E-SSAF – A Powerful E-Learning Tool for State-Space OA-RC and Gm-C Continuous Time Filter Synthesis

E. Doicaru, I.D. Nicolae, and E. Popescu

***Abstract*—** This paper introduces a software support tool for an advanced electronics laboratory session on filter synthesis. The application provides the students with a handy tool for determining the topological elements of active analogue continuous-time filters, described by the arbitrary transfer functions. The purpose is to obtain filters with small sensitivities, low noise level at the output and high dynamic ranges. The synthesis method implemented in the application is the method of intermediate transfer functions (IFs). The method was chosen because it can provide the optimization of the above mentioned performances at the abstract level of IFs not at the topological level. The application selects the IFs that yield structures with optimum performance and give their topological elements. The tool is specifically designed for students' use, providing a friendly, intuitive interface.

***Keywords*—** E-learning tool, Synthesis, Active Filters, Low Sensitivities

I. INTRODUCTION

In the VLSI era, as mixed analogue/digital circuits of increasing complexity are implemented into a single IC, the need for low-sensitivities low-noise filter performance becomes even greater. A major continuous-time filters application is direct signal processing, especially for medium dynamic range applications, in cases where high speed and/or low power dissipation are needed. Good examples of such applications include disk drivers read channel, high-speed data links, phase-locked loops, telephony, wireless communication systems, and IF receiver.

In this context, we felt the need for a tool that could be used by our electronics graduate students in laboratory settings for automated synthesis of analogue continuous-time filters. This is how E-SSAF tool was born (Automatic *S*tructural *S*ynthesis of *A*ctive Analogue Continuous-time High-order *F*ilters). Our objective was to provide an innovative approach of conducting interactive computer-based laboratory sessions, that help the learner understand the subject and offer additional motivation. With this in mind, we have designed and implemented a powerful tool that helps students determine the topological elements of active analogue continuous-time filters. Being conceived specifically as an e-learning tool, E-SSAF is characterized by a particular ease of use, a friendly interface and the presence of contextual help, that comes in very handy for the students.

More specifically, E-SSAF can be used to determine the topological elements of active analogue continuous-time filters, described by an arbitrary transfer function $t(s)=p(s)/e(s)$, where *e(s)* is the natural mode (pole) polynomial (of order *m*) and *p(s)* is the transmission zero polynomial (of order *m*). The purpose was to obtain small sensitivities for the filters, a very low noise level and good dynamic range. The synthesis method used is the intermediate transfer functions (IFs) method, [1].

The method consists of obtaining an *n*-th transfer function, *t*(s), using a structure of n resistively interconnected integrators using the intermediate transfer functions selected from the given *t(s)* following the general structure presented in Fig. 1. All integrators are assumed to be identical.

The application selects the IFs that yield structures with optimum performance. Even if the synthesis method was developed for the OA-RC structure, it can be easily used for the synthesis of the Gm-C structures. The method's great advantage consists in the possibility of filter performance optimization at the abstract level of intermediate transfer functions instead of the usual way (at topological level).

The rest of the paper is structured as follows: section 2 sketches an overview of the mathematical grounds of our approach. Section 3 introduces the design of E-SSAF, outlining the principles and algorithms the tool is based on. Section 4 briefly presents the structure of the application and some implementation details, illustrating some of its possible uses. Finally, section 5 concludes the paper, outlining future research directions.

Manuscript received December 28, 2006: Revised version received May 7, 2007. This work was supported by the CNCSIS Research Projects no. 29C/08.05.2007, theme no. 5, CNCSIS code no. 602 and no. 167.

E. Doicaru is with the Electronics Department, University of Craiova, Bvd. Decebal 107, 200440, Craiova, ROMANIA (corresponding author to provide phone: 00 40 251 435 724; fax: 00 40 251 438 198; e-mail: dmilena@electronics.ucv.ro)

I. D. Nicolae is with the Software Engineering Department, University of Craiova, Bvd. Decebal 107, 200440, Craiova, ROMANIA (e-mail:nicolae_ileana@software.ucv.ro)

E. Popescu is with the Software Engineering Department, University of Craiova, Bvd. Decebal 107, 200440, Craiova, ROMANIA (e-mail: popescu_elvira@software.ucv.ro).

