MINISTERUL EDUCAȚIEI NAȚIONALE
UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTY OF ELECTRONICS, TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Ing. Alexandru-Cristian Lodin

PhD Thesis

Summary

METHODS TO CONVERT ANALOG FILTERS DESCRIBED BY A NETLIST TO DIGITAL FILTERS

Scientific Advisor,

Prof.dr.ing Corneliu RUSU

1 Introduction

The last decades have witnessed an unprecedented development of the computer systems that has marked an exponential growth of the computational speed and storage space. Aside digital systems, the signal processing technologies have undergone a similar evolution, clearly noticeable in all areas: signal reconstruction, adaptive filtering, transforms, sensors, processing of signals obtained from sensors, medical devices, sensor arrays, image and optical processing, security and protection, intruder detection, forensics, optical signal processing, computer aided analysis and synthesis of circuits or devices, algorithms implementation, numerical methods, bio-electronics, compressive sampling, wireless networks, acoustic sensors, audio and speech processing.

The advances in CMOS technology has paved the way for digital implementations and product realized in analog electronics are now implemented with digital technology. Thus there is interest to design a digital implementation for any analog device. Sometimes the design process is time consuming and expensive, as engineers must know and consider both the analog and digital domain. Besides, for an accurate design, the frequency specifications must be provided, which is not always the case. It may happen that only the circuit diagram of analog implementation is available. This is the case addressed by the present work.

Starting from the idea that an analog circuit design can be converted to a digital filter, the thesis presents two different methods to achieve this goal: Analog Filter Netlist to Digital Filter Statements Approach and The State-Space Based Approach.

2 Purpose of the Thesis

In the case of IIR (Infinite Impulse Response) digital filters, the most often used design method is to convert the digital filter specifications into analog low-pass prototype filter specifications, to determine the analog low-pass filter transfer function $H_a(s)$ meeting these specifications and then to transform it into the desired digital filter transfer function $H(z)$. This approach has been widely used for several reasons as follows:

- analog approximation techniques are highly advanced;
- they usually yield closed-form solutions;
- extensive tables are available for analog filter design;
- many applications require the digital simulation of analog filters.

The basic idea behind the conversion of an analog prototype transfer function $H_a(s)$ into a digital IIR transfer function $H(z)$ is to apply a mapping from the s-domain to the z-domain such that the essential properties of the analog frequency response are preserved. This implies that the mapping function should be such that the imaginary $j\Omega$ axis in the s-plane be mapped onto the unit circle in z-plane, thus there will be a direct relationship between the two frequency variables in the two domains, and a stable analog transfer function will be transformed into a stable digital filter. To this end, the most widely used methods are forward approximation, backward approximation, and bilinear transform (trapezoidal approximation). Other popular approaches are matched z-transform, impulse invariance and step invariance.

All these methods use a mathematical description of analog filters, either in time or frequency domain, using concepts like transfer function, frequency response, impulse response, step response, etc. However, for many engineers analog filters mean certain circuits or a netlist of components, their connections, and maths are seldom used. Although, for digital filters hardware implementation is one of main choices, a set of statements in software must be written in designing or implementation phase of any digital filter.

Nevertheless, the digital description of analog filters has been used for many years. Starting few decades ago, analog filters have been simulated using digital computers, which proceeded on discrete implementations. These discrete implementations have been used to obtain a digital filter. Indeed, for instance in the SPICE's transmission-line component are used to simulate the ideal z^{-1} delay elements of digital filters like the integrator blocks are used to simulate the $1/s$ terms for an analog filter.

The direct design of a digital filter to meet pre-specified characteristics is a well-known subject. However, finding frequency response characteristics parameters of an analog filter, for enabling the digital filter design, is usually a very challenging task, especially when only the diagram is available. Strong knowledge of the subject and refined techniques are needed to extract the pre-specified characteristics from the diagram of an analog filter.

The goal of this work is to design an automated conversion of an analog filter described by a diagram to a digital filter. We mention that the proposed technique which converts the analog filter to the digital filter should be available as a software code (e.g. MATLAB or Python) for a general purpose computer and the software code should be easy to manipulate. We can also consider the following:

- for such a procedure, human interaction with the software has to be accepted, but it should be reduced as much as possible;

- any analog circuit may be described using a diagram where the components are drawn or a set of connected icons displayed on the screen.

