MODELING THE QUANTIZATION EFFECTS IN DIGITAL FILTERS VIA SPICE-FAMILY PROGRAMS

Dalibor BIOLEK, Viera BIOLKOVÁ, Zdeněk KOLKA

Abstract: The paper describes an unusual approach of employing the SPICE-family circuit simulation programs in a simple analysis of digital filters, including the quantization effects. For explanation and demonstration, the evaluation version of OrCad PSpice v.15.7 is used which is available free on the Internet.

Keywords: Digital filter, quantization, OrCad, PSpice, modeling, analysis.

1. INTRODUCTION

The design and subsequent analysis of digital filters is well supported by MATLAB combined with the Signal Processing Toolbox. In addition, owners of the Filter Design Toolbox can easily include quantization effects in the design. Additional features are enabled by exporting the designed realization structures into Simulink.

However, not everyone can avail themselves of such expensive software tools. That is why the paper is focused on the analysis of digital filters, including quantization effects, via the SPICE-family programs, namely via the evaluation version of OrCad PSpice from v. 15.7, which is freely available. Compared with MATLAB, another advantage is the graphical tool in the form of schematic editor, enabling easy drawing of filter realization structures with subsequent analysis of frequency responses from the input into an arbitrary node of filter structure. This feature is especially desirable for dynamic range optimization and scaling. The filter structure is created by connecting the blocks of delay, coefficient multiplication, and signal summation.

The techniques of digital filter analysis via Micro-Cap program are described in [1]. These procedures were transferable to the OrCad PSpice program only for idealized analysis, i.e. without considering the quantization effects. The reason was the absence of special mathematical functions for quantization modeling in the OrCad version 10.5 and olders. However, the set-up has changed beginning with version 15.7. In this paper, methods of modeling the quantization effects in OrCad PSpice v. 15.7 are described both for the AC and the transient analyses of digital filters.

2. MODELING THE QUANTIZATION EFFECTS IN ORCAD PSPICE V. 15.7

Starting with version 15.7, OrCad PSpice includes the CEIL, FLOOR, and INTQ functions for behavioral modeling. The FLOOR function returns an integer value of the argument [2]. It can be useful for modeling various methods of data quantization in fixed-point representation. Three of them are shown in Fig. 1 [3], [4].

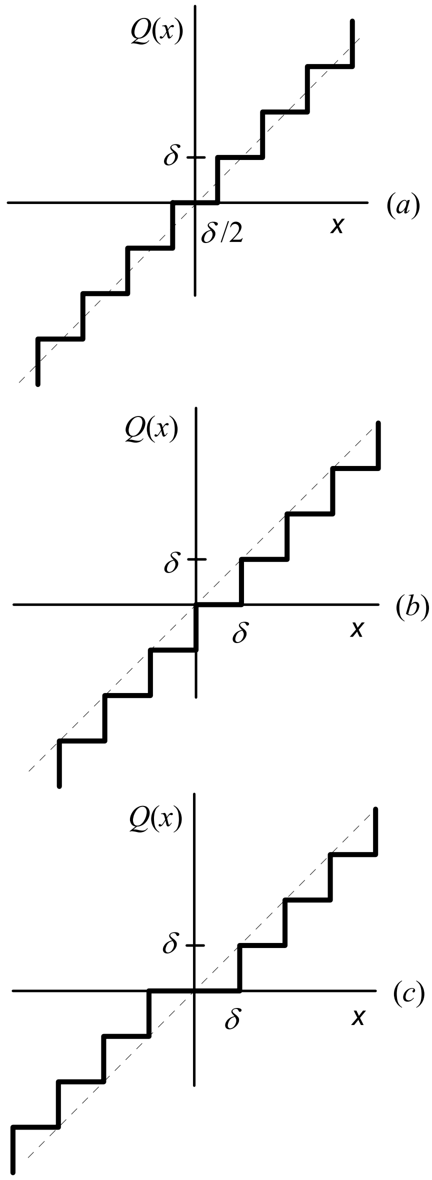


Fig. 1 Quantization methods [3]: (*a*) rounding, (*b*) two's complement truncation, (*c*) one's- complement and sign-magnitude truncation. Here $\delta = 2^{-N}$, *N* is the number of bits without the sign bit

