

Symbolic Analysis of an Analog Active Filter as Path for Conversion to Digital Filter

Alexandru Lodin

Signal Processing Group, Basis of
Electronics Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
alexandru.lodin@gmail.com

Lacrimioara Grama

Signal Processing Group, Basis of
Electronics Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Lacrimioara.Grama@bel.utcluj.ro

Corneliu Rusu

Signal Processing Group, Basis of
Electronics Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Corneliu.Rusu@bel.utcluj.ro

Abstract— Recently have been reported methods to deliver a digital filter from an analog active filter, described only by its circuits diagram. The proposed approaches have been implemented in MATLAB and Python, and they were based on state-space conversion from analog to digital domain. Based on the Python approach, we show in this paper how to compute the transfer function of a large order analog active filter. The analog filter is described only by its circuits diagram. Finally, the analog filter is converted to the corresponding digital filter, having similar frequency gain and phase characteristics.

Keywords— active filter, symbolic network analysis, state-space, Python

I. INTRODUCTION

The last decades have witnessed a high demand for a digital implementation for almost any analog device. Sometimes only the circuit diagram of the analog implementation is available. This is the case addressed by the present work.

We show in this paper how to compute the transfer function of a large order analog active filter. The analog filter is described only by its circuits diagram. Finally, the analog filter can be converted to the corresponding digital filter, having similar frequency gain and phase characteristics.

The paper is organized in the following manner. Section II recalls the symbolic analysis of analog circuits and conversion of analog filters described by their diagrams to digital filters. The method for computing network function, based on the State-Space approach is discussed in Section III. The Python version of State-Space approach is presented in Section IV.

II. RELATED WORK

A. Symbolic analysis of analog circuits

Frequency domain analysis is well established, but it is not well suited to instability detection, particularly when parasitic components are present [1]. One can compute and draw the response as a function of frequency using PSpice or other computer-aided circuits analysis software. Unfortunately, it may be possible that a high and narrow peak is missed. This might be the result that the set of discrete points does not pick all the information where the analysis is performed. To avoid, it is advisable to calculate

the poles and find whether oscillations can occur. The poles and the zeros can be found as roots of polynomials or computed directly from system matrices. Suitable methods are part of symbolic network analysis.

Depending on whether all, some of, or none of networks elements are represented by symbols, we have three types of symbolic network functions [2]:

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

In this work we present a method to generate type 3 symbolic network functions based on formulation of state-space equations. This software has been developed when we studied the conversion of analog filters (described by their diagrams) to digital filters.

B. Conversion of analog filters described by their diagrams to digital filters

In order to achieve the analog to digital conversion of a filter described only by its diagram, the *Analog Filter Netlist to Digital Filter Statements* (AFN-DFS) has been proposed in [3]. Although the implementation of this method is straightforward, it requires a small step-size for integration, thus a high sampling frequency is needed.

To overcome this drawback, the state-space based approach for converting an active analog filter described by a circuit diagram or a netlist to a digital filter has been proposed [4]. We first compute the state-space of the active analog filter, then a conversion from analog domain to digital domain is used. The implementation of the state-space approach was realized using MATLAB [5] and it has been tested extensively for passive filters.

The MATLAB implementation has encountered several difficulties in an attempt to convert large active circuits, especially those related to computational complexity. To overcome, the Python implementation has been developed [6]. Compared to the previous version, we have reduced to half the execution time and we have improved memory management. Also we have taken advantage of a couple of signal processing toolkits in order to complete the final stages of the algorithm.