In both cases the corresponding netlist of components may be provided.

In the first part of the thesis an approach of converting an analog filter described by a diagram or a netlist to a digital filter has been proposed. With the Analog Filter Netlist to Digital Filter Statements approach, a software code for the digital filter, corresponding to a previous implementation in analog domain, may be available rapidly. However, this approach has some drawbacks, the main one being the fact that it cannot provide a systematic method to deliver all the required description equations. Also, it may happen that this code is not optimal; later on, the code might be optimized for specific target processors.

In the second part of the thesis, we present a state-space based approach that provides an automatic procedure to convert an analog filter described by a diagram or a netlist to a digital filter described in system function form. The main three parts of the procedure are the following:

- the state-space of the analog filter is computed based on the analog circuit netlist;
- conversion from analog domain to digital domain is used;
- the digital filter is delivered in system function form.

After the presentation of theoretical aspects regarding the state-space description of systems, and a short description of two analog to digital conversion methods, all three steps involved in the conversion process are described in detail. With a couple of examples the validity of the method is verified.

To implement the state-space based method, two different approaches are discussed. The first one delivers a MATALAB software that is capable to perform the conversion for analog passive and active filters. Implementation details for each part of the procedure were presented pointing out the performance trough experimental results.

The second approach solves the conversion problem for large active filters and it was developed in Python high-level programming language. The Python approach was developed in order to overcome several limitations of the first version, to improve the program execution time and memory management. The most important optimizations of the proposed approach were described in detail. Experimental results for two large active filters were also presented.

3 The Analog Filter Netlist to Digital Filter Statements Approach

The The Analog Filter Netlist to Digital Filter Statements Approach (AFN-DFS) approach of converting an analog filter described by a netlist to a digital filter consists of the following steps:

1. every component of the netlist is replaced by its corresponding companion model, like in any software application used for simulation;

2. for the corresponding companion model, we write the equations that describe in the time domain the behavior of every component of the circuit; since the description of the behaviour of an analog component is done using continuous-time equations, all these equations must be discretized to obtain a discrete-time system of equations; this system of discrete-time equations describes the corresponding digital filter;

3. we can finally provide a software code which is nothing else than a transcription in software of the system of equations; this gives the software implementation of the digital filter, and this digital filter corresponds to the initial analog filter.

Consequently, using these remarks, one can use a previous implementation in analog domain and a software code for the digital filter may be available. One can recognize that the knowledge of state-space is very useful when implementing this approach. Moreover, this suggests that the state-space approach can lead certainly to an implementation by giving an adequate strategy of deriving the equations. Furthermore, having a state-space description of either analog or digital filter, one can obtain any canonical structure of digital filter. Furthermore, having a state-space description of either analog or digital filter, one can obtain any canonical structure of digital filter.

It should be noticed that we can apply other discretization of the branch equations for capacitors and inductors, e.g.

- Backward Euler algorithm: $\mathbf{x}_{n+1} = \mathbf{x}_n + hf(\mathbf{x}_{n+1}, t_{n+1})$;

- Trapezoidal algorithm: $\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}[f(\mathbf{x}_n, t_n) + f(\mathbf{x}_{n+1}, t_{n+1})]$;
- Runge-Kutta algorithms, multistep algorithms, Gear algorithms.

The main issue when implementing such algorithms is that the equations contain more variables for time index $n+1$, i.e. these equations are self-consistent in variable-index $n+1$. To solve the corresponding system of equations we cannot apply a one-step updating rule such as for forward Euler formula as presented above; we need to elaborate a rather complicated approach. But the AFN-DFS approach has been developed having in mind a minor interaction with the user.

It should be added that almost all formulas used to obtain a discretize form of the derivatives or of the higher order derivatives are developed by to be correct for a certain class of polynomials. Unfortunately such formulas are not appropriate for the linear time invariant systems, where the eigen functions are the complex exponential.

