About Analog Filter Netlist to Digital Filter Statements Approach

Lăcrimiora GRAMA, Alexandru LODIN, Corneliu RUSU
Technical University of Cluj-Napoca,
Faculty of Electronics, Telecommunications and Information Technology, Signal Processing Group
G. Barițiu 26-28, RO-400027, Cluj-Napoca, Romania
Corresponding author: Lacrimioara.Grama@bel.utcluj.ro

***Abstract*—The goal of this paper is to detail the Analog Filter Netlist to Digital Filter Statements approach used to obtain the digital filter from the netlist of the analog filter. This method is done by replacing each component by its corresponding companion model and then discretizing the set of equations which describe in the time domain the behavior of all the circuit's components. Finally, the software implementation of the digital filter consists in the transcription of the discrete-time system of equations. We show that the selection of step-size might be sensitive problem for this approach.**

Keywords—analog filter; digital filter; netlist

I. Introduction

Infinite Impulse Response Filters (IIR) are often designed by using analog filters described in transform domain. The analog filters are then converted to digital filters using appropriate transformation from analog frequency domain to digital frequency domain. However, to convert the analog filter to the digital filter for these standard methods we need to know in advance the transfer function of the analog filter.

Recently, two methods were proposed to obtain the digital filter directly from the netlist of the analog filter. The first one is the Analog Filter Netlist to Digital Filter Statements (AFN-DFS) approach, where every component is replaced by its corresponding companion model and a software code for the digital filter may be available rapidly [1, 2, 3, 4]. The second one is the state-space (SS) based approach, which allows automated conversion of an analog filter described by a diagram or a netlist to a digital filter described in system function form [5, 6].

In this work, we shall present in details the AFN-DFS approach. The paper is organized as follows. In Section II related works are revised and in Section III the AFN-DFS method is illustrated for an analog multiple notch filter. The advantages and the drawbacks of the method are the subject of Section IV.

II. Related Works

Any analog circuit may be described using a diagram where components are drawn or a set of connected icons displayed on the screen. In both cases the corresponding netlist of components is available.

The discrete models of the analog filters to obtain digital filters has been the focus of many previous works, such as [7, 8]. However, to obtain a digital filter directly from the netlist of an analog filter has been the goal of only recent works.

The Analog Filter Netlist to Digital Filter Statements Approach of converting an analog filter described by a netlist to a digital filter consists of the following steps:

1. First 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 behavior 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 then 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.

III. AFN-DFS Method for the Multiple Notch Filter

We shall illustrate the AFN-DFS method by an analog filter which is a multiple notch filter, previously implemented in television receivers (Fig. 1) [9]. Note that the AFN-DFS method has been applied for other simple analog circuits in [1, 2, 3, 4].

The output of the filter is the voltage across R8. In case that one would like to use standard techniques to convert from *s*-domain to *z*-domain, the transfer function in *s*-domain of the analog filter must be known in advance. As one can see, it is not an easy task to find the transfer function of this analog filter.

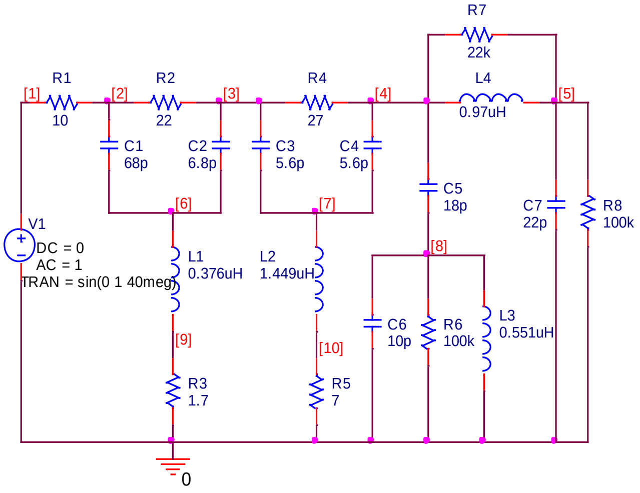

Fig. 1. *Analog test filter with passive elements [5].*

One can write the Kirchoff Current Laws:

