Άσκηση 2.3

* ΙIR 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]





* ΙIR 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]

****

****

* ΙIR 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]



