# AN APPROACH TO PASSIVE CIRCUIT SYNTHESIS USING GENETIC ALGORITHMS IN MATLAB – BAND-PASS FILTER

**Vladislav Petrov Durev**

Department of Electronics, Technical University of Sofia, Kl. Ohridski 8 Blvd., 1000 Sofia,Bulgaria, phone +359 888 492 802, e-mail:v_p_durev@yahoo.com

*An approach to passive circuit synthesis in MATLAB is proposed in the present paper. The Modified Nodal Analysis is used for the construction of the admittance circuit matrix. A circuit of passive band-pass filter is used for the demonstration of the method. The algorithm is easy to implement in MATLAB environment. A methodology for the verification of the developed approach is proposed together with structural-parametric optimization of the circuit.*

**Keywords:** Genetic Algorithm (GA), Modified Nodal Analysis (MNA), Circuit Synthesis

## 1. INTRODUCTION

The successful implementation of the electronic devices and reduction the costs for redesign and production are guaranteed by the accuracy of the corresponding computer models. Very important problem that calls for solution is the automatic creation of a passive electronic circuit for a given input characteristic, for example the circuit frequency response. The genetic algorithms (GA) [1,2,3] are very applicable in circuit synthesis because this is a multiple variables optimization problem. The space of search of the genetic algorithm depends only on the range of variation of the input independent variables. In the present paper a general approach to passive circuit synthesis is developed and compared to the results, obtained via similar methods [1,3]. The presented example of passive band-pass filter synthesis is applicable and with a good accuracy. It is based on a general electronic circuit model, developed using the possibilities of MATLAB and GA toolbox [4].

### 1.1 General Concept of the Genetic Algorithms

The genetic algorithm is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms(also known as evolutionary computation) that use techniques inspired by evolutionary biology such as *inheritance, mutation, selection,* and *crossover*(also called *recombination*).

There is a variety of examples for multiple variable optimizations in microelectronics. The geometric technology parameters for a given technology can be optimized to achieve given electrical characteristics [5]. The structure of the circuits can also be optimized as well as the values of the components [1,3,6]. The main advantage of the GA is that they do not need preliminary information or other knowledge for the investigated problem. Only the definition of the purpose function is important for realizing the effective global search.

### 1.2 Modified Nodal Analysis

The Modified Nodal Analysis (MNA) [7] is an extension of the nodal analysis method that not only determines the circuit's node voltages (as in the classical nodal analysis), but also some branch currents. The circuit elements without admittance description are defined by the corresponding component equations. As a result, the circuit matrix order increases, but MNA does not impose restrictions on the element types and is very applicable for circuit description with programming language. The MNA equations can be formulated (represented in a computer program) automatically in a simple, comprehensive manner. Once formulated, the system of equations has to be solved. This can be easily done using mathematical software like MATLAB with most of the operations over matrices incorporated. In this paper the MNA is applied to circuits with passive elements and independent sources.

## 2. STRUCTURAL-PARAMETRIC SYNTHESIS OF A BAND-PASS FILTER

A *PSpice* [8] model of a passive band-pass filter is presented in Fig. 1 and its frequency response is presented in Fig. 2 [1]. The resistor $R_s$ represents the internal resistance of the voltage source and $R_l$ is the load resistor. The input voltage source $V_{in}$ and the resistors $R_s$ and $R_l$ are fixed in the structure, i.e. the structure optimization will be performed only for the components $L_b$, $L_c$, $L_d$, $L_g$, $L_l$, $C_f$, $C_h$, $C_j$ and $C_k$. The circuit has eight nodes including the ground node GND (bp1, bp2, bp3, bp4, bp5, bp6, bp7 and GND). The GA makes the difference between the initial frequency response and the optimized frequency response minimal. The independent input variables for the structural-parametric synthesis are the nodes of the circuit, excluding the GND node and the values of the components $L_b$, $L_c$, $L_d$, $L_g$, $L_l$, $C_f$, $C_h$, $C_j$ and $C_k$. The range of variation for every node is from 0 (GND) to 7 (number of nodes excluding the GND node). The range of variation for every component value is from 0 to 12. The type of the component – inductance, capacitor or resistor is fixed for this example. The initial values of the model parameters for the circuit in Fig. 1 are: Lb = 0.2924, Lc = 0.4298, Ld = 5.5722, Lg = 0.03716, Ll = 0.01847, Cf = 3.1074, Ch = 0.2497, Cj = 23.881, Ck = 4.3. The initial parameters of the algorithm are:

NIND = 200
MAXGEN = 1000
NVAR = 27
PRECI = 200
GGAP = 0.7

where [4]:

NIND – number of individuals
MAXGEN – maximum number of iterations
NVAR – number of variables

PRECI – generation gap for the population

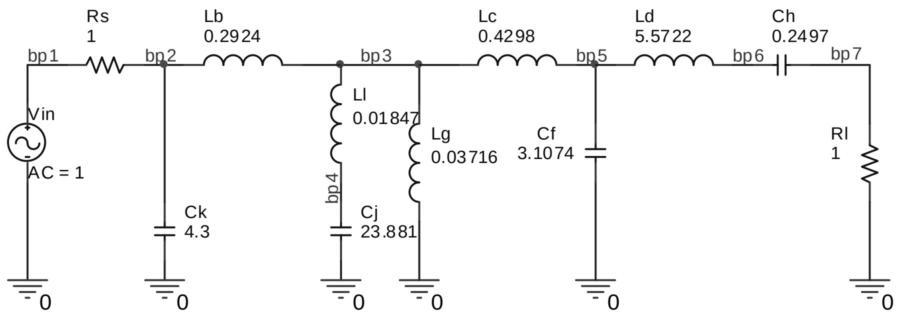

**Fig.1 PSpice model of passive band-pass filter – initial circuit**

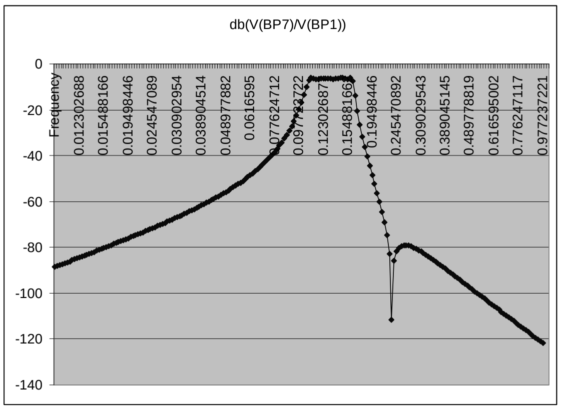

**Fig.2 Frequency response of the initial circuit of the band-pass filter**

The frequency response in Fig. 2 is used as an input data for the optimization program in MATLAB and it is stored in Microsoft Excel format .xls. All the resulting output data, generated from the optimization program is also stored in .xls format. A *FieldD* matrix is used to fix the ranges of the input variables [4]:

```
            %N1_1
FieldD = [PRECI              …
          0                  …
          NODES_NUMBER-1 …
          1                  …
          0                  …
          1                  …
          1                  …];
```

NODES_NUMBER represents the number of the nodes in the circuit including the GND node. For the structural-parametric optimization the algorithm varies the incident nodes for each of the components from 0 (GND) to (NODES_NUMBER – 1) as well as the component values. As the node values of $V_{in}$, $R_s$ and $R_l$ are fixed, the other nine components $L_b$, $L_c$, $L_d$, $L_g$, $L_l$, $C_f$, $C_h$, $C_j$ and $C_k$ are to be optimized in the circuit. Each of the components has two pins and one value, so the number of the variables for the structural synthesis is twenty-seven, i. e. NVAR = 27. The input matrix *Input* is created for the circuit which is with a *PSpice* netlist-like structure:

```
Input = [1  1                 2                 1;
         2  round(N11(ix))  round(N12(ix))  N10_1(ix);  % 0.2924;
         2  round(N21(ix))  round(N22(ix))  N10_2(ix);  % 0.01847;
         2  round(N31(ix))  round(N32(ix))  N11_1(ix);  % 0.03716;
         2  round(N41(ix))  round(N42(ix))  N11_2(ix);  % 0.4298;
         2  round(N51(ix))  round(N52(ix))  N12_1(ix);  % 5.5722;
         3  round(N61(ix))  round(N62(ix))  N12_2(ix);  % 4.3;
         3  round(N71(ix))  round(N72(ix))  N13_1(ix);  % 23.881;
         3  round(N81(ix))  round(N82(ix))  N13_2(ix);  % 3.1074;
         3  round(N91(ix))  round(N92(ix))  N14_1(ix);  % 0.2497;
         1  (NODES_NUMBER-1) 0              1];         % Load
```

