Journal of Engineering Sciences and Innovation
Volume 7, Issue 1 / 2022, pp. 87-96

Technical Sciences
Academy of Romania
www.jesi.astr.ro

F. Electrical, Electronics Engineering, Computer Sciences and Engineering

Received ***10 August 2021*** *Accepted* ***7 March 2022***
Received in revised form ***2 December 2021***

State-space approach for symbolic analysis of an analog passive filter and its conversion to digital filter

ALEXANDRU LODIN, LACRIMIOARA GRAMA, CORNELIU RUSU*

Technical University of Cluj-Napoca, Cluj-Napoca, Romania

Abstract. The conversion of an analog filter described by a wiring diagram into another digital filter, having the same frequency characteristics was recently solved by a Python implementation. In this paper we will use this application for symbolic analysis of a passive filter having multiple components. Its conversion to digital filter is also described. Through experimental results we prove that the magnitude and phase characteristics are similar for the active analog filter and for the corresponding digital filter.

Keywords: filter, symbolic network analysis, state-space, Python.

1. Introduction

One of the most used techniques in electronics and in adjacent fields is the analysis of circuits in the frequency domain. However, this method is not the most suitable for the analysis of instability, for example in the case of the presence of parasitic components. Computer-assisted circuit analysis software such as PSpice or other applications in the same family, can compute the frequency response and graphically represent the gain and phase characteristics. A high and narrow peak in the gain characteristic may be omitted. This situation can occur for example because the discrete set of frequencies is not well chosen and does not capture all the information for an accurate analysis.
To solve this issue, it is recommended to find out the poles of the circuit and in this way, we can detect if oscillations can occur. Poles and zeros are the roots of the polynomials that make up the transfer function and can be calculated by matrix

*Correspondence address: Corneliu.Rusu@ieee.org

calculus. These methods are known to be part of the symbolic analysis of the network.

We have three types of symbolic network functions [1] and this classification considers whether all the network elements are represented by symbols, only some or none of them:

1) fully symbolic network functions;
2) partially symbolic network functions;
3) rational functions of *s* with numerical coefficients.

In this paper we will focus on the third type of symbolic analysis and calculate the transfer function using the state space approach. This kind of interest was developed in our team when the conversion of analog filters into digital filters was pursued, but later it proved that it can be used in the symbolic analysis of analog filters.

We shall exemplify by a case of a passive filter having multiple components. Given the increased number of components, obtaining the transfer function and pole-zero configuration prove to be extremely difficult tasks. Fig. 1 presents the multiple notch passive analog filter (small signal analysis). The nodal voltage of component R_8 was selected as output and the input is represented by the voltage source V_1. These results have not yet been published.

