University of Zagreb

Faculty of Electrical Engineering and Computing

JURASOFT

Filterpart and Filtersyn 1.0

Filter Partition and Synthesis Programs for the Elliptic Filter Design

Matlab GUI Program

U S E R S G U I D E

Zagreb, 2006.

CONTENTS

1 GENERAL INFORMATION

1.1 About Programs

On using *Jurasoft Filter Partition and Synthesis Program Pack* to design Elliptic filters in more advanced way. The program pack consists of two parts:

Part-I: FILTERPART program
Part-II: FILTERSYN program

Both programs will be described in details in this Manual. Programs use the design procedures that are documented in [1] in more details. Programs build passive ladder LC filters terminated in both ends by resistors that have very low sensitivity to component tolerances in the pass-band, according to Orchard's theorem [2]. In this way obtained ladder networks can realize whole or only one part of the standard elliptic filter's transfer function.

Those two programs are programmed as a MATLAB's Graphical User Interface (GUI) and should run under the MATLAB environment. Standalone run-time versions of the programs are not yet available. It is recommended for users to use MATLAB Version 7 and later. It is highly recommended that the resolution of the display is at least 1024x769 pixels.

1.2 Program Setup

Using Windows Explorer create the new directory, for example:

```
C:\Program Files\MATLAB71\Filterpart10
```

or open the existing directory in the MATLAB path, for example:

```
C:\Program Files\MATLAB71\work
```

Then simply copy all the files from the installation disk into the chosen directory. The file names of the files contained on the disk are:

```
1.  readme.txt
2.  usersguidejsfilterpart.doc (this file)
3.  example4.m
4.  filterpart.fig
5.  filterpart.m
6.  filterpart_help.html
7.  filtersyn.fig
8.  filtersyn.m
9.  filtersyn_help.html
10. fig1a.jpg
11. fig1b.jpg
12. fig1c.jpg
13. fig1d.jpg
```

Note: If you copy files from CD-Rom it is advisable to turn of the read only attribute. This can be done using Windows Explorer.

To proceed with installation suppose that you have installed (copied) *Jurasoft Filter Partition and Synthesis Program Pack* files into the newly created directory having the name

```
C:\Program Files\MATLAB71\Filterpart10
```

After starting MATLAB check if *yours working directory* is the directory, which contains copied program files. For example you can type:

```
>pwd
```

MATLAB notifies you, which is your current working directory. If it is not the directory containing your program files, you have two possibilities:

(*i*) change to the directory containing above program files. For example you can type:

```
>cd 'C:\Program Files\MATLAB71\Filterpart10'
```

(*ii*) check to see if your new directory containing above program files is in the MATLAB search path. If it is not you can simply add a new directory to the search path by typing at the command prompt:

```
>path(path, 'C:\Program Files\MATLAB71\Filterpart10')
```

Now you are able to run programs from any MATLAB working directory.

Note: If you want to add `'C:\Program Files\MATLAB71\Filterpart10'` path permanently to the MATLAB search path, type: `pathtool` (`pathtool` opens the Set Path dialog box, a graphical user interface you use to view and modify the MATLAB search path). Alternatively, you can select Set Path directly from the File Menu in the MATLAB desktop.

2 PART-I: ELLIPTIC FILTER PARAMETERS USING FILTERPART

Adding the Jurasoft FILTERPART program to your tool suite adds a new filter design technique of ladder LC filters terminated in both ends by resistors. Use the FILTERPART program to develop filters with Elliptic or Cauer-Chebyshev (CC) filter responses that meet more complex requirements than those you can design using standard MATLAB functions. While the designs using standard MATLAB functions are available as command line functions, the graphical user interface (GUI) of Jurasoft FILTERPART makes the design process more clear and easier to accomplish.

2.1 How to start FILTERPART program

There are two main approaches in using the FILTERPART program:

i) "Standard design" of an *N*th-order normalized prototype elliptic analog lowpass filter: To generate **all** the zeros (Z), poles (P), and gain (K) of an elliptic filter open JS Filterpart enter from the MATLAB command prompt:

```
>[Z,P,K] = filterpart
```

When FILTERPART program starts user sees a GUI in Fig. 1, which opens in the **centre** of the screen. Upon exit the program returns Z, P and K of the desired transfer function.

In the program GUI (see Fig. 1) user can easily define: filter order N, Amax (dB) of ripple in the pass-band, a stop-band Amin (dB) down, or stop-band edge frequency Fstop (the pass-band edge frequency Fpass equals to unity). User then pushes "Pole/Zero Calculate" button, and simply exits the program. This procedure of calculating P, Z and K resembles using standard built-in MATLAB function ELLIPAP, the only difference is in that the user has very comfortable way of defining specifications parameters.

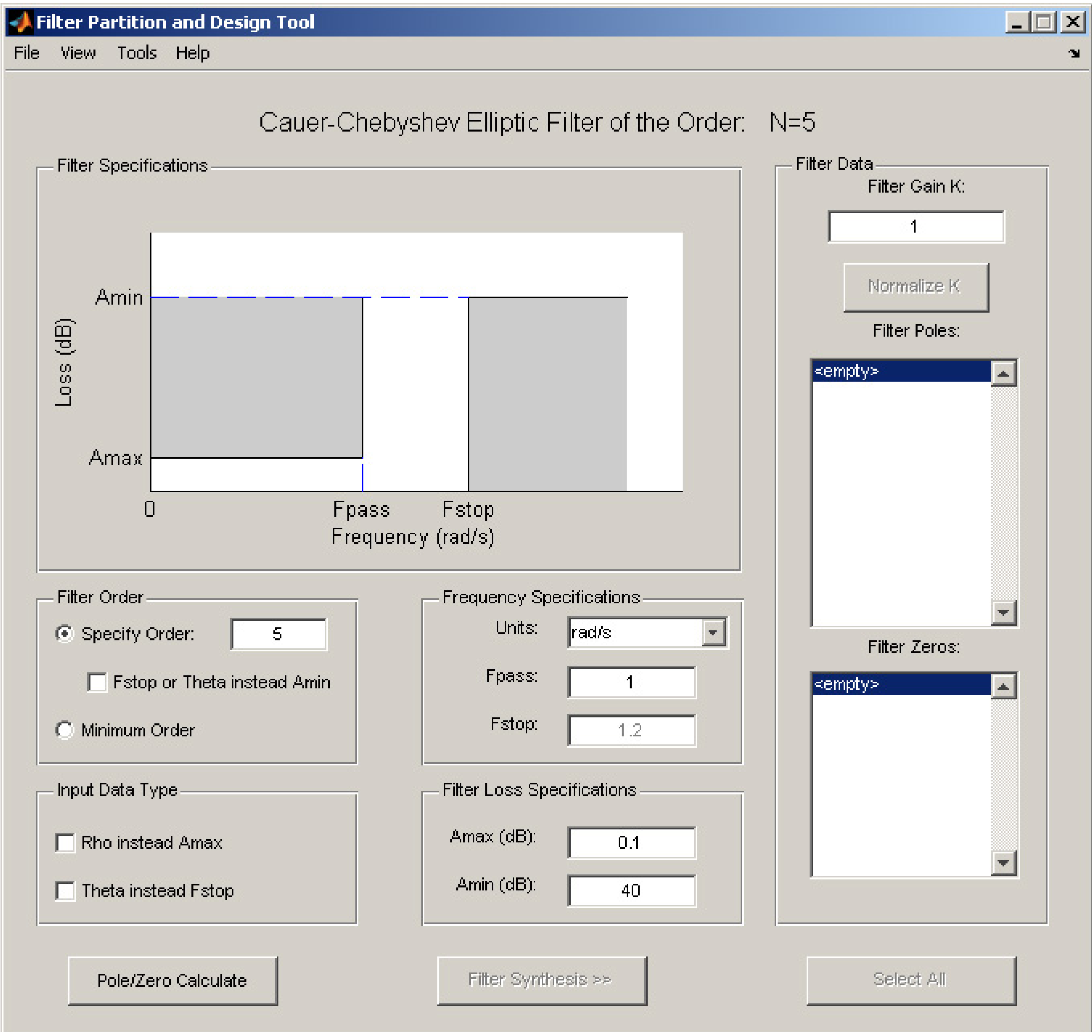


Fig. 1 JS FILTERPART program when initially open.

ii) "Filter Partition" design of an *N*th-order normalized prototype elliptic analog low-pass filter: To obtain **partial** zeros, and poles, and gain of a lower-than-Nth-order part of an elliptic low-pass filter transfer function, open JS Filterpart from the MATLAB command prompt by entering:

```
>filterpart
```

The GUI as in Fig. 1 starts. By pushing the “Pole/Zero Calculate” button the user first calculates all poles and zeros and the gain of the elliptic filter which meets the given specifications. After calculation of poles and zeros, those are presented in “Filter Poles” and “Filter Zeros” list boxes, and the gain is shown in “Filter Gain K” edit text box. List boxes display a list of items and enable users to select one or more items. User then has an opportunity to select desired poles and zeros and then push “Normalize K.” Using “Export Selected P/Z and K” command from the File Menu, the program exports variables Z, P, and K to the MATLAB ‘base’ workspace. If user selected all poles and zeros from theirs List boxes by pressing “Select All” push button, the exported Z, P, and K were identical to those in approach *i*) Standard design of an N^{th}-order normalized prototype elliptic analog LP filter.

2.2 Defining Filter Specifications Parameters

In this section various parts and panels of the FILTERPART GUI, which is shown in Fig. 1 are described. “Filter Specifications” panel shown in Fig. 2 helps users to understand the meaning of edit text boxes inside input data panels. The user can define the filter design data in the form of specifications using input data panels named “Frequency Specifications” and “Filter Loss Specifications” shown in Fig. 3.

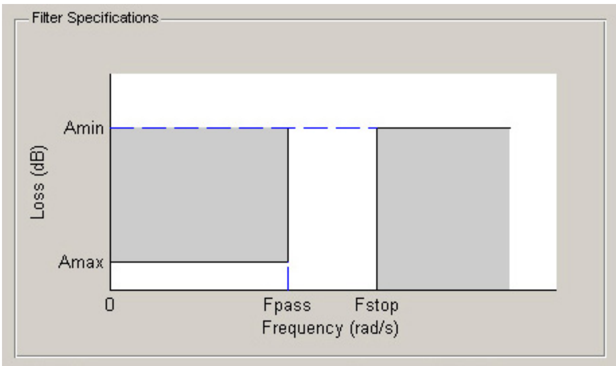


Fig. 2 Filter Specifications panel.

Edit text boxes (controls) are white rectangles in Fig. 3, containing numbers that the users can edit. Note that in “Frequency Specifications” and “Filter Loss Specifications” panels’ text labels nearby edit text boxes (controls) correspond to those in “Filter Specifications” panel shown in Fig. 2.

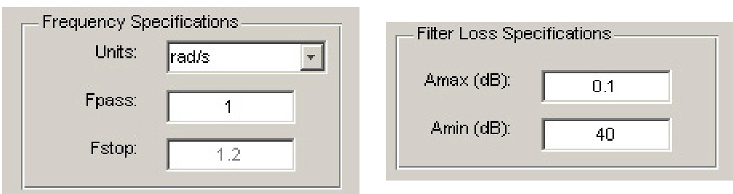


Fig. 3 Frequency and Filter Loss Specifications panels.

Users define values for the filter design using edit text controls. Once user defines the value of some parameter using edit text control, the program converts it to the double precision number and uses it with full precision in future calculations. On the other hand, the program displays only 5 to 6 decimal places in the display or in the edit text. That means the program has rounded the displayed value to 5 to 6 decimal places, but still performs calculations in full precision of 15 or more decimal places. Once users write the value into an edit text box the program accepts the value when user goes to other edit text box, or when presses enter.

GUI users can use the following available keyboard accelerators to modify the content of an edit text and to provide design data to the program:

- **Ctrl+X** – Cut
- **Ctrl+C** – Copy
- **Ctrl+V** – Paste
- **Ctrl+H** – Delete last character
- **Ctrl+A** – Select all

2.3 Filter Order

There are two mutually exclusive options for determining the filter order when you design an elliptic filter: *i*) Specify order—user enters the filter order in a text box. *ii*) Minimum order—the program determines the minimum order of the filter by itself.

In “Filter Order” panel shown in Fig. 4 user can specify the order N of the filter to design, or can instruct the program to calculate minimum order of the filter which satisfies given specifications.

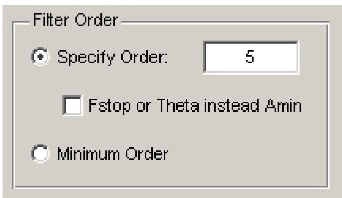
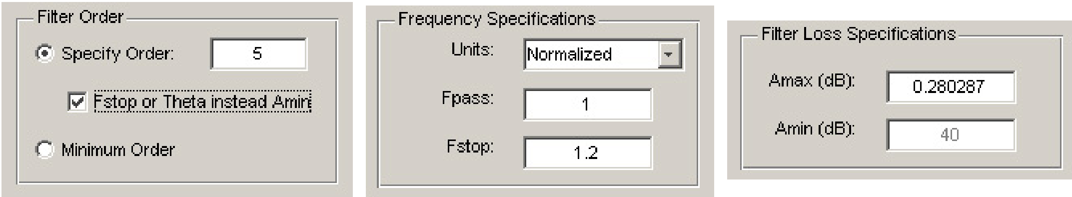


Fig. 4 Filter Order panel.

Note that the order *N* of the filter is always reported in the title on the top of the program as shown in Fig. 5.

Fig. 5 Filter Order report.

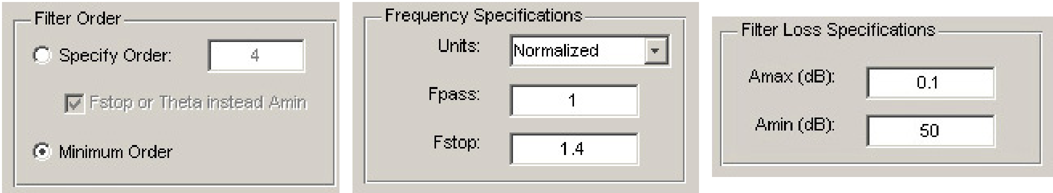
If user checks the “Fstop or Theta instead Amin” checkbox as shown in Fig. 6(a) then Fstop frequency specifications parameter can be defined as input parameter [see Fig. 6(b)], whereas the value of Amin(dB) parameter can not be entered because it is disabled for the input [see Fig. 6(c)]. Comparing to the case in Fig. 1, where “Fstop or Theta instead Amin” checkbox is not checked, on the contrary, a user is able to define Amin but not Fstop as the starting design parameter (Fstop edit text box is disabled in Fig. 1).



(a) (b) (c)

Fig. 6 (a) Filter Order panel's "Fstop or Theta instead Amin" check box is checked. (b) Influence on the "Frequency Specifications" panel; (c) Influence on the "Filter Loss Specifications" panel.

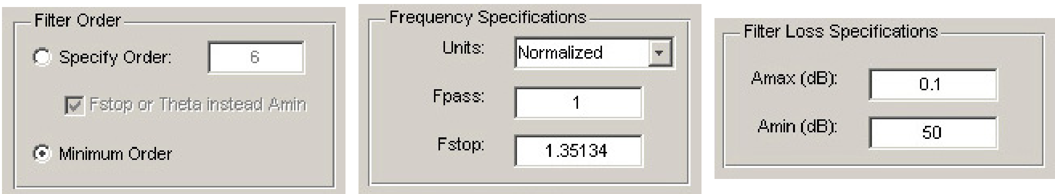
If the "Minimum Order" radio button is selected, a user can enter all specifications values as shown in Fig. 7. Note in Fig. 7(a) that the "Specify Order" edit-text box is disabled (dimmed) thus a user cannot input the filer order *N*.



(a) (b) (c)

Fig. 7 (a) The "Minimum Order" radio button is selected. (b)-(c) All specifications input fields are enabled.

After pushing "Pole/Zero Calculate" button program calculates the filter order and the minimum value of normalized Fstop frequency. The program presents those parameters as shown in Fig. 8. Besides, the program presents poles, zeros, and gain of the calculated elliptic filter in the appropriate panel.



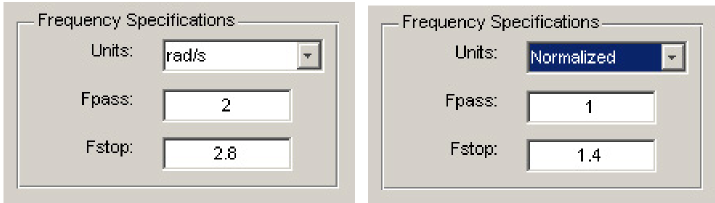
(a) (b) (c)

Fig. 8 (a) The "Minimum Order" radio button is selected and the "Pole/Zero Calculate" button has been pushed. The filter order was calculated and presented (dimmed). (b) Fstop was calculated and presented. (c) Remains the same.

Recall that the presented values are rounded to 5 decimal places, and that in the calculations the program uses more than 15 decimal places.

2.4 Various Input Data Types

The program allows the user to input filter design data in various ways. For example if user knows only real (denormalized) pass-band and stop-band edge frequencies from filter specifications, those can be very simply normalized using the program. The user can simply type in denormalized values while the Units Pop-up Menu is in position "rad/s" and then chooses "Normalized" Units. The program then calculates and displays new normalized values for Fpass and Fstop frequencies. The program normalizes both Fpass and Fstop to Fpass, where the new Fpass frequency equals to unity and the new Fstop equals to Fstop/Fpass (without any dimension).

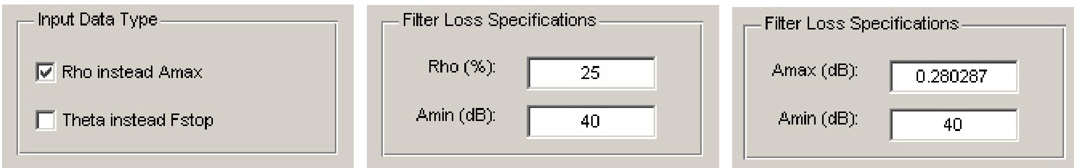


(a) (b)

Fig. 9 (a) Frequency Specifications panel's Units Pop-up Menu is in general position "rad/s". (b) When user chooses "Normalized" units, program calculates new normalized values of Fpass (=1) and Fstop.

The "Input data Type" panel shown in Fig. 10(a) and Fig. 11(a) enables the users to set various input data or to recalculate input data from one form to other. The options in the panels change to let you set (or see) the values that define your filter.

If "Rho instead Amax" check box is checked as shown in Fig. 10(a), instead Amax (dB) a user is able to enter the reflection factor Rho in percentage as shown in Fig. 10(b). The label beside the edit text box is "Rho (%)". When "Rho instead Amax" check box is unchecked the corresponding Amax (dB) value is displayed as shown in Fig. 10(c). The label changes to "Amax (dB)".



(a) (b) (c)

Fig. 10 (a) Input Data Type panel's "Rho instead Amax" check box is checked. (b) Influence on the "Filter Loss Specifications" panel when checked; (c) when not checked.

