FDSZ - Filter Designer in "s" or "z"

By Dr. Antonio Carlos Moreirao de Queiroz
COPPE-Programa de Engenharia Eletrica
Universidade Federal do Rio de Janeiro
CP 68504
21945-970 Rio de Janeiro, RJ
Brazil

E-mail: acmq@ufrj.br

Documentation for version 2.0

The FDSZ program computes approximations in Laplace or Z transform for continuous-time or discrete-time modulus filters. It uses a simple graphical user interface, with all the commands accessible from a menu in the frequency response window.

Installation:

Only the executable file is required.

Approximations:

The program can compute:
- Low-pass filters.
- High-pass filters.
- Symmetrical band-pass filters.
- Symmetrical band-reject filters.

With the approximations:
- Butterworth: Maximally flat polynomial filter.
- Chebyshev: Maximally selective polynomial filter, with uniform passband attenuation ripple.
- Inverse Chebyshev: Same selectivity of the Chebyshev filter, but with attenuation ripple at stopband and maximally flat passband.
- Elliptic: Maximally selective rational filter, with uniform attenuation ripple at passband and stopband.
- Irregular: This approximation is a generalization of the other approximations.

Irregular Approximations:

When this approximation is selected, a new dialog box appears when the design button is pressed, where it is possible to specify the following parameters of the low-pass prototype that the program uses to compute all the approximations. Following is a summary of the meaning of the parameters in the window:

Number of attenuation zeros at zero: Controls the degree of flatness of the gain function at DC. Must be odd for odd degrees and even for even degrees.

Number of transmission zeros at infinity: Controls the slope of the filter rolloff at high frequency, and the number of finite transmission zeros. Must be odd for odd degrees and even for even degrees.

Extreme values of the normalized characteristic function (fx): Control the attenuations at each passband minimum. The default values (+/-1 in alternation) produce a filter with uniform passband ripple. Zero values produce double attenuation zeros. It is also possible to specify directly the attenuations at the minima instead of the fx values. The passband minima are counted from low to high frequency.

Extreme values of the inverted and reflected normalized characteristic function (fy): Control the attenuations at each stopband maximum. The default values (+/-1 in alternation) produce a filter with uniform stopband ripple. Zero values produce double transmission zeros. It is also possible

to specify directly the attenuations at the maxima instead of the fy values. The stopband maxima are counted from high to low frequency.

Filters with double attenuation zeros or double transmission zeros can be designed in this way, by setting appropriate extreme values (fx, fy) to 0. The only restriction about fx and fy values is that three consecutive values (with the fy being considered in reverse order) cannot increase or decrease in monotonic way, including extra zero values at zero and infinity for odd-order filters. For even-order filters, the first fx and the first fy control the attenuations at DC and infinity respectively.
The normalized low-pass characteristic function is:
K(jw)/epsilon=alpha*X(w)/(w^n*Y(1/w))
And the inverted and reflected normalized characteristic function is:
K'(jw)/epsilon=alpha*Y(w)/(w^n*X(1/w))
where w is the normalized frequency, epsilon is a constant that controls the passband attenuation, alpha another constant that controls the stopband attenuation, n is the filter order (low-pass prototype), and X(w) and Y(w) are polynomials of degree n.
The program uses these functions in the design of all the approximations. Any low-pass prototype with real characteristic function, and a combination of single and double attenuation and transmission zeros can be generated in this way.
The most interesting of these approximations are:
Even-order approximations with double attenuation zeros at DC, avoiding the less than maximum DC gain that appears in Chebyshev or elliptic even-order approximations.
Even-order approximations with double transmission zeros at infinity, avoiding the constant gain at infinity of even-order elliptic or inverse-Chebyshev filters.
Filters with double attenuation zeros at the passband border, what produces better group delay and lower Q poles, and excellent sensitivity characteristics in LC doubly terminated realizations.
Filters with double transmission zeros, what leads to physically symmetrical or antimetrical structures in LC doubly terminated designs, and better stopband sensitivity characteristics in the general case.

The discrete-time approximations can be:

- Bilinear filters: Obtained from the application of the bilinear transformation s->(2/T)(z-1)/(z+1), where T is the sampling period, to a continuous-time prototype. This is the most usual method for the generation of discrete-time filters.
- LDI filters: Obtained from the application of the LDI transformation s->(1/T)(z-1)/(z^0.5), where T is the sampling period, to a continuous time prototype. Useful in the design of polynomial discrete-time filters, because it does not generate zeros at z=-1, as the bilinear transformation does for polynomial low-pass or band-pass filters, what can lead to simpler realizations. Usually a little less selective than the bilinear filters.