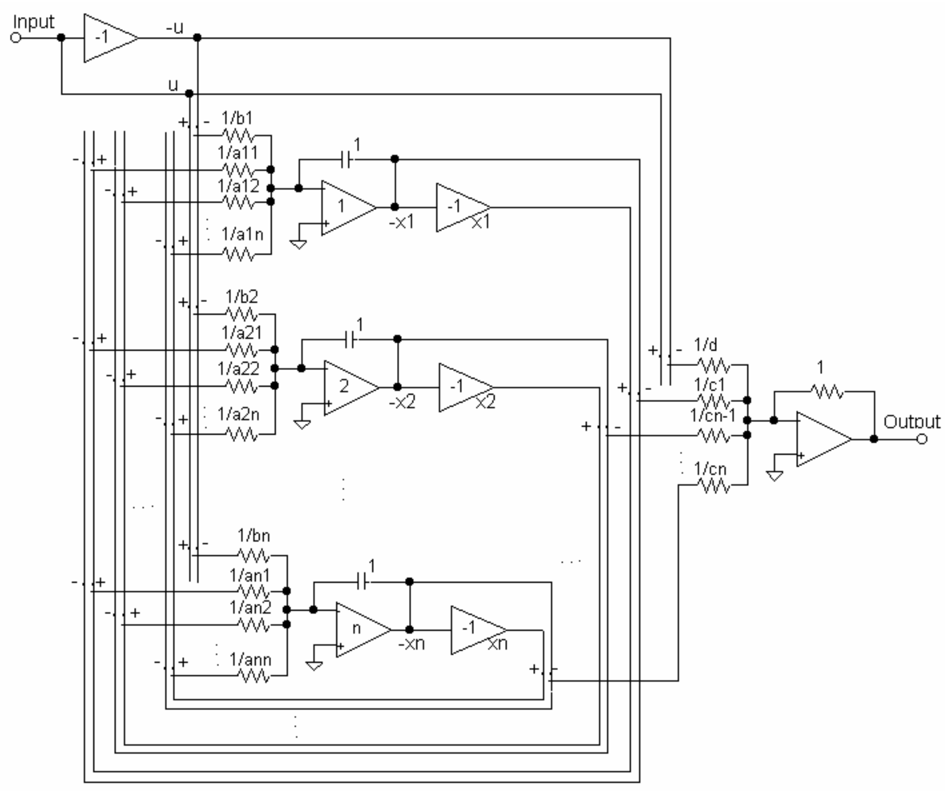

Fig. 1 Generic Filter Structure

II. MATHEMATHICAL GROUNDS

The above mentioned realization can be described by a state-variable formulation:

$$\begin{aligned} &s \cdot \mathbf{x}(s) = \mathbf{A} \cdot \mathbf{x}(s) + \mathbf{b} \cdot u(s) + \varepsilon(s) \\ &y(s) = \mathbf{c}^T \cdot \mathbf{x}(s) + d \cdot u(s) \end{aligned} \tag{1}$$

In relations (1) the vector $x(s)$ represents the circuit state (integrators outputs), the matrix **A** describes the interconnections between the n integrators, the vector **b** contains the coefficients that multiply the input signal u(s) in order to be applied to the integrators inputs, the vector **c** contains the coefficients required to form the output, the scalar d is the coefficient of the feed through component from input to output, and $\varepsilon(s)$ is the vector containing the noise component at the integrator inputs.

The dual sets of IFs, $\{f_i(s)\}$ and $\{g_i(s)\}$ are given by:

$$\begin{aligned} &f_i(\mathrm{s}) \overset{\Delta}{=} x_i(\mathrm{s})/u(\mathrm{s}) \\ &\mathbf{f}(\mathrm{s}) = (\mathrm{s} \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b} \\ &g_i(\mathrm{s}) \overset{\Delta}{=} y(\mathrm{s})/\varepsilon_i(\mathrm{s}) \\ &\mathbf{g}^{\mathrm{T}}(\mathrm{s}) = \mathbf{c}^{\mathrm{T}} \cdot (\mathrm{s} \cdot \mathbf{I} - \mathbf{A})^{-1} \end{aligned} \tag{2}$$

The first set, $\{f_i(s)\}$, contains the transfer function from the filter input to the integrator outputs, and the 2-nd set, $\{g_i(s)\}$, can be physically interpreted as the integrators noise gains.