The first column represents the component type: resistor – 1, inductor – 2 or capacitor – 3. The component type is fixed for this example. The fourth column represents the component values. N10_1 to N14_1 are the variables which represent every component value. The initial component values are shown as commented lines as last column. N11 to N92 are the variables, which represent every circuit node. An internal cycle calculates the *Input* matrix for every individual *ix*, as the independent input variables are vectors of generated values [4]. The node values are rounded, because the actual generated numbers are not integer numbers, which is needed for the number of the node representation. The components are not allowed to connect to node bp1, which is fixed.

The Y-matrix of the circuit is constructed on the base of the *Input* matrix using the MNA. The I- and U-matrices of the circuit are constructed and the matrix circuit equation is solved in the following way:

```
Y(1, NODES_NUMBER) = 1; %Inserting the voltage source into the matrices
Y(NODES_NUMBER, 1) = 1;
 I(NODES_NUMBER, 1) = Vin;
 U = pinv(Y)*I;
Vout_real(ix, 1) = real(U((NODES_NUMBER-1), 1)); %The output voltage is captured
Vout_imag(ix, 1) = imag(U((NODES_NUMBER-1), 1));
Vout_module(ix, 1) = abs(U((NODES_NUMBER-1), 1));
```

*Vout_real*, *Vout_imag* and *Vout_module* vectors are used to build the frequency response of the circuit. Once built, this frequency response is compared to the input frequency response and the GA is optimizing the difference between the two frequency responses to be minimal. The purpose function is presented in the following way:

```
if((frequency(i) > 0.105) && (frequency(i) < 0.106))
    g_fun = g_fun + 10e12*(abs(Vout_module - Vout_initial_module));
elseif((frequency(i) > 0.135) && (frequency(i) < 0.136))
    g_fun = g_fun + 10e12*(abs(Vout_module - Vout_initial_module));
elseif((frequency(i) > 0.168) && (frequency(i) < 0.169))
```

```
        g_fun = g_fun + 10e12*(abs(Vout_module - Vout_initial_module));
    end

    g_fun = g_fun + abs(Vout_module - Vout_initial_module);
```

The points from three character ranges are summed with coefficients of 10e12 – two ranges from the two steep areas of the frequency response from Fig. 2 and one range in the pass band. The other points are also summed with coefficients of 1, which is needed for the best match. The goal is *g_fun* to be zero (0). A simple verification of the algorithm can be performed first, fixing the parameters of the initial circuit from Fig. 1 into the matrices *FieldD* and *Input*. The result for the *g_fun* should be constant 0 for every iteration. Once the verification is performed successfully, the values of the independent variables are controlled from the GA in order to find the solution, which best fits the input requirements. The last generated *Input* matrix for this optimization is:

```
Input = [1.00000  1.00000   2.00000   1.00000000000000
         2.00000  0         2.00000   0.29702504487341
         2.00000  5.00000   7.00000   8.37458641961925
         2.00000  0         6.00000   3.51318350988257
         2.00000  3.00000   0         9.75398857974119
         2.00000  3.00000   6.00000   10.95486065172660
         3.00000  2.00000   3.00000   0.25697844864614
         3.00000  6.00000   5.00000   1.56241583053041
         3.00000  7.00000   6.00000   0.82198891629322
         3.00000  0         2.00000   5.08412189035098
         1.00000  7.00000   0         1.00000000000000];
```

The corresponding circuit is shown in Fig. 3. The frequency response of the initial circuit in Fig. 1 and the frequency response of the optimized circuit, shown in Fig. 3, are presented in Fig. 4.