Realizations:

The program designs filters in cascade of biquadratic sections. After prompting the user for the selection of pole ordering and pole-zero pairing, it designs an equalized cascade, where the maximum gain to each biquad output is equalized to 1.
The designer must first determine the order of the poles to be assigned to the biquads, by pointing with the mouse cursor one of the pole boxes and typing the number of a pole. The complex-conjugate, if existent, will be automatically added. The pole numbers can be conveniently obtained with the cursor in the poles and zeros window.
After this, the zeros shall be paired with the poles, by clicking the left mouse button in the squares corresponding to the desired pole-zero pairings. Again, complex-conjugates are automatically added.
The program designs a biquad cascade with equalized gains, and plots the gains from the input to each biquad output. The designer can then evaluate

if the ordering and pole-zero pairing chosen is satisfactory, and try new designs if not.
The program can also compute the coefficients for a particular discrete design, shown in the biquad.bmp file. The internal signal levels are not equalized.

Plots:

The program plots the frequency response and the poles and zeros diagram for the designed filters, and can compare one design with others, designed in the same session.
In the frequency response plot, the magnitude in decibels, phase in degrees, and group delay in seconds of the transfer function can be plotted. The plotted scales always refer to the magnitude plot. The frequency scale can be linear or logarithmic, in Hertz or rd/s.
In the poles and zeros plot, poles and zeros are plotted as "x" and "o" respectively, in linear scale.
Note that the default frequency unit is rad/s, and this affects the designs. It can be changed to Hertz in the frequency response parameters window.
The poles and zeros are always plotted in rad/s.
The scales in the frequency response plot can be changed directly with the following shortcuts:
"+", "-": Change the gain scale.
"a", "r": Change the frequency scale.
"<", ">": Move the frequency scale.
"g": Toggles the grid.
"m": Toggles the limits.
Vertical cursors: Move the gain scale.
And in the poles and zeros plot:
"+", "-": Change scale.
Cursor keys: Move the plot.
There is a zoom function in both plots activated by moving the mouse with the left button pressed, from the top left to the lower right.
The "z" key is a zoom-out function, that returns the scales to the limits before the last zoom-in.
The central mouse button or the space key controls a cursor, that can also be moved with the cursor keys in the frequency response plot, and points to the singularity that is closer to the mouse cursor in the poles and zeros plot.

Characteristic function and transducer function:

It is possible to plot the curves of the inverse of the characteristic function of the filter, 1/K(s), or its discrete-time equivalent 1/K(z). K(s) is related to the transfer function by the Feldtkeller equation K(s)K(-s)+1=H(s)H(-s), where H(s) is the continuous-time transducer function, inverse of the continuous-time transfer function: H(s)=1/T(s). In the nomenclature used in the program, H(s)=E(s)/P(s) and K(s)=F(s)/P(s). F(s) (and F(z)) is not available externally, but E(s) and P(s) (or E(z) and P(z)) can be saved.

Synthesis algorithm:

All the designs are made by optimization of a low-pass normalized prototype, followed by frequency transformations.
The program first computes a normalized low-pass characteristic function K(jw), as epsilon*alpha*X(w)/(w^n*Y(1/w)) using an optimization algorithm. "n" is the prototype order, "epsilon" controls the passband ripple, "alpha" controls the stopband ripple, and X(w) and Y(w) are two polynomials of degree n. These values are listed at the start of the synthesis process.
From this K(jw), K(s) is obtained, and H(s) for the low-pass prototype is computed by the resolution of Feldtkeller's equation.
From this H(s), the final continuous filter is obtained by a frequency transformation, and, if selected, a discrete-time filter is obtained by another transformation.

All the frequency transformations are done in the poles and zeros of the filter, to reduce numerical problems.

Normalization:

The continuous filters are listed and saved in frequency-normalized form, to avoid overflow in high-frequency filters.
The normalization factor is at the end of the listings and files saved.
The discrete-time filters are not normalized. Discrete-time filters with high ratio of sampling frequency to operating frequency may result imprecise in the ratio of polynomials form. The poles and zeros, however, are accurate.

Availability:

The FDSZ program is available at http://www.coe.ufrj.br/~acmq/programs.

Licensing:

The use of the FDSZ is free for academic, noncommercial purposes. For other uses, please contact the author.

Changes:

Version 1.0a: Better irregular filter design dialog box, zoom-out in the frequency response.
Version 1.0b: Corrected the design of first-order biquads, and formulae in this text.
Version 1.0d: Space works as the middle button of the mouse.
Version 2.0: Windows version. The older versions are not supported, but still work in the same way.

Last update: 7/7/2019

Antonio Carlos M. de Queiroz