FACTA UNIVERSITATIS
Ser: **Elec. Energ.** Vol. 25, N° 1, April 2012, pp. 59 - 68
DOI: 10.2298/FUEE1201059P

AUTOMATED KNOWLEDGE BASED FILTER SYNTHESIS USING MODIFIED CHEBYSHEV POLYNOMIALS OF THE FIRST KIND

Vlastimir D. Pavlović[1], Miroslav D. Lutovac[2], Maja M. Lutovac[3]

[1]University of Niš, Faculty of Electronic Engineering, Serbia
[2]University Singidunum, Belgrade, Serbia
[3]Lola Institute, Belgrade, Serbia

Abstract. *The paper presents the automated design of active RC and programmable filters. The approximating function is derived using Chebyshev orthogonal polynomials of the first kind. Optimization is performed using symbolic manipulation of expressions inputted into computer algebra system. The code is in the form of template notebook, and the user should specify the minimal number of numeric values, such as the number of second-order filter sections, the pass-band edge frequency, the reflection coefficient, and the preferred values of component values.*

Key words: *active filter, approximation, computer algebra systems, filter synthesis, knowledge representation*

1. INTRODUCTION

Analog filters are frequency-selective electrical circuits that are used to pass or reject a single sinusoidal signal component or a portion of the signal frequency spectrum [1]. Many different technologies may be used to implement analog filters. Also, many different physical components can be used for implementation. In practice, the values of filter components will diverge from the ideal and the filtering results may be different from the expected frequency response.

It is important to choose a filter structure and filter components with low sensitivity to the errors in the intended implementation technology in order to provide high production yield and low production cost. For that reason, the filter approximation, the design and tuning procedure should be appropriate for the intended technology. Therefore, the filter approximation, that is a mathematical description by the system of equations, should be appropriate for optimization and centering. Automated synthesis, tolerance analyses, and redesign procedure for improving the production yield or robustness to temperature changes are of great importance for production of efficient filters.

Received December 4, 2011
Corresponding author: Vlastimir D. Pavlović
University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia
E-mail: vlastimir.pavlovic@elfak.ni.ac.rs

At an early stage in the design process it is important to estimate the effect of component tolerances and parasitic elements, and also to estimate the changes affected by the temperature and aging [2, 3].

Numeric filter design software offers many solutions, and the proposed implementations may be inadequate for the chosen implementation technology [1]. For example, classic filter approximations may be inadequate to guarantee that the implemented filters will work well with real components or for temperature changes. Software environment with appropriate additional algorithms, as reasoning-based intelligent system, is the only solution for automated successful filter production that combines mathematical models (approximation), synthesis (computation of component values), and analysis of component imperfections. Currently, over thirty mathematics software products are widely applied. These mathematics software products can be divided into two categories. The first type of software environments are called computer algebra systems (CAS) such as Mathematica [4] that features on mathematical analysis, symbolic computation, formula reasoning and are capable of computing the analytical solution to mathematical problems. The second category is the numerical analysis software such as Matlab.

The main benefit of canned software, such as FDATOOL in Matlab, is that one does not have to know the mathematical theory in order to apply the numerical methods for solving real-life problems. On the other hand, computer algebra systems are well suited for teaching analog systems and solving real-life engineering problems and they are just examples of the theory applied for practical solving circuit devices [5]. Symbolic formulation of microwave circuit equations by a computer algebra system [4] can be presented from educational viewpoint [6], and the symbolic derivation of microwave network models can be incorporated in the numeric software tool for three-dimensional space electromagnetic circuit simulation.

Knowledge for filter synthesis can be accessible as script code so that the user can modify, improve, optimize commands for filter synthesis, such as the software described in [1] and [7]. Numeric only software tools can be usually used for the design and synthesis without possibility to change or adapt the software code for specific applications.

In Section 2 we first define the knowledge for generating approximating function. In section 3, we present how the knowledge is built into the computer algebra system. In Section 4, we illustrate automated synthesis of active opamp RC filters.