$$\begin{aligned} i_{12}(t)-i_{23}(t)-i_{26}(t)&=0\\ i_{23}(t)-i_{36}(t)-i_{34}(t)-i_{37}(t)&=0\\ i_{34}(t)-i_{47}(t)-i_{48}(t)-i_{45L}(t)-i_{45R}(t)&=0\\ i_{45L}(t)+i_{45R}(t)-i_{50C}(t)-i_{50R}(t)&=0\\ i_{26}(t)+i_{36}(t)-i_{69}(t)&=0\\ i_{37}(t)+i_{47}(t)-i_{710}(t)&=0\\ i_{48}(t)-i_{80C}(t)-i_{80R}(t)-i_{80L}(t)&=0\\ i_{69}(t)-i_{90}(t)&=0\\ i_{710}(t)-i_{100}(t)&=0 \end{aligned} \tag{1}$$

then the Branch Equations:

$$\begin{aligned} v_1(t)-v_2(t)&=R1\cdot i_{12}(t)\\ v_2(t)-v_3(t)&=R2\cdot i_{23}(t)\\ v_9(t)&=R3\cdot i_{90}(t)\\ v_3(t)-v_4(t)&=R4\cdot i_{34}(t)\\ v_{10}(t)&=R5\cdot i_{100}(t)\\ v_8(t)&=R6\cdot i_{80R}(t)\\ v_4(t)-v_5(t)&=R7\cdot i_{45R}(t)\\ v_5(t)&=R8\cdot i_{50R}(t) \end{aligned} \tag{2}$$

$$\begin{aligned} L1\cdot i'_{69}(t)&=v_6(t)-v_9(t)\\ L2\cdot i'_{710}(t)&=v_7(t)-v_{10}(t)\\ L3\cdot i'_{80L}(t)&=v_8(t)\\ L4\cdot i'_{45L}(t)&=v_4(t)-v_5(t) \end{aligned} \tag{3}$$

$$\begin{aligned} C1\cdot\left[v'_2(t)-v'_6(t)\right]&=i_{26}(t)\\ C2\cdot\left[v'_3(t)-v'_6(t)\right]&=i_{36}(t)\\ C3\cdot\left[v'_3(t)-v'_7(t)\right]&=i_{37}(t)\\ C4\cdot\left[v'_4(t)-v'_7(t)\right]&=i_{47}(t)\\ C5\cdot\left[v'_4(t)-v'_8(t)\right]&=i_{48}(t)\\ C6\cdot v'_8(t)&=i_{80C}(t)\\ C7\cdot v'_5(n)&=i_{50C}(t) \end{aligned} \tag{4}$$

The next step is to discretize above written equations from (1) to (4). The discretized Kirchoff current laws are as follows:

$$\begin{aligned} i_{12}(n)-i_{23}(n)-i_{26}(n)&=0\\ i_{23}(n)-i_{36}(n)-i_{34}(n)-i_{37}(n)&=0\\ i_{34}(n)-i_{47}(n)-i_{48}(n)-i_{45L}(n)-i_{45R}(n)&=0\\ i_{45L}(n)+i_{45R}(n)-i_{50C}(n)-i_{50R}(n)&=0\\ i_{26}(n)+i_{36}(n)-i_{69}(n)&=0\\ i_{37}(n)+i_{47}(n)-i_{710}(n)&=0\\ i_{48}(n)-i_{80C}(n)-i_{80R}(n)-i_{80L}(n)&=0\\ i_{69}(n)-i_{90}(n)&=0\\ i_{710}(n)-i_{100}(n)&=0 \end{aligned} \tag{5}$$

then the Branch Equations. In case of resistors they can be written straightforward:

$$\begin{aligned} v_1(n)-v_2(n)&=R1\cdot i_{12}(n)\\ v_2(n)-v_3(n)&=R2\cdot i_{23}(n)\\ v_9(n)&=R3\cdot i_{90}(n)\\ v_3(n)-v_4(n)&=R4\cdot i_{34}(n)\\ v_{10}(n)&=R5\cdot i_{100}(n)\\ v_8(n)&=R6\cdot i_{80R}(n)\\ v_4(n)-v_5(n)&=R7\cdot i_{45R}(n)\\ v_5(n)&=R8\cdot i_{50R}(n) \end{aligned} \tag{6}$$