The (*a*), (*b*), and (*c*) – type quantizations can be modeled in PSpice via FLOOR and other PSpice functions as follows:

$$Q(x) = \delta\ FLOOR(x/\delta + 0.5) \quad (1a)$$
$$Q(x) = \delta\ FLOOR(x/\delta) \quad (1b)$$
$$Q(x) = SGN(x)\delta\ FLOOR(ABS(x)/\delta) \quad (1c)$$

The above methods can be used for modeling the quantization of the filter coefficients, the input data, and the arithmetic rounding errors.

A concrete PSpice implementation by the user function ROUND can be as follows. In this demonstration, rounding method (*a*) is used:

```
.func round(x,N)
+{if(N<=0,x,2^(-N)*FLOOR(x*2^N+0.5))}      (2)
```

The number of bits N is defined by a global parameter. In Eq. (2), the user can discard the quantization after setting $N \leq 0$.

3. IMPLEMENTATION OF BASIC BLOCKS IN PSPICE

Since the basic circuit variables in the SPICE-family programs are voltages and currents, the summing block, the block of coefficient multiplication, and the block of delay by sampling period $T_S = 1/f_s$ can be implemented by controlled sources. For the popular PSpice program, the E-type controlled source can be applied:

Summing block:

```
Esum out 0 value={V(in1)+V(in2)}
```

Block of multiplying by constant A:

```
Emul out 0 in 0 {A}
```

Delay block:

```
Ez out 0 LAPLACE {V(in)} {exp(-s/fs)}
```

Both constant A, representing a concrete filter coefficient, and sampling frequency f_s must be defined as global variables by .param command.

Before the AC analysis of a digital filter compiled from the above blocks its input must be excited by an independent voltage source with the attribute AC = 1. After that, one will observe the frequency dependence of voltages at the outputs of respective blocks. With regard to the principal necessity of solving the linear model, it is necessary to disable the quantization of filter variables, except for the coefficients quantization. Full quantization can be used for the Transient analysis.

It is possible to model the above blocks as SPICE subcircuits and to assign them schematic symbols In CAPTURE. We get a powerful tool for digital filter analysis.

One possible form of simple PSpice library is given below:

```
*Digfil.lib - PSPICE library for digital filters
*analysis including the quantization effects

*quantization function, rounding
.func quant(x,N)
+{if(N<=0,x,2^(-N)*FLOOR(x*2^N+0.5))}
*quantization function, two's complement
*truncation
;.func quant(x,N)
+{if(N<=0,x,2^(-N)*FLOOR(x*2^N))}
*quantization function, ones' -complement and
*sign-magnitude truncation
;.func quant(x,N) {if(N<=0,x,
+SGN(x)*2^(-N)*FLOOR(ABS(x)*2^N))}

*summing block
*this subcircuit uses global parameter Nsum
*which MUST be defined in circuit file
.subckt SUM in1 in2 out params:Nsum={Nsum}
Esum out 0 value={quant(V(in1)+V(in2),Nsum)}
.ends

* multiplier
*this subcircuit uses global parameters Ncoef and
*Nmul which MUST be defined in circuit file
.subckt MUL in out
+params:coef=1 Ncoef={Ncoef} Nmul={Nmul}
Emul out 0
+value={quant(V(in)*quant(coef,Ncoef),Nmul)}
.ends

*delay block
*this subcircuit uses global parameter fs
*which MUST be defined in circuit file
.subckt DELAY in out
Ez out 0 LAPLACE {V(in)} {exp(-s/fs)}
Raux out 0 1T
.ends
```

4. DEMONSTRATION OF PSPICE SIMULATIONS

The following demonstration will be performed on the digital filter in Fig. 2 [1]. It is a cascade IIR 8th - order low-pass filter with a cut-off frequency of 3.4kHz and with a sampling frequency of 22.05kHz. The filter coefficients are given in the PSpice input file in **Appendix**. The results of AC analysis are in Fig. 3. Note that it is necessary to define filter coefficients by at least 9 bits in order to achieve therequired characteristic within the entire frequency range.