4 The State-Space Based Approach

Given the state-space canonical representation of an analog filter:

$$\left\{ \begin{array}{rcl} \dot{\mathbf{x}} & = & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} & = & \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{array} \right.$$

we can use it to evaluate the transfer function of the analog filter:

$$H(s) \quad = \quad \mathbf{C}\left(s\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D}.$$

From the transfer function of analog filter we can easily get the transfer function of digital filter, e.g. by bilinear transform or impulse invariance method; few slightly different variants are also available. Thus the state-space based approach for converting an analog filter described by a circuit diagram or a netlist to a digital filter described in system function form consists of three main parts:

I. First the state-space of the analog filter is computed based on diagram or netlist of the analog circuit;

II. Then a conversion from analog domain to digital domain is used;

III. Finally the digital filter is delivered in system function form.

In the following details regarding the main parts of the proposed approach will be described.

First based on the netlist data the network graph is constructed and its branches then numbered sequentially starting with the tree capacitances, followed by the tree resistances, voltage sources, link inductances, link resistances, and current sources. Next from the graph we can construct the cut-set matrix **Q**:

$$\mathbf{Q} = \begin{array}{c|ccc:ccc|} & C_t & R_t & E & L_l & R_l & J \\ C_t & & & & \mathbf{Q}_{lCL} & \mathbf{Q}_{lCR} & \mathbf{Q}_{lCJ} \\ R_t & & \mathbf{I} & & \mathbf{Q}_{lRL} & \mathbf{Q}_{lRR} & \mathbf{Q}_{lRJ} \\ E & & & & \mathbf{Q}_{lEL} & \mathbf{Q}_{lER} & \mathbf{Q}_{lEJ} \end{array}$$

The portion of the cut-set matrix corresponding to the link branches is designated $\mathbf{Q}_{l--}$. The elements of the cut-set matrix $\mathbf{Q}_l$ are obtained as follows:

$$q_{jk} = \begin{cases} 1, & \text{if branch } k \text{ belongs to cut-set } j \text{ and has the same reference direction;} \\ -1, & \text{if branch } k \text{ belongs to cut-set } j \text{ and has the opposite reference direction;} \\ 0, & \text{if branch } k \text{ does not belong to cut-set } j; \end{cases}$$

The procedure for obtaining the state equations is as follows:

1. Construct a proper tree for the network and number the branches;

2. Construct the topological matrix $\mathbf{Q}_l$ and partition it to provide the necessary set of submatrices;

$$\mathbf{Q}_l = \begin{bmatrix} \mathbf{Q}_{lCL} & \mathbf{Q}_{lCR} & \mathbf{Q}_{lCJ} \\ \mathbf{Q}_{lRL} & \mathbf{Q}_{lRR} & \mathbf{Q}_{lRJ} \\ \mathbf{Q}_{lEL} & \mathbf{Q}_{lER} & \mathbf{Q}_{lEJ} \end{bmatrix}$$

3. Construct the matrices $\mathbf{R}_l$ and $\mathbf{G}_t$ and calculate the resistive tree voltages and link currents ($\mathbf{v}_{tR}$ and $\mathbf{i}_{lR}$). The resistive tree voltages $\mathbf{v}_{tR}$ and link currents $\mathbf{i}_{lR}$ are evaluated as:

$$\begin{bmatrix} \mathbf{v}_{tR} \\ \mathbf{i}_{lR} \end{bmatrix} = \mathcal{H} \begin{bmatrix} \mathbf{v}_{tC} \\ \mathbf{i}_{lL} \end{bmatrix} + \widehat{\mathcal{H}} \begin{bmatrix} \mathbf{v}_{tE} \\ \mathbf{i}_{lJ} \end{bmatrix}$$

where:

$$\mathcal{H} = \begin{bmatrix} -\mathbf{Q}^t_{lRR} & \mathbf{R}_l \\ \mathbf{G}_t & \mathbf{Q}_{lRR} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{Q}^t_{lCR} & 0 \\ 0 & \mathbf{Q}_{lRL} \end{bmatrix}$$

$$\widehat{\mathcal{H}} = \begin{bmatrix} -\mathbf{Q}^t_{lRR} & \mathbf{R}_l \\ \mathbf{G}_t & \mathbf{Q}_{lRR} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{Q}^t_{lER} & 0 \\ 0 & -\mathbf{Q}_{lRJ} \end{bmatrix}.$$