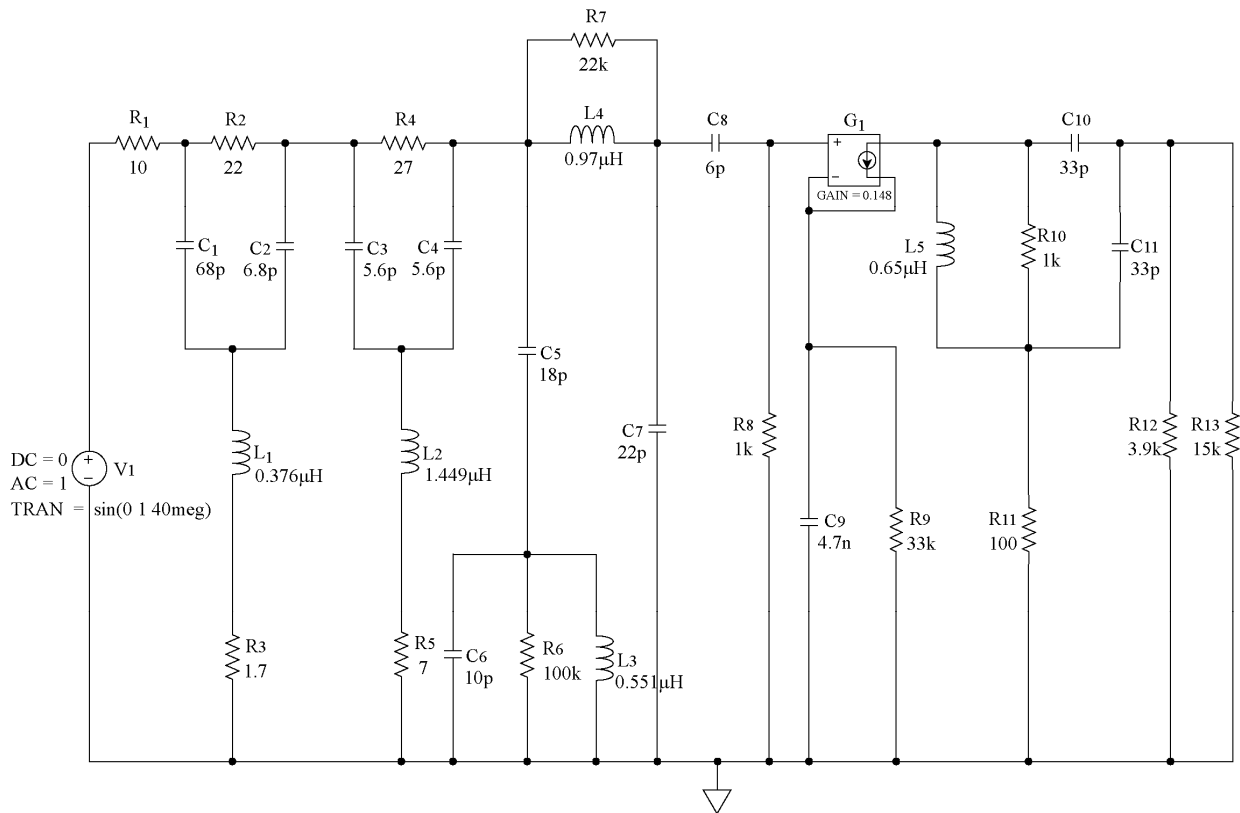


Fig. 1. Analog test active filter.

III. METHOD FOR COMPUTING NETWORK FUNCTION – THE STATE-SPACE APPROACH

To generate type 3 symbolic network functions we shall use the state-equation formulation for active filters. This is actually the same procedure as for networks containing dependent sources [4]. In the case of a network containing dependent sources, the tree current vector can be partitioned into four vectors corresponding to tree capacitance, tree resistance, voltage source, and dependent voltage source. Also the link current vector can be partitioned into inductance, link resistance, current source and dependent current source. The cut-set matrix \mathbf{Q} is modified accordingly.

The state-space procedure is performed as follows:

- 1) The link branch voltages are expressed in terms of tree branch voltages;
- 2) Capacitance currents and inductance voltages are expressed in terms of tree-branch voltages and link currents;
- 3) The resistive tree-branch currents and resistive link voltages are expressed in terms of tree-branch voltages and link currents;
- 4) The state equations are obtained after the vectors relating the resistive quantities and dependent sources are eliminated.

Details of the derivation of the equations can be found in [2], [7] and a complete list of computations is available in [8].

IV. THE PYTHON VERSION OF STATE-SPACE APPROACH

The Python version was designed as an object-oriented application. Starting with the first step of the method (which is collecting data from the netlist description), every component is stored as data of a single object which represents actually the network graph. In this way we do not need arrays to store the necessary network information and also no further arrangement is required.

The network graph is constructed as soon as the data is available from the file. For the resulting graph object, we have available all necessary graph operations by default. Thus, detecting the proper tree of the network was a fast and easy step to implement.

Matrix handling from this point on was achieved by creating routines to improve computing speed and memory allocation. Although all interpreted programming languages have performance related issue, we have used several modules. These are designed specifically to handle mathematical operations very fast.