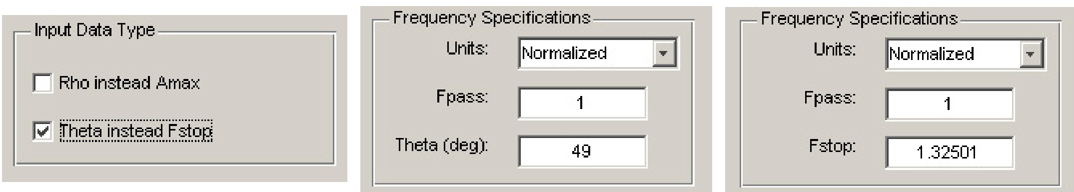
The program calculates Amax (dB) from given Rho (%) using:

$$A_{\max}(\mathrm{dB}) = -10\log\{1-[0.01\cdot\rho(\%)]^2\}\,, \tag{1}$$

or calculates Rho(%) from known Amax(dB) using:

$$\rho(\%) = 100\cdot\sqrt{1-10^{-A_{\max}(\mathrm{dB})/10}}\ . \tag{2}$$

If "Theta instead Fstop" check box is checked as shown in Fig. 11(a), instead Fstop a user is able to enter the Theta in degrees as shown in Fig. 11(b). Note that the Units have been automatically set to the "Normalized". The label beside the edit text box is "Theta (deg)". When "Theta instead Fstop" check box is unchecked the corresponding Fstop value is displayed as shown in Fig. 11(c). The label changes to "Fstop".



(a) (b) (c)

Fig. 11 (a) Input Data Type panel's "Theta instead Fstop" check box is checked. (b) Influence on the "Frequency Specifications" panel when checked; (c) when not checked.

The program calculates Fstop from given Theta(°) using:

$$F_{stop} = \frac{1}{\sin \Theta(^\circ)}, \tag{3}$$

or calculates Theta(°) from known Fstop using:

$$\Theta(^\circ) = \arcsin\left(\frac{1}{F_{stop}} \right). \tag{4}$$

Recall that the presented values are rounded to 5 decimal places, and that in the calculations the program uses more than 15 decimal places.

2.5 "Pole/Zero Calculate" Button and Pole/Zero Selection

When all design data in specifications are defined pushing "Pole/Zero Calculate" button calculates corresponding poles, zeros and gain values of the elliptic filter and fill them into "Filter Data" panel's list boxes on the right side of the GUI. Note that the Filterpart program shows strings "<empty>" before the poles and zeros are calculated.

The user has an opportunity to select desired poles and zeros in theirs list boxes simply by a mouse click, or can select them all by pushing "Select All" push button below. How users can select poles and zeros is the question how users select multiple items in a list box. List box multiple selection follows the standard for most systems:

- **Control**-click left mouse button — noncontiguous multi-item selection
- **Shift**-click left mouse button — contiguous multi-item selection
- Click left mouse button on one item inside one list box, release the button, and then **Control**-**A**— selects all items (or user can press "Select All" push button below).

Users must use one of these techniques to select desired poles and zeros.

Once a user selected desired poles and zeros there are many possible ways to proceed to the next step. In the next step user can:

i) draw magnitude in dB of the transfer function that corresponds to the selected poles and zeros,
ii) draw the pole-zero plot that corresponds to the selected poles and zeros,
iii) using the File Menu command "Export Selected P/Z and K", export the poles and zeros to the MATLAB 'base' workspace for the subsequent synthesis of the filter network,
iv) instead exporting selected P, Z and K user can proceed to the filter synthesis with those selected P, Z, and K, simply by pushing "Filter Synthesis>>" button below.

More on filter synthesis is described below in the "Part II: Filter Elements Synthesis Using FILTERSYN".

2.6 Menu Commands

Beside push buttons such as “Pole/Zero Calculate” and “Filter Synthesis”, user can use Tools Menu commands to start the same actions.

User can use the View Menu commands to present the magnitude of the transfer function that corresponds to the selected poles and zeros, pole-zero plot of selected poles and zeros or can select to see the general filter specifications. Besides, a user has an opportunity to use a Context Menu, by clicking the right mouse button over the axis area, which offers the same commands as the View Menu. (The axis area is the area of the GUI figure where filter specifications, magnitude or pole-zero plots are presented.)

If user wants to see filter magnitude in the more details user has an opportunity to use the View Menu’s “Zoom Window” as shown in Fig. 12. Zoom Window can be used to change X and/or Y Axis Min and/or Max values of the current axis. This zooming window can be applied to scale the filter pole-zero diagram, as well.

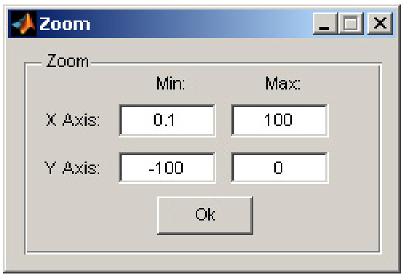


Fig. 12 Zoom Window.

When choosing to draw the magnitude of the transfer function that corresponds to the selected poles and zeros the FILTERSYN program checks if the user has correctly selected poles and zeros in complex-conjugate pairs. If not, the program takes no actions but warns the user with the following warning box, as shown in Fig. 13.

Fig. 13 Warning dialog box: select complex numbers in conjugate pairs.

After pushing an “Ok” button user has an opportunity to more carefully select poles and zeros once again. For the pole-zero plot the program does not check if poles and zeros are selected in complex-conjugate pairs.

Using Help Menu item “Jurasoft Filterpart Help” user can obtain help in html format. Using “About Jurasoft Filterpart” the About dialog box appears providing to the user with the most important data about the program and about the author. The About dialog box is displayed in Fig. 14.

Fig. 14 About Jurasoft Filterpart dialog box

On the File Menu user can choose “Print” to print the whole GUI figure (the whole program window as user can see) Finally, there exist “Exit” command to exit the program.

2.7 *Exiting the Program*

When user wants to exit the FILTERPART program by selecting File Menu “Exit” command the program displays the dialog box as shown in Fig. 15.

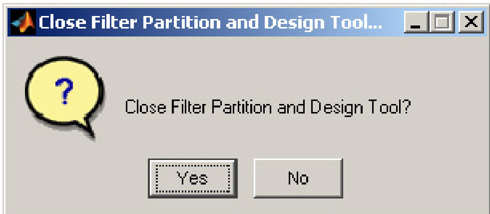


Fig. 15 Exit dialog box

In the dialog box user has an opportunity not to exit from the program by choosing “No” answer. This prevents user from accidentally closing the program and loosing design data. Furthermore, the program checks if the other application FILTERSYN is open, as well, and if it is, closes both applications.

2.8 *Launching Synthesis*

When user selects desired poles and zeros for the synthesis user can draw the pole-zero plot as a first check. Besides user can draw the magnitude of the transfer function to check if it has a maximum gain of 0dB. To correct the magnitude gain when its maximum value is other than 0dB, user has to recalculate the gain K by pressing “Normalize K” button. Now everything is ready to start the synthesis procedure.

To start the synthesis procedure user has the following two opportunities:

i) use “Export Selected P/Z and K” from the File Menu to export selected poles, zeros and gain. The program names the workspace variables by Z, P, and K. User can later start the program FILTERSYN from command prompt by entering:

```
>filtersyn(Z,P,K)
```

ii) simply press “Filter Synthesis >>” button to start the synthesis procedure.

When pushing “Filter Synthesis >>” button to start the synthesis procedure, the synthesis is started with the selected poles and zeros. Then the FILTERSYN, program checks if the user has correctly selected poles and zeros in complex-conjugate pairs. If not, the program takes no actions but warns the user with the warning dialog box as in Fig. 13: “Please select conjugate complex pairs!”. After pushing an “Ok” button, user has an opportunity to more carefully select poles and zeros once again.

2.9 Comparison of FILTERPART to standard functions for elliptic-filter design

In MATLAB there exist two standard functions ELLIPORD and ELLIPAP for elliptic filter design. In this section the comparison of those two functions to the program FILTERPART is presented. The function ELLIPORD—Elliptic filter order selection is run from the MATLAB command prompt by entering:

```
>[N, Wn] = ellipord(Fpass, Fstop, Amax, Amin, 's')
```

and returns the order *N* of the lowest order analog elliptic filter that loses no more than Amax (dB) in the passband and has at least Amin (dB) of attenuation in the stopband. Fpass and Fstop are the passband and stopband edge frequencies, in radians/second. ELLIPORD also returns Wn, the elliptic natural frequency to use with ELLIPAP to achieve the specifications.

The function ELLIPAP—Elliptic analog low-pass filter prototype is run from the MATLAB command prompt by entering:

```
>[Z,P,K] = ellipap(N,Amax,Amin)
```

and returns the zeros, poles, and gain of an *N*-th order normalized prototype elliptic analog lowpass filter with Amax (dB) of ripple in the passband and a stopband Amin (dB) down.

FILTERPART program is better than above two standard MATLAB functions because:

- FILTERPART comprises functionality of both ELLIPORD and ELLIPAP, therefore only FILTERPART function can be used. The FILTERPART is programmed as GUI and therefore is user-friendlier than those two standard MATLAB functions.
- Instead of MATLAB standard ELLIPAP function, FILTERPART uses two ultra high-speed built in functions for calculating elliptic filters transfer functions using Landens’ transformations presented by H. J. Orchard in [3];
- When in “Minimum filter Order” mode, FILTERPART first uses ELLIPORD to calculate minimum filter order, and then calculates Z, P, and K using ELLIPAP. In addition FILTERPART calculates and presents new Fstop value as in Fig. 8(b);
- When in mode “Specify Order” FILTERPART uses two built in Orchard’s ELLIPAP functions to calculate Z, P, and K. In addition FILTERPART has two input possibilities: (N, Amax, Amin) and (N, Amax, Fstop). FILTERPART calculates and presents the remaining fourth parameter Fstop and Amin for the two possibilities, respectively;

- FILTERPART can input specifications data in various forms: e.g. Theta(°) instead Fstop and Rho(%) instead Amax(dB). It can convert one input data to other;
- FILTERPART provides the ability for the user to simply select desired poles and zeros, and present them graphically, as well as the corresponding transfer function magnitude. Finally it can calculate the new gain K value to obtain maximum of the magnitude at 0dB;
- FILTERPART can simply export poles, zeros and gain K to the MATLAB 'base' workspace, or it can immediately call the filter synthesis program FILTERSYN.

Because of many advantages program FILTERPART can regularly be used in MATLAB where it can fully replace ELLIPORD and ELLIPAP when designing analog lowpass prototype filters of elliptic type. With FILTERPART the user does not need any filter tables any more (e.g. Zverev [4]), or any other design programs (e.g. MICROSIM Filter design program) to generate poles and zeros of elliptic filters. Everything you need is in FILTERPART.

Note: The *even* *N*th-order elliptic filter generated by MATLAB function ELLIPAP (and FILTERPART) has *N*/2 transfer function zeros, whereas the one tabulated in Zverev [4] has *N*/2-1 zeros. For example, 6^{th}-order CC filter transfer function in Zverev has only 2 transfer function zeros, whereas the filter obtained in MATLAB, has even 3 zeros. There exist no such difference for the *odd*-order transfer functions.

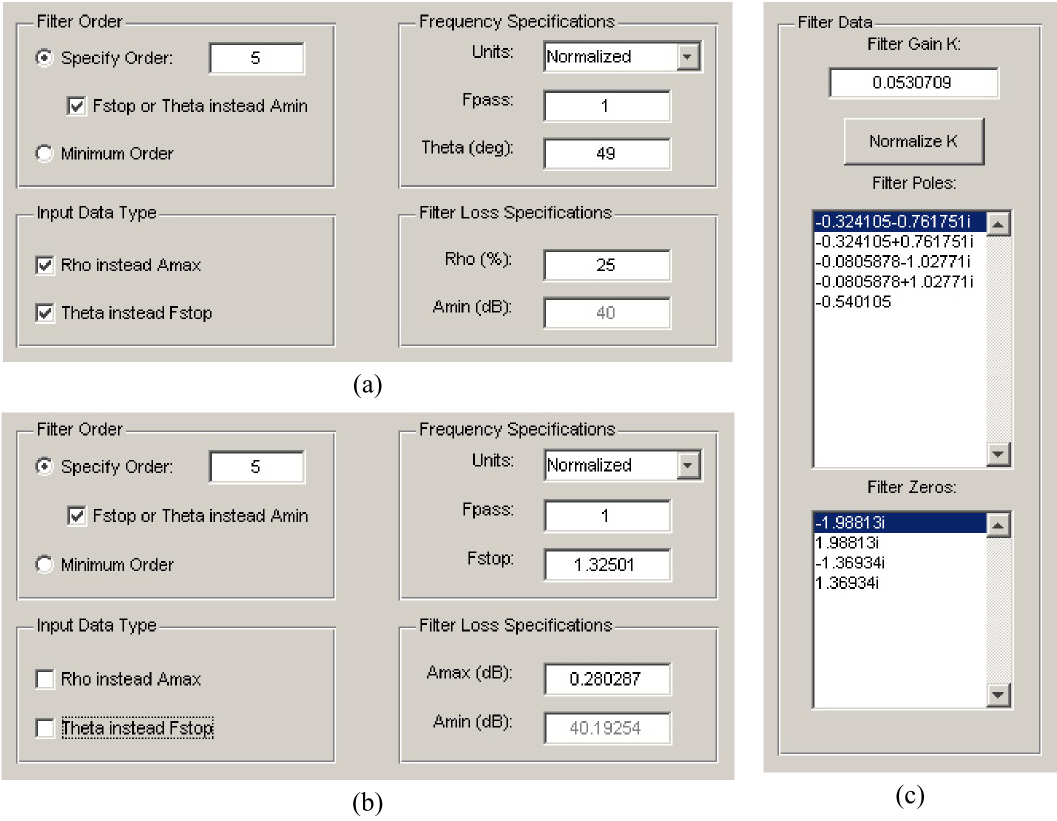
2.10 Examples

Example 1: "Standard Filter Design" generating **all** pole and zero values for the elliptic filters.

In this example we shall calculate all parameters of the standard filter CC 5 25 49. Because we know the filter order N=5 we select the "Specify Order" radio button for this example. Furthermore, to enable the input of the proper starting values we also have to check "Fstop or Theta instead Amax", "Rho instead Amax" and "Theta instead Fstop" check boxes. Now we can enter the values corresponding to CC 5 25 49 as shown in Fig. 16(a). If we want to see the corresponding Fstop and Amax(dB) values we can simply uncheck the "Rho instead Amax" and "Theta instead Fstop" check boxes as shown in Fig. 16(b) and read the values.

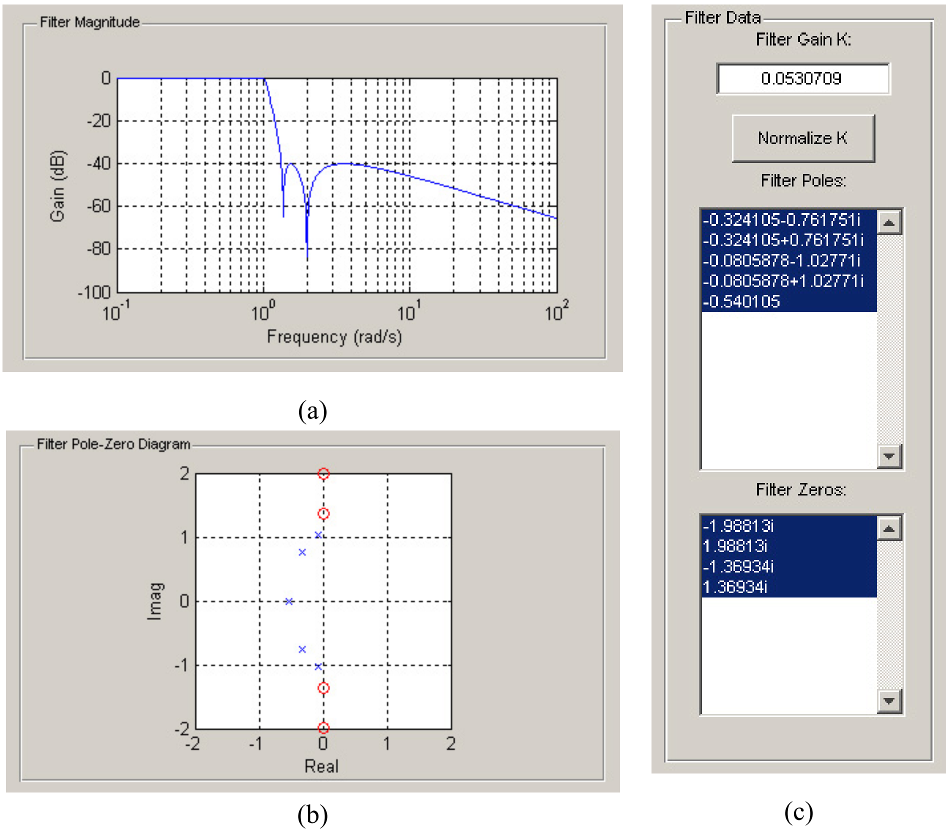
After pushing "Pole/Zero Calculate" button program calculates the value for Amin (dB) (dimmed) [see Fig. 16(b)] and fills in poles, zeros and gain K in "Filter Data" panel as shown in Fig. 16(c).

Once the poles and zeros of the standard elliptic CC 5 25 49 filter are calculated user can push "Select All" button to select all poles and zeros [see Fig. 17(c)] and then see the corresponding magnitude as in Fig. 17(a), or the corresponding pole-zero plot as in Fig. 17(b). Note that in this case no normalization of gain K is needed because the magnitude has its maximum at 0dB. This can be verified by zooming the magnitude characteristic.



(a) (b) (c)

Fig. 16 (a) Example of calculating CC 5 25 49 filter. (b) User can simply read the corresponding Fstop and Amax (dB) values. After pushing “Pole/Zero Calculate” button program calculates the value for Amin (dB) (dimmed) and (c) program presents poles, zeros and gain.

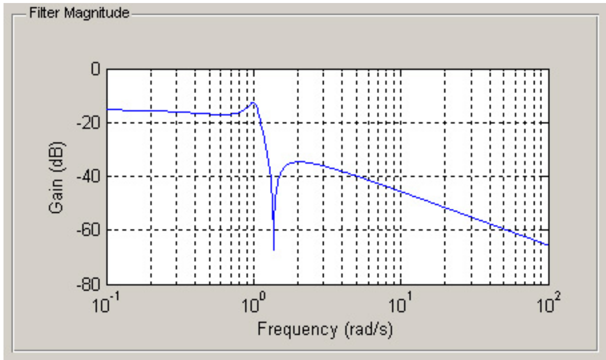


(a) (b) (c)

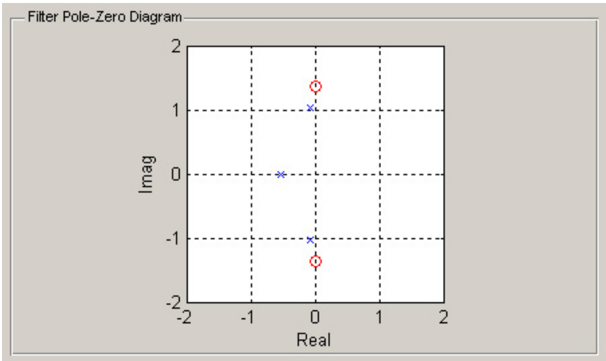
Fig. 17 (a) Magnitude of the CC 5 25 49 filter. (b) Pole zero diagram. (c) All poles and zeros selected.

Example 2: “Filter Partition Design” generating (some) **partial** pole and zero values (selected by user) for the (N-2)nd–order part of the elliptic filters transfer function.