## 3. CONCLUSIONS

A GA optimization approach to passive circuit synthesis of passive band-pass filter has been developed in the present paper. The synthesis procedure is realized using the GA toolbox in MATLAB. The obtained results are in agreement with previously published passive circuit synthesis optimizations and have a good accuracy. The methodology can be optimized for better results mainly using the improved expression for the purpose function.

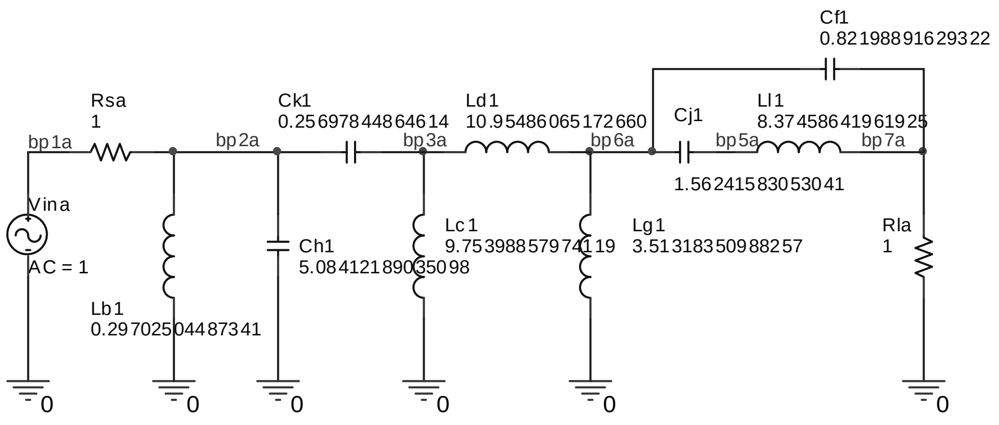

**Fig.3 The band-pass filter circuit after the structural-parametric GA optimization**

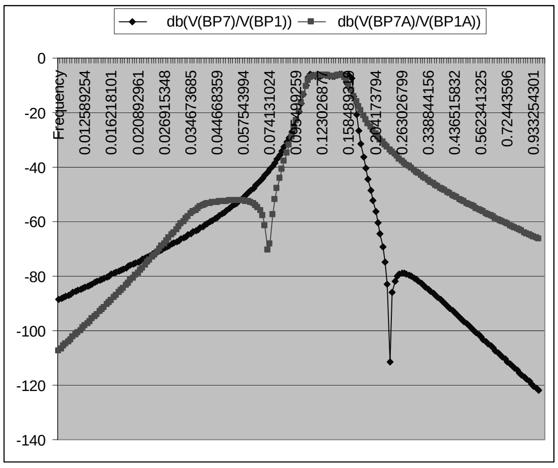

**Fig.4 Comparison between the frequency response of the initial and the GA built circuit**

## 5. References

[1] Grimbleby, J.B., *Automatic Analogue Network Synthesis using Genetic Algorithms*, IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95), Sheffield, 12–14 Sept. 1995, IEE Conference Publication No.414, pp. 53–58.

[2] Sivanandam, S.N., S.N.Deepa, *Introduction to Genetic Algorithms*, Springer, 2008

[3] Zebulum, R., M. Pacheco, M. Vellasco, *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*, CRC Press, 2002.

[4] Chipperfield, A., P. Fleming, H. Pohlheim, C. Fonseca, *Genetic Algorithm TOOLBOX for use with MATLAB*, User's Guide Version 1.2, Department of Automatic Control and Systems Engineering, University of Sheffield.

[5] Gadjeva, E., V. Durev, M. Hristov, D. Pukneva, *Optimization of geometric parameters of spiral inductors using genetic algorithms*, 13th International Conference Mixed Design of Integrated Circuits and Systems, 22-24 June, 2006, Gdynia, Poland.

[6] Durev, V.P., E.D. Gadjeva, *Passive Circuit Synthesis using Genetic Algorithms in MATLAB*, The 9th WSEAS International Conference on Evolutionary Computing(EC'08), 2-4 May, 2008, Sofia, Bulgaria

[7] Vlach, J., K.Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold Company, New York, 1893.

[8] *PSpice User's Guide*, Cadence Design Systems, Inc.,2000.