V. EXPERIMENTAL RESULTS

A. Framework

The filter presented in Fig. 1 was used as a test circuit in order to verify the functionality of the new implementation.

The test analog filter is a large order analog active notch filter, previously implemented in television receivers [9] using a bipolar junction transistor. It consists of several passive components (resistances, capacitances and inductances) and it does not contain any capacitances and

voltage sources loops or inductances and current sources cut-sets. The analog filter also contains an active element such as voltage controlled current source to model the bipolar junction transistor.

In order to validate the results of the proposed method, PSpice simulations of the analog filter are illustrated in Fig. 2. They are magnitude and phase of the frequency response for the analog filter.

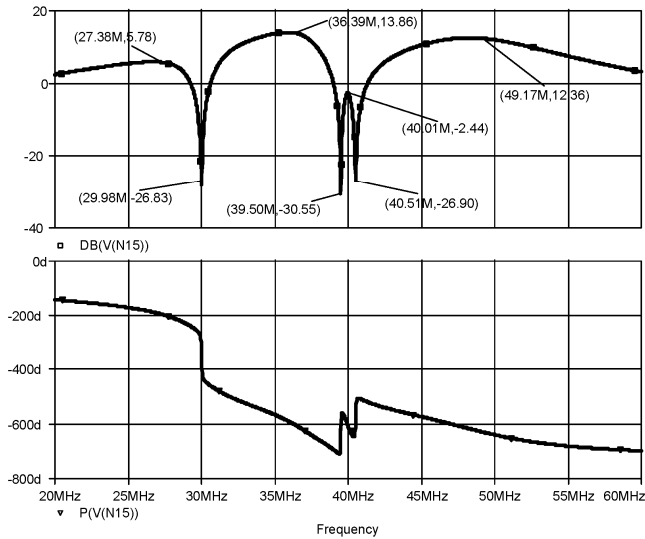


Fig. 2. Magnitude and phase of the frequency response for the analog filter, obtained using PSpice simulations.

B. Implementation

The network graph can be constructed based on the generated netlist file as shown in Fig. 3. The order in which the nodes are listed was used for the branches directions. The proper tree depicted in Fig. 4 was generated as the minimal spanning tree of the network graph. As one can see, the procedures are similar to those presented in [4], [5].

The obtained cut-set matrix results immediately. The partitioned matrix Q_l is then obtained by extracting the parts that are related to the link-branches. Both matrices are extremely large and they cannot be shown here.

Next the state-space equations can be formulated based on all variables involved determined in the previous steps. Finally the state-space equations are used in order to evaluate the transfer function of the analog filter:

$$H_a(s) = \sum_{k=0}^M B_k s^k / \sum_{k=0}^N A_k s^k \quad (1)$$

The coefficients of the analog transfer function, when the input is the input voltage source and the output is the voltage across R_{13} , are shown in Table I.

TABLE I. ANALOG FILTER COEFFICIENTS

k	B_k	A_k
16	0	1
15	0	$8.83316217 \cdot 10^{10}$
14	$-1.048576 \cdot 10^6$	$9.23185357 \cdot 10^{20}$
13	$-1.39630012 \cdot 10^{25}$	$1.85902155 \cdot 10^{30}$
12	$-6.04173528 \cdot 10^{35}$	$1.00092634 \cdot 10^{39}$
11	$-7.88177359 \cdot 10^{45}$	$6.52054237 \cdot 10^{47}$
10	$-3.09233966 \cdot 10^{55}$	$2.28597190 \cdot 10^{56}$
9	$-1.24110154 \cdot 10^{63}$	$8.62007622 \cdot 10^{64}$
8	$-4.95059499 \cdot 10^{72}$	$2.19423054 \cdot 10^{73}$
7	$-6.04607811 \cdot 10^{79}$	$5.40839948 \cdot 10^{81}$
6	$-2.54084396 \cdot 10^{89}$	$1.01635370 \cdot 10^{90}$
5	$-7.60237098 \cdot 10^{95}$	$1.62276170 \cdot 10^{98}$
4	$-3.97882375 \cdot 10^{105}$	$2.19567534 \cdot 10^{106}$
3	$5.93766476 \cdot 10^{111}$	$1.89870168 \cdot 10^{114}$
2	$6.58816874 \cdot 10^{120}$	$1.70415545 \cdot 10^{122}$
1	$4.24766622 \cdot 10^{124}$	$1.26140269 \cdot 10^{129}$
0	$3.73135107 \cdot 10^{122}$	$2.34099982 \cdot 10^{135}$