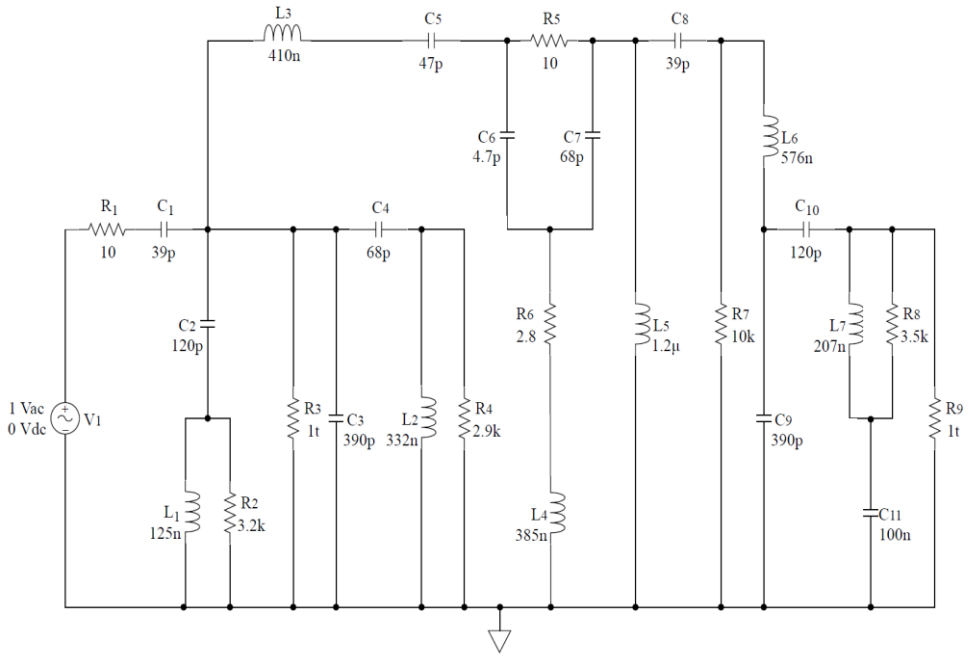

Fig. 1. The passive filter.

The paper is organized as follows: Section 2 presents a few implementation details of the proposed method, then Section 3 describes the analog filter and its characteristics. Also, the conversion of the analog filter to digital filter is discussed.

2. Implementation details

Our approach is as follows. We first compute the state-space of the analog filter, then a conversion from state-space to filter transfer description is realized. The derivation of equations and further details on the methods implementation can be found in [2] and [3], as well as in the references cited therein. The first implementation of the state-space approach was achieved using MATLAB, useful especially in case of passive filters. Also, a Python implementation of the state-space approach for both passive and active filters was developed and tested.

Although MATLAB has achieved satisfactory results in terms of the result of passive circuit conversion, with the consideration of multicomponent active filters, several disadvantages of this implementation have emerged. The difficulty arose due to the way the input data was structured. To obtain the network graph associated with a given active circuit, we need several processing steps. These require additional processing time, especially in the case of higher order filters. Inevitably some optimizations were a necessity and so came the implementation in Python.

One of the key aspects of the implementation consists in generating the graph associated to the network, for this purpose the data collected from the netlist description is stored as a graph object. A dedicated Python library was used, which can upload and store networks in standard and non-standard data formats, can generate various types of random and classic networks, thus facilitating the analysis of the network structure. The information in the netlist file is available as edges in the graph object for the current network. Such an edge has the following attributes [3]:

- Input node: first node in the network element netlist statement;
- Output node: second node from the netlist statement;
- Label: component unique alpha-numeric combination;
- Weight: network element weight, selected from a predefined look-up table;
- Data: dictionary type container that holds the nominal value, component type and in the case of a dependent source this container will store command node names.

The advantage of this data organization is that all the information in the netlist file will be collected and stored for further processing without the need of an additional data structure or other processing steps. After reading the netlist file, the graph associated with the network is available, and each component is assigned a weight so that the tree can be built immediately.

By reducing the number of steps involved in obtaining the network graph, the execution time has been significantly reduced. In the case of the multiple notch passive filter shown in Figure 1, the execution time of processing the netlist data has been reduced significantly from 0.2101 s in MATLAB, to 0.0044 s in Python.

The maximum amount of memory used during execution at a given time can give us an additional indication. For the implementation in Python, in the case of the

mentioned circuit, the memory is used more efficiently, from 316 kB to 12,288 kB, this being achieved by eliminating the need to create more data structures. The network graph of the multiple notch passive analog filter is shown in Fig. 2.