Given a transfer function t(s), IF synthesis is based on choosing a set of linearly independent functions, $\{f_i(s)\}$, having identical denominator polynomials, e(s), and arbitrary numerator polynomials of degree less than n. From this set we can obtain the {**A**, **b**, **c**, d} parameters using the following relationships:

$$\begin{aligned} &\mathbf{A} = \mathbf{F} \cdot \mathbf{E} \cdot \mathbf{F}^{-1},\ \mathbf{b} = \mathbf{F} \cdot \mathbf{1}, \\ &\mathbf{c}^{\mathrm{T}} = \mathbf{t}^{\mathrm{T}} \cdot \mathbf{F}^{-1}, d = t_{n+1}, \\ &t(\mathrm{s}) = \mathbf{t}^{\mathrm{T}} \cdot \mathbf{v}(\mathrm{s}) + t_{n+1}, \\ &\mathbf{f}(\mathrm{s}) = \mathbf{F} \cdot \mathbf{v}(\mathrm{s}), \\ &\mathbf{g}(\mathrm{s}) = \mathbf{G} \cdot \mathbf{v}(\mathrm{s}), \\ &v_i(\mathrm{s}) = 1/(\mathrm{s} - e_i),\ i = \overline{1, n}, \\ &\mathbf{G}^{\mathrm{T}} = \mathbf{H} \cdot \mathbf{F}^{-1}. \end{aligned} \tag{3}$$

where **t** is a vector containing the n residues of $t(s)$ at the poles, t_{n+1} is the residue at $s = \infty$, **F** is a matrix containing the residues of the *f* functions evaluated at the poles, **G** is a matrix of the residues of the *g* functions, e_i are the roots of $e(s)$, **E** is the diagonal matrix having the natural modes e_i as its elements, $\mathbf{1} = (1, 1, ...1)^T$ and **H** is a diagonal matrix formed from the residues of $t(s)$.

The sensitivities to the **A**, **b**, **c**, *d* topological parameters, the integrator gain γ_i and the operational amplifier gain $\mu_i(s)$ are described by relations (4).

$$\begin{aligned} &S_{A_{ij}}^{t(\mathrm{s})} = g_i(\mathrm{s})\cdot f_i(\mathrm{s})\cdot A_{ij}/t(\mathrm{s}),\, S_{b_i}^{t(\mathrm{s})} = g_i(\mathrm{s})\cdot b_i/t(\mathrm{s}),\\ &S_{c_i}^{t(\mathrm{s})} = f_i(\mathrm{s})\cdot c_i/t(\mathrm{s}),\ S_d^{t(\mathrm{s})} = d/t(\mathrm{s}),\\ &S_{\gamma_i(\mathrm{s})}^{t(\mathrm{s})} = f_i(\mathrm{s})\cdot g_i(\mathrm{s})\cdot \mathrm{s}/t(\mathrm{s}),\ S_{\mu_i(\mathrm{s})}^{t(\mathrm{s})} = S_{\gamma_i(\mathrm{s})}^{t(\mathrm{s})}\cdot S_{\mu_i(\mathrm{s})}^{\gamma_i(\mathrm{s})} \end{aligned} \tag{4}$$

In the terms of {**A**, **b**, **c**, d} elements the gain of the integrator i is proportional to the sum of the i row elements of matrix **A** and i element of vector **b**:

$$\sum_{j=1}^{n}\left|A_{ij}\right| + \left|b_i\right|.$$

It follows that "good" realizations have row sums that are almost equal and small in value.