On the other hand, for inductors and capacitors the derivatives have to be substituted by finite differences. In case of forward Euler algorithm, the system of equations $x'=f(\mathbf{x},t)$ is replaced by $\mathbf{x}_{n+1}=\mathbf{x}_n+h\cdot f(\mathbf{x}_n,t_n)$ Thus the companion model for inductors and capacitors can be discretized as follows:

$$\begin{aligned} L1\cdot\left[i_{69}(n+1)-i_{69}(n)\right]&=h\cdot\left[v_6(n)-v_9(n)\right]\\ L2\cdot\left[i_{710}(n+1)-i_{710}(n)\right]&=h\cdot\left[v_7(n)-v_{10}(n)\right]\\ L3\cdot\left[i_{80L}(n+1)-i_{80L}(n)\right]&=h\cdot v_8(n)\\ L4\cdot\left[i_{45L}(n+1)-i_{45L}(n)\right]&=h\cdot\left[v_4(n)-v_5(n)\right] \end{aligned} \tag{7}$$

$$\begin{aligned}
C1\cdot\left[v_2(n+1)-v_2(n)-v_6(n+1)+v_6(n)\right]&=h\cdot i_{26}(n)\\
C2\cdot\left[v_3(n+1)-v_3(n)-v_6(n+1)+v_6(n)\right]&=h\cdot i_{36}(n)\\
C3\cdot\left[v_3(n+1)-v_3(n)-v_7(n+1)+v_7(n)\right]&=h\cdot i_{37}(n)\\
C4\cdot\left[v_4(n+1)-v_4(n)-v_7(n+1)+v_7(n)\right]&=h\cdot i_{47}(n)\\
C5\cdot\left[v_4(n+1)-v_4(n)-v_8(n+1)+v_8(n)\right]&=h\cdot i_{48}(n)\\
C6\cdot\left[v_8(n+1)-v_8(n)\right]&=h\cdot i_{80C}(n)\\
C7\cdot\left[v_5(n+1)-v_5(n)\right]&=h\cdot i_{50C}(n)
\end{aligned} \tag{8}$$

Here h is the integration step-size and the derivatives have been substituted with the forward difference formula [10]. For every statement in the netlist, we have at least one finite difference equation. While the netlist describes an analog filter, the set of difference equations describes a digital filter. We have a set of 28 equations with 28 unknowns, where $v_1(n)$ is the input signal and $v_5(n)$ is the output signal.

We need some careful manipulation to solve this system. Given the capacitors voltages and the inductors currents at time index n, we compute the rest of voltages and currents from the circuits. Then we update the capacitance voltages and the inductors currents at the time index n+1. The script of the MATLAB file is presented in Table I and this script differs from the script suggested in [5].

Actually, any iterative procedure might be a source of divergence. Indeed, the forward Euler algorithm is sensitive to the settings of parameters, especially to the settings of the step-size. The forward Euler algorithm is numerically unstable when large step-sizes are taken. Basically, this instability is due to the widely different magnitudes of the eigenvalues of the circuit [11].

In case of implementation presented (Table I), the AFN-DFS approach is sensitive with respect to the selection of the step-size, as one can see in Figs. 2 and 3. If the step-size slightly changes from $h = 0.1/F_s$ to $h = 0.125/F_s$, the propose implementation may diverge.