4) Formulate the equations for the capacitive tree currents and inductive link voltages ($\mathbf{i}_{tC}$ and $\mathbf{v}_{lL}$) as:

$$\begin{bmatrix} \mathbf{i}_{tC} \\ \mathbf{v}_{lL} \end{bmatrix} = \mathcal{G} \begin{bmatrix} \mathbf{v}_{tC} \\ \mathbf{i}_{lL} \end{bmatrix} + \widehat{\mathcal{G}} \begin{bmatrix} \mathbf{v}_{tE} \\ \mathbf{i}_{lJ} \end{bmatrix}$$

where:

$$\mathcal{G} = \begin{bmatrix} 0 & -\mathbf{Q}_{lCL} \\ \mathbf{Q}^t_{lCL} & 0 \end{bmatrix} + \begin{bmatrix} 0 & -\mathbf{Q}_{lCR} \\ \mathbf{Q}^t_{lRL} & 0 \end{bmatrix} \mathcal{H}$$

$$\widehat{\mathcal{G}} = \begin{bmatrix} 0 & -\mathbf{Q}_{lCJ} \\ \mathbf{Q}^t_{lEL} & 0 \end{bmatrix} + \begin{bmatrix} 0 & -\mathbf{Q}_{lCR} \\ \mathbf{Q}^t_{lRL} & 0 \end{bmatrix} \widehat{\mathcal{H}}.$$

5) Generate the submatrices $\mathbf{C}_t$ and $\mathbf{L}_l$;

6) Formulate the state equations by further matrix manipulation:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{v}_{tC}\\ \mathbf{i}_{lL}\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{C}_t & 0\\ 0 & \mathbf{L}_l\end{bmatrix}^{-1}\mathcal{G}}_{\mathbf{A}}\begin{bmatrix}\mathbf{v}_{tC}\\ \mathbf{i}_{lL}\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{C}_t & 0\\ 0 & \mathbf{L}_l\end{bmatrix}^{-1}\widehat{\mathcal{G}}}_{\mathbf{B}}\begin{bmatrix}\mathbf{v}_{tE}\\ \mathbf{i}_{lJ}\end{bmatrix}.$$

To complete the state-space based approach we need two more steps:

7) Compute the system function of the analog filter from state-space equations;

8) Compute the system function of the digital filter from the system function of the analog filter.

For transforming an analog filter into the equivalent digital filter, the digital filter can be obtained in several ways, but here we briefly recall two of the most popular methods: bilinear transformation method and impulse invariance method.

In order to obtain the digital filter by use of the bilinear transformation, we can either

- Convert the continuous-time state-space system in matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ to the discrete-time system:

  $$\begin{cases} \mathbf{x}(n+1) &= \mathbf{A}_d\mathbf{x}(n) + \mathbf{B}_d\mathbf{u}(n) \\ \mathbf{y}(n) &= \mathbf{C}_d\mathbf{x}(n) + \mathbf{D}_d\mathbf{u}(n), \end{cases}$$

  then compute the transfer function of the digital filter from state-space representation;

- Or we can use the numerator and the denominator coefficients of the analog transfer function obtained previously and then apply the bilinear transform.

Using the impulse invariance method for analog-to-digital filter conversion, we need to get first the numerator and the denominator of the analog transfer function, then to transform the analog transfer function into the digital transfer function.

5 Software Implementation

Based on theoretical aspects of the state-space approach the software implementation workflow (MATLAB) illustrated in Figure 1, is composed of the following blocks:

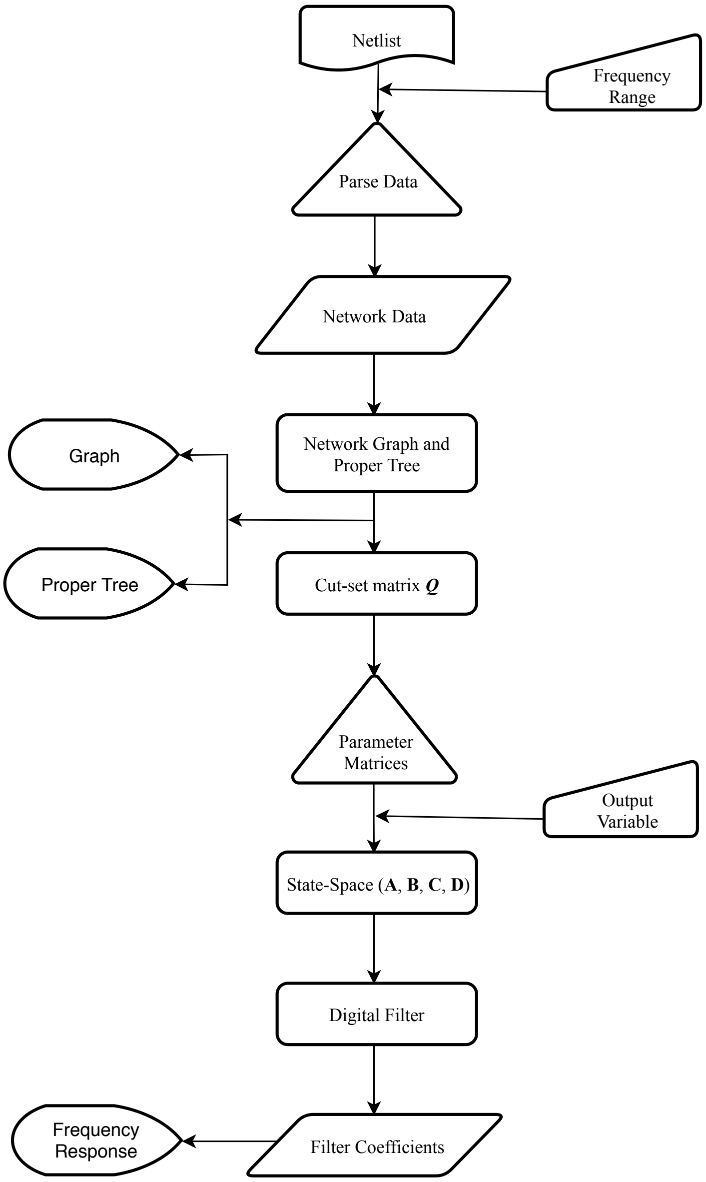

Figure 1: Proposed conversion method flowchart.

- *Netlist*: implementation start point represented by the netlist file input;
- *Frequency Range*: additional manual input step, required for specifying the analysis frequency range;

- *Parse Data*: extraction process block responsible with data collection;
- *Network Data*: process of preparing collected data;
- *Network Graph and Proper Tree*: block in charge of building the associated graph and detecting the proper tree;
- *Graph*: display block that depicts the network graph;
- *Proper Tree*: display block that illustrates the proper tree;
- *Cut-set Matrix Q*: processing block that handles the cut-set matrix computation;
- *Parameter Matrices*: extraction process that delivers link branch and parameter matrices;
- *Output Variable*: output variable manual selection step;
- *State-Space (A, B, C, D)*: process block responsible with building the input filter state-space;
- *Digital Filter*: the actual conversion from analog domain to digital domain is accomplished at this stage;
- *Filter Coefficients*: output block that delivers the corresponding digital filter coefficients;
- *Frequency Response*: display block that depicts the frequency response of the analog filter and its equivalent digital filter.

This software implementation of the state-space approach has presented satisfactory results in terms of conversion outcome. However by extending the range of test filters several implementation drawbacks have come to light. Collecting the netlist data and preparing the network graph computation requires several data structures, this results in an increased execution time and amount of required memory. Also we have observed that generating the proper tree necessitates a large number of steps which can be reduced.

Therefore several optimizations where implemented in the second version (Python) of the software implementation. One of the most important improvement is related to data collection and storage. Using an enhanced data layout all the information from the netlist file is collected and stored for further processing without requiring additional data structures and also without requiring any other processing steps.

Figure 2 illustrates the steps involved in the data collection block. The graph associated to the network is immediately available after reading all the netlist file content.