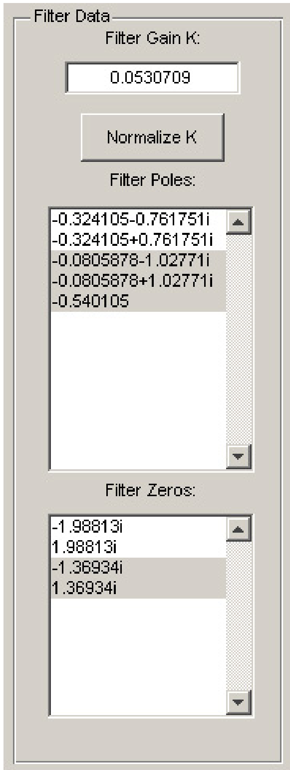
In this example we shall first calculate the parameters of the standard filter CC 5 25 49 in the same way as in the Example 1 above. Once the poles and zeros are calculated user can select desired poles and zeros as in Fig. 18(c) and then see the corresponding magnitude as in Fig. 18(a), or the corresponding pole-zero plot as in Fig. 18(b) of the part of the standard CC 5 25 49 filter. Note that in this case the normalization of the gain K is needed because the magnitude in Fig. 18(a) has its maximum less than 0dB.



(a)

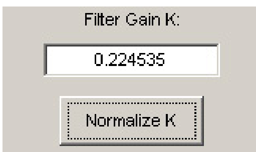


(b)

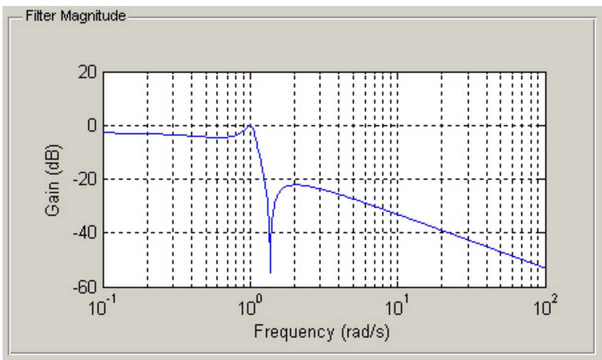


(c)

Fig. 18 (a) Magnitude of the part of the CC 5 25 49 filter.
(b) Partial pole-zero diagram. (c) Some poles and zeros selected.



(a)



(b)

Fig. 19 (a) By pressing Normalize K the new filter Gain K value is calculated and presented.
(b) When redrawing Filter Magnitude the maximum magnitude is 0dB.

User press “Normalize K” button and the new filter gain K is calculated (in double precision) and only 6 decimal places are shown as in Fig. 19(a). User has to initiate the plotting of the filters magnitude from the View Menu again and then the new magnitude as in Fig. 19(b) is redrawn showing that the maximum magnitude is now at 0dB.

In general, user can select any combination of poles and zeros. To be sure which combination was selected the user can first look at the pole zero diagram, because it presents only those that have been selected. After it user can check to see the magnitude corresponding to the selected poles, zeros and gain. Finally, when the gain K is normalized, user can again look at the magnitude (maximum).

Once the poles and zeros are selected and the optimum gain K calculated, using File Menu command “Export Selected P/Z and K”, the P, Z and K arrays could be exported to the MATLAB’s ‘base’ workspace. The program notifies that there are new variables by displaying them in the Command window as in Fig. 20.

```
Command Window

>> filterpart

Z =

        0 - 1.3693i
        0 + 1.3693i


P =

  -0.0806 - 1.0277i
  -0.0806 + 1.0277i
  -0.5401


K =

    0.2245
```

Fig. 20 MATLAB command window. The variables Z, P and K are exported to the ‘base’ workspace by pressing File Menu command “Export Selected P/Z and K”.

To check that those variables exist in the MATLAB workspace user can start the command “who” in the Command window after leaving the FILTERPART. The answer on the command “who” shows all variables in workspace as in Fig. 21.

```
Command Window

>> who

Your variables are:

K  P  Z

>>
```

Fig. 21 MATLAB command window. The variables Z, P and K exist in the ‘base’ workspace.

In the Part-II we proceed with filter synthesis.

3 PART II: FILTER ELEMENTS SYNTHESIS USING FILTERSYN

Adding the Jurasoft FILTERSYN program to your tool suite adds an Elliptic or Cauer-Chebyshev (CC) filters synthesis technique of ladder LC networks terminated in both ends by resistors. Whereas the FILTERPART has its counterparts in standard MATLAB functions such as ELLIPORD and ELLIPAP, the MATLAB in contrast does not posses any built-in synthesis program similar to FILTERSYN. Although, there exist many commercial programs for the filter synthesis (FILSYN, MICROSIM, etc.), none of them allow users to design a network from only a part of the transfer function, i.e. to design filters by “Filter Partition” as can be done using FILTERSYN.

The FILTERSYN program is capable of realizing only certain types of the networks (that realize elliptic filters, as networks in Zverev [4]). In the future it will be expanded to be able to design other standard networks. At the moment the FILTERSYN is the only existing synthesis program in the MATLAB.

3.1 How to start FILTERSYN program

To start the FILTERSYN program user has three possibilities:

i) Using FILTERPART with selected poles and zeros by pushing “Filter Synthesis>>” button (see section “Launching Synthesis” above),
ii) Use FILTERPART to export poles and zeros to the MATLAB workspace. Then type from the MATLAB command prompt (see section “Launching Synthesis” above):

```
>filtersyn(Z,P,K)
```

iii) Using poles and zeros from filter tables (e.g. Zverev). When there are no poles and zeros in the MATLAB workspace generated by any MATLAB program and user posses some other source of pole and zero values such as filter tables, user should write a script containing those pole and zero values and call a FILTERSYN program using that script. In that case FILTERPART program is not used. An example of such script is file “example4.m” (see more about scripts in MATLAB).

When FILTERSYN program starts user sees a GUI as in Fig. 22, which opens on top of the FILTERPART program slightly shifted to the right and to the top, off the screen **centre**.

There are two main approaches in using the FILTERSYN program:

i) “Standard design” of an *N*th-order normalized prototype elliptic analog low-pass filter: To generate the network, user has to select **all** the zeros and poles of an elliptic filter to start the synthesis using FILTERSYN.
ii) “Filter Partition” design of an *N*th-order normalized prototype elliptic analog low-pass filter: To generate the network user has to select **partial** zeros and poles, and recalculate the gain K of a lower-than-Nth-order part of an elliptic low-pass filter to start the synthesis using FILTERSYN.

Those two approaches slightly differ in some design steps.

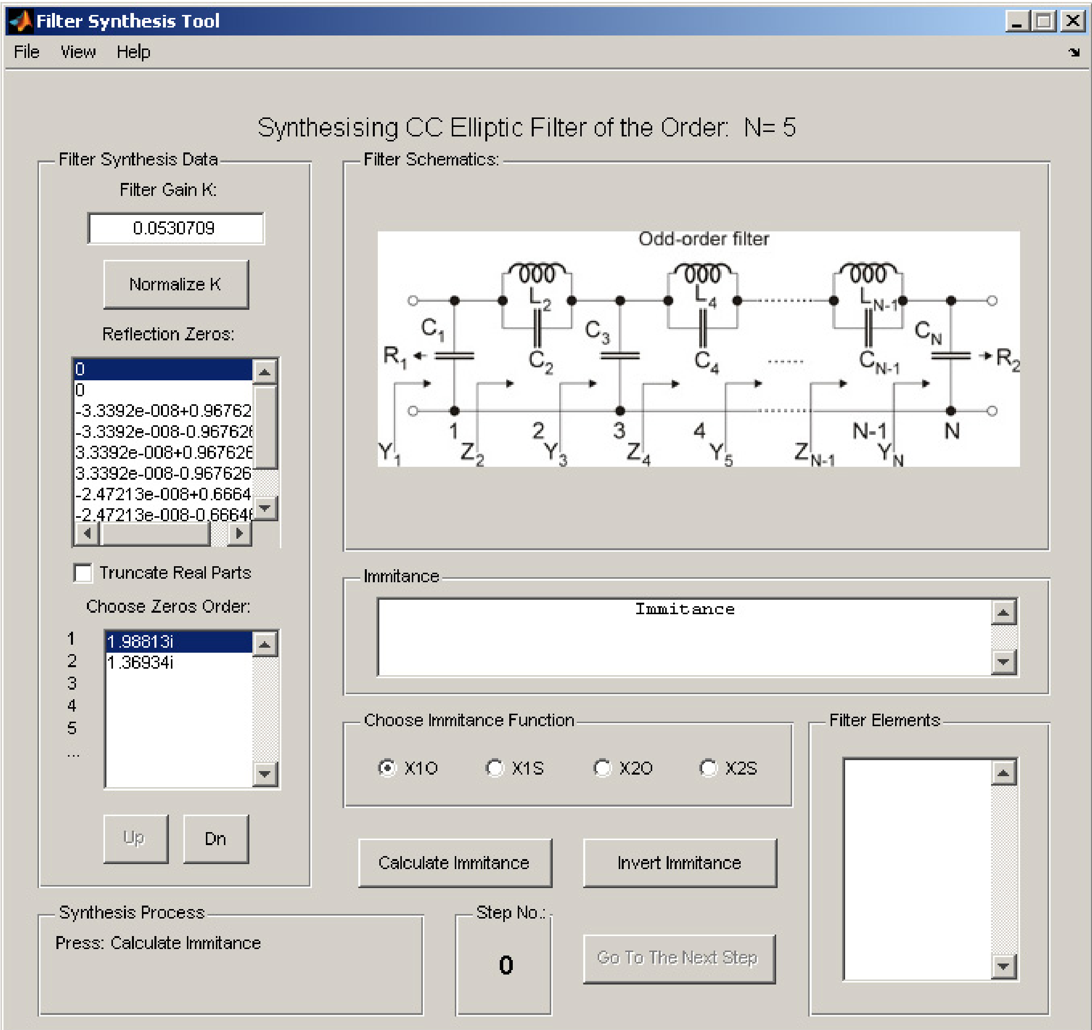


Fig. 22 JS FILTERSYN program when initially open.

3.2 *Description of the FILTERSYN program GUI*

In what follows various parts and panels of the FILTERSYN GUI, which is shown in Fig. 22 are described.

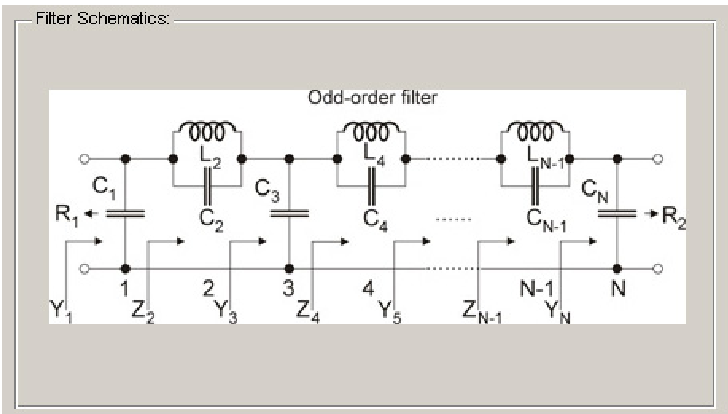


Fig. 23 Filter Schematics panel with initial network.

"Filter Schematics" panel shown in Fig. 23 presents the current filter network under synthesis. At the time of synthesis from View menu user can choose a schematic, and use it as help. User can choose among four schematics that are shown in Fig. 24.

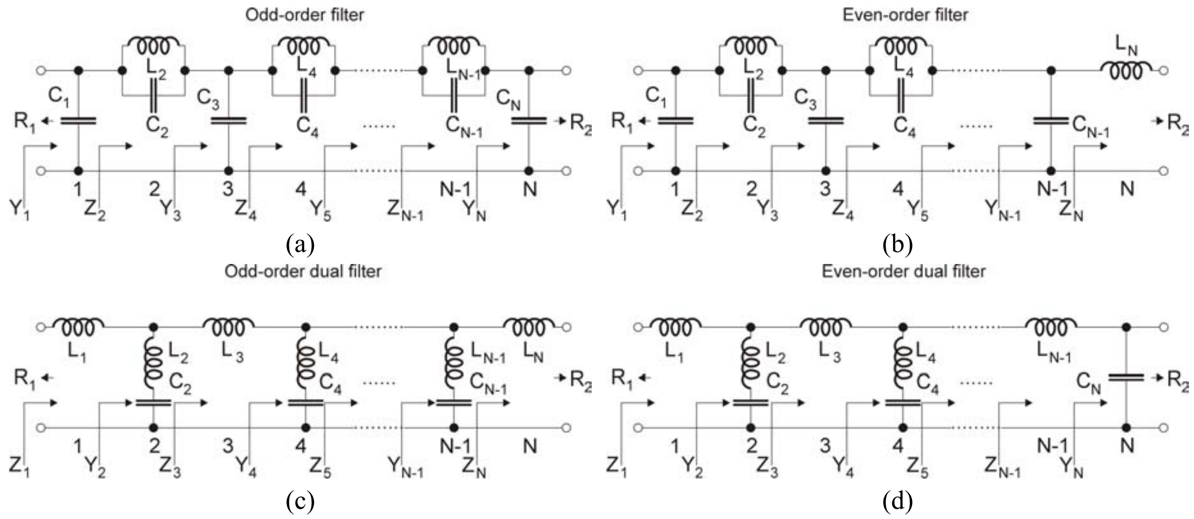


Fig. 24 Possible filter networks. (a) Odd-order filter. (b) Even-order filter. (c) Odd-order dual filter. (b) Even-order dual filter.

The FILTERPART program calculates only elements of the filter networks as shown in Fig. 24(a) and (b). If user wants to calculate elements of the dual circuits in Fig. 24(c) and (d), those have the same values as elements in the original networks at the same position but of the dual character. Thus, at the position 1, instead of shunt capacitor C_1 in the original network in Fig. 24(a), there exists series inductor L_1 in the dual network in Fig. 24(c), both having the same value. At the position 2, instead of C_2 in the original network, there exists L_2 in the dual network with the same values; L_2 in original has the same value as C_2 in dual network, and so on. There is only one difference in that the user has to calculate the value for R_2 in dual network as the reciprocal value of R_2 given in the original network. The same tabulation principle of element values is used in the most filter tables (e.g. Zverev [4]). Furthermore, at all schematics in Fig. 24 input and successive impedances Z_i or admittances Y_i are shown for the user to follow the synthesis procedure more easy.

At the time of initialisation, when the FILTERPART program starts, it calculates the square magnitude of the reflection coefficient Rho, i.e. $|\rho(j\omega)|^2$, from filter design data such as: filter poles P, zeros Z, and gain K. Recall that Z, P, and K are the input design data for the program. The reflection coefficient zeros are calculated at the time of initialisation, as well. Those are presented in the "Filter Synthesis Data" panel's List-box labelled "Reflection Zeros" as shown in Fig. 26(a). All together there are $2N$ zeros that are presented, and they are indeed zeros of the square magnitude of the reflection coefficient Rho $|\rho(j\omega)|^2$.

3.3 Menu Commands

User can use the View Menu commands to present the network schematics to be realized, the magnitude of the transfer function that corresponds to all poles and zeros in the synthesis, square magnitude of the reflection coefficient Rho, or the reflection coefficient zeros. If user wants to see filter magnitude, Rho square magnitude or Rho zeros diagram in more details user can start from the View Menu the “Zoom Window” as shown in Fig. 12 above.

Using the View menu user can choose to see the magnitude of the transfer function to be synthesized, as shown in Fig. 26(b). The user can simply check if the magnitude maximum is at 0dB. The procedure of the gain K normalization is available in the program FILTERSYN, to obtain the maximum of the magnitude to be at 0dB. When user press “Normalize K” button, the new values of reflection zeros are recalculated and presented. Furthermore, using the “Rho Square” command from the View menu user can choose to see the square magnitude of the reflection coefficient Rho, i.e. $|\rho(j\omega)|^2$, as shown in Fig. 26(c). It can provide the user with preliminary information about filters magnitude sensitivity to component tolerances in the pass-band.

On the File Menu user can choose “Export Elements” to export the calculated element values at any step of the synthesis into the MATLAB workspace, “Print” to print the whole GUI figure (the whole program window as user can see) and “Close” to exit the FILTERSYN program.

Using Help Menu item “Jurasoft Filtersyn Help” user can obtain this document in html or using “About Jurasoft Filtersyn” the About dialog box appears providing to the user with the most important data about the program and about the author. The About dialog box is displayed in Fig. 25.

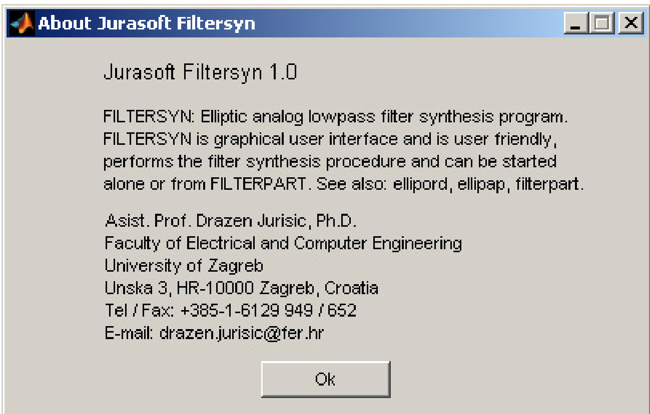
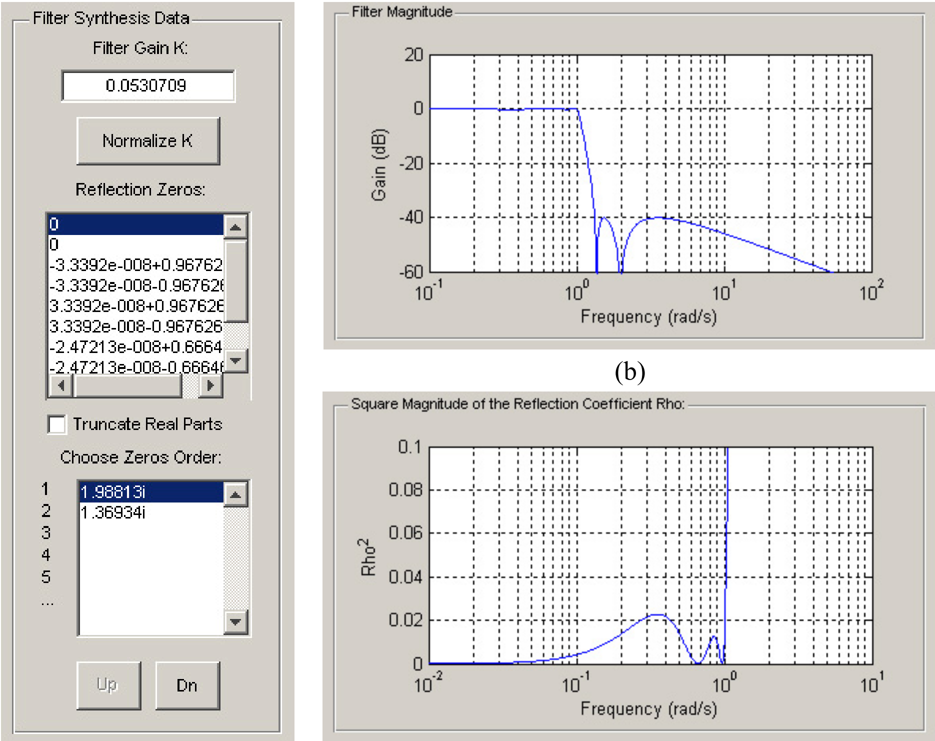


Fig. 25 About Jurasoft Filtersyn dialog box.