2. APPROXIMATION

Continuous-time active RC filters are suitable for integration into the analog front end of mixed mode VLSI (Very-Large-Scale Integration) chips, simple design using programmable analog and SC (Switched Capacitor) integrated filters, or as the classic analog filters such as active opamp RC (**op**erational **amp**lifier, Resistor and Capacitor). The design procedure is usually based on the sensitivity and tolerance analysis [8] – [14] in order to manufacture robust filters, to increase the production yield and to minimize the cost of mass production. The sensitivity and tolerance analysis allows the designer to predict variations of the filter performances and to predict the production yield (which is defined as a ratio of the number of manufactured filters satisfying the specification to the total number of manufactured filters) [1].

The most general form of low-pass prototype all-pole filter transfer function is

$$H_n(s) = \frac{K}{\prod_{r=1}^{n}(s - s_r)} . \tag{1}$$

The filter order is n, K is a constant to specify the attenuation at s=0 (for example 0 dB), and the poles of the transfer function are

$$s_r = \sigma_r + j\omega_r,\ r \in \{1,2,3,...,n\} . \tag{2}$$

The squared magnitude response is

$$H_n(j\omega)H_n(-j\omega) = \frac{1}{1+\varepsilon^2 A_n(\omega^2)} . \tag{3}$$

The denominator of the squared magnitude response is a polynomial in ω, with real constants. The parameter ε determines the attenuation at the pass-band edge, and it can be given as

$$\varepsilon^2 = \frac{\rho^2}{1-\rho^2} , \tag{4}$$

ρ is the pass-band reflection factor.

The characteristic function, $A_n(\omega)$, is normalized to 1 at the pass-band edge frequency, ω_p,

$$A_n(\omega_p) = 1 . \tag{5}$$

The squared magnitude response of proposed class becomes

$$A_n(\omega^2) = \sum_{r=0}^{n} b_{2r} T_{2r}(\omega) , \tag{6}$$

$T_{2r}(\omega)$ is the even $2r$-th-order Chebyshev orthogonal polynomial of the first kind, and b_{2r} are real coefficients.

The approximation procedure begins with the criteria of the minimum of the ratio of the reflected power and the transmitted power is obtained by minimizing the following integral

$$I_{\min}(\omega) = \int_0^1 p(\omega) A_n(\omega^2) d(\omega) . \tag{7}$$

In the following derivations, $p(\omega) = 1/\sqrt{1-\omega^2}$ is the weighting function. Firstly, we define a function using conditions of the proper prototype low-pass approximation for odd order

$$\begin{aligned}&\phi(b_0,b_2,b_4,b_6,...b_{2n},\lambda_0,\lambda_1) = \\ &\int_0^1 [\sum_{r=0}^{r=n} b_{2r} T_{2r}(\omega)]^2 d(\omega) - \lambda_0[\sum_{r=0}^{r=n} b_{2r} T_{2r}(0)] - \lambda_1[\sum_{r=0}^{r=n} b_{2r} T_{2r}(1) - 1] .\end{aligned} \tag{8}$$

where λ_0 and λ_1 are Legendre multiplications.

Next, we derive partial derivatives and set a system of equations that should be solved in terms of $b_0, b_2, b_4, b_6, ... , b_{2n}$

$$\frac{\partial}{\partial b_0} \phi\ (b_0, b_2, b_4, b_6, \ldots b_{2n},\ \lambda_0, \lambda_1) = h_0 b_0 - \lambda_0 T_0(0) - \lambda_1 T_0(1) = 0 \cdot \tag{9}$$

$$\begin{gathered}\frac{\partial}{\partial b_{2r}} \phi(b_0, b_2, b_4, b_6, \ldots b_{2n}, \lambda_0, \lambda_1) = \\ h_{2r} b_{2r} - \lambda_0 T_{2r}(0) - \lambda_1 T_{2r}(1) = 0,, r = 1, 2, 3, \ldots, n\end{gathered} \ . \tag{10}$$

$$\frac{\partial}{\partial \lambda_0} \phi(b_0, b_2, b_4, b_6, \ldots b_{2n}, \lambda_0, \lambda_1) = \sum_{r=0}^{r=n} b_{2r} T_{2r}(0) = 0 \cdot \tag{11}$$

$$\frac{\partial}{\partial \lambda_1} \phi(b_0, b_2, b_4, b_6, \ldots b_{2n}, \lambda_0, \lambda_1) = \sum_{r=0}^{r=n} b_{2r} T_{2r}(1) - 1 = 0 \cdot \tag{12}$$

