Chebyshev Type I Bandpass filter designed using FDESIGN.BANDPASS.

% All frequency values are normalized to 1.

Fstop1 = 0.18;         % First Stopband Frequency
Fpass1 = 0.2;          % First Passband Frequency
Fpass2 = 0.5;          % Second Passband Frequency
Fstop2 = 0.52;         % Second Stopband Frequency
Astop1 = 80;           % First Stopband Attenuation (dB)
Apass  = 1;            % Passband Ripple (dB)
Astop2 = 80;           % Second Stopband Attenuation (dB)
match  = 'passband';   % Band to match exactly

% Construct an FDESIGN object and call its CHEBY1 method.
h  = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass,
...
                      Astop2);
Hd = design(h, 'cheby1', 'MatchExactly', match);

% [EOF]
```

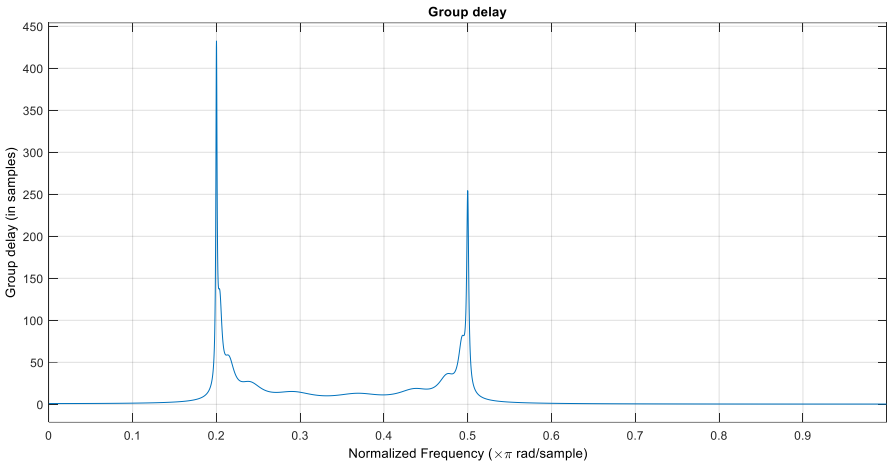

Magnitude Response (dB) and Phase Response
Magnitude (dB)
Phase (radians)
Normalized Frequency (×π rad/sample)
0
-100
-200
-300
-400
-500
-600
-700
32.682
22.979
13.276
3.573
-6.13
-15.833
-25.536
-35.24
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Group delay
Group delay (in samples)
Normalized Frequency (×π rad/sample)
800
700
600
500
400
300
200
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

❖ IIR με αναλογικό Elliptic

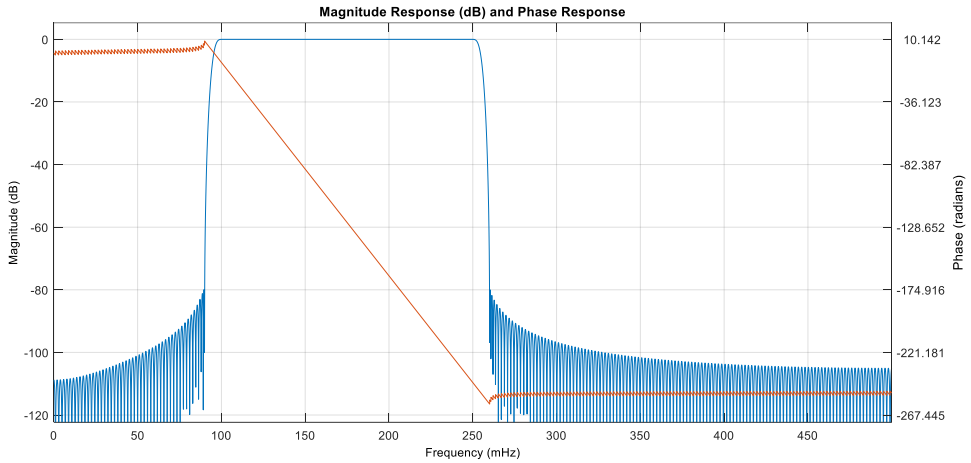
```
function Hd = elliptic
%ELLIPTIC Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.4 and Signal Processing Toolbox 8.0.
% Generated on: 09-May-2019 23:16:47

% Elliptic Bandpass filter designed using FDESIGN.BANDPASS.