3.4 *“Standard design” of a CC 5 25 49 filter*

We proceed with synthesis of a CC 5 25 49 filter, i.e. with the same example which we described in FILTERPART program. In this example we design a filter network using “standard design”, i.e. we calculate all network elements and those are elements of ladder LC network terminated in resistances in both ends. All poles and zeros of the transfer function are selected before the synthesis started and the obtained network should be of the 5th order.

Besides the synthesis example of CC 5 25 49 filter, in this section we continue with the description of the FILTERSYN program GUI.



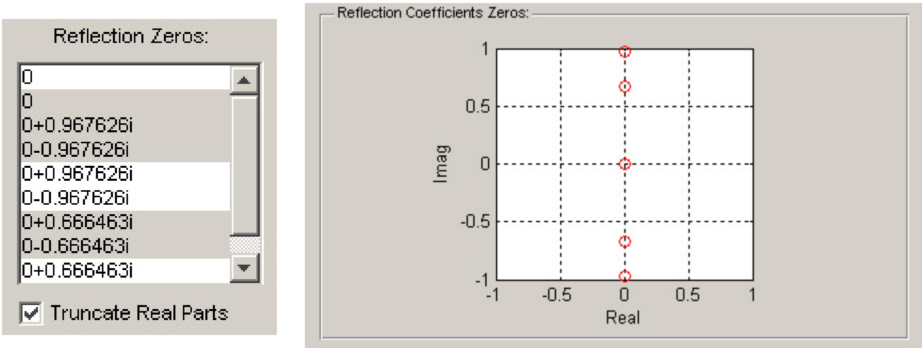
(a) (b) (c)

Fig. 26 (a) Zeros of the Reflection Coefficient Rho. (b) Magnitude of the CC 5 25 49 filter. (c) Square Magnitude of the Reflection Coefficient Rho.

In what follows the reflection coefficient Rho is denoted by $\rho(s)$, whereas its numerator is denoted by $F(s)$. The user has an opportunity to choose zeros of $\rho(s)$, i.e. "Reflection Zeros" [those are zeros of $F(s)$, as well] from the "multiple selection" List-box in Fig. 26(a). Reflection zeros should be chosen according to the following rules: because $F(s)$ does not need to be Hurwitz polynomial we are not constrained that all reflection zeros are in the left-half plane. The complex zeros of the $|\rho(j\omega)|^2$ appear in quadruplets, thus the only requirement is that if we choose one complex root, we have to choose its conjugate to obtain real coefficients of $F(s)$. The real zeros of $|\rho(j\omega)|^2$ appear only in pairs, e.g. if σ_1 is a zero then $-\sigma_1$ is zero, as well. Since only squared magnitude of reflection coefficient is given, the zero distribution of $F(s)$ is optional, as long as $F(s)\cdot F(-s)$ (and $s=j\omega$) represents the numerator of $|\rho(j\omega)|^2$. Thus $F(s)$ with all its zeros in the left-half plane is an acceptable zero distribution, the other can be obtained from this one by replacing any real-zero factor or pair of complex conjugate zeros by its mirror image, or a combination of such substitutions can be made. Recall that there are $2N$ zeros, and that we choose N of them.

In Fig. 26(a) a "single selection" List-box situated on bottom of the panel containing transfer function zeros at finite frequencies is presented. The program performs the network synthesis by full removal of poles at frequencies of those zeros in the order they are presented in "Choose Zeros Order" List-box. To every full pole removal at one finite frequency corresponds one parallel RC circuit in the network. User can choose another order of zeros by simply choosing a single zero in the List-box and shifting it up or down. This task is usually required when the synthesis procedure yields negative component values.

User has an opportunity to call “Rho zeros” command from the View menu, and the program draws in the *s*-plane, zeros selected in the “Reflection zeros” List-box. In this way user can see optically what zeros he selected. Note that Rho zeros in Fig. 26(a) have real parts that are very small, they are of the order 1e-8, so user can check on the “Truncate Real Parts” checkbox. The obtained zeros are then purely imaginary as shown in Fig. 27(a), and the corresponding zeros plot in the s-plane is in Fig. 27(b).



(a) (b)

Fig. 27 (a) Zeros of the Reflection Coefficient Rho without real part. (b) Reflection Coefficient Zeros plot.

When selected zeros, user can proceed to the synthesis (recall that user has to select *N* zeros and if they are complex they should be in complex conjugate pairs). The first synthesis step is calculating the reactance to be synthesized. User can choose among X1O, X1S, X2O or X2S from the “Choose Immitance Function” panel’s radio buttons as shown in Fig. 28. The X1 refers to the input reactance at the port 1, while the other port 2 is in open circuit (in the case “O”), or is in short circuit (in the case “S”). The X2 selection refers to the reactance at the port 2, etc. Next, user has to press “Calculate Immitance” button, and the immitance is calculated and shown in “Immitance” panel as in Fig. 31(d).

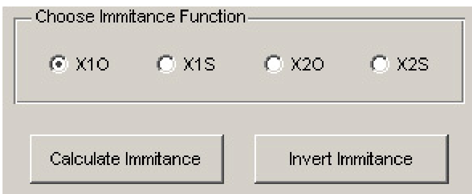


Fig. 28 Choose Immitance to calculate.

If user failed to correctly choose reflection coefficient zeros, for example if lower number of zeros has been chosen than the order of the filter *N*, or user accidentally failed to choose zeros in complex conjugate pairs, the program notifies those erroneous cases by the error messages as shown in Fig. 29, and the immitance is not calculated. After pushing an “Ok” button user has an opportunity to more carefully select Rho zeros once again. For the Rho zeros plot, the program does not check the number of zeros or if they are selected in complex-conjugate pairs

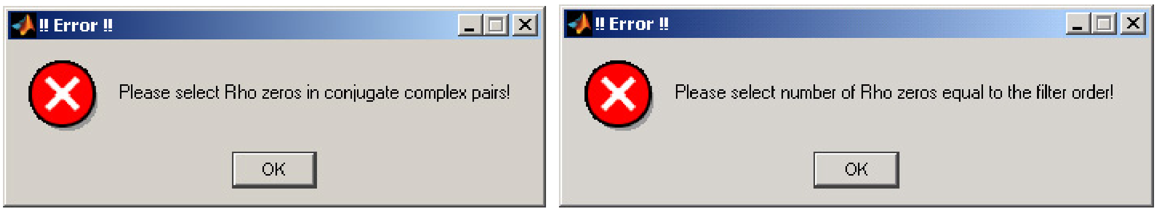


Fig. 29 Error messages when user failed to select the reflection zeros in proper way.

The message panel "Synthesis Process" shown in Fig. 30 notifies the user about the next step to be undertaken. In Fig. 30(a) the message presents a beginning and in Fig. 30(b) an end of the synthesis process. Some messages of internal steps of the synthesis process are presented in Fig. 31(a)-(c).

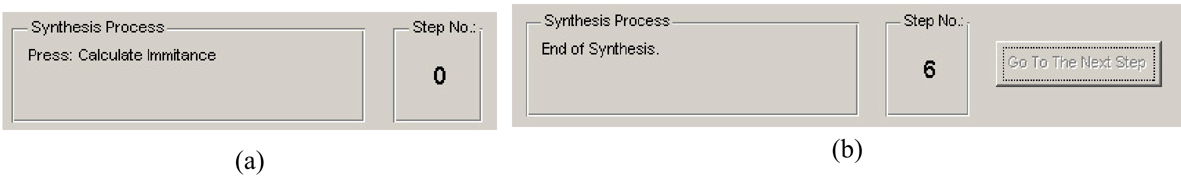


Fig. 30 Message panel "Synthesis Process". (a) At the beginning and (b) At the end of the synthesis.

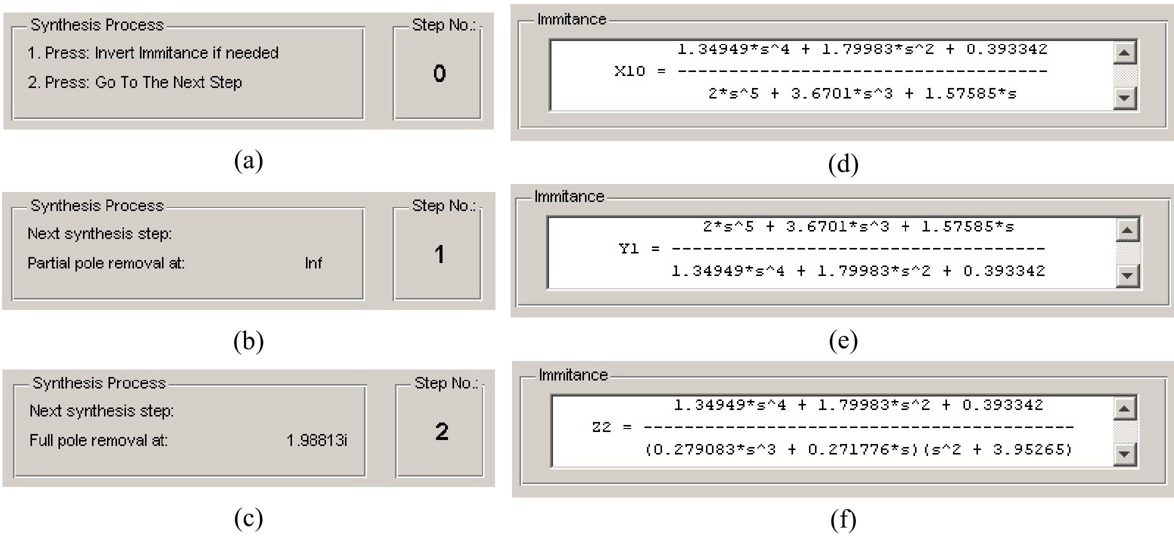
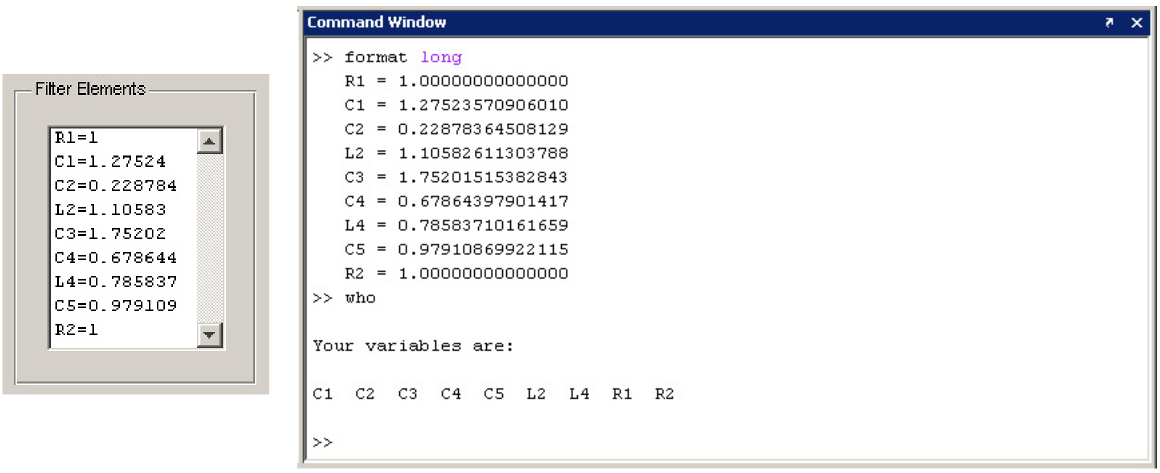


Fig. 31 (a)-(c) Message panel. (d)-(f) The calculated immitance is presented in "Immitance" panel.

The message panel in Fig. 31(a) instructs the user to first check the immitance shown in "Immitance panel" in Fig. 31(d) to see if it should be inverted. If the order of the polynomial in the numerator is lower than the order of the polynomial in the denominator user has to switch on the toggle button "Invert Immitance" and then proceed. When the user press "Go To The Next Step" push button and if the toggle button "Invert Immitance" is switched on, the new inverted reactance, i.e. susceptance is shown in "Immitance panel" as in Fig. 31(e). This is the susceptance Y_1 to be realized and the user can proceed. In the remaining figures the subsequent steps are presented. The message panel in Fig. 31(b) shows that the next operation on addmitance Y_1 will be partial pole removal at infinity, and as soon as this operation has been completed, new remaining reactance Z_2 is shown Fig. 31(f), etc. The presentation of the transfer function of the remaining immitance to be realized, helps user to keep track of the overall synthesis procedure. User can quickly see if all coefficients in the immitances have positive and real coefficients. If not, something went wrong and probably some other order of the zeros in “Choose Zeros Order” List-box in Fig. 26(a) should be arranged. After each step the calculated elements are shown in the element vales are presented in the “Filter Elements” panel as in Fig. 32(a). User can quickly see if element values are positive and real. If not, something went wrong and probably some other zeros order could fix the problem, as well.

Note that the highest order in the numerator in Y_1 is 5 and that it is the same as the network order. In some cases it can happen that the highest order is lower than the network order, and then the user has to calculate other more suitable immitance to be realized, e.g. X1S, X2O or X2S.

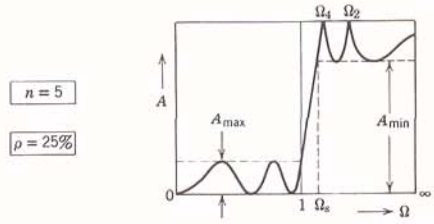
Once the all elements are calculated they are presented in the “Filter Elements” panel rounded to 6 decimal places as in Fig. 32(a). If user wants, those could be exported to the MATLAB’s ‘base’ workspace using File Menu command “Export Elements”. The program notifies that there are new variables by displaying them as in Fig. 32(b). To check that those variables exist in the MATLAB workspace user can start the command “who”, which shows all variables in workspace as in Fig. 32(b). Note that, if the “format long” command has been executed before, the elements in the command window are presented in 14 decimal places. Recall that, regardless of displaying only few decimal places, the program always performs calculations in double precision.



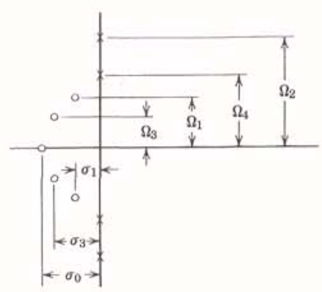
(a) (b)

Fig. 32 Filter elements. (b) MATLAB command window after pressing File Menu command “Export Elements”. The filter elements are exported and exist in the ‘base’ workspace as variables.

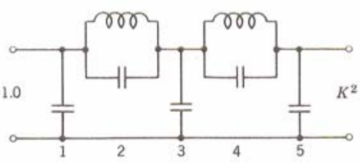
By presenting element values we conclude this example of network synthesis. Filter networks that realize calculated filters are shown in Fig. 34. If we compare the calculated values to the element values of the same circuit in some commercial filter tables we can conclude if our synthesis program has designed the same circuit. For example, in Zverev on pages 220-221 there are tables with pole-zeros values and filter elements of the CC 5 25 49 filter. The tables are repeated here in Fig. 33. Comparing our elements with those in tables we see that our synthesis program works correctly, because we have obtained the same elements.



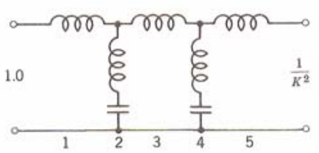
θ	Ω_K	A_{MIN}	σ_0	σ_1	σ_3	Ω_1	Ω_2	Ω_3	Ω_4
C	∞	∞	0.42450	-0.13118	-0.34343	1.0332	∞	0.6386	∞
1.0	57.2987	212.22	0.42454	-0.13115	-0.34342	1.0331	97.4775	0.6385	60.2470
2.0	28.6537	182.11	0.42465	-0.13109	-0.34341	1.0331	48.7389	0.6387	30.1274
3.0	19.1073	164.49	0.42484	-0.13098	-0.34340	1.0331	32.4927	0.6389	20.0893
4.0	14.3356	151.99	0.42511	-0.13082	-0.34338	1.0331	24.3697	0.6392	15.0716
5.0	11.4737	142.29	0.42545	-0.13063	-0.34335	1.0331	19.4959	0.6397	12.0620
6.0	9.5668	134.36	0.42587	-0.13039	-0.34332	1.0331	16.2468	0.6402	10.0565
7.0	8.2055	127.65	0.42637	-0.13010	-0.34329	1.0331	13.9260	0.6408	8.6247
8.0	7.1853	121.84	0.42695	-0.12978	-0.34324	1.0331	12.1854	0.6415	7.5516
9.0	6.3925	116.70	0.42761	-0.12941	-0.34319	1.0330	10.8316	0.6422	6.7175
10.0	5.7588	112.11	0.42834	-0.12899	-0.34313	1.0330	9.7486	0.6431	6.0507
11.0	5.2408	107.94	0.42916	-0.12854	-0.34307	1.0330	8.8625	0.6441	5.5057
12.0	4.8097	104.14	0.43006	-0.12804	-0.34299	1.0330	8.1241	0.6452	5.0520
13.0	4.4454	100.63	0.43104	-0.12749	-0.34291	1.0329	7.4993	0.6463	4.6684
14.0	4.1336	97.39	0.43210	-0.12690	-0.34282	1.0329	6.9638	0.6476	4.3401
15.0	3.8637	94.36	0.43325	-0.12627	-0.34272	1.0328	6.4997	0.6490	4.0559
16.0	3.6280	91.52	0.43448	-0.12560	-0.34261	1.0328	6.0936	0.6504	3.8076
17.0	3.4203	88.85	0.43580	-0.12489	-0.34248	1.0328	5.7353	0.6520	3.5888
18.0	3.2361	86.33	0.43720	-0.12413	-0.34235	1.0327	5.4168	0.6536	3.3946
19.0	3.0716	83.94	0.43870	-0.12333	-0.34220	1.0326	5.1318	0.6554	3.2212
20.0	2.9238	81.66	0.44029	-0.12248	-0.34204	1.0326	4.8753	0.6572	3.0654
21.0	2.7904	79.50	0.44197	-0.12160	-0.34187	1.0325	4.6433	0.6592	2.9246
22.0	2.6695	77.43	0.44375	-0.12067	-0.34168	1.0324	4.4323	0.6613	2.7970
23.0	2.5593	75.45	0.44562	-0.11970	-0.34147	1.0324	4.2397	0.6634	2.6807
24.0	2.4586	73.54	0.44759	-0.11869	-0.34125	1.0323	4.0631	0.6657	2.5743
25.0	2.3662	71.71	0.44967	-0.11763	-0.34101	1.0322	3.9007	0.6681	2.4767
26.0	2.2812	69.95	0.45185	-0.11654	-0.34074	1.0321	3.7507	0.6706	2.3868
27.0	2.2027	68.25	0.45413	-0.11540	-0.34046	1.0320	3.6119	0.6732	2.3038
28.0	2.1301	66.60	0.45653	-0.11422	-0.34015	1.0319	3.4829	0.6759	2.2270
29.0	2.0627	65.01	0.45904	-0.11301	-0.33981	1.0318	3.3629	0.6787	2.1556
30.0	2.0000	63.47	0.46166	-0.11175	-0.33945	1.0317	3.2508	0.6817	2.0892
31.0	1.9416	61.97	0.46440	-0.11045	-0.33906	1.0315	3.1460	0.6847	2.0274
32.0	1.8871	60.51	0.46727	-0.10911	-0.33864	1.0314	3.0476	0.6879	1.9695
33.0	1.8361	59.10	0.47027	-0.10773	-0.33819	1.0313	2.9553	0.6912	1.9154
34.0	1.7883	57.72	0.47339	-0.10631	-0.33770	1.0311	2.8683	0.6946	1.8646
35.0	1.7434	56.37	0.47666	-0.10485	-0.33718	1.0310	2.7864	0.6981	1.8170
36.0	1.7013	55.06	0.48006	-0.10336	-0.33662	1.0308	2.7089	0.7018	1.7722
37.0	1.6616	53.78	0.48361	-0.10182	-0.33601	1.0306	2.6356	0.7056	1.7299
38.0	1.6243	52.53	0.48731	-0.10025	-0.33535	1.0304	2.5662	0.7095	1.6901
39.0	1.5890	51.30	0.49117	-0.09864	-0.33465	1.0302	2.5003	0.7135	1.6525
40.0	1.5557	50.10	0.49519	-0.09699	-0.33390	1.0300	2.4377	0.7177	1.6170
41.0	1.5243	48.92	0.49939	-0.09531	-0.33309	1.0298	2.3781	0.7220	1.5833
42.0	1.4945	47.76	0.50376	-0.09359	-0.33222	1.0296	2.3213	0.7265	1.5515
43.0	1.4663	46.63	0.50832	-0.09183	-0.33129	1.0294	2.2672	0.7311	1.5213
44.0	1.4396	45.51	0.51307	-0.09004	-0.33029	1.0291	2.2154	0.7358	1.4926
45.0	1.4142	44.42	0.51803	-0.08822	-0.32922	1.0288	2.1660	0.7407	1.4654
46.0	1.3902	43.34	0.52320	-0.08636	-0.32807	1.0286	2.1187	0.7457	1.4396
47.0	1.3673	42.27	0.52859	-0.08446	-0.32684	1.0283	2.0733	0.5709	1.4150
48.0	1.3456	41.23	0.53422	-0.08254	-0.32551	1.0280	2.0299	0.7562	1.3916
49.0	1.3250	40.19	0.54010	-0.08058	-0.32410	1.0277	1.9881	0.7617	1.3693
50.0	1.3054	39.17	0.54624	-0.07860	-0.32258	1.0273	1.9480	0.7673	1.3481
51.0	1.2868	38.16	0.55266	-0.07658	-0.32096	1.0270	1.9095	0.7731	1.3279
52.0	1.2690	37.17	0.55937	-0.07453	-0.31922	1.0266	1.8724	0.7791	1.3087
53.0	1.2521	36.18	0.56639	-0.07246	-0.31736	1.0262	1.8366	0.7852	1.2903
54.0	1.2361	35.21	0.57374	-0.07036	-0.31537	1.0259	1.8021	0.7915	1.2728
55.0	1.2208	34.24	0.58144	-0.06823	-0.31323	1.0254	1.7689	0.7980	1.2561
56.0	1.2062	33.29	0.58952	-0.06608	-0.31095	1.0250	1.7368	0.8046	1.2402
57.0	1.1924	32.34	0.59799	-0.06390	-0.30850	1.0246	1.7057	0.8114	1.2250
58.0	1.1792	31.40	0.60690	-0.06170	-0.30588	1.0241	1.6757	0.8184	1.2104
59.0	1.1666	30.46	0.61626	-0.05948	-0.30308	1.0236	1.6467	0.8255	1.1966
60.0	1.1547	29.53	0.62612	-0.05725	-0.30008	1.0231	1.6185	0.8328	1.1834
θ	Ω_K	A_{MIN}	σ_0	σ_1	σ_3	Ω_1	Ω_2	Ω_3	Ω_4