The squared norm, h_r, of the orthogonal Chebyshev polynomial is defined by $T_{2r}(\omega)$, $r = 0,1,2,3,\ldots$

$$h_r = \int_0^1 p(\omega) T_{2r}(\omega) T_{2r}(\omega) d(\omega) \quad r = 0, 1, 2, 3, \ldots \tag{13}$$

The integration gives the values of h_r in terms of r, where

$$h_0 = \pi, \ r = 0 \ , \tag{14}$$

that is

$$h_r = \frac{\pi}{2}, \quad r = 1, 2, 3, \ \ldots \tag{15}$$

Let us design a filter of the order $n = 6$ with $\rho = 0.25$. The poles of the transfer function are computed using equations (4)-(12). The characteristic function of this filter is presented in Figure 1. It is important to note that attenuation oscillates around 0. This type of approximation cannot be implemented using a lossless LC filter because it has negative attenuation (gain) in the pass-band.

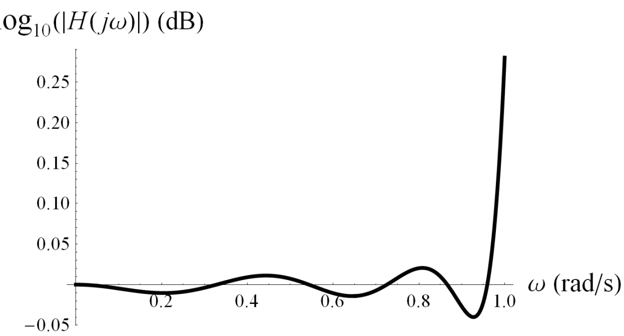

Fig. 1. Magnitude response in dB in the passband.

Figure 2 illustrates the attenuation, which increases monotonically in the stop-band because the transfer function is the so-called all-pole filter function.

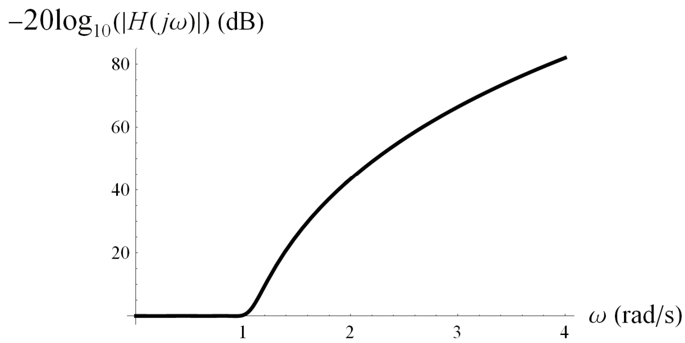

Fig. 2. Attenuation of the sixth-order filter.

3. Automated Building Knowledge for Generation of Transfer Function

The design procedure presented in some papers can be very simple, but it may not be so simple to implement that knowledge in the numeric software, especially when some special functions are used or the closed form solutions of the integration are required.

In this section we present how the knowledge from the previous section is built into a computer algebra system such as Mathematica.

First of all, the code in Mathematica is in the so-called notebooks that can be used as template for modification and simple adaptation to different applications. It is very important to specify the minimal number of parameters, so that the main code can be used with minimal changes. For example, we are going to design filters that consist of second-order sections (called biquads), and at the beginning of the code we specify the number of biquads that are restricted by implementation requirements, say the number of biquads is 3 (Figure 3).

```
nBiquads = 3;
n = 2 nBiquads;
b = Table[ToExpression[StringJoin["b", ToString[r]]], {r, 0, 2 n, 2}]

{b0, b2, b4, b6, b8, b10, b12}
```

Fig. 3. Initialization and automated generation of coefficients.

The filter order is two times larger, and all coefficients in equation (6) are generated as a list of variables. The list of symbols consists of the specified letter `b`, while the even numbers from 0 to $2n$ are appended to `b`. An input cell contains the code and output cell produces the result – the list of coefficients.