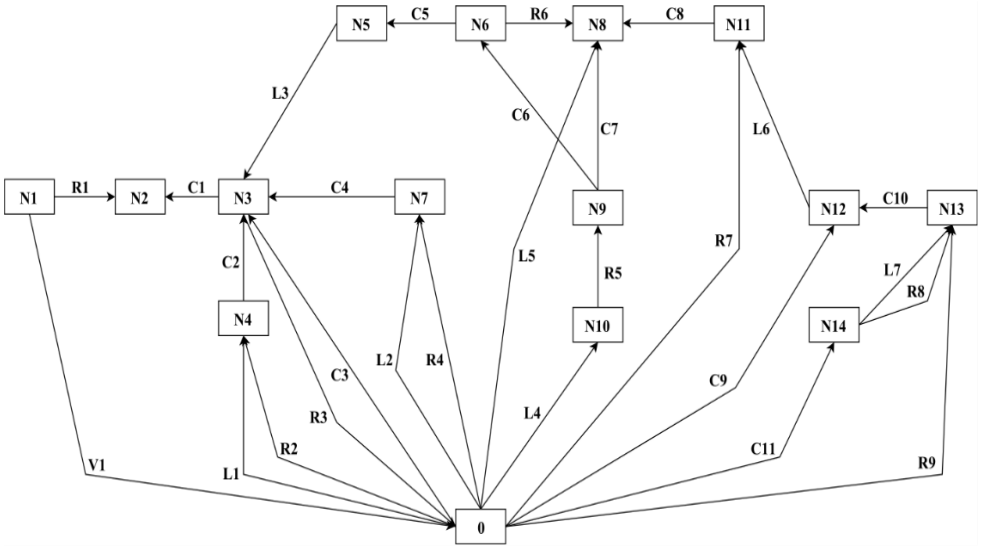

Fig. 2. The network of the passive filter.

In this implementation it is not necessary to remove excess elements to obtain the proper tree, as each network element is already assigned a weight depending on the type of component. For converting to an undirected graph and calculating the minimum spanning tree, the network graph object is used. As an example, the proper tree of the multiple notch passive analog filter is shown in Fig. 3. At this stage the tree and co-tree objects are properly defined, allowing the computation of the cut-set matrix $\mathbf{Q}_l$ to take place without any other preparation.

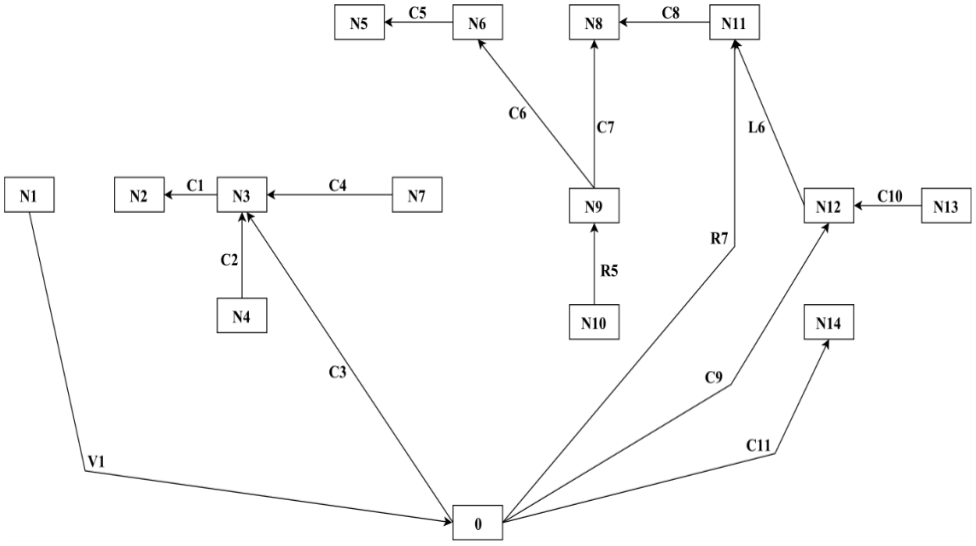

Fig. 3. The tree of the passive filter

For the generation of the **Q** matrix, the process does not undergo major changes compared to the standard methods used in the construction of state equations, methods that were also used for the implementation in MATLAB. The matrices containing the parameters are obtained as usual: before the state space matrices can be constructed, the relationship between the matrices describing the components and their connections must be represented.

After obtaining the state equations, no further implementation improvements are needed compared to the standard methods and thus the last blocks of the processing chain can be executed without any particular problems. In this way the transfer function of the analog filter is obtained. To have the equivalent digital filter, the bilinear transformation is usually applied, thus finalizing the second part of the implementation.

3. The analogue filter and its conversion to digital filter

Next after the state-space equations have been formulated, we evaluate the transfer function of the analog filter:

$$H_a(s) = \frac{\sum_{k=0}^{M} b_k s^k}{\sum_{k=0}^{N} a_k s^k} \tag{1}$$

The coefficients of transfer function of analog passive filter are presented in Table 1.