(a)



$K^2=1.0$							$K^2=\infty$						
C_1	C_2	L_2	C_3	C_4	L_4	C_5	C_1	C_2	L_2	C_3	C_4	L_4	C_5
1.456	0.0000	1.307	2.283	0.0000	1.307	1.456	0.7280	0.0000	1.3772	1.7073	0.0000	1.6431	1.4689
1.45584	0.00008	1.30650	2.28310	0.00021	1.30635	1.45571	0.7279	0.0001	1.3771	1.7071	0.0002	1.6429	1.4687
1.45563	0.00032	1.30627	2.28240	0.00084	1.30567	1.45511	0.7277	0.0003	1.3768	1.7065	0.0007	1.6420	1.4683
1.45528	0.00073	1.30588	2.28124	0.00190	1.30454	1.45411	0.7273	0.0007	1.3762	1.7056	0.0015	1.6406	1.4676
1.45480	0.00129	1.30534	2.27962	0.00338	1.30295	1.45272	0.7268	0.0012	1.3754	1.7043	0.0027	1.6387	1.4666
1.45418	0.00202	1.30464	2.27753	0.00528	1.30092	1.45092	0.7262	0.0019	1.3745	1.7025	0.0042	1.6362	1.4654
1.45342	0.00291	1.30379	2.27498	0.00762	1.29843	1.44873	0.7254	0.0028	1.3732	1.7004	0.0061	1.6332	1.4638
1.45252	0.00396	1.30279	2.27197	0.01038	1.29549	1.44615	0.7245	0.0038	1.3718	1.6980	0.0082	1.6296	1.4620
1.45149	0.00517	1.30163	2.26849	0.01357	1.29210	1.44316	0.7234	0.0049	1.3702	1.6951	0.0108	1.6254	1.4599
1.45031	0.00655	1.30031	2.26455	0.01720	1.28826	1.43978	0.7222	0.0062	1.3683	1.6919	0.0137	1.6208	1.4575
1.44900	0.00810	1.29883	2.26016	0.02127	1.28398	1.43599	0.7209	0.0077	1.3662	1.6883	0.0169	1.6155	1.4549
1.44754	0.00981	1.29720	2.25530	0.02579	1.27924	1.43182	0.7194	0.0093	1.3639	1.6843	0.0205	1.6097	1.4519
1.44594	0.01170	1.29542	2.24999	0.03075	1.27406	1.42724	0.7177	0.0111	1.3613	1.6799	0.0244	1.6034	1.4487
1.44421	0.01375	1.29347	2.24422	0.03617	1.26843	1.42227	0.7159	0.0131	1.3586	1.6751	0.0287	1.5965	1.4452
1.44233	0.01597	1.29137	2.23799	0.04206	1.26235	1.41690	0.7140	0.0152	1.3556	1.6700	0.0334	1.5891	1.4414
1.44030	0.01836	1.28910	2.23131	0.04841	1.25583	1.41113	0.7119	0.0175	1.3524	1.6645	0.0384	1.5811	1.4374
1.43814	0.02093	1.28668	2.22417	0.05523	1.24887	1.40497	0.7097	0.0200	1.3489	1.6586	0.0439	1.5726	1.4331
1.43582	0.02368	1.28409	2.21659	0.06254	1.24147	1.39841	0.7073	0.0226	1.3453	1.6524	0.0497	1.5636	1.4285
1.43337	0.02660	1.28135	2.20855	0.07035	1.23363	1.39146	0.7048	0.0254	1.3413	1.6458	0.0558	1.5540	1.4236
1.43077	0.02970	1.27844	2.20007	0.07865	1.22535	1.38411	0.7021	0.0284	1.3372	1.6388	0.0624	1.5439	1.4184
1.42801	0.03299	1.27537	2.19114	0.08747	1.21663	1.37637	0.6993	0.0316	1.3329	1.6315	0.0694	1.5333	1.4130
1.42512	0.03646	1.27213	2.18177	0.09682	1.20748	1.36823	0.6963	0.0349	1.3283	1.6237	0.0768	1.5221	1.4073
1.42207	0.04012	1.26873	2.17196	0.10671	1.19790	1.35970	0.6931	0.0385	1.3234	1.6156	0.0846	1.5104	1.4014
1.41887	0.04397	1.26516	2.16171	0.11715	1.18788	1.35077	0.6898	0.0422	1.3184	1.6072	0.0929	1.4982	1.3952
1.41552	0.04802	1.26142	2.15102	0.12815	1.17744	1.34145	0.6864	0.0461	1.3130	1.5984	0.1016	1.4854	1.3887
1.41201	0.05226	1.25752	2.13990	0.13975	1.16658	1.33173	0.6827	0.0503	1.3075	1.5892	0.1107	1.4722	1.3819
1.40835	0.05671	1.25344	2.12834	0.15194	1.15528	1.32162	0.6789	0.0546	1.3017	1.5797	0.1204	1.4585	1.3749
1.40453	0.06136	1.24918	2.11636	0.16475	1.14357	1.31112	0.6750	0.0592	1.2957	1.5698	0.1305	1.4441	1.3676
1.40055	0.06623	1.24475	2.10395	0.17821	1.13144	1.30022	0.6708	0.0639	1.2894	1.5596	0.1411	1.4293	1.3601
1.39641	0.07130	1.24015	2.09112	0.19234	1.11889	1.28892	0.6665	0.0689	1.2828	1.5490	0.1522	1.4141	1.3522
1.39210	0.07660	1.23536	2.07787	0.20715	1.10593	1.27724	0.6621	0.0742	1.2760	1.5380	0.1638	1.3983	1.3442
1.38764	0.08212	1.23040	2.06421	0.22269	1.09256	1.26515	0.6574	0.0796	1.2690	1.5267	0.1761	1.3820	1.3358
1.38300	0.08787	1.22524	2.05013	0.23897	1.07878	1.25268	0.6526	0.0853	1.2617	1.5151	0.1888	1.3652	1.3273
1.37819	0.09386	1.21991	2.03564	0.25603	1.06460	1.23980	0.6476	0.0913	1.2541	1.5031	0.2022	1.3479	1.3184
1.37321	0.10009	1.21438	2.02075	0.27391	1.05001	1.22653	0.6423	0.0975	1.2463	1.4908	0.2162	1.3301	1.3093
1.36806	0.10657	1.20866	2.00546	0.29265	1.03503	1.21287	0.6370	0.1040	1.2382	1.4781	0.2309	1.3119	1.3000
1.36272	0.11330	1.20274	1.98978	0.31228	1.01966	1.19880	0.6314	0.1108	1.2298	1.4651	0.2462	1.2932	1.2904
1.35720	0.12030	1.19662	1.97370	0.33286	1.00389	1.18434	0.6256	0.1179	1.2211	1.4518	0.2623	1.2740	1.2805
1.35150	0.12757	1.19030	1.95723	0.35443	0.98774	1.16947	0.6196	0.1253	1.2122	1.4381	0.2791	1.2543	1.2704
1.34561	0.13513	1.18377	1.94038	0.37705	0.97121	1.15421	0.6134	0.1330	1.2030	1.4241	0.2967	1.2342	1.2601
1.33952	0.14298	1.17703	1.92316	0.40078	0.95429	1.13854	0.6069	0.1410	1.1934	1.4098	0.3151	1.2136	1.2495
1.33324	0.15112	1.17007	1.90557	0.42570	0.93701	1.12246	0.6003	0.1494	1.1836	1.3952	0.3345	1.1926	1.2387
1.32675	0.15959	1.16289	1.88761	0.45187	0.91935	1.10598	0.5934	0.1581	1.1735	1.3803	0.3547	1.1711	1.2276
1.32006	0.16837	1.15548	1.86929	0.47939	0.90133	1.08908	0.5863	0.1631	1.1631	1.3650	0.3760	1.1492	1.2163
1.31315	0.17750	1.14784	1.85061	0.50833	0.88296	1.07177	0.5790	0.1768	1.1523	1.3494	0.3983	1.1269	1.2047
1.30603	0.18698	1.13995	1.83159	0.53881	0.86423	1.05405	0.5714	0.1868	1.1412	1.3336	0.4217	1.1041	1.1929
1.29868	0.19683	1.13182	1.81224	0.57094	0.84515	1.03590	0.5636	0.1972	1.1298	1.3174	0.4464	1.0809	1.1809
1.29111	0.20707	1.12343	1.79255	0.60485	0.82573	1.01733	0.5555	0.2081	1.1180	1.3009	0.4724	1.0573	1.1687
1.28329	0.21771	1.11477	1.77253	0.64069	0.80597	0.99832	0.5471	0.2195	1.1059	1.2842	0.4997	1.0333	1.1562
1.27524	0.22878	1.10584	1.75220	0.67860	0.78589	0.97888	0.5384	0.2314	1.0934	1.2671	0.5286	1.0089	1.1435
1.26693	0.24030	1.09663	1.73156	0.71878	0.76549	0.95900	0.5295	0.2439	1.0806	1.2498	0.5591	0.9841	1.1306
1.25836	0.25229	1.08712	1.71063	0.76142	0.74477	0.93867	0.5202	0.2570	1.0673	1.2322	0.5914	0.9589	1.1174
1.24952	0.26478	1.07730	1.68941	0.80677	0.72374	0.91788	0.5106	0.2707	1.0537	1.2143	0.6256	0.9333	1.1040
1.24040	0.27781	1.06716	1.66791	0.85508	0.70243	0.89662	0.5007	0.2852	1.0396	1.1961	0.6619	0.9074	1.0904
1.23099	0.29139	1.05668	1.64615	0.90666	0.68082	0.87489	0.4904	0.3004	1.0251	1.1777	0.7005	0.8811	1.0766
1.22128	0.30559	1.04585	1.62414	0.96185	0.65894	0.85267	0.4798	0.3164	1.0102	1.1591	0.7417	0.8545	1.0626
1.21126	0.32042	1.03466	1.60189	1.02105	0.63679	0.82995	0.4688	0.3333	0.9948	1.1402	0.7857	0.8276	1.0483
1.20091	0.33595	1.02307	1.57942	1.08472	0.61439	0.80671	0.4574	0.3511	0.9789	1.1210	0.8328	0.8003	1.0339
1.19022	0.35222	1.01108	1.55674	1.15338	0.59175	0.78295	0.4455	0.3700	0.9625	1.1016	0.8833	0.7727	1.0192
1.17916	0.36930	0.99865	1.53387	1.22766	0.56888	0.75863	0.4332	0.3900	0.9456	1.0820	0.9377	0.7448	1.0044
1.16774	0.38725	0.98576	1.51083	1.30829	0.54581	0.73374	0.4202	0.4113	0.9282	1.0622	0.9965	0.7166	0.9893
L_1	L_2	C_2	L_3	L_4	C_4	L_5	L_1	L_2	C_2	L_3	L_4	C_4	L_5



(b)

Fig. 33 Tables in Zverev: CC 5 25 49 filter.

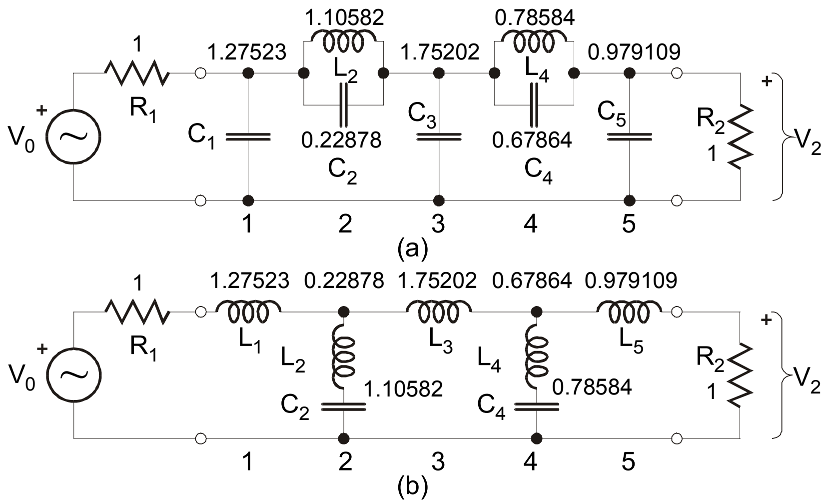


Fig. 34 (a) Doubly terminated LC-ladder realization of CC 5 25 49 filter that was originally calculated using FILTERSYN. (b) Dual network.

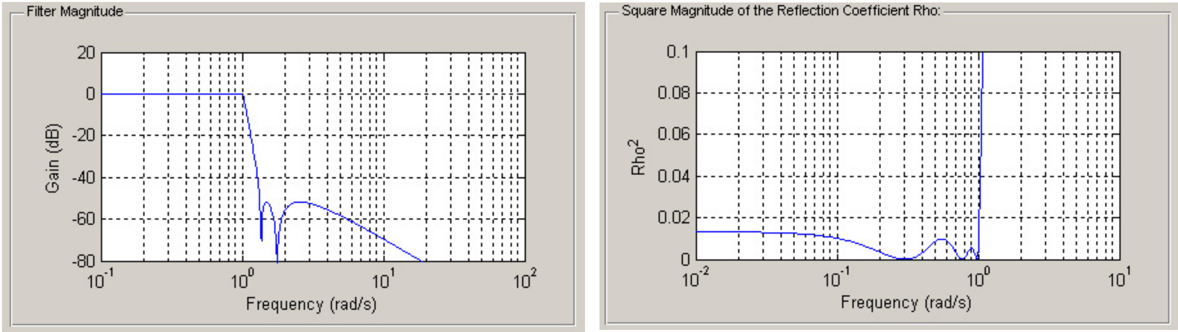
3.5 *“Standard design” of a CC 6 25 50 filter*

In this example we present standard design of the filter network when we start with **all** poles and zeros values from filter table (e.g. Zverev [4]). This corresponds to the case *iii*) of how to start FILTERSYN program. In that case a user should write a script containing poles and zeros values and call a FILTERSYN program from that script, thus the FILTERPART program is not needed. An example of the simple script file called “example4.m” is shown in Fig. 35. User had to read the pole zero values from the tables and write them down in one m-file (script). Note that the gain *K* can simply be set to unity, or any other value. That is convenient because FILTERSYN has a possibility to calculate the required *K* value for the 0 dB filter pass-band gain.

```
% These are the comment lines
% An example of P and Z values from tables
% for CC 6 25 50 filter
Z = [i*1.774078; -i*1.774078;...
     i*1.367782; -i*1.367782];
P = [-0.3960663+i*0.3447488;...
     -0.3960663-i*0.3447488;...
     -0.0587240+i*1.0187758;...
     -0.0587240-i*1.0187758;...
     -0.2151261+i*0.8318082;...
     -0.2151261-i*0.8318082];
K = 1.0;
% With defined Z, P, K user can start the synthesis
filtersyn(Z,P,K);
```

Fig. 35 Listing of the MATLAB script file for the filter synthesis with pole and zero values from filter tables (CC 6 25 50)

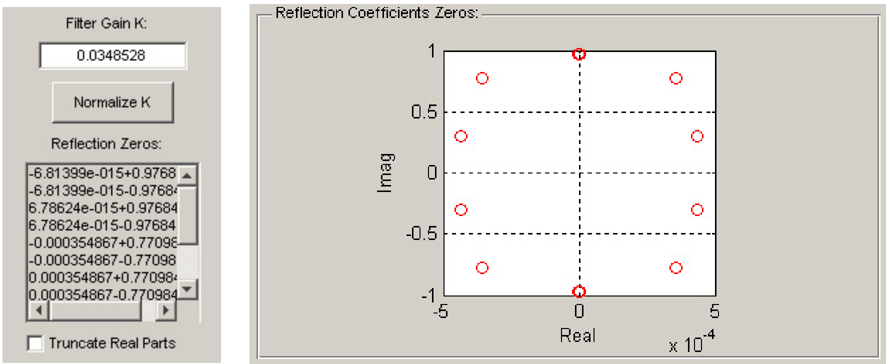
When the program FILTERSYN starts user has to press the “Normalize K” button, the program calculates the new gain *K* value and the new reflection zeros. User can check the magnitude of the transfer function as well as the square magnitude of the reflection coefficient Rho (Fig. 36).