The next input cell, Figure 4, defines the approximation according to equation (6) using automatically generated list of coefficients and special function – Chebyshev polynomial. Using the command Plot, the characteristic function is obtained as shown in Figure 1.

The next step is to create the function that can be used for optimization, see Figure 5. We may use the list of coefficients, and two additional parameters according to equation (8), but the polynomial is defined as unknown function pP of two arguments. The integral cannot be solved because the function pP is unknown. Nevertheless, the partial derivatives can be defined according to equations (9) and (10), as shown in Figure 6.

$$\text{a2} = \left(\sum_{r=0}^{n} (\text{b[[r + 1]] ChebyshevT[2 r, w]})\right) \text{ // Simplify // Flatten // Factor}$$

$$b0 - b10 + b12 - b2 + b4 - b6 + b8 + 50\,b10\,w^2 - 72\,b12\,w^2 + 2\,b2\,w^2 - 8\,b4\,w^2 + 18\,b6\,w^2 - 32\,b8\,w^2 - 400\,b10\,w^4 + 840\,b12\,w^4 + 8\,b4\,w^4 - 48\,b6\,w^4 + 160\,b8\,w^4 + 1120\,b10\,w^6 - 3584\,b12\,w^6 + 32\,b6\,w^6 - 256\,b8\,w^6 - 1280\,b10\,w^8 + 6912\,b12\,w^8 + 128\,b8\,w^8 + 512\,b10\,w^{10} - 6144\,b12\,w^{10} + 2048\,b12\,w^{12}$$

Fig. 4. Building the knowledge of approximation.

$$\phi = \int_0^1 \left(\left(\sum_{r=0}^{n} \text{b[[r + 1]] pP[2 r, w]}\right)\right)^2 dw - \text{L0} \sum_{r=0}^{n} (\text{b[[r + 1]] pP[2 r, 0]}) - \text{L1}\left(-1 + \sum_{r=0}^{n} (\text{b[[r + 1]] pP[2 r, 1]})\right)$$

$$\int_0^1 (\text{b0 pP[0, w] + b2 pP[2, w] + b4 pP[4, w] + b6 pP[6, w] + b8 pP[8, w] + b10 pP[10, w] + b12 pP[12, w]})^2\, dw -$$
$$\text{L0 (b0 pP[0, 0] + b2 pP[2, 0] + b4 pP[4, 0] + b6 pP[6, 0] + b8 pP[8, 0] + b10 pP[10, 0] + b12 pP[12, 0])} -$$
$$\text{L1 (-1 + b0 pP[0, 1] + b2 pP[2, 1] + b4 pP[4, 1] + b6 pP[6, 1] + b8 pP[8, 1] + b10 pP[10, 1] + b12 pP[12, 1])}$$

Fig. 5. General form of optimization function.

The unknown function is replaced by the corresponding polynomial in the last step, by substituting the head of the function (name of the function) by the appropriate symbol for the special function. This way we can identify the numbers associated to the parameters in equations (9) and (10).

```
eqB = Table[D[ϕ, b[[r + 1]]], {r, 0, n}] /. pP[x_, 0] → ChebyshevT[x, 0] /.
   pP[x_, 1] → ChebyshevT[x, 1] /. pP[x_, v_] → ChebyshevT[x, v]
```