The following sensitivity invariant can be demonstrated [2]:

$$\sum_i f_i(\mathrm{s})\cdot g_i(\mathrm{s}) = -\mathrm{d}\,t(\mathrm{s})/\mathrm{ds} \tag{5}$$

and one obtains a classical sensitivity result:

$$\sum_i S_{\gamma_i(\mathrm{s})}^{t(\mathrm{s})} = -\frac{\mathrm{s}}{t(\mathrm{s})}\cdot\frac{\mathrm{d}\,t(\mathrm{s})}{\mathrm{ds}} \tag{6}$$

Because the sum of the sensitivities is constant, the minimum is obtained when all sensitivities are equal and it can be derived for the minimum sensitivity realization:

$$f_i(s)\cdot g_i(s) = -\frac{1}{n}\cdot\frac{d\,t(s)}{ds},\ \forall i=\overline{1,n} \tag{7}$$

Noise signals injected at integrator inputs can be modeled by $\varepsilon(s)$, and if we assume that the noise signals have white spectra with equal densities N_i^2, the output noise will have a power spectrum given by:

$$P_{n0}(\omega) = N_i^2\cdot\sum_i\left|g_i(\mathrm{j}\cdot\omega)\right|^2 \tag{8}$$

with a RMS level of:

$$\left\|P_{n0}(\omega)\right\| = N_i^2\cdot\sqrt{\sum_i\left\|g_i(\mathrm{j}\cdot\omega)\right\|_2^2} \tag{9}$$

where $\left\|g_i(j\cdot\omega)\right\|_2^2 = \sqrt{\int_{-\infty}^{\infty}\left|g_i(j\cdot\omega)\right|^2\cdot d\omega}$.

This description can be adapted to the Gm-C filter. The following relation gives the current through a single capacitor of a Gm-C filter:

$$\begin{aligned} C_i\cdot\dot v_{C_i} &= G_{m1i}\cdot v_1 + G_{m2i}\cdot v_2 + \ldots +\\ &+ G_{mni}\cdot v_n + G_{mbi}\cdot u + \varepsilon_i \end{aligned} \tag{10}$$

where v_i, $i = 1 \div n$, represents the voltage across the capacitor i, u is the input voltage and ε_i is the total noise current through capacitor.

By scaling Eq. (10) with C_i we obtain:

$$\begin{aligned} \dot v_{C_i} &= (G_{m1i}/C_i)\cdot v_1 + (G_{m2i}/C_i)\cdot v_2 + \ldots +\\ &+ (G_{mni}/C_i)\cdot v_n + (G_{mbi}/C_i)\cdot u + \varepsilon_i/C_i \end{aligned} \tag{11}$$

Interconnecting n capacitors with transconductors can be conveniently described by the same state-variable formulation (1):

$$\begin{aligned} &s\cdot\mathbf{x}(s) = \mathbf{A}\cdot\mathbf{x}(s) + \mathbf{b}\cdot u(s) + \varepsilon_{\mathbf{S}}(s)\\ &y(s) = \mathbf{c}^T\cdot\mathbf{x}(s) + d\cdot u(s) \end{aligned} \tag{12}$$

where the states x_i are represented by capacitor voltages, matrix element A_{ij} is implemented by a transconductor with input voltage x_j and the output voltage across the i^{th} capacitor x_i, the vector element b_i is implemented by transconductor from the input u to state i, c_i is the multiplication coefficient of state i required to form the output voltage y, d is the multiplication coefficient of the input voltage u and ε_S can model the current noise at the capacitor i.

So the meaning of $\{f_i(s)\}$ IFs set is the same as in the OA-RC filter synthesis and the physical meaning of $\{g_i(s)\}$ IFs set is the noise gain to the capacitor i at output. We can conclude that all the results obtained in the OA-RC filter synthesis can be applied to the Gm-C filter.

III. Principles and Algorithms behind E-SSAF Tool

The tool is used to find the synthesis IF sets leading to low sensitivities of filters, low noise performance and also good dynamic ranges and for the determination of topological elements of active analogue continuous-time filters.