The filter impulse response as a result of Transient analysis is in Fig. 4. The filter coefficients are quantized to 9 bits. A comparison of two cases is given, for rounding free arithmetic operation (case (*a*)), and for 7-bit quantization of the data (case (*b*)). For the case (*b*), one can observe the parasitic limit cycle operation. Note that the waveforms represent the continuous-time representation of discrete-time signal. That is why one must allow for the steady-state values within a concrete sampling action.

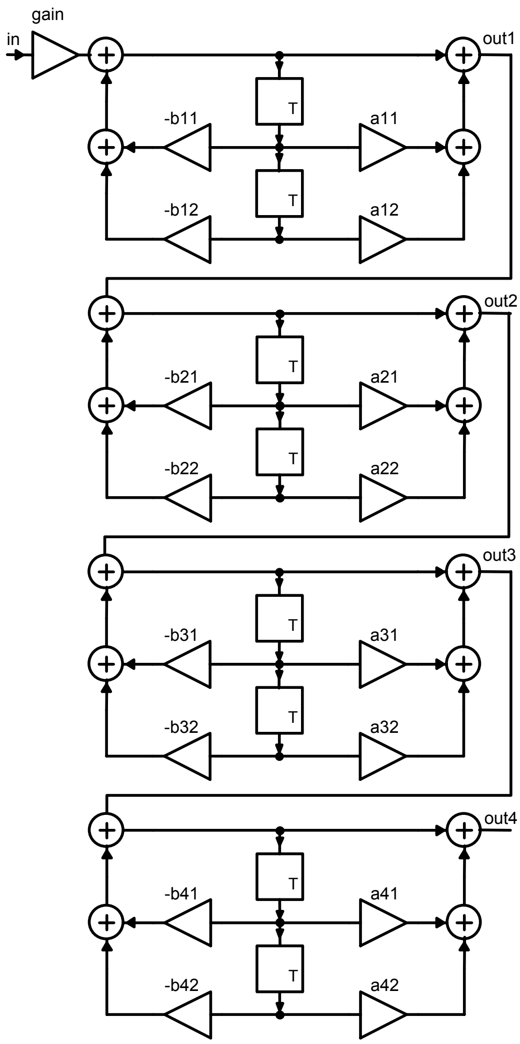


Fig. 2 IIR filter of LP type [1]