$$\Big\{2\,b0 - \frac{2\,b10}{99} - \frac{2\,b12}{143} - \frac{2\,b2}{3} - \frac{2\,b4}{15} - \frac{2\,b6}{35} - \frac{2\,b8}{63} - L0 - L1,$$
$$-\frac{2\,b0}{3} - \frac{206\,b10}{9009} - \frac{98\,b12}{6435} + \frac{14\,b2}{15} - \frac{38\,b4}{105} - \frac{26\,b6}{315} - \frac{134\,b8}{3465} + L0 - L1,$$
$$-\frac{2\,b0}{15} - \frac{46\,b10}{1365} - \frac{106\,b12}{5355} - \frac{38\,b2}{105} + \frac{62\,b4}{63} - \frac{34\,b6}{99} - \frac{158\,b8}{2145} - L0 - L1,$$
$$-\frac{2\,b0}{35} - \frac{6\,b10}{85} - \frac{358\,b12}{11\,305} - \frac{26\,b2}{315} - \frac{34\,b4}{99} + \frac{142\,b6}{143} - \frac{22\,b8}{65} + L0 - L1,$$
$$-\frac{2\,b0}{63} - \frac{326\,b10}{969} - \frac{46\,b12}{665} - \frac{134\,b2}{3465} - \frac{158\,b4}{2145} - \frac{22\,b6}{65} + \frac{254\,b8}{255} - L0 - L1,$$
$$-\frac{2\,b0}{99} + \frac{398\,b10}{399} - \frac{54\,b12}{161} - \frac{206\,b2}{9009} - \frac{46\,b4}{1365} - \frac{6\,b6}{85} - \frac{326\,b8}{969} + L0 - L1,$$
$$-\frac{2\,b0}{143} - \frac{54\,b10}{161} + \frac{574\,b12}{575} - \frac{98\,b2}{6435} - \frac{106\,b4}{5355} - \frac{358\,b6}{11\,305} - \frac{46\,b8}{665} - L0 - L1\Big\}$$

Fig. 6. Partial derivatives and substitutions.

The partial derivatives to other parameters can be realized in a similar way, see Figure 7.

```
eqL = {
  D[ϕ, L0] /. pP[x_, 0] -> ChebyshevT[x, 0],
  D[ϕ, L1] /. pP[x_, 1] -> ChebyshevT[x, 1]
 }

{-b0 + b10 - b12 + b2 - b4 + b6 - b8, 1 - b0 - b10 -
```

Fig. 7. Partial derivatives of other parameters.

The list of all partial derivatives, denoted eqs in Figure 8, is obtained using the command Join. In order to form a system of equation, all partial derivatives should be set to be equal to 0. The solving of automatically derived system of equations is possible using the command Solve, as illustrated in Figure 8.

```
sol1 = Solve[eqs, parameters] // Flatten
```

$$\left\{b0 \to \frac{145\,685}{1\,572\,864},\ b2 \to \frac{37\,903}{196\,608},\ b4 \to \frac{555\,713}{3\,145\,728},\ b6 \to \frac{22\,933}{131\,072},\ b8 \to \frac{232\,883}{1\,572\,864},\ b10 \to \frac{52\,003}{393\,216},\ b12 \to \frac{260\,015}{3\,145\,728},\ L0 \to -\frac{1024}{135\,135},\ L1 \to \frac{1}{45}\right\}$$

Fig. 8. Solving the system of equations.

Numeric values of all parameters after optimization of the integral (7) are known, and the characteristic function and the transfer function can be obtained by simple substitution of the parameters into equation (6). The characteristic function is obtained as shown in Figure 9

```
aw = a2 /. sol1 // Simplify
```

$$\frac{w^2\left(-3003 + 51\,051\,w^2 - 277\,134\,w^4 + 646\,646\,w^6 - 676\,039\,w^8 + 260\,015\,w^{10}\right)}{1536}$$

Fig. 9. Approximation after substitution of the coefficients with solved values.

The passband ripple is specified by entering the value of the reflection coefficient. According to equation (3), the characteristic function is defined in Figure 10

```
kw = (1 + e2 * aw) /. w → I w
```

$$1 - \frac{w^2\left(-3003 - 51\,051\,w^2 - 277\,134\,w^4 - 646\,646\,w^6 - 676\,039\,w^8 - 260\,015\,w^{10}\right)}{23\,040}$$

Fig. 10. Characteristic function in the s domain.

The transfer function contains poles from the left-half-plane in the s domain, see Figure 11.

```
rw = w /. Sort[NSolve[kw == 0, WorkingPrecision → 64]];
rwl = rw[[1 ;; Length[rw] / 2]]
```

```
-0.538203 - 0.274387 i
-0.538203 + 0.274387 i
-0.385336 - 0.760416 i
-0.385336 + 0.760416 i
-0.137409 - 1.05048 i
-0.137409 + 1.05048 i
```

Fig. 11. Selection of the poles from the left s plain.