(a) (b)

Fig. 36 (a) Magnitude of the CC 6 25 50 filter. (c) Square Magnitude of the Reflection Coefficient Rho.

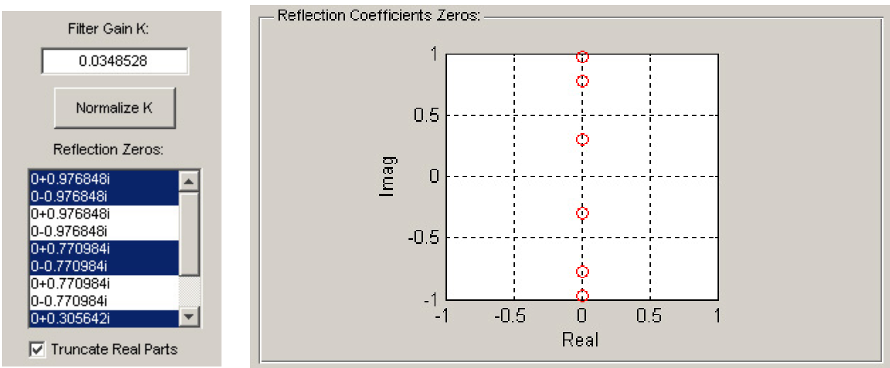
Note that the maximum magnitude of the transfer function is at 0dB. The gain *K* and zero values are shown in the list-box in Fig. 37(a). User can press Ctrl-A over the list box to select all reflection zeros and draw them in the form of pole-zero plot in Fig. 37(b).



(a) (b)

Fig. 37 (a) Calculated gain K and Zeros of the Reflection Coefficient Rho.
(b) Reflection Coefficient Zeros plot.

Now user can truncate real parts of zeros that are negligible, select 6 Rho zeros and present them in the form of pole-zero plot (see Fig. 38).



(a) (b)

Fig. 38 (a) Zeros of the Reflection Coefficient Rho without real parts.
(b) Reflection Coefficient Zeros plot.

With those data the synthesis of the network's input immitance can start. Note that in this case of the even (6^{th}-) order transfer function we use the impedance function “X1S”, which is of the same (6^{th}-) order, as well. We do not use the “X1O” because it is of the lower

(5th-) order. The “Invert Immitace” should be switched on because the denominator of “X1S” is of larger order than the numerator, thus we proceed with the realization of admittance Y_1. The remaining steps are the same as in previous example of synthesizing the CC 5 25 49 filter, and therefore will not be described here again. Finally, the obtained element values are presented in “Filter Elements” panel and exported in the workspace as shown in Fig. 39.

```
Command Window
>> example4
   R1 = 1.00000000000000
   C1 = 1.22816468989686
   C2 = 0.30102349833603
   L2 = 1.05549014030932
   C3 = 1.73950872342663
   C4 = 0.59131817422678
   L4 = 0.90395123920597
   C5 = 1.78997858222624
   L6 = 0.89588096590517
   R2 = 0.59999368473716
>>
```

Fig. 39 Filter elements in the MATLAB workspace as variables.

Filter networks that realize calculated filters are shown in Fig. 40.

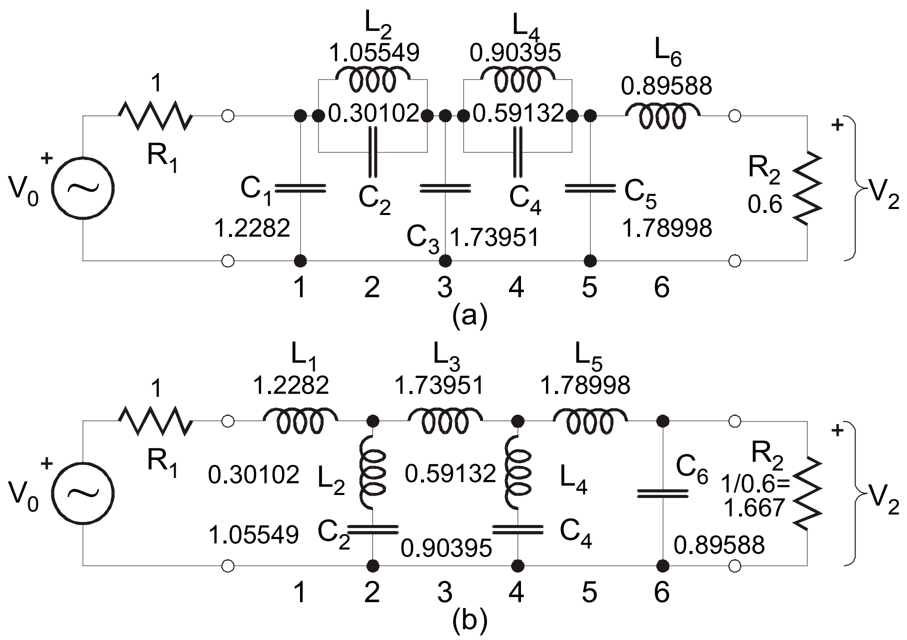
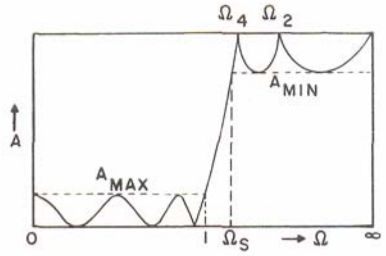


Fig. 40 (a) Doubly terminated LC-ladder realization of CC 6 25 50 filter that was originally calculated using FILTERSYN. (b) Dual network.

If we compare the calculated values to the element values of the same circuit in filter tables Zverev we can conclude that our synthesis program has designed the same circuit. In Zverev on pages 251-253 there are tables with pole-zeros values and filter element values of the CC 6 25 50 filter. The tables are repeated here in Fig. 41.

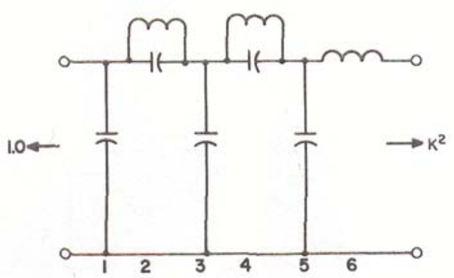
n = 6

ρ = 25%

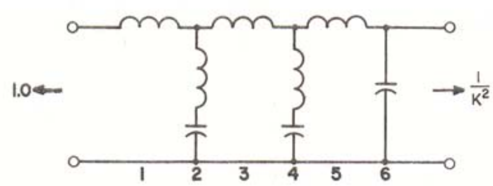


θ	Ω_S	A_{MIN}	σ_1	σ_3	σ_5	Ω_1	Ω_2	Ω_3	Ω_4	Ω_5
C	∞	∞	-0.0907744	-0.2480003	-0.3387747	1.0236117	∞	0.7493358	∞	0.2742759
11.0	5.422373	136.13	-0.0892271	-0.2467529	-0.3411936	1.0234213	7.922792	0.7530737	5.624706	0.2770783
12.0	4.975749	131.57	-0.0889329	-0.2465125	-0.3416569	1.0233847	7.263825	0.7537870	5.160811	0.2776167
13.0	4.598265	127.37	-0.0886130	-0.2462500	-0.3421617	1.0233447	6.706339	0.7545634	4.768680	0.2782039
14.0	4.275108	123.48	-0.0882675	-0.2459650	-0.3427085	1.0233013	6.228589	0.7554031	4.432936	0.2788406
15.0	3.995415	119.86	-0.0878963	-0.2456573	-0.3432974	1.0232544	5.814626	0.7563064	4.142305	0.2795272
16.0	3.751039	116.46	-0.0874995	-0.2453264	-0.3439290	1.0232041	5.452491	0.7572736	3.888329	0.2802645
17.0	3.535748	113.26	-0.0870769	-0.2449721	-0.3446036	1.0231502	5.133037	0.7583050	3.664543	0.2810531
18.0	3.344698	110.25	-0.0866287	-0.2445938	-0.3453217	1.0230927	4.849152	0.7594009	3.465915	0.2818938
19.0	3.174064	107.39	-0.0861548	-0.2441910	-0.3460838	1.0230315	4.595218	0.7605618	3.288476	0.2827873
20.0	3.020785	104.67	-0.0856552	-0.2437634	-0.3468903	1.0229667	4.366743	0.7617880	3.129050	0.2837345
21.0	2.882384	102.08	-0.0851298	-0.2433104	-0.3477418	1.0228980	4.160091	0.7630799	2.985065	0.2847363
22.0	2.756834	99.61	-0.0845787	-0.2428314	-0.3486389	1.0228254	3.972284	0.7644380	2.854418	0.2857938
23.0	2.642462	97.24	-0.0840019	-0.2423258	-0.3495823	1.0227489	3.800865	0.7658627	2.735370	0.2869078
24.0	2.537873	94.97	-0.0833993	-0.2417930	-0.3505724	1.0226683	3.643786	0.7673544	2.626475	0.2880796
25.0	2.441895	92.79	-0.0827709	-0.2412323	-0.3516101	1.0225837	3.499325	0.7689138	2.526516	0.2893103
26.0	2.353536	90.69	-0.0821168	-0.2406431	-0.3526961	1.0224948	3.366027	0.7705412	2.434463	0.2906012
27.0	2.271953	88.66	-0.0814368	-0.2400245	-0.3538312	1.0224016	3.242651	0.7722373	2.349441	0.2919535
28.0	2.196422	86.70	-0.0807311	-0.2393759	-0.3550161	1.0223041	3.128134	0.7740027	2.270699	0.2933688
29.0	2.126320	84.81	-0.0799996	-0.2386962	-0.3562517	1.0222020	3.021559	0.7758379	2.197588	0.2948485
30.0	2.061105	82.98	-0.0792423	-0.2379847	-0.3575389	1.0220953	2.922133	0.7777435	2.129549	0.2963942
31.0	2.000308	81.20	-0.0784593	-0.2372404	-0.3588788	1.0219839	2.829163	0.7797202	2.066092	0.2980077
32.0	1.943517	79.47	-0.0776505	-0.2364622	-0.3602722	1.0218676	2.742043	0.7817687	2.006790	0.2996906
33.0	1.890370	77.79	-0.0768159	-0.2356493	-0.3617203	1.0217464	2.660243	0.7838896	1.951268	0.3014450
34.0	1.840548	76.15	-0.0759555	-0.2348003	-0.3632242	1.0216201	2.583292	0.7860837	1.899195	0.3032727
35.0	1.793769	74.56	-0.0750694	-0.2339142	-0.3647850	1.0214885	2.510774	0.7883516	1.850277	0.3051761
36.0	1.749781	73.00	-0.0741577	-0.2329898	-0.3664041	1.0213515	2.442321	0.7906943	1.804254	0.3071573
37.0	1.708362	71.49	-0.0732202	-0.2320256	-0.3680826	1.0212091	2.377602	0.7931123	1.760893	0.3092188
38.0	1.669312	70.01	-0.0722571	-0.2310204	-0.3698221	1.0210609	2.316322	0.7956066	1.719987	0.3113631
39.0	1.632449	68.56	-0.0712684	-0.2299727	-0.3716239	1.0209069	2.258218	0.7981780	1.681350	0.3135930
40.0	1.597615	67.14	-0.0702541	-0.2288808	-0.3734897	1.0207469	2.203049	0.8008273	1.644814	0.3159113
41.0	1.564662	65.75	-0.0692144	-0.2277433	-0.3754209	1.0205808	2.150602	0.8035554	1.610227	0.3183212
42.0	1.533460	64.39	-0.0681492	-0.2265582	-0.3774193	1.0204083	2.100682	0.8063633	1.577454	0.3208259
43.0	1.503888	63.05	-0.0670586	-0.2253240	-0.3794868	1.0202293	2.053113	0.8092517	1.546369	0.3234289
44.0	1.475840	61.74	-0.0659428	-0.2240385	-0.3816252	1.0200435	2.007733	0.8122217	1.516862	0.3261339
45.0	1.449216	60.45	-0.0648019	-0.2226997	-0.3838365	1.0198508	1.964398	0.8152741	1.488829	0.3289449
46.0	1.423927	59.19	-0.0636358	-0.2213055	-0.3861229	1.0196509	1.922972	0.8184101	1.462178	0.3318662
47.0	1.399891	57.94	-0.0624449	-0.2198535	-0.3884867	1.0194438	1.883335	0.8216305	1.436822	0.3349022
48.0	1.377032	56.71	-0.0612292	-0.2183414	-0.3909301	1.0192290	1.845375	0.8249363	1.412684	0.3380578
49.0	1.355282	55.50	-0.0599888	-0.2167665	-0.3934557	1.0190064	1.808987	0.8283285	1.389693	0.3413381
50.0	1.334577	54.31	-0.0587240	-0.2151261	-0.3960663	1.0187758	1.774078	0.8318082	1.367782	0.3447488
51.0	1.314859	53.14	-0.0574350	-0.2134172	-0.3987646	1.0185369	1.740561	0.8353762	1.346891	0.3482958
52.0	1.296076	51.98	-0.0561220	-0.2116367	-0.4015537	1.0182893	1.708356	0.8390337	1.326965	0.3519854
53.0	1.278176	50.83	-0.0547852	-0.2097814	-0.4044369	1.0180330	1.677386	0.8427815	1.307952	0.3558246
54.0	1.261116	49.70	-0.0534249	-0.2078477	-0.4074176	1.0177676	1.647585	0.8466206	1.289804	0.3598207
55.0	1.244853	48.58	-0.0520415	-0.2058317	-0.4104994	1.0174927	1.618888	0.8505520	1.272479	0.3639819
56.0	1.229348	47.47	-0.0506353	-0.2037296	-0.4136863	1.0172082	1.591235	0.8545765	1.255935	0.3683166
57.0	1.214564	46.37	-0.0492066	-0.2015371	-0.4169825	1.0169136	1.564571	0.8586949	1.240135	0.3728344
58.0	1.200469	45.29	-0.0477559	-0.1992494	-0.4203927	1.0166088	1.538846	0.8629080	1.225044	0.3775453
59.0	1.187032	44.21	-0.0462836	-0.1968618	-0.4239216	1.0162932	1.514011	0.8672164	1.210630	0.3824605
60.0	1.174224	43.14	-0.0447904	-0.1943690	-0.4275746	1.0159667	1.490022	0.8716207	1.196863	0.3875919
θ	Ω_S	A_{MIN}	σ_1	σ_3	σ_5	Ω_1	Ω_2	Ω_3	Ω_4	Ω_5

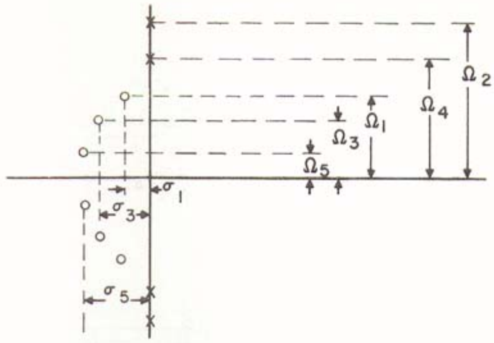
(a)



(b)



(c)



θ	C_1	C_2	L_2	C_3	C_4	L_4	C_5	L_6
	K²=0.6000							
C	1.476	0.00000	1.330	2.353	0.00000	1.412	2.216	0.8855
11.0	1.465	0.01210	1.317	2.323	0.02280	1.387	2.195	0.8860
12.0	1.463	0.01442	1.315	2.317	0.02717	1.382	2.191	0.8861
13.0	1.460	0.01695	1.312	2.311	0.03194	1.377	2.187	0.8862
14.0	1.458	0.01969	1.309	2.304	0.03712	1.371	2.182	0.8863
15.0	1.455	0.02264	1.306	2.297	0.04270	1.365	2.177	0.8865
16.0	1.452	0.02582	1.303	2.289	0.04869	1.358	2.172	0.8866
17.0	1.449	0.02921	1.299	2.281	0.05510	1.352	2.166	0.8867
18.0	1.446	0.03282	1.296	2.272	0.06193	1.344	2.160	0.8869
19.0	1.442	0.03666	1.292	2.263	0.06919	1.336	2.154	0.8870
20.0	1.439	0.04073	1.288	2.253	0.07689	1.328	2.147	0.8872
21.0	1.435	0.04502	1.283	2.243	0.08503	1.320	2.140	0.8873
22.0	1.431	0.04956	1.279	2.232	0.09362	1.311	2.132	0.8875
23.0	1.426	0.05433	1.274	2.221	0.1027	1.302	2.125	0.8877
24.0	1.422	0.05935	1.269	2.209	0.1122	1.292	2.117	0.8879
25.0	1.417	0.06462	1.264	2.197	0.1222	1.282	2.108	0.8881
26.0	1.412	0.07014	1.258	2.184	0.1327	1.271	2.099	0.8883
27.0	1.407	0.07592	1.253	2.171	0.1438	1.260	2.090	0.8885
28.0	1.402	0.08197	1.247	2.158	0.1553	1.249	2.081	0.8887
29.0	1.397	0.08828	1.241	2.144	0.1674	1.237	2.071	0.8890
30.0	1.391	0.09488	1.234	2.129	0.1800	1.225	2.061	0.8892
31.0	1.385	0.1018	1.228	2.114	0.1932	1.213	2.051	0.8894
32.0	1.379	0.1089	1.221	2.098	0.2070	1.200	2.040	0.8897
33.0	1.373	0.1164	1.214	2.082	0.2214	1.186	2.029	0.8900
34.0	1.366	0.1242	1.207	2.066	0.2364	1.173	2.017	0.8902
35.0	1.359	0.1323	1.199	2.049	0.2521	1.159	2.006	0.8905
36.0	1.352	0.1407	1.191	2.031	0.2685	1.144	1.994	0.8908
37.0	1.345	0.1495	1.183	2.013	0.2855	1.129	1.981	0.8911
38.0	1.337	0.1586	1.175	1.995	0.3034	1.114	1.969	0.8914
39.0	1.330	0.1681	1.167	1.976	0.3220	1.099	1.956	0.8917
40.0	1.322	0.1780	1.158	1.957	0.3414	1.083	1.942	0.8920
41.0	1.314	0.1882	1.149	1.937	0.3616	1.066	1.928	0.8924
42.0	1.305	0.1989	1.139	1.917	0.3828	1.050	1.914	0.8927
43.0	1.296	0.2100	1.130	1.896	0.4049	1.033	1.900	0.8930
44.0	1.288	0.2215	1.120	1.875	0.4280	1.015	1.885	0.8934
45.0	1.278	0.2335	1.110	1.854	0.4522	0.9977	1.870	0.8937
46.0	1.269	0.2459	1.100	1.832	0.4775	0.9796	1.855	0.8941
47.0	1.259	0.2589	1.089	1.809	0.5040	0.9612	1.840	0.8945
48.0	1.249	0.2724	1.078	1.786	0.5317	0.9424	1.824	0.8949
49.0	1.239	0.2864	1.067	1.763	0.5608	0.9233	1.807	0.8952
50.0	1.228	0.3011	1.055	1.739	0.5914	0.9038	1.791	0.8956
51.0	1.217	0.3163	1.044	1.715	0.6236	0.8840	1.774	0.8960
52.0	1.206	0.3322	1.032	1.690	0.6574	0.8538	1.757	0.8964
53.0	1.194	0.3488	1.019	1.665	0.6931	0.8433	1.739	0.8969
54.0	1.183	0.3661	1.006	1.640	0.7308	0.8225	1.721	0.8973
55.0	1.170	0.3842	0.9932	1.614	0.7707	0.8014	1.703	0.8977
56.0	1.158	0.4031	0.9798	1.588	0.8129	0.7799	1.684	0.8981
57.0	1.145	0.4230	0.9660	1.561	0.8577	0.7581	1.665	0.8986
58.0	1.131	0.4438	0.9518	1.534	0.9053	0.7360	1.646	0.8990
59.0	1.118	0.4656	0.9373	1.506	0.9561	0.7136	1.627	0.8995
60.0	1.104	0.4885	0.9223	1.478	1.010	0.6909	1.607	0.8999
θ	L_1	L_2	C_2	L_3	L_4	C_4	L_5	C_6

(d)

Fig. 41 Tables in Zverev: CC 6 25 50 filter underlined by red line.