The mathematical analysis emphasizes the fact that the best realizations are those obtained by orthogonal, orthonormal and derivative IFs sets. The E-SSAF tool generates these IFs sets. An IF set is orthogonal if all functions from the set are linearly independent and their inner product, given by (13), is zero.

$$f_i \bullet f_j = \int_{-\infty}^{\infty} f_i(j\cdot\omega)\cdot\bar f_j(j\cdot\omega)\cdot d\omega,\ i,j=\overline{1,n} \tag{13}$$

An IF set is orthonormal if all $\{f_i(\mathrm{s})\}$ are orthogonal and $\left\|f_i\right\|_2 = 1,\ \forall i = 1,n$. Such sets can be obtained from any set of independent $\{f_i(s)\}$ by choosing the Gram-Schmidt orthonormalisation procedure [3]. Firstly the orthogonal IF set is determined using the function set as the starting point $u_i = \mathrm{s}^{i-1}/e(\mathrm{s})$:

$$
\begin{aligned}
&v_1 = u_1 \\
&v_2 = u_2 - \lambda_{21} \cdot v_1 \\
&v_3 = u_2 - (\lambda_{31} \cdot v_1 + \lambda_{32} \cdot v_2) \\
&\ldots \\
&v_n = u_2 - (\lambda_{n1} \cdot v_1 + \lambda_{n2} \cdot v_2 + \ldots + \lambda_{n,n-1} \cdot v_{n-1})
\end{aligned}
\tag{14}
$$

with

$$\lambda_{ij} = (u_i \bullet v_j)/(v_j \bullet v_j) = (u_i \bullet v_j\}\Big/\left(\|v_j\|_2\right)^2 \tag{15}$$

The inner products and the norm of functions are determined using the residue theorem. The rational form of functions makes this possible. So the application uses the following relations:

$$
\begin{aligned}
u_i \bullet v_j &= \int_{-\infty}^{\infty} R_{ij}(\omega) \cdot \mathrm{d}\omega = \\
&= 2 \cdot \pi \cdot \mathrm{j} \cdot \sum_{\substack{m \\ \mathrm{Im}(pol_m > 0)}} rez(R_j, pol_m)
\end{aligned}
\tag{16}
$$

$$\|v_j\| = \sqrt{2 \cdot \pi \cdot \mathrm{j} \cdot \sum_{\substack{m \\ \mathrm{Im}(pol_m > 0)}} rez(R_j, pol_m)}$$

The function residues are determined in conformity with the classical definition [3]. The orthonormal functions are determined with:

$$f_i = v_i / \|v_i\|_2 \tag{17}$$

Then the topological elements are determined according to (3). Their sensitivities are evaluated and the sensitivity criteria are displayed:

$$
\begin{aligned}
&S_A = \max_{i,j,\omega \in R_a} \mathrm{Real}(\mathrm{S}_{A_{ij}}^{t(s)}),\ S_b = \max_{i,\omega \in R_a} \mathrm{Real}(\mathrm{S}_{b_i}^{t(s)}) \\
&S_c = \max_{i,\omega \in R_a} \mathrm{Real}(\mathrm{S}_{c_i}^{t(s)}),\ S_d = \max_{\omega \in R_a} \mathrm{Real}(\mathrm{S}_{d}^{t(s)}) \\
&S_\gamma = \max_{i,\omega \in R_a} \mathrm{Real}(\mathrm{S}_{\gamma_i}^{t(s)}).
\end{aligned}
\tag{18}
$$

The noise is computed using the relation:

$$P_{zg} = \left\| \|P_{n0}(\omega)\|_2 \right|_{N_i=1} = \sqrt{\sum_i \|g_i(\mathrm{j} \cdot \omega)\|_2^2} \tag{19}$$

Since sensitivities are proportional to $\{f_i(s)\}$, it results that various sensitivities go to zero in the passband. Note that both magnitude and phase sensitivities are forced to zero, where only magnitude sensitivity is zero in a ladder. The net effect is, however, not as good as this would suggest because not all sensitivities are simultaneously forced to zero and those that are not zero become larger that they would be for a minimum sensitivity structure.