Squared pole frequencies and pole Q-factors are computed using code illustrated in Figure 12.

```
wp2 = Table[rwl[[i]] rwl[[i + 1]], {i, 1, Length[rwl] - 1, 2}] // Chop;
N[%, 16] // MatrixForm
vp2 = Table[-rwl[[i]] - rwl[[i + 1]], {i, 1, Length[rwl] - 1, 2}] // Chop;
```

```
0.3649503915020269
0.7267164987467748
1.122388432121841
```

$$Qp = N\left[\frac{\sqrt{wp2}}{vp2}\right];$$

```
% // MatrixForm
```

```
0.56123
1.10615
3.855
```

Fig. 12. Squared pole frequencies and pole Q-factors.

4. Automated Synthesis

The number of possible realization structures is practically infinite. The knowledge based design should be carried on by precise rules. Therefore, the best filter structures are built into specific containers. For example, there are several filter structures that are selected according to the pole Q-factor [1]. In the case when the universal filter section is more preferable than specific sections, general purposes biquads can be used.

Suppose that the implementation requirement is to use the simplest biquad with the minimal number of components (one operational amplifier, two resistors, and two capacitors), the capacitors from the standard set of values, and with the minimal ratio of resistor values. The next step is to define a function for synthesis, see Figure 13. The function returns the component values in terms of the pole-frequency and the pole Q-factors. One capacitance can be chosen from the standard list, while parameter P can be selected in optimization.

```
designLPLQ[Qp_, wp_, Cx_, P_] := Module[
  {C2, C4, R1, R3},
  C4 = Cx;
  C2 = 2 Qp^2 Cx (1 + (1 + P^2)/(2 P));
  R1 = 1/(wp Cx Qp (1 + P));
  R3 = P R1;
  {R1, C2, R3, C4}];
```

Fig. 13. Function for synthesis

The design procedure is very simple when computer algebra system is used. Firstly, all components can be expressed as a function of unknown parameter P. By solving the equation so that the value of the second capacitor is equal to the value from the standard list of capacitance values, the parameter P is computed. The whole list of standard capacitance values can be searched until the optimal value for P is obtained. The implementation requirement is to have P from the range 0.1<P<10, but the value closest to 1 is the best solution. Using the proposed procedure, all component values are computed for the frequency normalization to 4100 Hz, so that the passband edge frequency is 4100 Hz, see Figure 14. The filter topology and design procedure are the same as in [1].

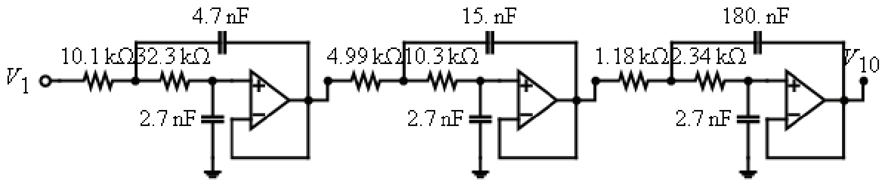

Fig. 14. Synthesized the sixth-order filter.

All capacitances connected to the + node of the amplifiers are chosen to be 2.7 nF, while other capacitances are obtained by optimization. All P parameters are from the range 2<P<3.3. In order to prevent nonlinear distortion at the outputs of amplifiers, that is the amplitude of the sinusoidal signal cannot be larger at any amplifier output than at the filter input, the design provides no gain at any biquad outputs (important when large signals are processed [1]). The maximal attenuation in the passband is less than 12 dB (important when large signals are processed [1]). The proposed method provides the filter design with small resistances ratio (P<3.3), predefined values of 50% of used capacitance (2.7 nF), the minimal value of the maximal attenuation in the passband at biquad outputs (less than 12 dB). The whole procedure is automated and only a minimal number of parameters have to be defined, for example the number of biquads and filter specification.

The next step can be the sensitivity analysis. Since all parameters are defined as symbols, the sensitivity analysis is simple because the sensitivity functions are already programmed and they can be used as build-in knowledge [1]. Instead of optimization resistance ratio, the gain-sensitivity can be optimized [1], or the sensitivity to element values [5].