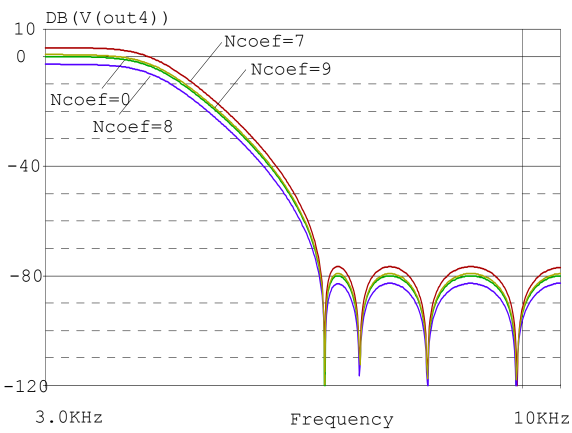
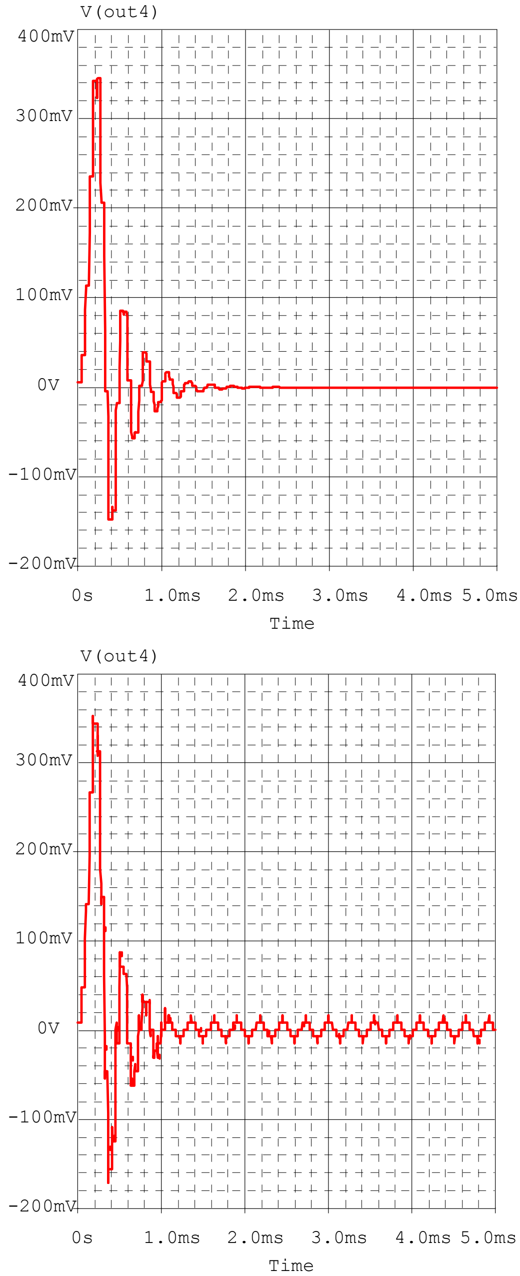


Fig. 3 Frequency responses of filter in Fig. 2 for several numbers of bits for coefficient quantization



(*a*)

(*b*)

Fig. 4 Impulse responses of filter in Fig. 2 for N_{coef}=9 and (*a*) rounding free arithmetic operation, (*b*) for 7 bits quantization

5. ANALYSIS OF QUANTIZATION NOISE PROPAGATING TO THE FILTER OUTPUT

Regarding the quantization effects, in addition to coefficient quantization, the ratio of powers of output and input noises is frequently analyzed. Denote this ratio as *NNR* (Noise to Noise Ratio). It is useful to differentiate between two kinds of sources of quantization noise: noise at the filter input, generated by analog-to-digital converter, and quantization noise due to multiplying the digital filter signals and filter coefficients. In the first case, the complete filter is placed between the noise source and the output. In the latter, the transfer path of noise starts at the output of the appropriate multiplier and leads to the filter output.

However, the methodology of *NNR* computation is identical in both cases. In the first step, we find the transfer function $K(z)$ between the source of the noise and the filter output. Then the *NNR* is evaluated by means of the formula [5]

$$NNR = \frac{1}{2\pi j}\oint_L z^{-1}K(z)K(z^{-1})dz = \sum_{z_p} res\{z^{-1}K(z)K(z^{-1})\} \tag{3}$$

The symbol L represents a curve of integration, specified as a circle of unity radius, centered in the origin of the z-domain complex plane. According to the residue theorem [6], the right-side sum represents the sum of residua of complex function within the braces {} in the poles of this function, which are located inside the curve of integration.