Table 1. The coefficients of the transfer function of the passive filter.

k	b_k	a_k
1	0	1
2	0	1.0164e+11
3	0	2.0116e+21
4	0	4.9766e+30
5	2.7839e+34	7.5753e+38
6	6.3369e+44	1.4354e+48
7	1.2557e+52	1.0764e+56
8	9.2973e+61	1.6320e+65
9	1.0542e+69	7.3865e+72
10	4.3835e+78	9.2402e+81
11	2.3859e+85	2.5332e+89
12	6.6812e+94	2.6699e+98
13	8.8587e+89	3.8630e+105
14	-2.2422e+99	3.4612e+114
15	1.0012+106	1.5713e+121
16	1.5749+112	1.1443e+130
17	2.5143+117	2.0844e+135
18	-2.2041e+116	9.1605e+131
19	-5.3244e+109	3.7168e+123

The pole-zero configuration of the passive filter has been computed and is shown in Table 2.

Table 2. The pole-zeros of the passive filter.

i	z_i	p_i
1	-2.2749e+10 + 0.0000e+00i	-7.6055e+10 + 0.0000e+00i
2	-1.3021e+06 + 2.5820e+08i	-2.2735e+10 + 0.0000e+00i
3	-1.3021e+06 - 2.5820e+08i	-2.8196e+09 + 0.0000e+00i
4	-2.5355e+06 + 2.1045e+08i	-2.9515e+06 + 3.0191e+08i
5	-2.5355e+06 - 2.1045e+08i	-2.9515e+06 - 3.0191e+08i
6	-2.8509e+06 + 1.8899e+08i	-2.8512e+06 + 2.3999e+08i
7	-2.8509e+06 - 1.8899e+08i	-2.8512e+06 - 2.3999e+08i
8	3.2088e+04 + 0.0000e+00i	-4.3325e+06 + 2.2314e+08i
9	8.7839e+03 + 3.0203e+04i	-4.3325e+06 - 2.2314e+08i
10	8.7839e+03 - 3.0203e+04i	-4.4002e+06 + 2.1441e+08i
11	-2.4828e+04 + 1.7210e+04i	-4.4002e+06 - 2.1441e+08i
12	-2.4828e+04 - 1.7210e+04i	-2.6076e+06 + 2.0084e+08i
13	-1.3971e-04 + 0.0000e+00i	-2.6076e+06 - 2.0084e+08i
14	1.2246e-10 + 0.0000e+00i	-6.0017e+05 + 6.9532e+07i
15	-	-6.0017e+05 - 6.9532e+07i
16	-	-1.8220e+05 + 0.0000e+00i
17	-	-4.3981e-04 + 0.0000e+00i
18	-	-1.3496e-08 + 0.0000e+00i

The magnitude and the phase of the frequency response function, corresponding to the analog filter, are shown in Fig. 4. One can see that the analog filter has several rejection frequencies.

The analog filter can be converted into an equivalent digital filter. The sampling frequency is selected 10 times greater than the upper limit of the analog frequency range taken into consideration for the evaluation. Since the maximum frequency of interest is 60 MHz, the sampling frequency will be selected F_s = 600 MHz. The digital filter can be designed in many ways, here we show the results of the bilinear transform.

The system function of the digital filter is of the form:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \tag{2}$$

The coefficients of transfer function of digital filter are presented in Table 3.