5. Conclusion

The filter analysis and design theory are very well established in many textbooks and are available through canned numeric software. In this paper an automated knowledge based filter synthesis is presented. The basic idea is to build knowledge of computer algebra systems by coding formulas. All parameters are kept as symbols until the moment when the numeric values should be computed, say to plot frequency response.

The approximation function can be any rational function, and some parameters can be optimized for preferred criteria, such as the minimum of the ratio of the reflected power. The rational function can be replaced by appropriate function, such as Chebyshev polynomial. The knowledge of preferred filter structure has to be a part of filter synthesis knowledge; the most suitable structure is selected from the collection of schematics, with the description as system of equations and the corresponding transfer function. Higher order filter can be designed using simpler cascaded filter blocks. The decision is made on filter parameter or implementation requirements such as the simplest implementation, the element values from predefined or standard set of values, or the minimal sensitivity.

The future work is to develop software that will be available for the usage of web based computer algebra systems.

References

[1] M. D. Lutovac, D. V. Tošić, and B. L. Evans, Filter Design for Signal Processing Using MATLAB and Mathematica, Prentice Hall, New Jersey, USA, 2000.

[2] V. D. Pavlović, M. V. Popović, An Iterative Method for Lossy LC Ladder filter synthesis, Proceedings of European Conference on Circuit Theory and Design, Paris, pp. 185-190, 1987.

[3] B. D. Raković, V. D. Pavlović, A method of designing double terminated lossy ladder filters with incresed element tolerances, IEE Proceedings Pt G, Vol. 134, No. 6, pp. 285-291, 1987.

[4] S. Wolfram, The Mathematica Book, Wolfram Media, Champaign USA, 2003.

[5] M. Lutovac, Design of Analog and Digital Filters Using Computer Algebra Systems, Academic Mind, in Serbian, Belgrade, 2011.

[6] D. V. Tošić, V. V. Petrović, Formulation of microwave circuit equations: An educational viewpoint, Proc. of 9th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, TELSIKS 2009, Niš, Serbia, pp. 160-163, 2009.

[7] M. D. Lutovac and D. V. Tošić, SchematicSolver - Symbolic Signal Processing, Version 2.2, Miroslav Lutovac, Belgrade, Serbia, 2010.

[8] A. Ilic and V. D. Pavlovic, New class of filter functions generated most directly by Christoffel-Dorboux formula for Gegenbauer orthogonal polynomials, International Journal of Electronics, Taylor & Francis, Vol. 98, No. 1, pp. 61-79, 2011.

[9] V. D. Pavlović, An explicit form of all-pole filter function with decreasing envelope of the summed sensitivity function, Int. J. Circ. Theor. Appl., Vol. 39, No. 5, pp. 515–531, 2011.

[10] V. D. Pavlovic and A. Ilic, Filter functions generated most directly by the Christoffel-Dorboux formula for Jacoby polynomials, International Journal of Electronics, Taylor & Francis, to be published TEBT-2011-0133, 2011.

[11] M. Lutovac, V. D. Pavlović, Efficient multirate EMQF digital filters // Electronics and Electrical Engineering. – Kaunas: Technological. – Vol. 119, No. 3. – pp. 39–44, 2012.

[12] V. D. Pavlović, M. M. Lutovac, M. D. Lutovac, Automated knowledge-based filter synthesis using modified Legendre approximation and optimization of summed sensitivity, Int. J. Reasoning-based Intelligent Systems, Vol. 4, Nos. 1/2, pp.17-27, 2012.

[13] V. D. Pavlović and Ć. B. Dolićanin, (2011), 'Mathematical Foundation for the Christoffel-Darboux Formula for Classical Orthonormal Jacobi Polynomials Applied in Filters', Scientific Publications of The State University of Novi Pazar Ser. A: Appl. Math. Inform. and Mech, Vol. 3, No. 2, pp.139–151, 2011.

[14] V. D. Pavlovic, New class of filter functions generated directly by the modified Christoffel-Darboux formula for classical orthonormal Jacobi polynomials, Int. J. Circuit Theory and Applications, 2012, (in press, DOI: 10.1002/cta.1817), 2012.