As proven in [7], for Mth-order digital FIR filter, the *NNR* is equal to the sum of squares of its coefficients h_i:

$$NNR = \sum_{i=0}^{M} h_i^2 \tag{4}$$

In the case of IIR filters or high-order FIR filters, the computation of *NNR* according to (4) can be complicated. In such cases, we can use another method, which starts from the frequency response of the transfer path between the source of noise and the filter output:

$$NNR = \frac{1}{f_s}\int_0^{f_s}\left|K(f)\right|^2 df = \frac{2}{f_s}\int_0^{f_s/2}\left|K(f)\right|^2 df \tag{5}$$

The frequency response can be easily obtained by AC analysis. After that, the simulation program must compute the mean square of the absolute value of frequency response within the frequency region from 0 to $f_s/2$ or from 0 to f_s, if appropriate.

An economical realization structure of a 15th-order linear-phase FIR filter is in Fig. 5 [1]. The sampling frequency is 11025Hz. The filter has 16 "symmetrical" coefficients. In Fig. 5, half of them are implemented:

a0= 0.011932832718619923,
a1= 0.0043636441772756003,
a2= -0.0081892506039719284,
a3= 0.073487283106366652,
a4= 0.027003232751724886,
a5= -0.21586100424115826,
a6=-0.14953110748229956,
a7= 0.26565882334944552.

Fig. 6 summarizes the results of AC analysis in the frequency range up to one half of the sampling frequency. The upper curve illustrates the running average (represented by the „*avg*“ function) of the absolute value of the square of frequency response. According to (5), its value for the frequency fs/2 should be equal to the *NNR*. The value scanned from PROBE is 0.291776, which is – within the frame of all digits above – exactly the same result as those generated from the formula (4).

The method described is general, enabling a fast determination of *NNR* from an arbitrary node of the realization structure to the filter output.

6. CONCLUSION

The above facilities of digital filter analyses via the SPICE-family programs have been verified for many simulation problems. Additional details, e.g. how to simulate large digital structures via

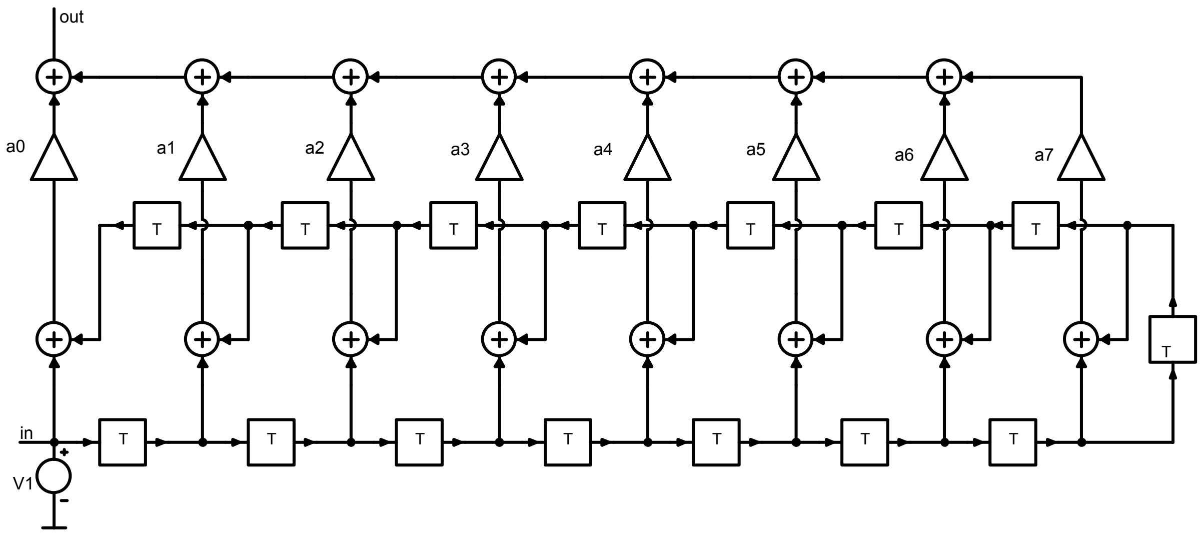


Fig. 5 15th-order bandpass FIR filter

evaluation version of Micro-Cap program, etc., are described in [1].