From (7) one can see that if the minimum sensitivity realization exists, then $\{f_i(s)\}$ must divide $t'(s)$ and one expects some sort of useful "reciprocity" because it is known that the dual sets $\{f_i(s)\}$ and $\{g_i(s)\}$ of a good system ($\{\tilde{f}_i\} = \{g_i\}$) should yield another good one [4]. This suggests that if a unique best system exists, it must be self-dual. In the E-SSAF application, $\{f_i\}$ are chosen so that the factors of $t'(s)$ missing from $f_i(s)$ (that could be the factors of $g_i(s)$ in the case of minimum-sensitivity realizations) can appear in f_{n-i} :

$$f_i \cdot f_{n-i} = t'(\mathrm{s}) \tag{20}$$

We have also chosen the factors of $t'(s)$ in order that all $\{f_i(s)\}$ will be band pass-like functions. This was done because some preliminary results in investigation of minimum-sensitivity structures indicate that for band pass filter, the minimum sensitivity realization must be composed of band pass sub-filters. Owing to the "reciprocity", the total output noise of these realizations is very close to that of the best ladder simulation circuit.

Unlike orthogonal and orthonormal IFs that provide a single set of n functions f_i, in the DIFs case we can obtain a lot of n function sets. The application generates all DIFs sets and selects one or more DIFs sets, following the student options. Therefore we will first present the way DIFs sets are generated and then the way the optimum variant according to the student-selected criteria is chosen.

The first step in the DIFs sets determination is to obtain the derived transfer function $t'(s)$ and next, by inspecting the transmission zero polynomial order for $t'(s)$, to determine if it has roots. After normalisation, required by application formalism, the derivative transfer function roots are determined.

The DIFs cannot be generated in two cases: i) if the transmission zero polynomial order of $t'(s)$ is less than n; ii) if in all sets of IFs we have at least a function with the transmission zero polynomial order greater than the natural mode (pole) polynomial $e(s)$ (such a function is not physically possible).

The next step is to separate the real roots from the complex ones. Because all IF transmission zero polynomial coefficients must be real numbers, the conjugated complex roots are grouped in binomials. Then, the monomials (resulted from real roots) and binomials are binary coded. So the simultaneous generation of f_i and f_{n-i} DIFs of a set becomes easier.

The number of monomials and binomials possible combinations is reduced by generating only the combinations that yield polynomials with order less than n and also, the monomials and binomials of the transmission zero polynomial of $t'(s)$ that are not used in the current combination must yield a polynomial with real coefficients with an order less that n.

These combination variants, representing all possible physical DIFs variants, are combined once again in *n/*2 groups. The complementary groups are also generated in order to form a complete set of *n* DIFs. The functions that generate the combinations use the *backtracking* algorithm [6]. Each of the binary numbers generated like this represents the index into a monomial table or a binomial table containing the respective polynomials formed from the $t'(s)$ transmission zero polynomial roots.

IV. Implementation and Use of E-SSAF

The tool was implemented in C++ using Visual Studio 2005 Integrated Development Environment. Microsoft Foundation Classes have been used for the graphical user interface.

The general structure of the application is presented in Fig. 2. We can thus identify several component modules:

a) The Input Module that provides a graphical interface for the introduction of user data (e.g. polynomial degree, zeroes and poles, etc.). Contextual help is provided to support the student to understand both the application functionalities and the theoretical method used for the filter synthesis;

b) The Output module that provides a graphical interface for the results preview. It also offers additional information that accompanies the results in order to facilitate their understanding. This module is used for displaying filter topological parameters and various intermediate data (f, g intermediate functions set, residues, filter topology, etc.);

c) The Core Module that performs the actual computations (intermediate functions set, residues, sensitivities, operations with matrices and polynomials etc).

First the student is presented with the welcome screen, which provides a glimpse into the capabilities of the application, as seen in Fig. 3.