3.6 "Filter partition" design of a CC 6 25 50 filter

In this short example we present the design case of the filter network when we start with poles and zeros values from filter table (e.g. Zverev). The difference to the example above is in that we realize only **part** of the transfer function; therefore it is the "filter partition" design rather than "standard design". As in the previous example, user writes a script containing pole and zero values, and calls a FILTERSYN program from that script. Note that some of the poles and zeros that will be realized by a separate biquad are simply left out. In our example those are the poles having middle pole Q factor and zeros near the cut-off frequency, although other possibilities can be used. An example of the simple script file called "example5.m" is shown in Fig. 35. Note that it was readily edited from script "example4.m" by simply deleting the lines containing superfluous zeros and poles. The RLC ladder network to be synthesised is of the 4th order and has only one zero pair.

```
% These are the comment lines
% An example of partial P and Z values from tables
% for CC 6 25 50 filter
Z = [i*1.774078; -i*1.774078];
P = [-0.3960663+i*0.3447488;...
     -0.3960663-i*0.3447488;...
     -0.0587240+i*1.0187758;...
     -0.0587240-i*1.0187758];
K = 1.0;
% With defined Z, P, K user can start the synthesis
filtersyn(Z,P,K);
```

Fig. 42 Listing of the MATLAB script file for the filter partition synthesis
with pole and zero values from filter tables (part of CC 6 25 50)

The procedure can be accomplished in the same way as above but with one exception: real parts of the reflection coefficient zeros cannot be neglected, thus we are not allowed to check on the "Truncate Real Parts" check box. Thus there arise an opportunity to choose different combinations of the reflection coefficient zeros. This situation will be demonstrated in the following example. In the following example we will convey the whole design procedure "by partition" on the 7th-order CC filter in more detail.

3.7 "Filter partition" design of a CC 7 25 50 filter

We proceed with design of a CC 7 25 50 filter using both FILTERPART program to calculate design data and FILTERSYN program to synthesize a network. In this example we design a filter network using "filter partition design", i.e. we calculate elements of ladder LC network terminated in resistances in both ends for the **part** of the CC 7 25 50 filter. Not all poles and zeros are selected before the synthesis starts and the obtained network should be of the 5th order. For the purpose of this example FILTERPART program looks as in Fig. 43. The program calculates Amin = 64.36148 dB; this is the same vale as in Zverev.

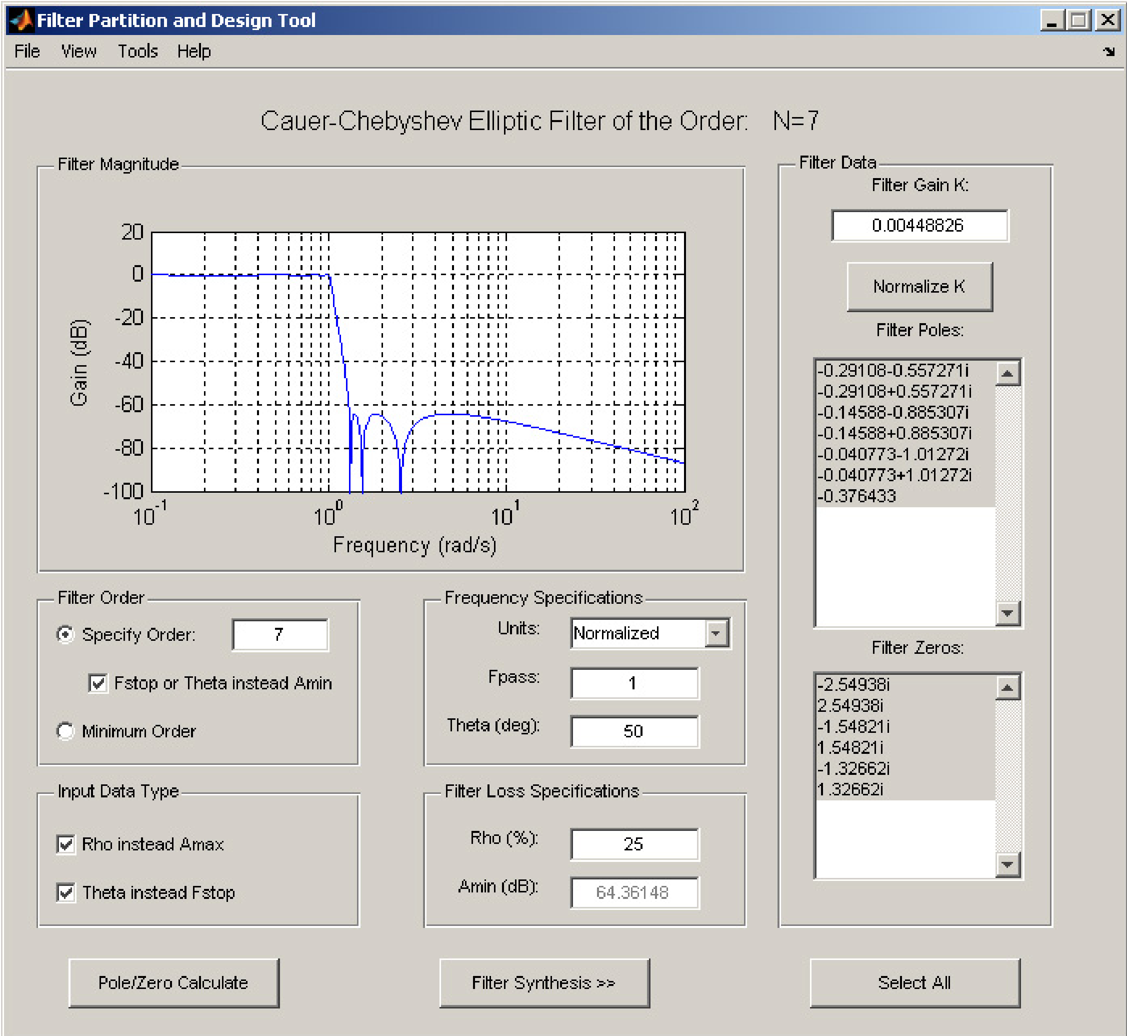
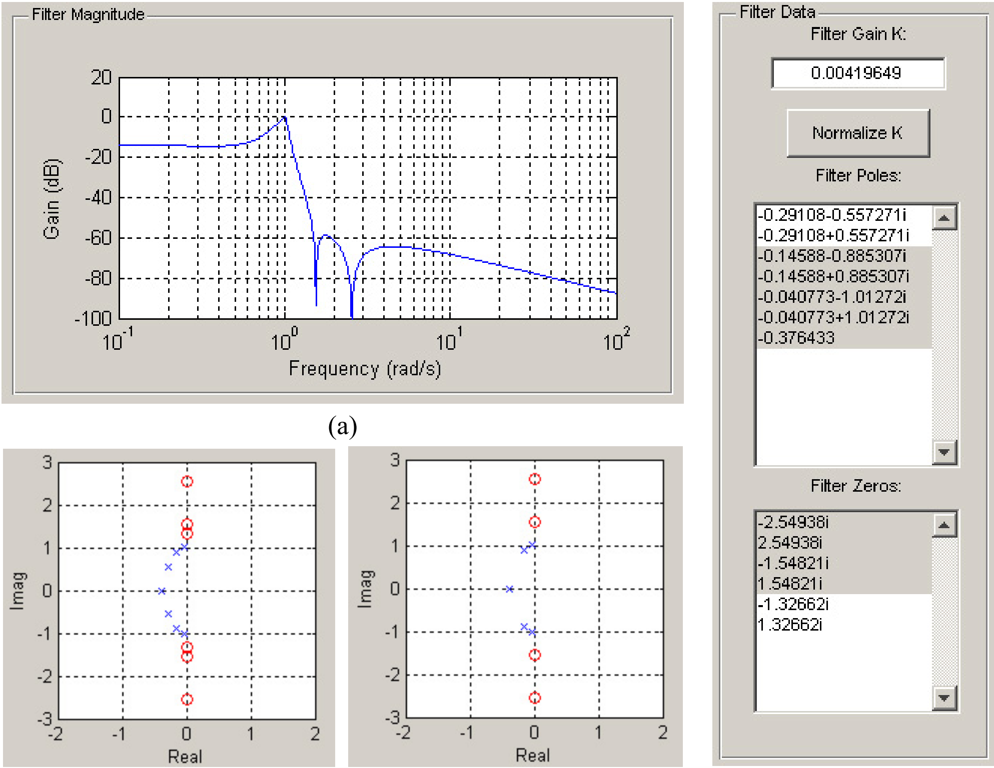


Fig. 43 FILTERPART program when calculating CC 7 25 50 filter.

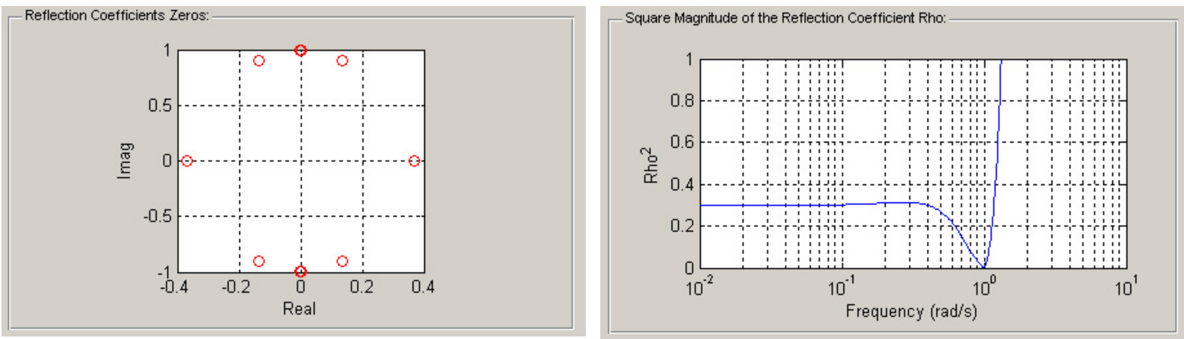
All poles and zeros of the CC 7 25 50 filter are shown in Fig. 44(b). To carry on the partial design of the filter, user has to simply select those poles and zeros to be realized by a ladder LC network, normalize the gain K and then proceed to the filter synthesis. Note that user can select many combinations for the filter partition synthesis, and in our example we choose poles and zeros as shown in Fig. 44(d). Magnitude, and pole-zero plot of partial poles and zeros to be realized by the ladder part of the CC 7 25 50 filter are shown in Fig. 44(a) and (c), respectively. Note the new gain K value and the maximum 0dB of the magnitude. The user can now press “Filter Synthesis” button, and the FILTERSYN program starts.



(b) (c) (d)

Fig. 44 FILTERPART program: (a) Magnitude of the part of the CC 7 25 50 filter. (b) All poles-and zeros diagram. (c) Partial pole-zero diagram. (d) Some poles and zeros selected and appropriate *K*.

When user starts synthesis there are all together 10 reflection-coefficient zeros available for the user to choose in the program FILTERSYN. All zeros are shown in Fig. 45(a), whereas the corresponding square magnitude of the reflection coefficient Rho is shown in Fig. 45(b). Note that the Rho-squared in Fig. 45(b) is much larger than in Fig. 26(c) of the CC 5 25 49 filter, because the network to be realized in this example is the magnitude of some general curve, whereas the magnitude curve in example in Fig. 26(c) is of the standard elliptic filter type. (Note the difference in scale of the ordinates.)



(a) (b)

Fig. 45 FILTERSYN program: (a) Plot of all reflection coefficient Rho zeros. (b) Square magnitude of the Rho.

User can conclude that the greater sensitivity of the networks magnitude will be obtained when the network realizes only a part of the CC transfer function, which is only a general curve, than when realizes a whole CC filter. Since user has 5 zeros to choose there are four possible ways in that user can do it. The possible combinations are shown in Fig. 46.

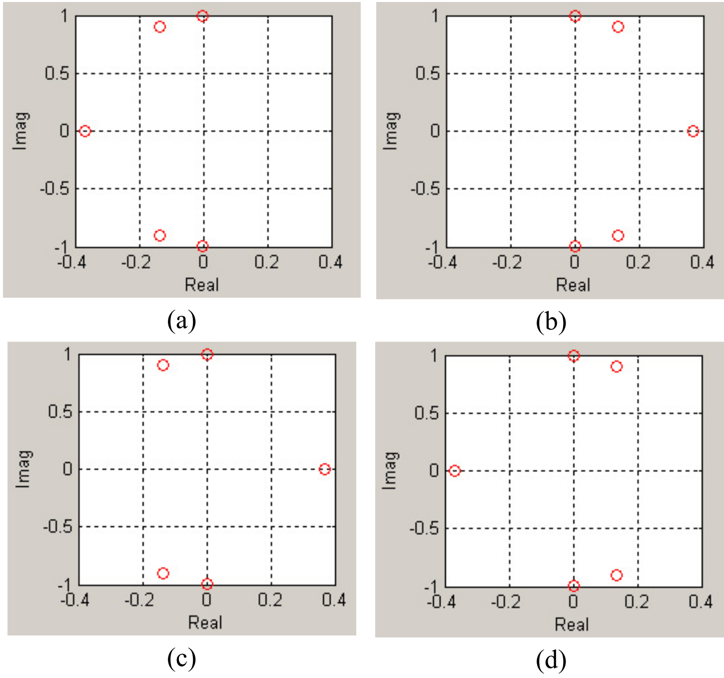


Fig. 46 Choosing reflection coefficient zeros. (a) All in the left *s*-plane. (b) All in the right *s*-plane. (c) Alternating in the *s*-plane. (d) Alternating in the *s*-plane in the other way.

To continue with our synthesis we choose the all four cases and program FILTERSYN calculate elements that are presented in the Table I. The component values are:

Table I Element values of the filter design by partition.

case *a*)	case *b*)
R1 = 1.00000000000000	R1 = 1.00000000000000
C1 = 15.86080834376006	C1 = 1.31574358272734
C2 = 2.50828233784809	C2 = 0.13242795993136
L2 = 0.06134165373321	L2 = 1.16185574944410
C3 = 162.61355519756177	C3 = 1.61720379606577
C4 = 43.44363724674439	C4 = 0.08132225948874
L4 = 0.00960318862708	L4 = 5.13017525275284
C5 = 93.82783563928089	C5 = 0.12921900894860
R2 = 0.01127497598109	R2 = 0.01127497598109
case *c*)	case *d*)
R1 = 1.00000000000000	R1 = 1.00000000000000
C1 = 2.29654052480764	C1 = 2.68076928526630
C2 = 0.06237533876417	C2 = 0.40425134724055
L2 = 2.46671504607182	L2 = 0.38061020126138
C3 = 0.41854058066976	C3 = 49.09204270909306
C4 = 0.01279808877116	C4 = 27.17577792174195
L4 = 32.59841766899359	L4 = 0.01535181235027
C5 = 0.02234015886684	C5 = 191.51307781772775
R2 = 0.01127497598109	R2 = 0.01127497598109

Important note: There exist two solutions for the terminating resistor R_2 that satisfy the design procedure: those are the first value: $R_2^{I}=R_2$ and the second: $R_2^{II}=1/R_2$. Thus the second value user can simply calculate by reciprocating the first R_2 value. The one of the values R_2 is calculated by the program and presented. Which of the two values of R_2 should be used is different from the case to case. For example, although the program presents for the four design cases *a*)–*d*) in the Table I the same value R_2=0.011275, *the user has to decide himself* if the value for the terminating resistor is R_2 or $1/R_2$. Thus, in the example above in Table I in cases *a*) and *d*) there should be R_2^{I}=0.011275, whereas in the cases *b*) and *c*) there should be R_2^{II}=1/0.011275=88.6918.

In the case *b*) we obtain the component values having minimum component spread. So we can choose such a combination of the reflection coefficient zeros in the *s*-plane to realize network with minimum component spread as the design criterion. The RLC-ladder network that corresponds to the case *b*) in this example is shown in Fig. 47(a). With this example the filter synthesis by partition is concluded.

In the process of synthesis there exist another one possibility in choosing different order of the transfer function zeros realizations, and it can yield another four combinations of network element. This case is not presented in this example. Anybody who wishes can readily try it by himself, to calculate many different filter networks using this very fast synthesis program. Thus, synthesis is really an ambiguous process and our program gives a user an opportunity to quickly synthesise high quality filter networks.

In the "filter partition" design we decompose the filter transfer function $T(s)$ into a product of two parts:

$$T(s)=T_{Ladder}(s)\cdot T_{Biquad}(s)\ . \qquad (5)$$

Those two parts, that decompose a 7[th]-order transfer function $T(s)$ can be realized as shown in Fig. 47.

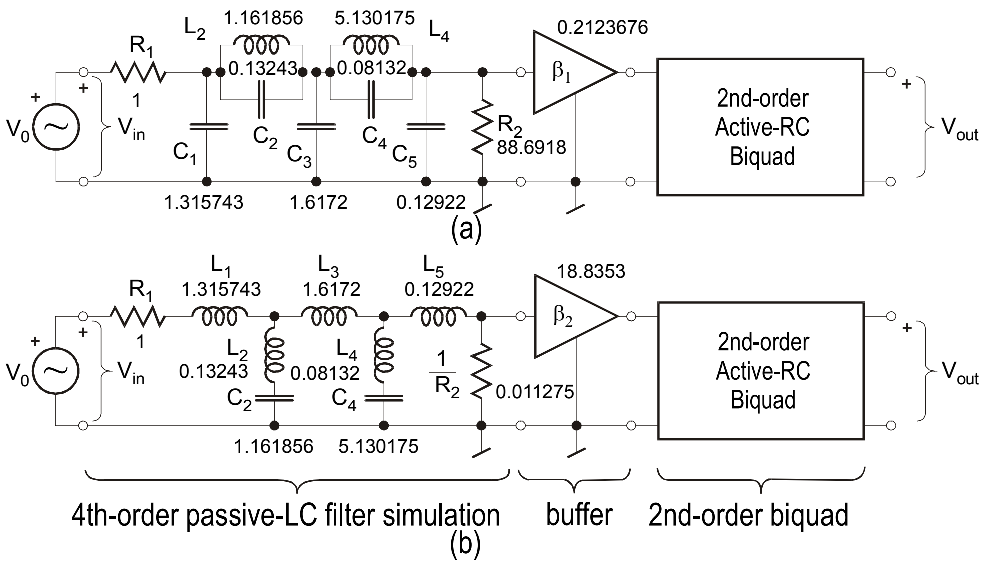


Fig. 47 7[th]-order elliptic filter realized by filter partition.
(a) The realization with elements of ladder RLC directly from the program. (b) The ladder RLC is dual.