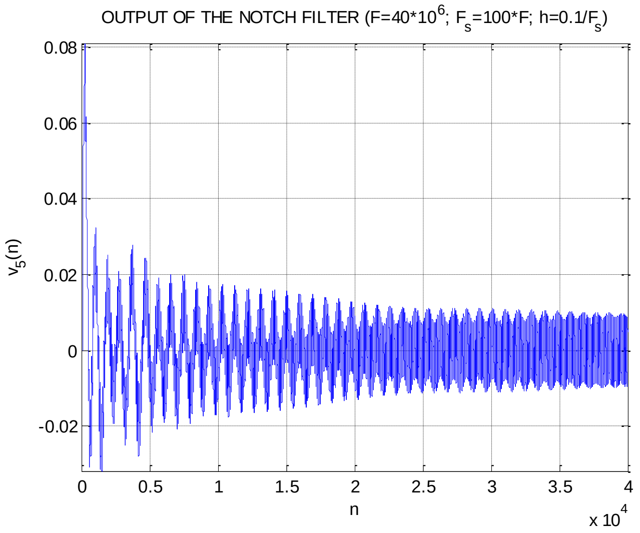

Fig. 2. *Output of the digital filter when the step-size is h = 0.1/Fs.*

TABLE I. THE MATLAB CODE FOR THE AFN-DFS METHOD

```
F = 40*10^6; Fs = 100*F;
N = 40000; h = 0.1/Fs;
v1 = sin(2*pi*F/Fs*[0:N-1 N]);
v2 = zeros(1,N+1);  v3 = zeros(1,N+1);
v4 = zeros(1,N+1);  v5 = zeros(1,N+1);
v6 = zeros(1,N+1);  v7 = zeros(1,N+1);
v8 = zeros(1,N+1);  v9 = zeros(1,N+1);
v10 = zeros(1,N+1);
i69 = zeros(1,N+1);  i710 =z eros(1,N+1);
i80L = zeros(1,N+1); i45L = zeros(1,N+1);
i12 = zeros(1,N+1);  i23 = zeros(1,N+1);
i34 = zeros(1,N+1);  i50R = zeros(1,N+1);
i45R = zeros(1,N+1); i80R = zeros(1,N+1);
i100 = zeros(1,N+1); i90 = zeros(1,N+1);
i26 = zeros(1,N+1);  i36 = zeros(1,N+1);
i37 = zeros(1,N+1);  i47 = zeros(1,N+1);
i50C = zeros(1,N+1); i48 = zeros(1,N+1);
i80C = zeros(1,N+1);
R1 = 10; R2 = 22; R3 = 1.7; R4 = 27; R5 = 7;
R6 = 100*10^3; R7 = 22*10^3; R8 = 100*10^3;
C1 = 68*10^-12; C2 = 6.8*10^-12; C3 = 5.6*10^-12;
C4 = 5.6*10^-12; C5 = 18*10^-12; C6 = 10*10^-12;
C7 = 22*10^-12;
L1 = 0.376*10^-6; L2 = 1.449*10^-6;
L3 = 0.551*10^-6; L4 = 0.97*10^-6;
for n = 1:N
   i12(n) = (v1(n)-v2(n))/R1;
   i23(n) = (v2(n)-v3(n))/R2;
   i34(n) = (v3(n)-v4(n))/R4;
   i45R(n) = (v4(n)-v5(n))/R7;
   i50R(n) = v5(n)/R8;
   i80R(n) = v8(n)/R6;
   i100(n) = i710(n);
   i90(n) = i69(n);
   i26(n) = i12(n)-i23(n);
   i36(n) = i69(n)-i26(n);
   i50C(n) = i45L(n)+i45R(n)-i50R(n);
   i37(n) = i23(n)-i36(n)-i34(n);
   i47(n) = i710(n)-i37(n);
   i48(n) = i34(n)-i47(n)-i45L(n)-i45R(n);
   i80C(n) = i48(n)-i80R(n)-i80L(n);
   v9(n) = R3*i90(n);
   v10(n) = R5*i100(n);
   i69(n+1) = i69(n)+h/L1*(v6(n)-v9(n));
   i710(n+1) = i710(n)+h/L2*(v7(n)-v10(n));
   i80L(n+1) = i80L(n)+h/L3*v8(n);
   i45L(n+1) = i45L(n)+h/L3*(v4(n)-v5(n));
   v5(n+1) = v5(n)+h/C7*i50C(n);
   v8(n+1) = v8(n)+h/C6*i80C(n);
   v4(n+1) = v4(n)+v8(n+1)-v8(n)+h/C5*i48(n);
   v7(n+1) = v7(n)+v4(n+1)-v4(n)-h/C4*i47(n);
   v3(n+1) = v3(n)+v7(n+1)-v7(n)+h/C3*i37(n);
   v6(n+1) = v6(n)+v3(n+1)-v3(n)-h/C2*i36(n);
   v2(n+1) =v 2(n)+v6(n+1)-v6(n)+h/C1*i26(n);
end;
v5min = min(v5);
v5max = max(v5);
figure(); plot(v5); grid;
axis([0 N v5min v5max])
xlabel('n'); ylabel('v_5(n)');
title('OUTPUT  OF  THE  NOTCH  FILTER  (F=40*10^6;
F_s=100*F; h=0.1/F_s)')
```

However, when the step-size is $h = 0.1/F_s$, the output of the digital filter is much smaller than the input, thus the digital filter is indeed a notch filter. Note that the magnitude of the output is consistent with the results presented in [5].