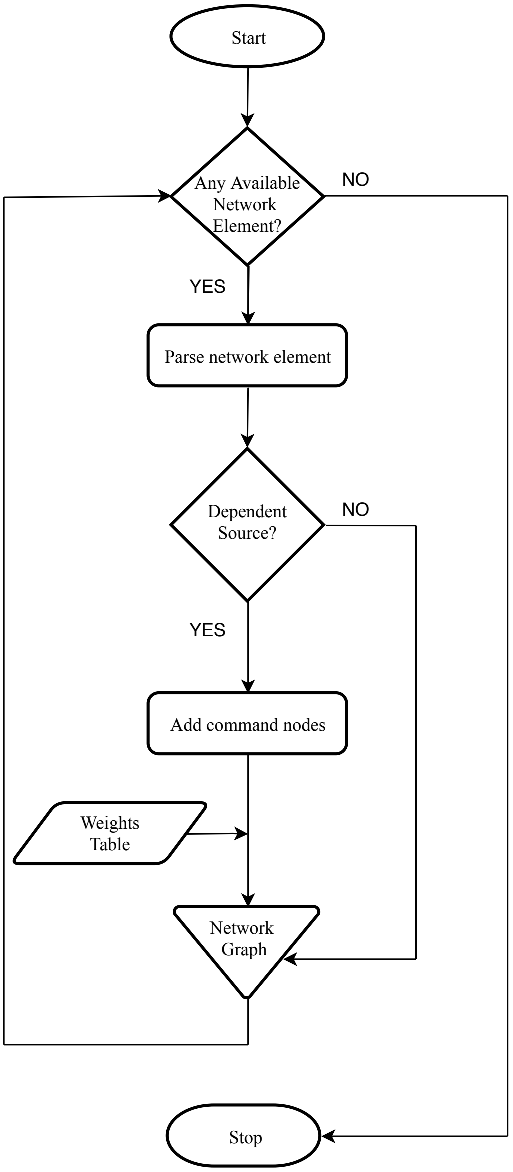

Figure 2: Improved Parse Data processing block flowchart.

Associating a weight to each component is also handled in this stage therefore the proper tree can be detected immediately.

Reducing the number of steps involved in obtaining the network graph, has resulted in improving the execution time of the method. Code execution times indicate an improvement of a couple of orders of magnitude. The performance gap between the two implementations will increase as the number of network elements is higher. In this case also memory is used more efficiently, by eliminating the necessity to create multiple data structures. Memory profiling shows the amount of memory in use during the parse data block execution, for all of the three networks considered.

6 Personal Contributions

This PhD thesis presents a state-space based approach that provides an automatic procedure to convert an analog filter, described by a netlist, to a digital filter described in system function form. One of the main goals of this work was to provide a software tool capable of delivering the digital correspondent of an analog filter. The most important contributions of this work are the following:

- applying the AFN-DFS approach on a multiple notch filter,
- show that the AFN-DFS approach is useful when one implements the forward Euler formula.
- reformulating the state-space equations for networks with only passive elements for computer implementation;
- reformulating the state-space equations for networks with both passive and active elements for computer implementation;
- obtaining and comparing PSpice simulations results with the frequency characteristics of the digital filters obtained after conversion for both case studies.
- developing the entire process chain illustrated in the overview flowchart diagram;
- proposing an automated proper tree detection method;

- testing and comparing results with PSpice simulations, as to validated the outcomes obtained from the implemented software tool.
- defining limitations in terms of speed and memory management of the first software version;
- proposing and developing an enhanced software implementation in order to overcome the limitations of the first version ;
- evaluating the efficiency of the approach using several test networks;
- testing and comparing results with PSpice simulations, for a couple of active filters.

7 List of Publications

The research activity has materialized in three ISI indexed publications. The proposed approach, implementation details and experimental results have been published in scientific journals and presented at international conferences.

- **A. Lodin**, L. Grama, C. Rusu, and J. Takala, “Systematic method to convert analog filters to digital filters,” in *Proceedings of IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, Taipei, Taiwan, 16–18 october 2013, pp. 189–194, ISSN: 2162-3562, [ISI/WoS].
- **A. Lodin**, L. Grama, and C. Rusu, “Symbolic analysis of an analog active filter as path for conversion to digital filter,” *Carpathian Journal of Electronic and Computer Engineering*, vol. 11, no. 2, pp. 8–12, 2018, DOI: 10.2478/cjece-2018-0011.
- **A. Lodin**, L. Grama, and C. Rusu, “From bulky analog active filters to digital filters,” in *2018 International Symposium on Electronics and Telecommunications ISETC 2018*, Timisoara, Romania, 8–9 november 2018,
DOI: 10.1109/ISETC.2018.8583912.