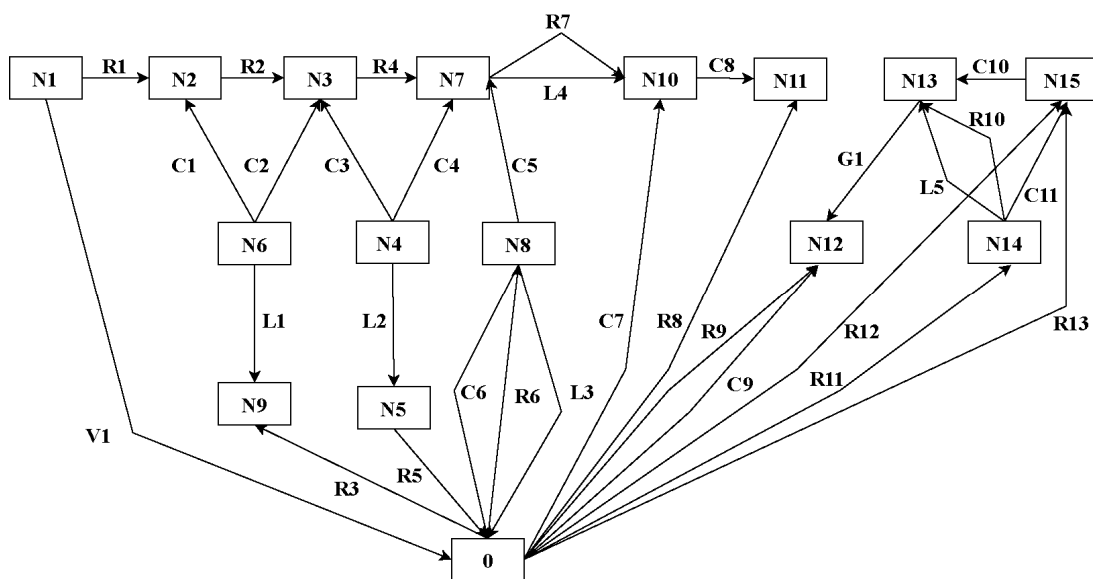


Fig. 3. Network graph representation of the filter shown in Fig. 1.

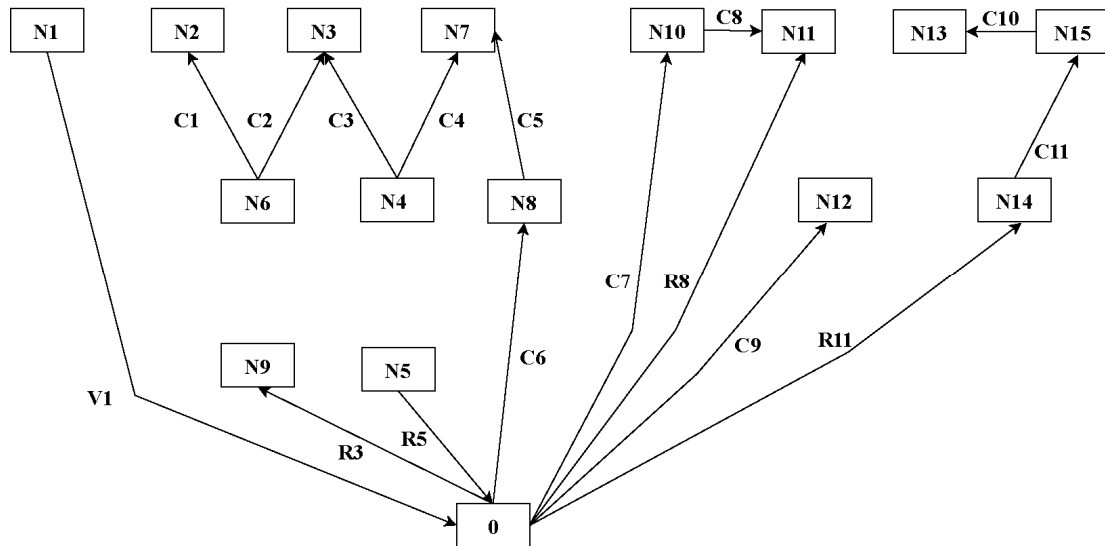


Fig. 4. The proper tree of the filter shown in Fig. 1.