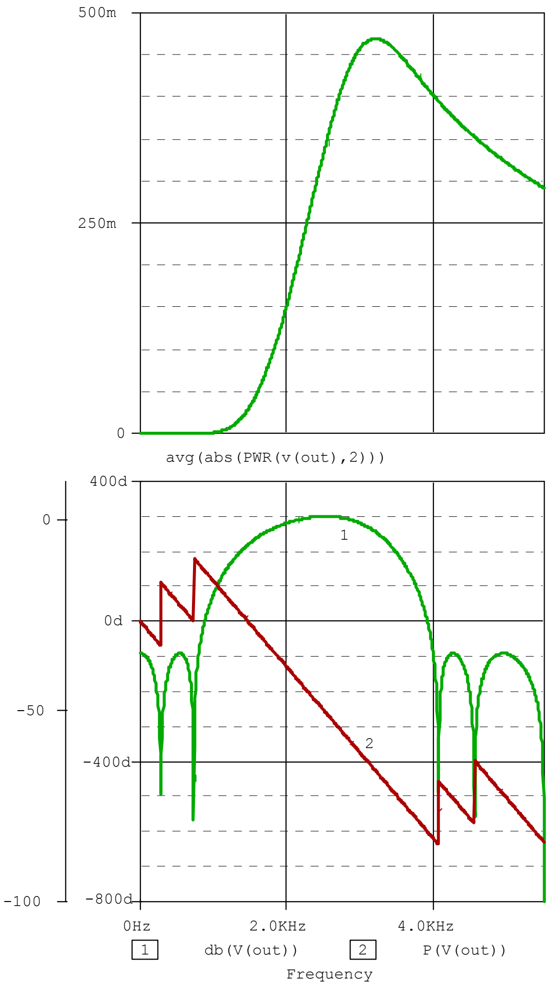


Fig. 6 Frequency responses of the filter from Fig. 5

7. ACKNOWLEDGMENT

This work is supported by the Grant Agency of the Czech Republic under grants No. 102/05/0771 and No. 102/05/0277, and by the research programmes of BUT MSM0021630503, MSM0021630513, and UD Brno MO FVT0000403.

References

[1] BIOLEK, D., ABUETWIRAT, I. F.: Analysis of digital filters via Spice-Compatible programs, 2006. Elektrorevue, 2006/26, ISSN 1213-1539, http://www.elektrorevue.cz/clanky/06026/english.htm.

[2] PSpice A/D Reference Guide. Product Version 15.7, July 2006.

[3] MITRA, S. K.: Digital Signal Processing. A Computer-Based Approach. Third Edition. McGraw Hill, 2006. ISBN 007-124467-0.

[4] DAVÍDEK, V., PŠENIČKA, B.: Tools for discrete-time signal processing. Exercises. ČVUT Praha, 1994. ISBN 80-01-01139-9 (In Czech).

[5] AHMED, N., NATARAJAN, T.: Discrete time Signals and Systems. Reston, Prentice Hall, 1983.

[6] Cauchy integral theorem and residue theorem: http://en.wikipedia.org/wiki/Residue_theorem

[7] ABUETWIRAT, I.F.: Finite Impulse Response Filter. Diploma work, Inst. of Microelectronics, BUT, 2006.