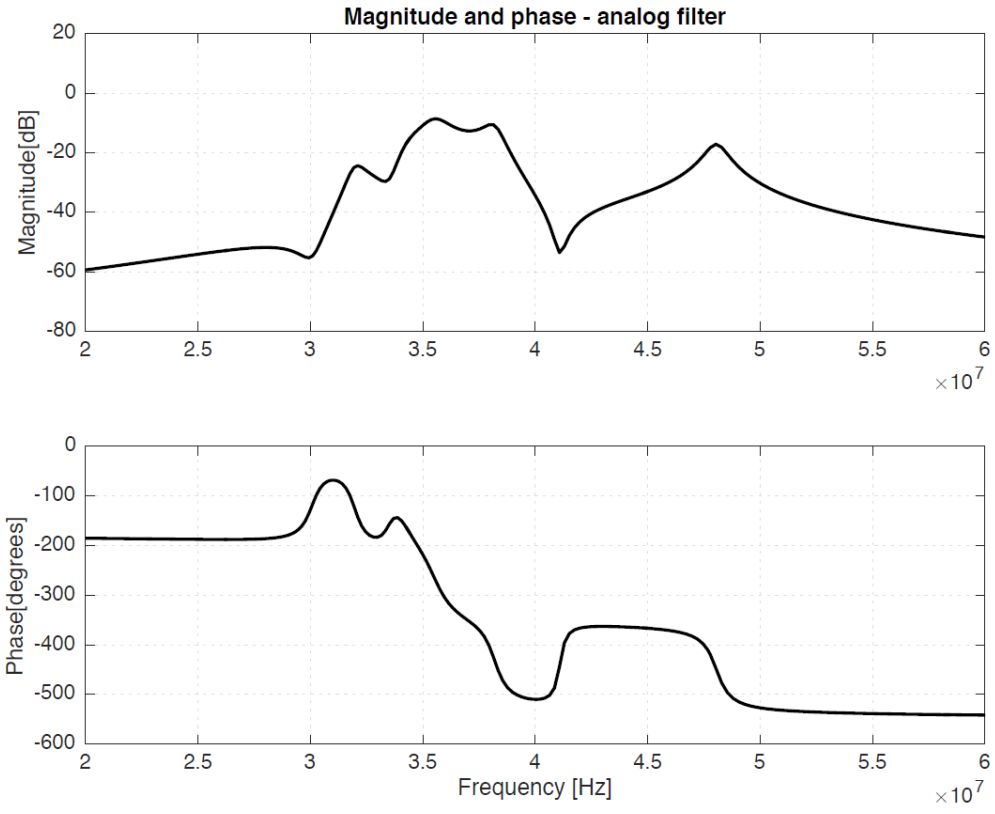

Fig. 4. Magnitude and phase of the passive filter.

Table 3. The coefficients of the digital filter.

k	b_k	a_k
1	1.9890e-04	1
2	-1.3120e-03	-9.5859e+00
3	3.0445e-03	4.2041e+01
4	-1.0054e-03	-1.0829e+02
5	-8.7245e-03	1.7019e+02
6	1.6388e-02	-1.3347e+02
7	-3.6451e-03	-5.7915e+01
8	-2.3878e-02	3.0151e+02
9	3.0547e-02	-3.8387e+02
10	-1.1463e-03	2.0793e+02
11	-2.9124e-02	7.7192e+01
12	2.4277e-02	-2.3585e+02
13	2.0877e-03	1.9335e+02
14	-1.5136e-02	-6.3942e+01
15	8.4986e-03	-2.1376e+01
16	6.2477e-04	3.4855e+01
17	-2.7013e-03	-1.7980e+01
18	1.1883e-03	4.7602e+00
19	-1.8134e-04	-5.4639e-01

The poles and zeros of the digital filter have been computed and they are shown in Table 4.

Table 4. The pole-zeros of the digital filter.

k	z_k	p_k
1	-1.0033e+00 + 0.0000e+00i	-9.4858e-01 + 0.0000e+00i
2	-1.0000e+00 + 3.3527e-03i	-9.8435e-01 + 0.0000e+00i
3	-1.0000e+00 - 3.3527e-03i	-6.4908e-01 + 0.0000e+00i
4	-9.9659e-01 + 0.0000e+00i	5.9125e-01 + 7.9677e-01i
5	-9.4861e-01 + 0.0000e+00i	5.9125e-01 - 7.9677e-01i
6	6.8499e-01 + 7.2353e-01i	7.1824e-01 + 6.8401e-01i
7	6.8499e-01 - 7.2353e-01i	7.1824e-01 - 6.8401e-01i
8	7.7505e-01 + 6.1997e-01i	7.4742e-01 + 6.4521e-01i
9	7.7505e-01 - 6.1997e-01i	7.4742e-01 - 6.4521e-01i
10	8.1242e-01 + 5.6818e-01i	7.9226e-01 + 5.9734e-01i
11	8.1242e-01 - 5.6818e-01i	7.9226e-01 - 5.9734e-01i
12	1.0362e+00 + 0.0000e+00i	7.6351e-01 + 6.2559e-01i
13	1.0226e+00 + 2.8334e-02i	7.6351e-01 - 6.2559e-01i
14	1.0226e+00 - 2.8334e-02i	9.7158e-01 + 2.2825e-01i
15	9.9182e-01 + 3.5329e-02i	9.7158e-01 - 2.2825e-01i
16	9.9182e-01 - 3.5329e-02i	1.0000e+00 + 2.1214e-04i
17	9.6745e-01 + 1.5571e-02i	1.0000e+00 - 2.1214e-04i
18	9.6745e-01 - 1.5571e-02i	9.9932e-01 + 0.0000e+00i