C. Comparison of the experimental results

To compare PSpice results with Python approach simulations results first the transfer function of the analog filter using the state-space description was computed. Then, as illustrated in Fig. 5, the magnitude and the phase of the frequency response function, corresponding to the analog filter, was plotted. One can see that the analog filter has several rejection frequencies.

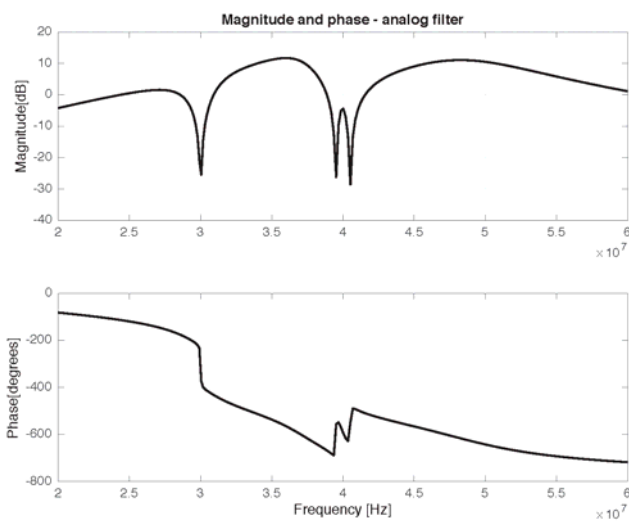


Fig. 5. Magnitude and phase of the frequency response for the analog filter (Python approach).

The rejection frequencies of the analog filter and the corresponding attenuation for PSpice and Python simulations are shown in Tables II and III.

It can be easily observed that the results are quite similar.

The small differences can be justified by the fact that PSpice models of real passive and active components are not ideal. On the other hand, we have assumed ideal components in the derivation of the method.

TABLE II. PSpice APPROACH

Rejection frequency [MHz]	Attenuation [dB]
$F_{r1} = 27.38$	$A_1 = 5.78$
$F_{r2} = 29.98$	$A_2 = -26.83$
$F_{r3} = 36.39$	$A_3 = 13.86$
$F_{r4} = 39.50$	$A_4 = -30.55$
$F_{r5} = 40.01$	$A_5 = -2.44$
$F_{r6} = 40.51$	$A_6 = -26.90$
$F_{r7} = 49.17$	$A_7 = 12.36$

TABLE III. PYTHON APPROACH

Rejection frequency [MHz]	Attenuation [dB]
$F_{r1} = 27.38$	$A_1 = 1.54$
$F_{r2} = 29.98$	$A_2 = -25.43$
$F_{r3} = 36.39$	$A_3 = 11.60$
$F_{r4} = 39.50$	$A_4 = -26.19$
$F_{r5} = 40.01$	$A_5 = -4.40$
$F_{r6} = 40.51$	$A_6 = -28.54$
$F_{r7} = 49.17$	$A_7 = 19.94$

D. Experimental results for conversion to digital filter

In this experiment the analog filter is transformed 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. This is the usual selection in the simulation of a sampling process.

The PSpice output file states that the maximum frequency of interest is 60 MHz, thus for our case the sampling frequency will be $F_s = 600$ MHz.

Obtaining the digital filter can be achieved in several manners, but we present here the results of one of the most popular methods: the bilinear transform. Illustrated in Fig. 6 are the frequency response characteristics of the digital filter, converted using the bilinear transformation. The digital filter coefficients are presented in Table IV.

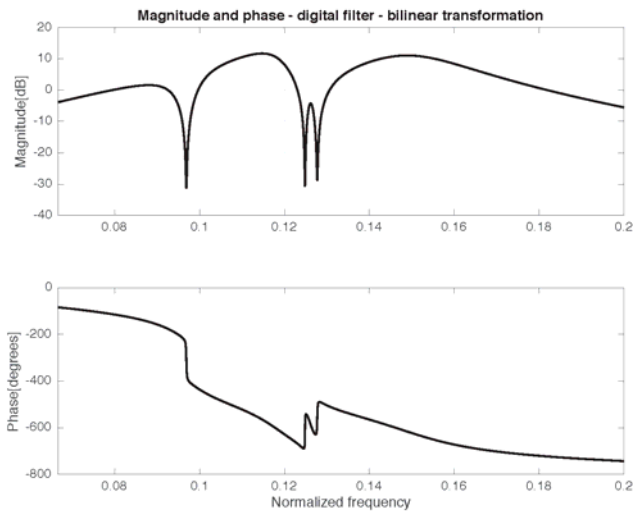


Fig. 6. Magnitude and phase of the frequency response of the digital filter, obtained using bilinear transform (Python approach).