- **A. Lodin**, D. Moga, N. Stroia, D. Petreus, V. Muresan, R. A. Munteanu, and L. Vladareanu, “Modelling and simulation of a remote controlled mechatronic device,” *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 1, pp. 255–260, 1 june 2019, DOI: http://dx.doi.org/10.21533/pen.v7i1.372.g262, ISSN: 2303-4521.

- L. Grama, **A. Lodin**, C. Rusu, and J. Takala, “Method to convert analog filters to digital filters,” in *Proceedings of International Symposium on Image and Signal Processing and Analysis ISPA 2013*, Trieste, Italy, 4–6 september 2013, pp. 438–443, ISSN: 1845-5921, [ISI/WoS].

- C. Rusu, **A. Lodin**, A. Lodin, and J. Astola, “Ordering minimum-phase sets: Numerical properties and systematic searches,” in *Proceedings of European Signal Processing Conference EUSIPCO 2010*, Alborg, Denmark, 23–27 august 2010, pp. 1286–1290, ISSN: 2076-1465, [ISI/WoS].

- L. Grama, **A. Lodin**, and C. Rusu, “About analog filter netlist to digital filter statements approach,” *Carpathian Journal of Electronic and Computer Engineering*, vol. 9, no. 1, pp. 30–33, 2016, ISBN: [ProQuest].

- L. Grama, **A. Lodin**, and C. Rusu, “About converting active analog filters to digital filters,” *Acta Techina Napocensis - Electronics and Telecommunications*, vol. 57, no. 4, pp. 36–39, 2016, ISBN: [ProQuest].

- **A. Lodin**, “Systematic procedure for finding the minimum-phase sequence from a given set,” in *Novice Insights in Electronics, Communications and Information Technology*, Cluj-Napoca, Romania, 2010, pp. 26–32.

- A. Lodin, M. Ghiurcau, and **A. Lodin**, “Automatic iris location using hough transform,” *Carpathian Journal of Electronic and Computer Engineering*, vol. 3, no. 1, pp. 49–52, 2010, ISBN: [ProQuest].

- C. Farago, **A. Lodin**, and R. Groza, “An operational transconductance amplifier sizing methodology with genetic algorithm-based optimization,” *Acta Techina*

Napocensis - Electronics and Telecommunications, vol. 55, no. 1, pp. 15–20, 2014, ISBN: [ProQuest].

- D. Danilescu, **A. Lodin**, L. Grama, and C. Rusu, "Road anomalies detection using basic morphological algorithms," *Carpathian Journal of Electronic and Computer Engineering*, vol. 8, no. 2, pp. 15–18, 2015, ISBN: [ProQuest].

Grants member

- POSDRU/86/1.2/S/61810, "Sistem integrat de programe de masterat în domeniul ingineriei de sunet, imagine şi al aplicaţiilor multimedia - PROMISE," 2012 – 2013, Manager Adjunct P2: Prof. dr. ing. Corneliu RUSU.
- PN-II-PT-PCCA-2013-4-1762, 3/2014 (2014-2017), "Sistem inteligent de management, monitorizare și mentenanță a pavajelor și drumurilor folosind tehnici imagistice moderne - PAV3M," 2015–2016, Project director: Conf. dr. ing. Lăcrimioara GRAMA.
- PN-II-RU-TE-3571.10.2015, "Echipament de chimiohipertermie intraperitoneală, bazat pe paradigma sistemelor cyber-fizice (hiper-cps)," 2016, Project director: Prof. dr. ing. Daniel MOGA.
- PN-III-P2-2.1-BG-2016-0378, 54BG/2016, "Îmbunătățirea performanțelor unui robot prin analiza contextului ambiental din informații acustice - ROXAC," 2016, Project director: Conf. dr. ing. Lăcrimioara GRAMA.

Prizes

- 2nd Prize, "Image Processing and Analysis Student Competition," Brasov, Romania, 30 march 2013.
- 2nd Prize, "The 12th Symposium for Students in Electronics and Telecommunications - SSET 2016," Cluj-Napoca, Romania, 20 may 2016.