Element values in the Table I case *b*) are calculated by the program FILTERSYN and used for the RLC ladder network in Fig. 47(a). Those values can be used for the dual ladder network in Fig. 47(b), as well, using rules described above, and with the value of terminating resistor R_2 to be reciprocal to the value of R_2 in the original network in Fig. 47(a). Note that the resistor R_1=1, thus its reciprocal value is again equal to unity.

A drawback that the passive LC filter designer has to contend with is a given amount of signal attenuation in the passband that must be overcome with gain somewhere else. The designer of active inductor-simulated filters must accept the same drawback, even though his filter structure contains active elements that could, in different circumstances, provide gain. Thus, the additional amplifier in Fig. 47 referred to as buffer, provides not only gain but also isolation between the cascaded sections, i.e. between 4th-order passive LC filter simulation and 2nd-order biquad. The gain of a buffer in Fig. 47(a) is referred to as β_1.

Recall that the relation between the power transmission coefficient $t(s)$ and the voltage transfer function $T_{Ladder}(s)$ is given by [1]:

$$T_{Ladder}(s) = \frac{V_2(s)}{V_0(s)} = \frac{1}{2}\sqrt{\frac{R_2}{R_1}} \cdot t(s)\,, \tag{6}$$

where V_0 is the voltage of the generator at the input of the ladder LC network and V_2 is the voltage at the terminating resistor R_2. Power transmission coefficient is defined by:

$$\left|t(j\omega)\right|^2 = \frac{P_2}{P_{\max}}\,, \tag{7}$$

where P_2 is the power at the resistor R_2, and $P_{\max}$ the maximum deliverable power to R_2. When the maximum power is transferred from the generator to R_2, there is $\left|t(j\omega)\right|^2 = 1$, thus we have

$$\left|T_{Ladder}(j\omega)\right|_{\max} = \frac{1}{2}\sqrt{\frac{R_2}{R_1}}\,. \tag{8}$$

To correct the $\left|T_{Ladder}(j\omega)\right|_{\max}$ to have the value of 1 (or 0dB) we have to introduce additional amplification of

$$\beta_1 = \frac{2}{\sqrt{R_2}};\ \text{with}\ (R_1 = 1)\,. \tag{9}$$

Using (9) the gain β_1=0.2123676 for the circuit Fig. 47(a) is defined with the value R_2=88.6918 in this circuit. For the dual circuit in Fig. 47(b) the gain is designated by β_2 because it differs from the gain β_1 in its dual counterpart, only because of the value R_2 which is reciprocal. Thus the gain β_2=18.8353 is readily calculated using (9) with the appropriate value of R_2=1/88.6918 = 0.011275 of the circuit in Fig. 47(b).

The dual passive ladder RLC circuit in Fig. 47(b) can readily be realized using its active-RC representation in the form of DCR-circuit in Fig. 49. In DCR circuit every resistor corresponds to one inductor in the prototype ladder RLC circuit having the same value, capacitors in DCR correspond to resistors in RLC having the reciprocal values, and super-capacitors (D in DCR) correspond to capacitors in RLC circuit having the same values. DCR circuit that simulates the original RLC circuit has the same transfer function magnitude and thus the same maximum value. Although due to rules of transformation of RLC to DCR circuits, the DCR has the value of C_2=88.6918, the gain of the additional amplifier is the same as for the original RLC prototype in Fig. 47(b), i.e. β_2=18.8353, which provides the maximum 0dB magnitude of the $T_{Ladder}(s)$.

The gains of $T_{Ladder}(s)$ and $T_{Biquad}(s)$ are designated by k_{Ladder} and k_{Biquad}, respectively. The overall $T(s)$ in (5), thus has the gain $k = k_{Ladder} \times k_{Biquad}$. The user can readily read the gain k from the Fig. 43, when all zeros and poles for the CC 7 25 50 filter are calculated for the first time, as the gain k=0.00448826. When in filter partition design user selected poles and zeros to be realized by RLC ladder network, and after the user has normalized its gain to 0dB, the gain k_{Ladder} readily follows from Fig. 44(d), and has the value k_{Ladder}=0.00419649. Finally, user can calculate the remaining gain for the Biquad circuit by readily dividing $k_{Biquad} = k/k_{Ladder}$= 0.00448826/0.00419649 = 1.06952731295720. Note that the gain k_{Biquad} has been calculated with the highest accuracy by the division in the MATLAB workspace.

3.8 The Biquad circuit realization

The 2nd-order biquadratic part of the overall transfer function in the case of the elliptic filter realization by partition can be realized using well-known Tow-Thomas "Biquad" filter circuit with 4 opamps shown in Fig. 48.

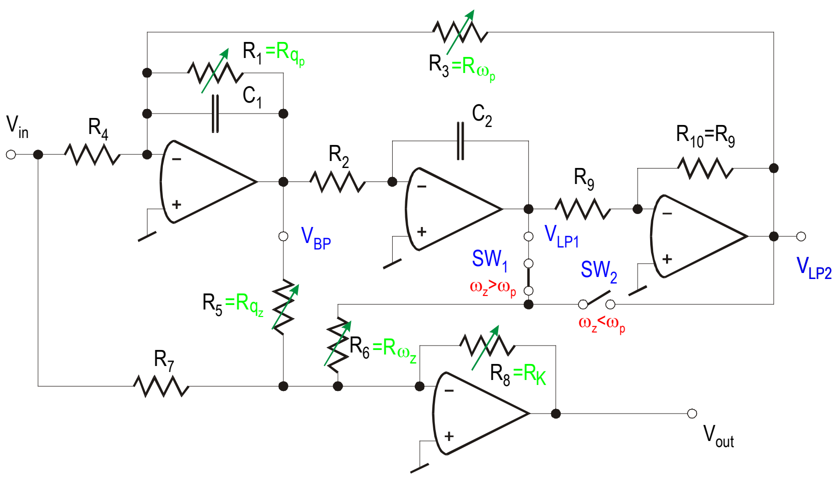


Fig. 48 The Biquad filter circuit.

The transfer function of the "Biquad" circuit in Fig. 48 is given by:

$$T_{\text{Biquad}}(s)=\frac{V_{out}}{V_{in}}=-\frac{R_8}{R_7}\cdot\frac{s^2+s\dfrac{\omega_p}{q_p}\left(1-\dfrac{R_7}{R_5}\cdot\dfrac{R_1}{R_4}\right)+\omega_p^2\left(1\pm\dfrac{R_7}{R_6}\cdot\dfrac{R_3}{R_4}\right)}{s^2+\dfrac{\omega_p}{q_p}s+\omega_p^2}. \tag{10}$$

The transfer function (10) can be equated to the general form:

$$T_{\text{Biquad}}(s)=\frac{V_{out}}{V_{in}}=-K\cdot\frac{s^2+s\dfrac{\omega_z}{q_z}+\omega_z^2}{s^2+\dfrac{\omega_p}{q_p}s+\omega_p^2}, \tag{11}$$

and we obtain:

$$\omega_p=\frac{1}{\sqrt{R_2R_3C_1C_2}},\ q_p=R_1\sqrt{\frac{C_1}{R_2R_3C_2}},\ \omega_z=\omega_p\sqrt{1\pm\frac{R_7}{R_6}\cdot\frac{R_3}{R_4}},\ K=\frac{R_8}{R_7}, \tag{12}$$

where ± is + when switch SW_1 is closed and SW_2 open (as in Fig. 48), – when SW_2 is closed (and SW_1 open). When + the $\omega_z>\omega_p$ can be realized, whereas when – the $\omega_z<\omega_p$ can be realized, whereas when both SW_1 and SW_2 are open then $\omega_z=\omega_p$.

The middle term in the numerator in (10) is equal to:

$$\frac{\omega_z}{q_z}=\frac{\omega_p}{q_p}\left(1-\frac{R_7}{R_5}\cdot\frac{R_1}{R_4}\right). \tag{13}$$

Note that if user realizes the finite transfer function zero, as in our example, the middle term in the numerator $T_{\text{Biquad}}(s)$ has to be zero, thus from (13) we have to calculate R_5.

From the relations describing the circuit in Fig. 48 we can readily see an important property of the biquad, i.e. that it can be orthogonally tuned. By this we mean that:

1. R_3 can be adjusted to the specific value of ω_p.
2. R_1 can then be adjusted to give the specified value of pole Q, q_p without changing ω_p, which has already been adjusted.
3. R_6 can then be adjusted to give the specified value of ω_z, without changing ω_p, and q_p that have already been adjusted.
4. R_5 can then be adjusted to give the specified value of zero Q, q_z without changing ω_p, ω_z, and q_p that have already been adjusted. (In the case of notch filter the value of R_5 provides that $q_z=\infty$, whereas in the all-pass circuit that there is $q_z=-q_p$.)
5. Finally, R_8 can be adjusted to give the desired value of $K=k_{Biquad}$ or pass-band gain for the circuit, without affecting ω_p, ω_z, q_p, or q_z which have already been set.

These steps are called tuning algorithm. This algorithm provides for *orthogonal* tuning, which is always preferred over the *iterative* tuning, especially when the filter is to be produced on a production line with a laser used to adjust each circuit element value.

The step-by-step design procedure for the Biquad part of the filter by partition in Fig. 47 can be performed as follows:

1. Choose normalized values $R_2 = R_4 = R_7 = R_9 = R_{10} = 1$, $C_1 = C_2 = 1$.
2. Using (12) calculate element values. Calculate $R_3 = 1/\omega_p^2$,
3. Calculate $R_1 = q_p/\omega_p$,
4. Calculate $R_6 = 1/(\omega_z^2 - \omega_p^2)$, because $\omega_p < \omega_z$ (the SW_1 should be closed). If SW_2 were closed then $\omega_z < \omega_p$ and we calculate $R_6 = 1/(\omega_p^2 - \omega_z^2)$,
5. Set $R_5 = R_1$, to have $q_z = \infty$,
6. Calculate $R_8 = K = k_{Biquad}$, to realize gain in the pass-band.

In our example we choose $R_2=R_4=R_7=R_9=R_{10}=1$, $C_1=C_2=1$, and then we can readily calculate $R_3= 2.53$, $R_1=R_5=1.7177$, $R_6 = 0.7328$, and finally $R_8=k_{Biquad}=1.06952$. Besides the tuning feasibility note how the design equations for the 4-opamp Biquad are simple and of the explicit form.

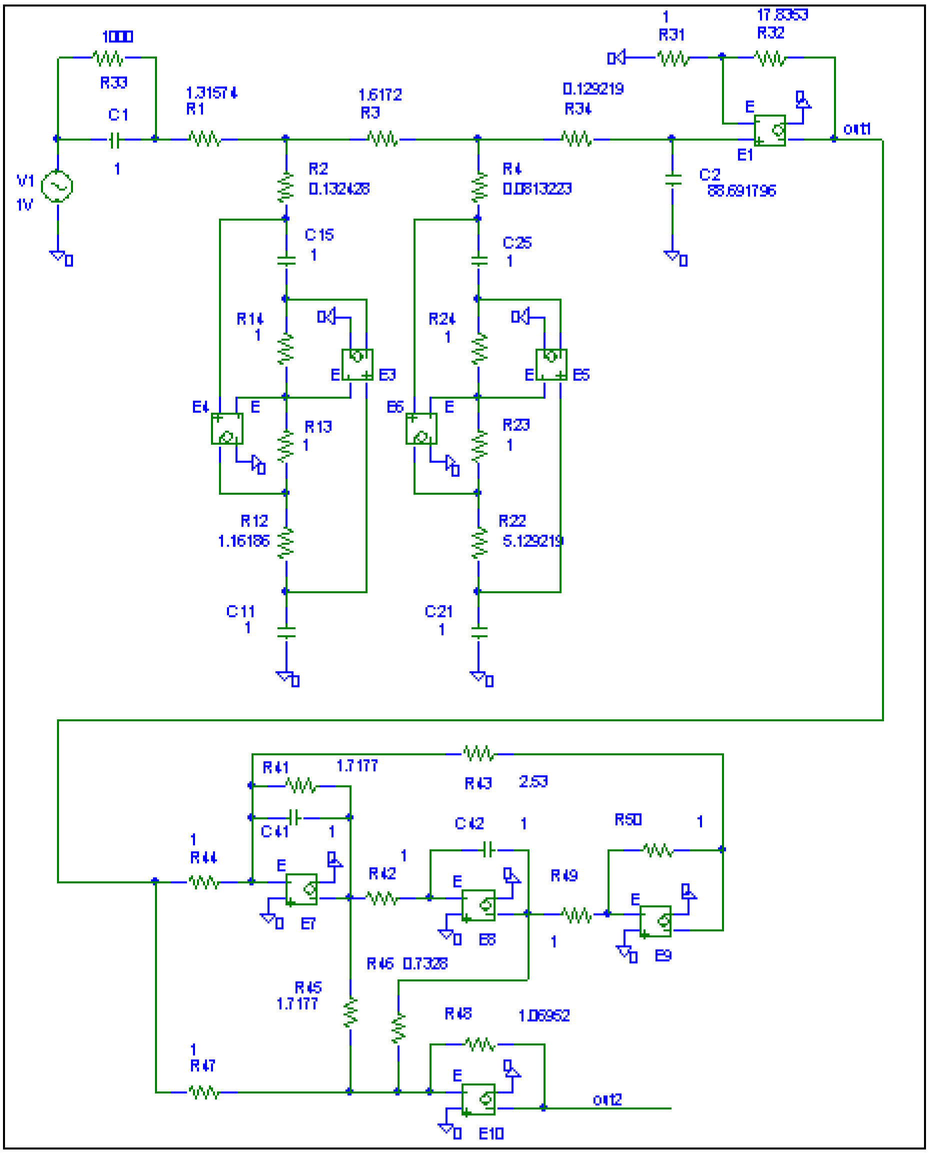


Fig. 49 PSpice schematic of CC 7 25 50 filter circuit realized by filter partition.

The whole CC 7 25 50 filter realized by filter partition is shown in Fig. 49 in the form of the PSpice schematic. Note that the simulated active ladder filter part (DCR filter) is cascaded to the biquad circuit part. Finally, the corresponding filter transfer function magnitude is shown in Fig. 50.

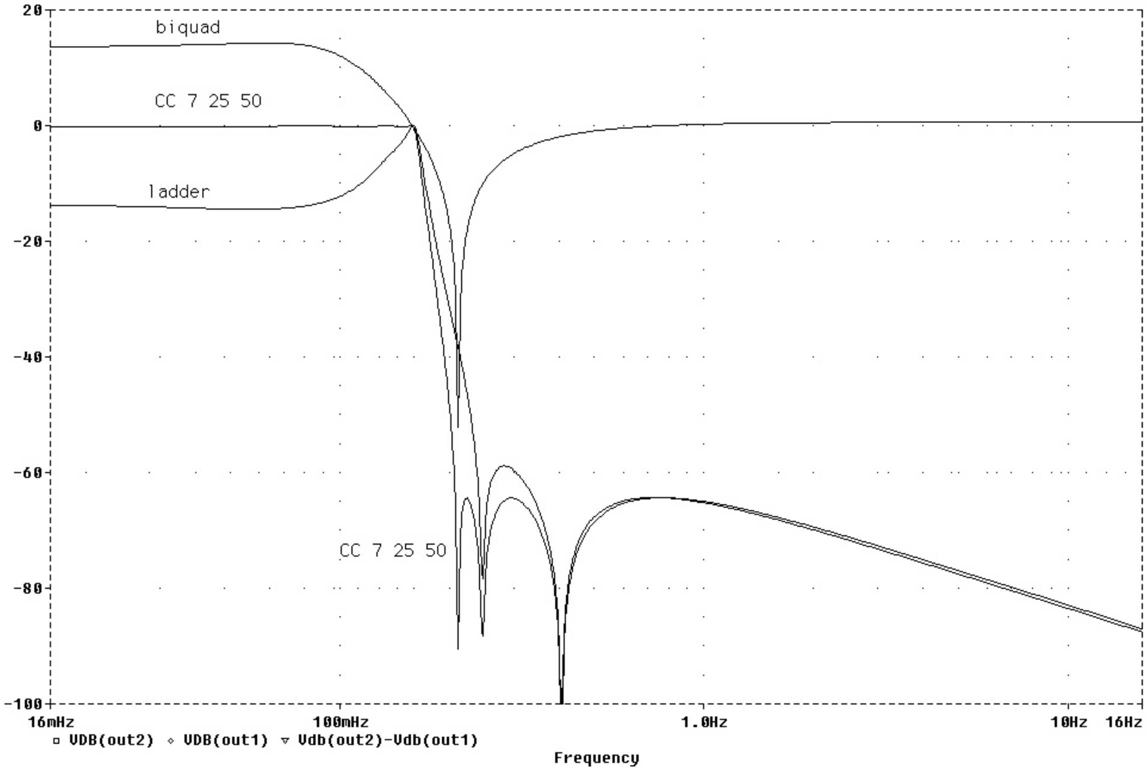


Fig. 50 Magnitude of the CC 7 25 50 filter circuit realized by filter partition.

4 CONCLUSION

A procedure for the design of low-sensitivity and tunable LP active-*RC* filters with finite zeros (of the elliptic type) is presented by partition of filters transfer function. The step-by-step design procedures are presented in examples for even and odd filter order using MATLAB GUI programs FILTERPART and FILTERSYN. In the design we have the following steps: (*i*) calculate transfer function poles and zeros using FILTERPART or read poles and zeros from some filter tables (e.g. Zverev); (*ii*) decompose the transfer function into two parts: ladder and biquad; (*iii*) optimize the gains of the filter transfer function parts to have maximum dynamic range; (*iv*) in the realization of the first part calculate passive RLC ladder LP prototype circuit using FILTERSYN. Then use active-RC realization that simulate passive RLC ladder prototype; (*v*) in the realization of the second part use well known design procedures of the biquad filter. The biquad filter has the possibility of functional tuning of its transfer function parameters. The whole filter designed by filter partition has low-sensitivity to component tolerances. Therefore, besides the low sensitivity performance, the newly introduced approach provides a good filter solution for manufacturing because it has the opportunity of tuning. The filter design method by partition is becoming more and more efficient as the filter order is becoming higher.

5 REFERENCES

[1] D. Jurišić, G. S. Moschytz and N. Mijat, "*Low-Sensitivity, Tunable Active-RC Filters Using Filter Partition*," to be submitted for publication in the IEEE Trans. on Circuits and Systems, vol. CAS-xx, no. xy, Month Year.

[2] H. J. Orchard, "Inductorless filters," *Electron. Letters*, vol. 2, pp. 224-225, June 1966.

[3] H. J. Orchard, A.N. Wilson, "*Elliptic Functions for Filter Design*," IEEE Trans. on Circuits and Systems—I, vol. CAS-44, no. 4, pp. 273-278, April 1997.

[4] A. I. Zverev, *Handbook of Filter Synthesis*, John Wiley and Sons, New York 1967.