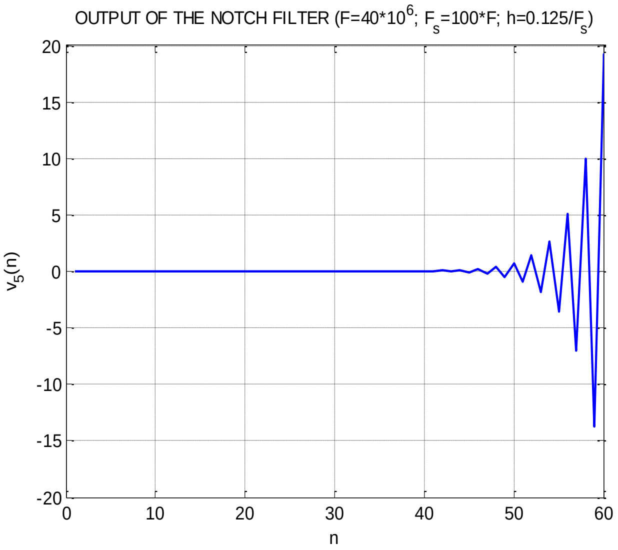

Fig. 3. *Output of the digital filter when the step-size is h = 0.125/Fs.*

IV. Advantages and Drawbacks of the AFN-DFS Method

One can recognize that the knowledge of state-space has been useful in the previous implementation since finally the succession of steps follows a state-space architecture. Moreover, this suggests that the state-space approach can lead certainly to an implementation by giving an adequate strategy of deriving the equations [6]. Furthermore, having a state-space description of either analog or digital filter, one can obtain any canonic structure of digital filter [12].

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

1. Backward Euler algorithm

$$\mathbf{x}_{n+1} = \mathbf{x}_n + f\left(\mathbf{x}_{n+1}, t_{n+1}\right)$$

2. Trapezoidal algorithm

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left[f\left(\mathbf{x}_n, t_n\right) + f\left(\mathbf{x}_{n+1}, t_{n+1}\right)\right]$$

3. Or more complex algorithms such Runge-Kutta algorithms, multistep algorithms, Gear algorithms [11].

The main issue when implementing such algorithms is that the equations contains 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 presented in Table II; we need to elaborate a rather sophisticated approach. But the AFN-DFS approach as well as SS approach have been developed having in mind a small interaction with the user [1].

To conclude the AFN-DFS approach is useful when one implements the forward Euler formula, but in this case a small step-size h is needed. This may lead to a large sampling frequency, which is not always desired.

V. Conclusion

In this paper we have recalled the AFN-DFS approach and we have discussed advantages and drawbacks of the method. The method is friendly and rather easy to implement; however, it sometimes need a small step-size for integration.

Acknowledgment

This work has been supported by Grant PAV3M PN-II-PT-PCCA-2013-4-1762, no. 3/2014 and by Grant PN-III-P2-2.1-BG-2016-0378, no. BG54/2016.

References

[1] C. Rusu, L. Grama, and J. Takala, “SPICE simulation of analog filters: A method for designing digital filters,” in Proc. of EUROCAST 2009, Las Palmas de Gran Canaria, Spain, Feb. 2009, pp. 179–180.

[2] C. Rusu, L. Grama, and J. Takala, SPICE Simulation of Analog Filters: A Method for Designing Digital Filters. Berlin / Heidelberg: Lecture Notes in Computer Science, Springer, 2009, vol. 5717, pp. 534–539.

[3] C. Rusu, L. Kovacs, and L. Grama, “From analog filters to digital filters through SPICE,” Carpathian Journal of Electronic and Computer Engineering, vol. 2, no. 1, pp. 13–16, 2009.

[4] C. Rusu, L. Grama, and J. Takala, “A SPICE method for designing digital filters from analog filters,” in Proc. of the 9th International Symposium on Electronics and Telecommunications (ISETC) 2010, Timisoara, Romania, Nov. 2010, pp. 407–410.

[5] L. Grama, A. Lodin, C. Rusu, and J. Takala, “Method to convert analog filters to digital filters,” in Proc. of ISPA 2013, Sep. 2013, pp. 451–456.

[6] A. Lodin, L. Grama, C. Rusu, and J. Takala, “Systematic method to convert of analog filters to digital filters,” in Proc. of 2013 IEEE Workshop on Signal Processing Systems, 2013, pp. 189–194.

[7] W. K. Chen, Ed., The Circuits and Filters Handbook. Boca Raton, Florida: CRC Press, 1995.

[8] R. J. Faehnrich, “Spice does digital filters,” EDN Design Ideas: September 29, http://www.edn.com/archives/1994/092994/20di1.htm, 1994, Accessed: January 5, 2011.

[9] M. Silisteanu, L. Cipere, and C. Constantinescu, Lucrari practice de depanare a receptoarelor de televiziune. Bucuresti: Editura Didactica si Pedagogica, Bucuresti, 1982.

[10] D. Kincaid and W. Cheney, Numerical analysis: matehmatics of scientific computing. Pacific Grove, California: Brooks/Cole Publishing Company, 1991.

[11] L. D. Chua and P.-M. Lin, Computer-Aided Analysis of Electronic Circuits. Englewood Cliffs, NJ: Prentice Hall, 1975.

[12] D. J. DeFatta, J. G. Lucas, and W. S. Hodgkiss, Digital Signal Processing: A System Design Approach. New-York: John Wiley & Sons, 1988.