TABLE IV. DIGITAL FILTER COEFFICIENTS

k	b_k	a_k
0	$-3.09 \cdot 10^{-2}$	1.00
1	$8.51 \cdot 10^{-2}$	-6.79
2	$3.96 \cdot 10^{-2}$	$2.07 \cdot 10^1$
3	$-3.70 \cdot 10^{-1}$	$-3.47 \cdot 10^1$
4	$2.78 \cdot 10^{-1}$	$2.73 \cdot 10^1$
5	$5.38 \cdot 10^{-1}$	$1.13 \cdot 10^1$
6	$-9.07 \cdot 10^{-1}$	$-5.55 \cdot 10^1$
7	$-1.00 \cdot 10^{-1}$	$6.35 \cdot 10^1$
8	1.10	$-2.72 \cdot 10^1$
9	$-5.49 \cdot 10^{-1}$	$-1.70 \cdot 10^1$
10	$-5.88 \cdot 10^{-1}$	$3.30 \cdot 10^1$
11	$6.36 \cdot 10^{-1}$	$-2.10 \cdot 10^1$
12	$4.56 \cdot 10^{-2}$	4.20
13	$-2.88 \cdot 10^{-1}$	3.02
14	$7.78 \cdot 10^{-2}$	-2.61
15	$4.86 \cdot 10^{-2}$	$8.49 \cdot 10^{-1}$
16	$-2.27 \cdot 10^{-2}$	$-1.10 \cdot 10^{-1}$

In conclusion of this last experiment, we can state that the frequency response characteristics of the digital filter generated using the bilinear transform are almost the same as for the analog filter described in the given netlist.

VI. CONCLUSION

In this paper we have shown how to compute the transfer function of a large order analog active filter. The analog filter was described only by its circuit diagram. Finally, the analog filter has been converted to the corresponding digital filter. Experimental results were also presented, and they show that the results are similar for the analog active filter (Pspice and Python simulations) and for the corresponding digital filter.

ACKNOWLEDGEMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI-UEFISCDI, project number PNIII-P2-2.1-PED-2016-1608, 222PED/2017, within PNCDI III.

REFERENCES

- [1] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 1993.
- [2] L. D. Chua and P.-M. Lin, *Computer-Aided Analysis of Electronic Circuits*. Englewood Cliffs, NJ: Prentice Hall, 1975.
- [3] C. Rusu, L. Grama, and J. Takala, "SPICE simulation of analog filters: A method for designing digital filters," in *Proceedings EUROCAST 2009*, Gran Canaria, Feb. 2009, pp. 179–180.
- [4] L. Grama, A. Lodin, C. Rusu, and J. Takala, "Method to convert analog filters to digital filters," in *Proceedings ISPA 2013*, Sep. 2013, pp. 451–456.
- [5] A. Lodin, L. Grama, C. Rusu, and J. Takala, "Systematic method to convert of analog filters to digital filters," in *Proceedings of 2013 IEEE Workshop on Signal Processing Systems*, 2013, pp. 189–194.
- [6] A. Lodin, L. Grama, and C. Rusu, "From bulky analog active filters to digital filters," in *Proceedings of the International Symposium on Electronics and Telecommunications 2018 - ISETC 2018*, 2018, pp. 1–4.
- [7] N. Balabanian and T. A. Bickart, *Electrical Network Theory*. New York: John Willey & Sons, 1969.
- [8] L. Grama, A. Lodin, and C. Rusu, "About converting active analog filters to digital filters," *Acta Technica Napocensis - Electronics and Telecommunications*, vol. 57, no. 4, pp. 36–39, 2016.
- [9] M. Silisteanu, L. Cipere, and C. Constantinescu, *Lucrari practice de depanare a receptoarelor de televiziune*. Bucuresti: Editura Didactica si Pedagogica, Bucuresti, 1982.