% All frequency values are normalized to 1.

Fstop1 = 0.18;     % First Stopband Frequency
Fpass1 = 0.2;      % First Passband Frequency
Fpass2 = 0.5;      % Second Passband Frequency
Fstop2 = 0.52;     % Second Stopband Frequency
Astop1 = 80;       % First Stopband Attenuation (dB)
Apass  = 1;        % Passband Ripple (dB)
Astop2 = 80;       % Second Stopband Attenuation (dB)
match  = 'both';   % Band to match exactly

% Construct an FDESIGN object and call its ELLIP method.
h  = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass,
...
                      Astop2);
Hd = design(h, 'ellip', 'MatchExactly', match);

% [EOF]
```

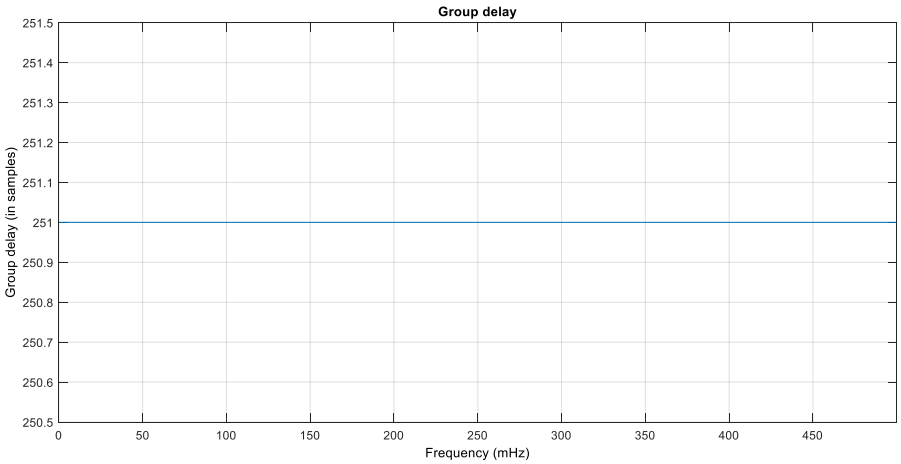




❖ FIR με Kaiser window

```
function Hd = kaiser
%KAISER Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.4 and Signal Processing Toolbox 8.0.
% Generated on: 09-May-2019 23:21:44

% FIR Window Bandpass filter designed using the FIR1 function.

% All frequency values are normalized to 1.

Fstop1 = 0.18;            % First Stopband Frequency
Fpass1 = 0.2;             % First Passband Frequency
Fpass2 = 0.5;             % Second Passband Frequency
Fstop2 = 0.52;            % Second Stopband Frequency
Dstop1 = 0.0001;          % First Stopband Attenuation
Dpass  = 0.057501127785;  % Passband Ripple
Dstop2 = 0.0001;          % Second Stopband Attenuation
flag   = 'scale';         % Sampling Flag

% Calculate the order from the parameters using KAISERORD.
[N,Wn,BETA,TYPE] = kaiserord([Fstop1 Fpass1 Fpass2 Fstop2], [0 1 0],
...
                             [Dstop1 Dpass Dstop2]);

% Calculate the coefficients using the FIR1 function.
b  = fir1(N, Wn, TYPE, kaiser(N+1, BETA), flag);
Hd = dfilt.dffir(b);

% [EOF]
```


Magnitude Response (dB) and Phase Response
Magnitude (dB)
Phase (radians)
Frequency (mHz)
0
-20
-40
-60
-80
-100
-120
10.142
-36.123
-82.387
-128.652
-174.916
-221.181
-267.445
0
50
100
150
200
250
300
350
400
450


Group delay
Group delay (in samples)
Frequency (mHz)
251.5
251.4
251.3
251.2
251.1
251
250.9
250.8
250.7
250.6
250.5
0
50
100
150
200
250
300
350
400
450