Appendix - PSpice circuit file: IIR 8-th order low-pass digital filter, full quantization

```
*
.options ITL4 100
.param
+fs 22.05kHz N 7 Ncoef 0 Nsum {N} Nmul {N}
.param a11 1.8880173933198123
+a12 0.99999999999998335
+b11 -0.32655932340433669
+b12 0.039446152567426181
+a21 1.2426043302792451
+a22 1.0000000000000131
+b21 -0.4147385903741958
+b22 0.15472564825866436
+a31 0.62610036993806006
+a32 0.99999999999999201
+b31 -0.57140840958280426
+b32 0.38161259426751648
+a41 0.3148879700393597
+a42 0.99999999999999423
+b41 -0.77820865449734911
+b42 0.74175331275429501
.param gain  0.0053733804257323015

;Vin in 0 DC 1 AC 1 ;input signal for step
                              * response and AC analysis
Ein in 0 value={stp(time)-stp(time-1/fs)}  ;input
                              *signal for impulse response
Xgain in in1 MUL params:coef={gain}
*stage No. 1
Xz11 11 12 DELAY
Xz12 12 13 DELAY
Xb11 12 m11 MUL params:coef={-b11} ;the
                                          *feedback path
Xb12 13 m12 MUL params:coef={-b12}
Xsum11 in1 s11 11 SUM
Xsum13 m11 m12 s11 SUM
Xa11 12 m13 MUL params:coef={a11}  ;the
                                          *forward path
Xa12 13 m14 MUL params:coef={a12}
Xsum12 11 s12 out1 SUM
```

```
Xsum14 m13 m14 s12 SUM
*end of stage No. 1
*stage No. 2
Xz21 21 22 DELAY
Xz22 22 23 DELAY
Xb21 22 m21 MUL params:coef={-b21} ;the
                              *feedback path
Xb22 23 m22 MUL params:coef={-b22}
Xsum21 out1 s21 21 SUM
Xsum23 m21 m22 s21 SUM
Xa21 22 m23 MUL params:coef={a21}  ;the
                               *forward path
Xa22 23 m24 MUL params:coef={a22}
Xsum22 21 s22 out2 SUM
Xsum24 m23 m24 s22 SUM
*end of stage No. 2
*stage No. 3
Xz31 31 32 DELAY
Xz32 32 33 DELAY
Xb31 32 m31 MUL params:coef={-b31} ;the
                              *feedback path
Xb32 33 m32 MUL params:coef={-b32}
Xsum31 out2 s31 31 SUM
Xsum33 m31 m32 s31 SUM
Xa31 32 m33 mul params:coef={a31}  ;the forward
                                      *path
Xa32 33 m34 mul params:coef={a32}
Xsum32 31 s32 out3 SUM
Xsum34 m33 m34 s32 SUM
*end of stage No. 3
*stage No. 4
Xz41 41 42 DELAY
Xz42 42 43 DELAY
Xb41 42 m41 MUL params:coef={-b41} ;the
                              *feedback path
Xb42 43 m42 MUL params:coef={-b42}
Xsum41 out3 s41 41 SUM
Xsum43 m41 m42 s41 SUM
Xa41 42 m43 MUL params:coef={a41}  ;the
                               *forward path
Xa42 43 m44 MUL params:coef={a42}
Xsum42 41 s42 out4 SUM
Xsum44 m43 m44 s42 SUM
*end of stage No. 4

;.step param Ncoef list 0 7 8 9
;.AC dec 1000 10 11.025k
.tran 0 5m skipbp
.probe
.inc digfil.lib
.end
```

Summary: The paper shows that digital filters can be successfully analyzed via programs, which are primarily designated for simulating the analog circuits, namely the SPICE-family circuit simulation programs. By means of techniques of behavioral modeling, we can include also the quantization effects into the models of digital filters.

The free evaluation version of OrCad PSpice v.15.7 is used for the implementation. However, one can utilize other similar SPICE-like programs with proper internal mathematical tools for modelling the quantization effects.

prof. Ing. Dalibor BIOLEK, CSc.[1)]
Ing. Viera BIOLKOVÁ[2)]
doc. Dr. Ing. Zdeněk KOLKA[2)]
[1)] UO Brno, Katedra elektrotechniky, Kounicova 65
612 00 Brno, Česká republika
E-mail: dalibor.biolek@unob.cz
[2)] FEKT VUT Brno, Ústav radioelektroniky, Purkyňova
118 612 00 Brno, Česká republika
E-mail: biolkova@feec.vutbr.cz
kolka@feec.vutbr.cz