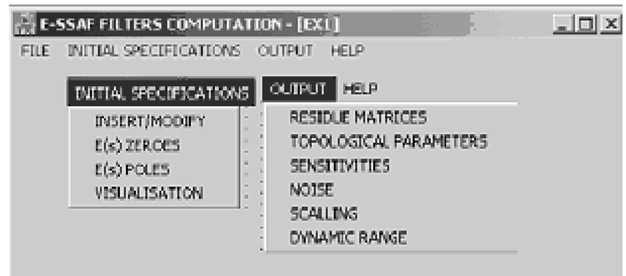

Fig. 3. E-SSAF – welcome screen

Next the student can input the initial specifications, either in the form of transfer functions or of poles/zeroes (see Fig. 4). She/he can then choose to visualize the generated intermediate functions, as can be seen in Fig. 5.

V. Conclusions

In today's VLSI era, when the need for low-sensitivities low-noise filter performances is an important aspect, the need for efficient software support tools has also become apparent. This is why we have devised a powerful tool that can be used in educational settings, helping the students during advanced electronics laboratory sessions, in order to determine the topological elements of active analogue continuous-time filters.

As future work, we intend to conduct experiments with our tool in real-world settings, in order to prove its usability and effectiveness.

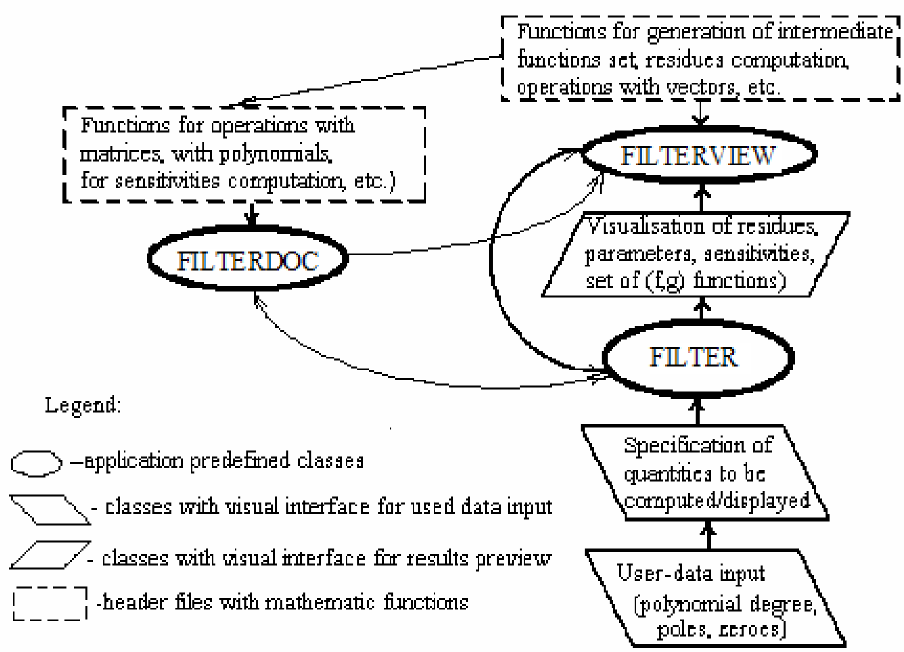

Fig. 2 General structure of E-SSAF

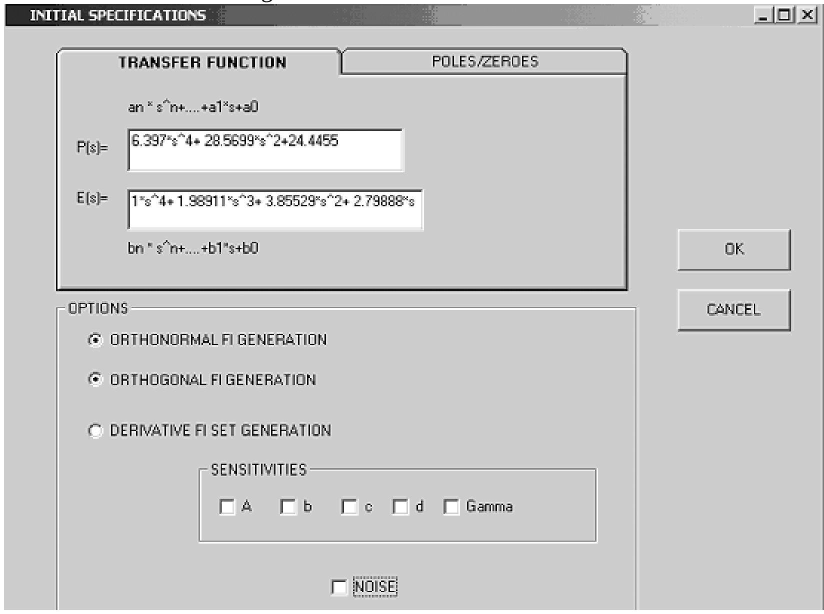

Fig. 4. Initial specifications

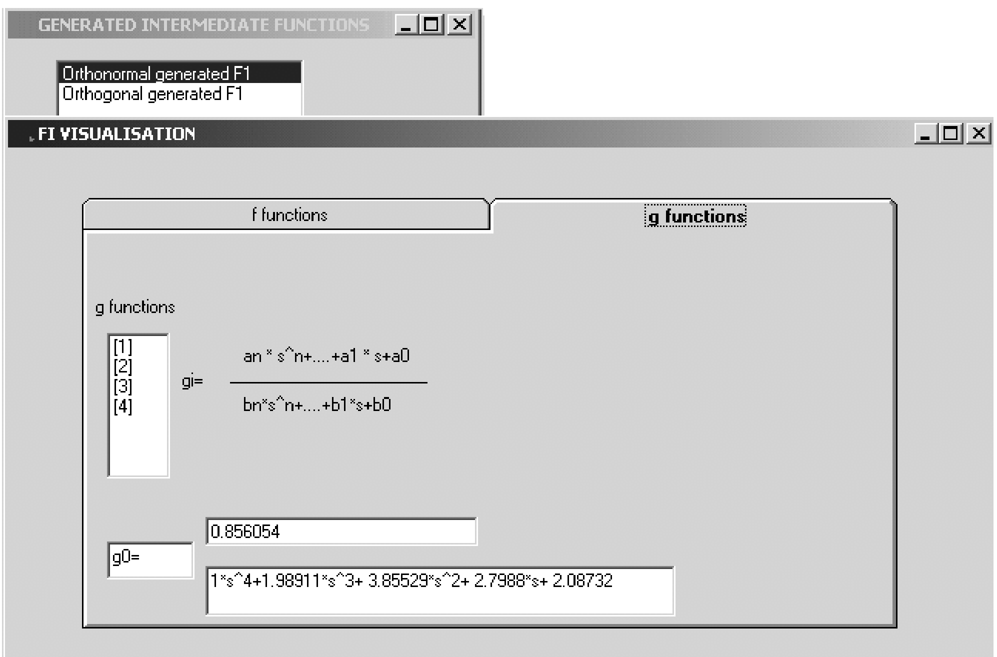

Fig. 5 Intermediate functions visualization

REFERENCES

[1] Snelgrove WM, Sedra AS, Synthesis and Analysis of State-Space Active Filters Using Intermediate Transfer Functions, *IEEE Transaction on Circuits and Systems*, vol. CAS-33, pp. 287-300, 1986

[2] Snelgrove WM, Sedra AS, A novel synthesis method for state-space active networks, *Proc. Midwest Symp. Circuits and Systems*, Toledo, 1980

[3] Sabac IG , *Special Mathematics* (in Romanian). Editura Didactica si Pedagogica. Bucharest, 1981

[4] Doicaru E (1998) The Optimum Automatic Synthesis at Active Analogue Continuous-Time High-Order Filters, *Proc. 6th International Symposium on Automatic Control and Computer Science*, 20-21, Vol. II Computer Science, Iasi, Romania, pp. 182-189 , Nov. 1998

[5] Press WH, Teukolski SA, Vetterling WT, Flannerry BP *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992

[6] Doicaru E, Barbulescu L, Dan C, (2007) *SSAF- A Powerful Tool for High Order Analog Continous Time*

Filter Synthesis, European Computing Conference, (in press at Springer Verlag).