The magnitude and the phase of the frequency response function, corresponding to the digital, are plotted in Fig. 5. Comparing Figs. 4 and 5, we can state that the frequency response characteristics of the digital filter generated using the bilinear transformation are almost the same as those of the multiple notch passive analog filter.

4. Conclusion

In this paper we discussed the implementation of a method for calculating the transfer function of a passive, high-order analog filter with frequent rejection multiples. The analog filter was described only by its small signal circuit diagram. Subsequently, the analog filter was converted to the corresponding digital filter. Experimental results have also been presented showing that the magnitude and phase characteristics are similar for the active analog filter and for the corresponding digital filter.

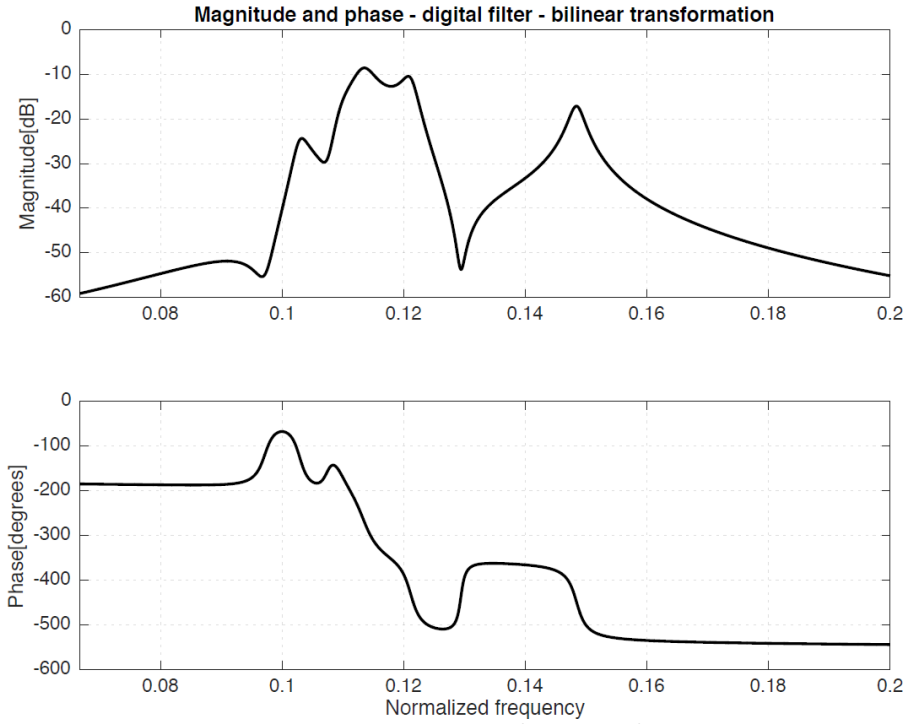

Fig. 5. Magnitude and phase of the digital filter

References

[1] Chua L. O. and Lin P.-M., *Computer-aided analysis of electronic circuits*, Englewood Cliffs, NJ, Prentice-Hall, 1975.

[2] Lodin A., Grama L., Rusu C., *Symbolic analysis of an analog active filter as path for conversion to digital filter*, Carpathian Journal of Electronic and Computer Engineering, **11**, 2, 2018, p. 8–12.
[3] Lodin A., Grama L., Rusu C., *Python implementation of the state-space method to convert analog filters described by a netlist to digital filters*, in Proceedings The 6th International Symposium on Electrical and Electronics Engineering (ISEEE), 2019, paper 47.